

Optimal System Architecture Synthesis for Advanced Aircraft Configurations

by

Paul R. Mokotoff

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in the University of Michigan
2026

Doctoral Committee:

Assistant Professor Gokcin Cinar, Chair
Ralph Jansen, Hybrid Electric Technical Integration Manager, NASA
Professor Joaquim R. R. A. Martins
Professor Peter Seiler

Paul R. Mokotoff

prmoko@umich.edu

ORCID iD: [0009-0006-4651-5597](https://orcid.org/0009-0006-4651-5597)

© Paul R. Mokotoff 2026

For the greater glory of God
(Ad Majorem Dei Gloriam)

Acknowledgments

While earning my PhD over the last four years, I have been blessed with countless opportunities to engage in meaningful research, meet new friends, and explore all that the city of Ann Arbor and the state of Michigan have to offer. Everyone acknowledged (and those I may have inadvertently forgotten) is and continues to be part of a massive support system, helping me navigate the complexities of life, inside and outside of academia. You all have helped me grow personally, professionally, and spiritually over the last four years, and I thank you for being a part of my PhD journey.

First, I'd like to thank Prof. Cinar for advising me during my PhD journey. It was a pleasure to be your first PhD student and watch the lab blossom with talent during these past four years. Thank you for all the research and teaching opportunities you provided me, as well as integrating me into the EAP community.

Second, I'd like to thank the other members of my PhD Thesis Committee: Ralph Jansen, Prof. Martins, and Prof. Seiler. I was lucky to interact with each of my committee members in meaningful ways during my PhD. Ralph, thank you for all the technical expertise you provided during our monthly meetings with NASA. Your vast EAP knowledge-base and keen eye towards certifying future aircraft concepts was incredibly helpful. Prof. Martins, it was a pleasure to be your student in AEROSP 588 and collaborate on the Boeing MBSA&E Project. You were also influential in recruiting me to come to Michigan, and I am thankful for the time we spent together before I decided to pursue a PhD here. Prof. Seiler, I enjoyed working with you during the NAF Future Ready Scholars Program in 2025.

There are two other faculty members within the Aerospace Engineering Department that I want to acknowledge. Prof. Cesnik, thank you for being my AEROSP 543 instructor, supporting GSAC and all student organizations in the department, and for the kindness you have shown me during my time in Ann Arbor. I enjoyed getting to know you better and look forward to seeing you at St. Francis on Sundays! I will also never forget the AEROSP 543 prelim you gave me. Prof. Shaw, thank you for being my AEROSP 513 and AEROSP 518 instructor. Aside from the many visits in Office Hours, I appreciate you taking time to speak with me about being an academic.

There are many organizations that funded my research over the last four years and provided key technical expertise. Without you, I would not have been able to successfully complete the PhD program. First, I would like to thank NASA for funding the EAP research that became central to my PhD: Amy Chicatelli, Karin Bozak, Ralph Jansen, Gaudy Bezos-O'Connor, Jonathan Litt, Nolan LaMarche, Jonah Sachs-Wetstone, Andrew Meade, and Dennis Rohn. I am incredibly thankful for the opportunity to work with you all. Amy, thank you for our discussions about my future career plans and support outside of the EAP Project. Gaudy, thank you for all your technical expertise and visiting Michigan. I also enjoyed meeting Alex at St. Mary Student Parish while he was on mission! Second, I want to thank Boeing for funding the MBSA&E Project. I had the privilege of working with Alex Carrere, Sean Wakayama, Mingxuan Shi, Ran Engelbeck, Joerg Gablonsky, and Ron Plybon (of GE Aerospace), and thoroughly enjoyed the time we spent together at the 2026 AIAA SciTech Conference as well as the Technical Interchange Meeting at NASA Glenn in 2025. Third, I want to thank RTX, specifically Mike Ikeda and Katie Kirsch, and the Graham Sustainability Institute at the University of Michigan for funding a project on hybrid electric regional airline operations. I enjoyed visiting RTX in Fall 2023 and seeing the work they are undertaking to electrify future advanced propulsion systems.

Although he did not fund my work, Philip Abolmoali was an amazing mentor for my AEROSP 740 project (and a fellow Long Islander!). I had the pleasure of meeting him in Dayton in 2023, and am thankful that we still remain in-touch.

Next, I'd like to thank everyone from the IDEAS Lab. To my fellow PhD students, past and present – Max Arnson, Huseyin Acar, Arenas Wang, Miranda Stockhausen, Emma Cassidy, Nathan Wongprapinkul, and Triet Ho – thank you for your friendship over these past four years. I have enjoyed growing together as researchers, and am thankful for the network of support that we have begun to establish for each other. I also want to acknowledge my (under)graduate collaborators and mentees – Marcel Barbeau, Yilin Deng, and Nawa Khailany. Thank you for your enthusiasm, commitment to pursuing research, and for teaching me more about mentoring than I taught you about research.

I also had a tremendous network of friends throughout the Aerospace Engineering Department at the University of Michigan. I'd like to thank "MDO & Co." – Elliot Kimmel, Javier Maestre, Eytan Adler, Hannah Hajdik, Ali Gray, Sabet Seraj, Alex Coppeans, Galen Ng, Safa Bakhshi, Alex Kleb, Jasmin Lim, Miles McGruder, Bernardo Pacini, Ella Wu, and Alex Kerlee – for being my closest friends in the department during the last four years. Thank you for including me in all the MDO Lab outings and dinners, making me feel like

I belonged, and introducing me to “Frisbee Fridays.” I also want to thank Nathan Pharis, Michael Pilipchuk, Adam Bussey, Taehyeun Kim, Jasmine Chang, Nihal Simha, and Ryan Foster for their friendship, impromptu conversations, and service on GSAC.

My PhD was also a time of significant spiritual growth, thanks to the vibrant Catholic community in Ann Arbor. To the St. Mary Student Parish Jesuits – Frs. Bobby Karle, Dan Dixon, Joel Medina, Kyle Shinseki, Pat Casey, and Joe Wagner – thank you for your tireless dedication to building God’s Kingdom and witness to the faith, offering the Sacraments, and connecting me closer to God via Ignatian Spirituality. I will miss our conversations in the Sacristy before Daily Mass. I also want to thank a few close friends from the Grad/YP group – Calvin Kerns, Oscar Arguedas, Alex Quizon, Casey Cummings, Luke Bugada, Andrew Morris, and Joseph Beckett – as well as those that served with me on the Lead Team. I am continually amazed at how everyone lives out their Catholic faith, and was inspired by your love for our Lord. I also want to thank the Daily Mass Sacristans – Rainey Lamey, Regina George, & Toni Morales – as well as the Sunday 8:30 am Choir for making St. Mary Student Parish feel like home. I especially want to thank Rainey for all the Sacristan and Eucharistic Minister training, and for allowing me to serve God and my parish in ways that I never could have imagined before coming to Michigan. Lastly, I want to thank Cynthia George, my Spiritual Director, and the person that I consider as my first friend in Ann Arbor. I am incredibly blessed to have been your directee and received guidance from you during this time of tremendous spiritual growth, more than I could have ever imagined before coming to SMSP. Thank you for your patience, care, persistence, wisdom, and steadfast demonstration of what it means to build a relationship with our triune God who is Father, Son, and Holy Spirit.

I was lucky to curl at the Detroit Curling Club with some truly amazing and talented teammates – Ann Swisshelm, Sean Silver, Quentin Way, Mark Lazar, and Elizabeth Demers. Thank you for taking a chance on “the new kid without a team,” making me feel included, helping me improve on the ice, and for cleaning up the many shots I missed. Although Greg Jorgensen and I were never on a league team together, we shared many memories on the ice at the Glowspiel in Leamington, as well as many off the ice at DCC. I am also indebted to you for explaining the sport to Bailey when she came to watch. I also want to thank Jeff & Maggie Peplinski, Ben Glotzhober, Taha Jamil, Craig Gonzalez, Don Bleaney, and Kara Szpondowski for their friendship on and off the ice.

Although I’ve been in Michigan, my mentors and friends from Syracuse University continued to tirelessly support me. Dan Oluwalana has been my PhD partner-in-crime.

Thank you for all your support as we navigated our PhD journeys together, and for hosting me when I visited you. Aly Rote has also been a huge support mechanism. Thank you for the many phone calls, long discussions, and hosting me in Pensacola. Lastly, I want to thank Prof. John Dannenhoffer for advising me during all five years at Syracuse University, providing me with limitless research opportunities, and showing me what a rewarding career in academia can be like. Thank you for recruiting me to come to Syracuse University in 2016, believing in me when I lost hope (remember MAE 341?!), and helping me grow as a researcher, teacher, and mentor. Thanks to Prof. Dannenhoffer, I also had the pleasure of working with Bob Haimes, Nitin Bhagat, Darcy Allison, Dean Bryson, and Ryan Durscher – they have been constant supporters throughout my undergraduate and graduate careers as well.

One of the biggest blessings during my PhD journey was meeting my girlfriend, Bailey, and all the time we've spent together for almost two years. Bailey, I love you! Thank you for being my #1 fan, my adventure buddy, my Michigan connoisseur, my meal-prep partner, and, most importantly, my girlfriend and my love. I am thankful for the adventures we've been on together (especially to Long Island and Northern Michigan), weekends in Grand Rapids, date nights we've shared (especially our Autumn dates and fancy dinners), and all the memories we've made together. I am eager to continue journeying together and cannot wait to see the good paths that God has planned for both of us. Thank you for the constant gift of yourself and the blessing that you are in my life.

Most importantly, I want to thank my family for all their love and support throughout these past four years – I love you! I am beyond blessed to have two amazing parents, Denise and Mark, that formed me into the person that I am today and made tremendous sacrifices to raise Brian and I. Thank you for the weekly FaceTime calls, pep talks, and deliveries of New York Bagels when you visited Ann Arbor. I am forever indebted to both of you for the love you have shown me. I also want to thank my brother, Brian, for his constant love and support, and the deliveries of homemade Italian Flag Cookies. I always looked forward to our weekly FaceTime calls, talking politics, joking around, hearing about your Autocross races, and “shooting the breeze” on whatever else was going on in our lives. We are due for another vacation together (Club Med again?). I would also be remiss if I did not give a shout-out to Savannah & Clover — boop boop!

Table of Contents

Dedication	ii
Acknowledgments	iii
List of Figures	xi
List of Tables	xiv
List of Appendices	xvii
Abstract	xviii
Chapter	
1 Introduction	1
1.1 Nascent Advanced Aircraft Designs	4
1.1.1 NASA’s SUSAN	5
1.1.2 Elysian E9X	6
1.2 Computational Tools for Advanced Aircraft Design	7
1.3 Aircraft Performance and Propulsion System Analysis	11
1.3.1 Hybridization Factors	12
1.3.2 Graph-based Methods	13
1.4 Generalized Aircraft Performance Requirements	15
1.4.1 Existing Certification Regulation Adaptations	16
1.5 System Architecture Synthesis	18
1.5.1 Graph-Based Techniques	18
1.5.2 Multidisciplinary Design Optimization	22
1.5.3 Constraint Programming	25
1.5.4 Branch-and-Bound Methods	26
1.5.5 Integrated Multidisciplinary Design Optimization and Model-Based Systems Engineering Methods	27
1.5.6 Alternative Approaches	30
2 Literature Gaps and Research Objectives	32
2.1 Literature Gaps	32
2.1.1 Nascent Advanced Aircraft Designs	32

2.1.2	Computational Tools for Advanced Aircraft Design	33
2.1.3	Propulsion System Analysis Frameworks	34
2.1.4	Generalized Flight Performance Requirements	35
2.1.5	System Architecture Synthesis	36
2.1.6	Summary of Literature Gaps	38
2.2	Research Questions and Hypotheses	39
2.2.1	Central Research Question	39
2.3	Thesis Outline	41
3	Graph-based Propulsion System Analysis Framework	44
3.1	Introduction	44
3.1.1	Research Questions, Hypotheses, and Experiments	44
3.1.2	Chapter Outline	46
3.2	Graph-based Propulsion System Analysis (GPSA) Framework Construction	47
3.2.1	Component Classification	48
3.2.2	Architecture Construction	49
3.2.3	Operation Definition	53
3.2.4	Power Flow Simulation	63
3.3	GPSA Framework Validation (Experiment 2.0.1)	68
3.3.1	Experimental Setup	68
3.3.2	Test A: Conventional	75
3.3.3	Test B: Parallel Hybrid	79
3.3.4	Test C: Series Hybrid	89
3.3.5	Test D: Series/Parallel Hybrid Electric	96
3.3.6	Test E: Distributed Electric Propulsion	101
3.3.7	Test F: Multiple Sinks	108
3.3.8	Conclusions	114
3.4	Chapter Summary	117
4	Parametric Aircraft Sizing Tool	120
4.1	Introduction	120
4.1.1	Research Questions, Hypotheses, and Experiments	120
4.1.2	Chapter Outline	122
4.2	Aircraft Sizing Tool Modules	123
4.2.1	Computational Procedure	124
4.2.2	Aircraft Sizing	126
4.2.3	Mission Analysis	132
4.2.4	Segment Analysis Algorithm	142
4.3	Aircraft Sizing Tool Validation (Experiment 2.0.2)	143
4.3.1	Experimental Setup	143
4.3.2	Results: Aircraft Sizes	145
4.3.3	Conclusions	147
4.4	Electrified Aircraft Design Space Exploration (Experiment 2.0.3)	148

4.4.1	Experimental Setup	148
4.4.2	Results: System- and Subsystem-Level Performance	151
4.4.3	Conclusions	152
4.5	Chapter Summary	153
5	Generalized Engine Inoperative Performance Requirements	156
5.1	Introduction	156
5.1.1	Research Questions, Hypotheses, and Experiments	157
5.1.2	Chapter Outline	164
5.2	Parameter Suitability Test (Experiment 3.0.1)	164
5.2.1	Experimental Setup	164
5.2.2	Results	168
5.2.3	Conclusions	172
5.3	Examining the Relationship between the SEPLF and CGLF	173
5.3.1	Mathematical Justification of Approximate Equivalence between the SEPLF and CGLF	178
5.4	Generalized Performance Requirement Construction (Experiment 3.0.2) . .	182
5.4.1	Fleet-Level SEPLF Trends	182
5.4.2	Constructing Generalized Engine Inoperative Performance Require- ments	184
5.4.3	Generalized and FAA Climb Performance Requirement Comparison	188
5.4.4	Conclusions	190
5.5	Generalized Engine Inoperative Correction Factor Construction (Experiment 3.0.3)	191
5.5.1	Propulsion System Failure Mapping to a Correction Factor	191
5.5.2	Constructing an Engine Inoperative Correction Factor	192
5.5.3	Experimental Setup	193
5.5.4	Results: Upper Maximum Allowable Correction Factor	193
5.5.5	Results: Lower Maximum Allowable Correction Factor	196
5.5.6	Sensitivity Study	197
5.5.7	Conclusions	198
5.6	Advanced Aircraft Sizing Study (Experiment 3.1.1)	199
5.6.1	Aircraft Modeling	200
5.6.2	Constraint Diagram Generation and Sizing Point Comparison . . .	202
5.7	Chapter Summary	208
6	Bi-Level Optimization Framework	212
6.1	Introduction	212
6.1.1	Research Questions, Hypotheses, and Experiments	212
6.1.2	Chapter Outline	217
6.2	Bi-Level Optimization Framework	217
6.2.1	Propulsion Architecture Optimization Problem and Solver	219
6.2.2	Power Management Optimization Problem and Solver	222

6.2.3	Convergence to an Optimum	223
6.2.4	Fault Tree Analysis	224
6.2.5	Propulsion Architecture Selection	233
6.3	Optimization Framework Validation (Experiment 4.0.1)	242
6.3.1	SUSAN Modeling Approach	243
6.3.2	Performance-Safety Tradeoffs for Fixed Propulsion System Architectures	246
6.3.3	Performance-Safety Tradeoffs using the Bi-Level Optimization Frame- work	258
6.4	Heuristics Benchmarking (Experiment 4.0.2)	265
6.4.1	Experimental Setup	266
6.4.2	Results	266
6.4.3	Conclusions	269
6.5	Optimization Framework and Heuristic Robustness (Experiment 4.0.3) . .	270
6.5.1	Experimental Setup	271
6.5.2	Results	271
6.5.3	Conclusions	280
6.6	Chapter Summary	281
7	Conclusions	285
7.1	Introduction	285
7.1.1	Chapter Outline	286
7.2	Answers to Research Questions	286
7.2.1	Research Question 2.0: Graph-based Propulsion System Analysis Framework and Parametric Aircraft Sizing Tool	286
7.2.2	Research Question 3.0: Generalized Engine Inoperative Performance Requirements	288
7.2.3	Research Question 4.0: Bi-Level Optimization Framework	291
7.2.4	Research Question 1.0: Overarching Thesis	295
7.3	Key Lessons Learned for Advanced Aircraft Design	297
7.4	Contributions	298
7.5	Future Work	301
7.5.1	GPSA Framework	301
7.5.2	Parametric Aircraft Sizing Tool	302
7.5.3	Generalized Engine Inoperative Performance Requirements	303
7.5.4	System Safety Analysis	304
7.5.5	Bi-Level Optimization Framework	304
7.5.6	Broader Systems Architecting Problems	306
	Appendices	308
	Bibliography	336

List of Figures

1.1	Examples of advanced propulsion systems to be integrated in a future aircraft concept.	3
1.2	NASA’s SUSAN aircraft concept [1].	5
1.3	Takeoff segments prescribed by the FAA [2].	16
2.1	Thesis structure.	42
3.1	Parallel hybrid propulsion system architecture.	47
3.2	General component types in a propulsion system.	48
3.3	Parallel hybrid electric propulsion architecture schematic.	49
3.4	Flow direction definition on the PHE architecture.	53
3.5	Generic junction with n sources and m sinks.	56
3.6	Conventional propulsion system architecture.	70
3.7	Parallel hybrid propulsion system architecture.	71
3.8	Series hybrid propulsion system architecture.	72
3.9	Series-Parallel hybrid propulsion system architecture.	73
3.10	Distributed electric propulsion system architecture.	74
3.11	Series hybrid propulsion system with power off-take.	75
3.12	Conventional propulsion system architecture.	76
3.13	Parallel hybrid propulsion system architecture.	80
3.14	Series hybrid propulsion system architecture.	89
3.15	Series-Parallel hybrid electric propulsion architecture schematic.	96
3.16	Distributed electric propulsion architecture schematic.	101
3.17	Series hybrid electric propulsion architecture with power off-take schematic. . .	108
4.1	FAST’s overarching computational procedure, generated using OpenMDAO [3] (FAST, however, is not implemented in OpenMDAO).	125
4.2	Electric machine sizing data from Ref. [4].	129
4.3	FAST’s mission analysis module.	133
4.4	Theoretical drag coefficient for a windmilling engine, reproduced from Ref. [5]. .	137
4.5	Drag coefficient correction, reproduced from Ref. [5].	138
4.6	Flap effectiveness parameter, reproduced from Ref. [6].	139

4.7	Propulsion system architectures examined in the electrified aircraft design study for Experiment 2.0.3. Each row/column corresponds to: Fuel; Battery; Turboshift Engine; Electric Generator; Electric Motor; Propeller; and Thrust Power Sink. .	150
5.1	Mission profile for OEI diversion.	166
5.2	Climb performance impacts due to a propulsion system failure parameterized by the SEPLF or IPALF. The black diagonal line is a reference line indicating $CGLF = SEPLF$ or $CGLF = IPALF$, respectively.	171
5.3	Percent error for the predicted CGLF for a given SEPLF.	176
5.4	Cumulative distribution function of percent error between the SEPLF and CGLF for <i>all</i> operating points during climb.	177
5.5	Normalized specific excess power components allocated for climbing (plotted as circles) and accelerating (plotted as crosses) for different takeoff and climb speeds. The takeoff speed is represented by the point's color.	180
5.6	SEP loss fraction distribution, $\Delta V = 25.72$ m/s.	183
5.7	Sigmoid Curves constructed to represent FAA Part 25 climb performance requirements.	186
5.8	Absolute error between FAA and proposed flight performance requirements for 2-engine aircraft.	189
5.9	Candidate engine inoperative correction factor curve fits.	195
5.10	Sensitivity of RMSE with respect to the parametric curves and maximum allowable engine inoperative correction factor.	197
5.11	Baseline constraint diagram to match literature. The feasible region is shaded in green.	203
5.12	Constraint diagrams using the proposed climb requirements and engine inoperative correction factor. The feasible region is shaded in green.	205
6.1	Extended design structure matrix illustrating the bi-level optimization framework's overarching workflow.	218
6.2	Example system architecture for FTA algorithm.	230
6.3	Schematic digitized to estimate the cable lengths, adapted from Ref. [7].	244
6.4	Baseline propulsion architecture.	247
6.5	Weight breakdowns for all configurations examined.	250
6.6	Pareto Frontiers for different propulsion system failure modes synthesized under a fixed architecture with power management optimization only. System architectures corresponding to each point are labeled at the top).	256
6.7	Pareto Frontiers for different propulsion system failure modes synthesized under the bi-level optimization framework. System architectures corresponding to each point are labeled at the top.	262
6.8	Improved Double Redundancy propulsion system architecture from the bi-level optimization framework.	263
6.9	Batteries included in the propulsion system architectures.	272
6.10	Electric generators included in the propulsion system architectures.	272

6.11	Electric motors included in the propulsion system architectures.	273
6.12	Redundancy levels included in the propulsion system architectures.	273
6.13	Fuel burn distribution.	274
6.14	Magnified final fuel burn distribution.	274
6.15	Failure rate distribution.	275
6.16	Batteries included in the propulsion system architectures.	276
6.17	Electric generators included in the propulsion system architectures.	276
6.18	Electric motors included in the propulsion system architectures.	277
6.19	Redundancy levels included in the propulsion system architectures.	277
6.20	Fuel burn distribution.	278
6.21	Magnified final fuel burn distribution.	279
6.22	Failure rate distribution.	279
6.23	Cable length distribution.	280
6.24	Magnified final cable length distribution.	280
A.1	Turboelectric propulsion architecture schematic.	308
A.2	NASA SUSAN propulsion architecture schematic (simplified).	310
A.3	Series hybrid propulsion architecture schematic with multiple battery recharging options.	312
C.1	SEP loss fraction distributions.	320
C.2	Absolute error between FAA and proposed flight performance requirements for 3-engine aircraft.	323
C.3	Absolute error between FAA and proposed flight performance requirements for 4-engine aircraft.	324
C.4	Candidate engine inoperative correction factor curve fits.	326
C.5	Constraint diagrams using the proposed climb requirements and engine inoperative correction factor. The feasible region is shaded in green.	332
D.1	Propulsion system architectures analyzed in prior performance-safety trade studies.	335

List of Tables

1.1	Features and capabilities of presently available aircraft design tools.	8
1.2	Climb performance requirements in FAA Part 25 regulations.	15
1.3	Engine inoperative correction factors for conventional aircraft configurations. . .	18
3.1	Downstream power flow in a PHE architecture	68
3.2	Downstream power flow in a conventional architecture (all values reported in kW). .	76
3.3	Upstream power flow in a conventional architecture (all values reported in kW). .	77
3.4	Power flow using the framework developed in Ref. [8] (all values reported in kW). .	78
3.5	Benchmarking metrics for the conventional propulsion system.	79
3.6	Upstream power flow in a PHE architecture without battery charging (all values reported in kW).	81
3.7	Upstream power flow in a PHE architecture with battery charging (all values reported in kW).	83
3.8	Power flow using the framework developed in Ref. [8] (all values reported in kW). .	84
3.9	Power flow using the framework developed in Ref. [8] (all values reported in kW). .	85
3.10	Benchmarking metrics for the parallel hybrid propulsion system.	88
3.11	Downstream power flow in a SHE architecture (all values reported in kW). . . .	91
3.12	Equations solved in each downstream SHE analysis iteration.	91
3.13	Upstream power flow in a SHE architecture (all values reported in kW).	92
3.14	Downstream power flow using the framework from Ref. [8] (all values reported in kW).	93
3.15	Benchmarking metrics for the series hybrid propulsion system.	96
3.16	Downstream power flow in a SHE architecture (all values reported in kW). . . .	98
3.17	Upstream power flow in a series/parallel hybrid electric architecture (all values reported in kW).	99
3.18	Power flow using the framework developed in Ref. [8] (all values reported in kW). .	100
3.19	Benchmarking metrics for the series/parallel hybrid electric propulsion system. .	101
3.20	Downstream power flow in a DEP architecture (all values reported in kW). . . .	103
3.21	Upstream power flow in a DEP architecture (all values reported in kW).	104
3.22	Power flow using the framework developed in Ref. [8] (all values reported in kW). .	105
3.23	Benchmarking metrics for the distributed electric propulsion system.	108
3.24	Downstream power flow with multiple sinks (all values reported in kW).	110

3.25	Upstream power flow with multiple sinks (all values reported in kW).	111
3.26	Benchmarking metrics for the series hybrid propulsion system with power off-take.	114
4.1	Assumptions made in FAST’s base-level segment analysis. Higher fidelity codes can be embedded into FAST to override the listed assumptions.	134
4.2	Key performance and design parameters for modeling the ERJ175LR and SUSAN concepts.	145
4.3	Baseline aircraft sizing and performance parameters	146
4.4	Embraer ERJ175LR variant sizing and performance parameters.	151
5.1	Aircraft configurations and variants considered during the operational analysis.	165
5.2	Takeoff speeds and increments (both in m/s) simulated under OEI conditions. Note that $V_{Clib} = V_{Tko} + \Delta V$	166
5.3	Statistical performance metrics recorded to assess parameter suitability.	172
5.4	Sample statistics across <i>all</i> operating points during climb.	177
5.5	Relative climb/acceleration fraction contributions.	180
5.6	Selected quantities across the SEPLF distributions for $\Delta V = 25.72$ m/s.	183
5.7	Median SEPLFs from the operational analysis mapped to the corresponding FAA Part 25 regulations.	184
5.8	Optimized Sigmoid Curve parameters for all FAA Part 25 performance requirements.	187
5.9	Engine inoperative correction factors as a function of the SEPLF.	192
5.10	Curves fits tested while constructing the engine inoperative correction factor. . .	193
5.11	Engine inoperative curve constructions, $\hat{\kappa}_{max} = 2.25$	194
5.12	Engine inoperative curve constructions, $\hat{\kappa}_{max} = 2.05$	196
5.13	Model calibration for the notional E9X. All masses are in units of kg.	201
5.14	SEPLFs for each novel engine inoperative scenario.	202
5.15	Constraint diagram comparison at a 4 kN/m^2 wing loading.	203
5.16	Performance parameters and weights for the redesigned E9Xs. Percentages listed are the relative difference with respect to the notional E9X.	206
5.17	Performance parameters and weights for the redesigned E9Xs. Percentages listed are the relative difference with respect to the notional E9X.	208
6.1	Electrical component performance parameters.	244
6.2	Estimated cable efficiency after digitizing the propulsion system schematic (all in %).	245
6.3	Calibration factors applied to model SUSAN in FAST.	245
6.4	Weight comparison between the SUSAN model in FAST and literature (all weights in kg).	246
6.5	System-level weight breakdowns (all in kg). The percent difference relative to the baseline configuration is included in parenthesis for each redundant configuration.	248
6.6	Component failure rates and probabilities.	251
6.7	Failure intensities (per flight hour) for each configuration under varying failure modes.	252

6.8	Quantitative probability terms defined in FAA Advisory Circular 25.1309-1B.	253
6.9	System safety constraints for benchmarking and corresponding optimized propulsion system architectures.	258
6.10	Pareto Hypervolume from <i>fixed</i> architecture optimization.	258
6.11	Component catalog used in the bi-level optimization framework.	259
6.12	Hyperparameters utilized for the Simulated Annealing Optimizer.	261
6.13	Electric motors that fail from two electric generators failing.	263
6.14	System safety constraints for benchmarking and corresponding optimized propulsion system architectures. Optimized architectures in <i>italics</i> represent configurations with improved system safety relative to the fixed architecture benchmarks.	264
6.15	Pareto Hypervolume from <i>fixed</i> architecture optimization.	264
6.16	System-level performance and safety baselines from the bi-level optimization framework.	267
6.17	Optimized system-level performance and computational resources required, 50 individuals per generation.	267
6.18	Optimized system-level performance and computational resources required, 100 individuals per generation.	268
B.1	Notional ERJ175LR design mission profile.	314
B.2	Notional SUSAN design mission profile.	315
C.1	Generic diversion mission profile. All airspeeds are reported as true airspeeds.	316
C.2	Lift-to-drag ratio correction factors for 2-engine aircraft under one engine inoperative conditions. ΔV represents the difference between the takeoff and climb speeds, previously outlined in Tab. 5.2.	317
C.3	Lift-to-drag ratio correction factors for 4-engine aircraft under one engine inoperative conditions. ΔV represents the difference between the takeoff and climb speeds, previously outlined in Tab. 5.2.	317
C.4	Selected quantities across the SEPLF distributions for $\Delta V = 0.00$ m/s.	321
C.5	Selected quantities across the SEPLF distributions for $\Delta V = 5.14$ m/s.	321
C.6	Selected quantities across the SEPLF distributions for $\Delta V = 10.29$ m/s.	321
C.7	Selected quantities across the SEPLF distributions for $\Delta V = 15.43$ m/s.	322
C.8	Selected quantities across the SEPLF distributions for $\Delta V = 20.58$ m/s.	322
C.9	Notional E9X sizing parameters. Values with asterisks were either assumed or modeled using information outside of Ref. [9].	327
C.10	Assumed notional E9X design mission profile. Airspeeds listed in m/s are denoted as true or equivalent airspeeds.	328
C.11	Notional E9X constraint diagram modeling parameters. Values with asterisks were either assumed or modeled using information outside of Ref. [9].	329
C.12	Constraint diagram comparison at a 6 kN/m ² wing loading.	331
D.1	Parameters to model SUSAN in FAST.	333
D.2	Estimated cable lengths (all in m).	334

List of Appendices

A	Additional Electrified Propulsion Architectures Defined for Analysis the GPSA Framework	308
B	Design Mission Profiles Simulated in Experiments 2.0.2 and 2.0.3	314
C	Generalized Engine Inoperative Performance Requirements	316
D	Modeling NASA’s SUSAN Aircraft	333

Abstract

The commercial aviation industry generates approximately 2.5% of worldwide CO₂ emissions annually, resulting in up to 1,800 megatonnes of CO₂ emitted per year by 2050. Electrified propulsion systems provide the most promising short-term path to reduce CO₂ emissions because of its ability to be integrated in existing tube-and-wing aircraft designs while also maintaining compatibility with existing airport infrastructures. Many possible electrified propulsion system architectures exist, requiring multidisciplinary design optimization to identify ones that maximize system-level performance and meet the necessary safety regulations. However, the increased architectural complexity of electrified propulsion systems requires configuration-agnostic design methods to be developed, enhancing traditional conventional aircraft design methods. In this thesis, a unified propulsion system architecture optimization and synthesis environment was developed to account for nominal aircraft performance, off-nominal propulsion system failure scenarios, and system-level safety constraints in a configuration-agnostic manner, and consisted of three overarching contributions.

First, a graph-based propulsion system analysis framework was developed, providing a formal method to represent and analyze advanced propulsion systems in a configuration-agnostic manner. The framework abstracts a propulsion system as a graph with component constitutive models at each vertex and power transfer relationships along each edge, overcoming topological restrictions in prior state-of-the-art propulsion system analysis methods. Uni- and bi-directional power flows are simulated using a fixed point iteration, enabling power to be propagated across branches of a propulsion system with different path lengths. Integrating the GPSA Framework with a parametric aircraft sizing tool provides a configuration-agnostic aircraft design environment, enabling system-level tradeoffs between aircraft with disparate propulsion system architectures to be quantified.

Second, currently published FAA flight performance requirements for aircraft with discrete, homogeneously sized propulsors were generalized to accommodate advanced propulsion systems with heterogeneously sized or distributed propulsors and their corresponding critical failure modes. A monotonic, bounded mapping between the specific excess power loss and

FAA-published flight performance requirements was constructed to reproduce the climb gradient regulations for conventional aircraft while extending the same flight performance logic to accommodate advanced aircraft concepts and their propulsion systems. To enable seamless integration into the aircraft sizing process, traditional $n/(n - 1)$ engine inoperative correction factors were also generalized as a function of the failure mode severity. An advanced aircraft sizing study revealed that, for aircraft with a propulsion system failure less severe than a legacy one-engine inoperative condition on a conventional aircraft, sizing under the generalized flight performance requirements will reduce propulsion system over-sizing. System-level performance may improve, but depends on the operational constraints imposed.

Third, a bi-level optimization framework was constructed to assemble redundant, safety-feasible propulsion system architecture topologies utilizing embedded, problem-specific heuristics, while also identifying the optimal power management strategy. The optimization framework decomposes a mixed integer optimization problem into its discrete variables, finding the best system architecture, and its continuous variables, finding the optimal component sizes and power management strategy. Depending on the failure mode, the bi-level optimization framework assembles propulsion system architectures with no worse system-level performance and possibly improved system-level safety relative to fixed architecture baselines and those synthesized using stochastic optimization methods with problem-agnostic heuristics. In some cases, system-level safety is improved with minimal system-level performance penalty via asymmetric cross-connection schemes generated by the optimization framework.

Together, the computational frameworks developed in this thesis support the design and optimization of future aircraft concepts and their advanced propulsion system architectures in a configuration-agnostic manner while accounting for nominal aircraft performance, off-nominal propulsion system failure scenarios, and system-level safety constraints.

Chapter 1

Introduction

The commercial aviation industry connects passengers and cargo across the world, making regional and transcontinental travel more available and accessible. Transporting passengers and cargo, however, requires fossil fuels to be combusted, contributing to worldwide CO₂ emissions. Each year, the aviation industry emits approximately 2.5% of the world’s total CO₂ emissions [10], corresponding to approximately 900 megatonnes emitted in 2018 [11]. As the demand for air travel continues to grow, CO₂ emissions are projected to at least double relative to 2018 levels by 2050 [12].

In response to the world’s increasing temperatures and anthropogenic climate change, non-governmental organizations and airlines have established ambitious goals to minimize or fully eliminate their CO₂ emissions [11]. There are four broad methods to realize such emissions goals: 1) technology advancements; 2) operational improvements; 3) alternative fuels; and 4) market-based measures (namely, carbon offsets). Operational improvements are broadly applicable across the national and global aerospace systems, but provide minimal emission savings. Alternative fuels, such as sustainable aviation fuel or biofuels, enable reduced emissions during aircraft operations, but are still generated during the fuel production process. Market-based measures permit CO₂ emissions during airline operations and require alternative sequestration methods to be effective. Given that carbon is still emitted at a large-scale in the aforementioned method, radical technology advancements must be made in the short-term to effectively reduce the aviation industry’s CO₂ emissions.

Radical technology advancements include developing novel airframes and integrating alternative propulsion systems into future aircraft concepts. Novel airframes under study include a transonic truss-braced wing and “Flying V” concept [13], as well as a blended wing body and boxed-wing configurations [14]. While some blended wing body aircraft are flying today, they have yet to be certified by any regulatory authorities, limiting their ability to penetrate the commercial aviation market for now. Other novel aircraft concepts have been

primarily designed on paper and, in some occasions, developed experimentally, but have not been advanced enough for commercial aircraft design. Therefore, a significant reduction in emissions must come from alternative propulsion systems and their novel operational modes.

In the literature, there is a focus on electrified propulsion systems, with a particular emphasis on fully- and hybrid-electric propulsion systems [15]. Electrified propulsion systems promise to improve energy efficiency and reduce operating costs by partially or fully transmitting electrical power between components in the propulsion system. Additional advantages of electrified propulsion systems include decoupling rotational motion between the turboshaft engine and fan for optimal component performance [16], and reducing the thermal/acoustic signature [17], serving as an attractive option for defense-related missions including surveillance and reconnaissance. Thus, future aircraft concepts leveraging electrified [18, 19] and distributed [20–22] propulsion systems enable more efficient and quiet flight, taking positive steps towards reducing commercial aviation’s CO₂ and community noise emissions. In other contexts, aircraft electrification may also refer to flight control actuation or power generation for aircraft subsystems [23]. However, the remainder of this thesis focuses on aircraft electrification in a propulsion-based context.

While hydrogen-powered aircraft have been explored in recent years [24, 25], additional infrastructure at existing airports is needed to distribute and store liquid hydrogen, preventing widespread utilization of such aircraft [26]. Therefore, electrified propulsion systems provide the most promising short-term path forward for reducing commercial aviation’s CO₂ emissions while maintaining compatibility with existing airport infrastructures, and are the focus of this thesis.

A majority of aircraft designs with electrified propulsion systems penetrate the regional airline market, focusing on concepts with 19 to 80 passengers [9, 19, 27–31], with fewer studies focused on single-aisle aircraft designed to carry about 150-200 passengers [7, 32, 33]. Since state-of-the-art batteries have a lower gravimetric specific energy (the energy stored per unit mass) than kerosene-based fuels [34–36], a significant weight penalty is incurred for installing electrified systems onboard the aircraft [37], particularly for fully-electric aircraft. Therefore, it is most common to design an advanced aircraft concept with a hybrid electric propulsion system, boosting the aircraft’s power during takeoff and/or climb and enabling the gas turbine engine to be marginally downsized for more efficient operations during cruise [18]. Integrating an electrified propulsion system into an aircraft is not a trivial challenge, and requires significant efforts to select an optimum propulsion system architecture and account for all system- and subsystem-level interactions between aircraft systems.

Components in an electrified propulsion system can be arranged in multiple ways, resulting in many different propulsion system architectures. A non-exhaustive selection of electrified propulsion system architectures are illustrated in Fig. 1.1. All nine propulsion system architectures illustrated are assembled using only seven components: 1) fuel; 2) battery; 3) gas turbine engine; 4) electric generator; 5) electric motor; 6) propeller/fan; and 7) power off-take. Thus, there exists a combinatorial design space, even without including redundant components to reduce the rate of system-level failures.

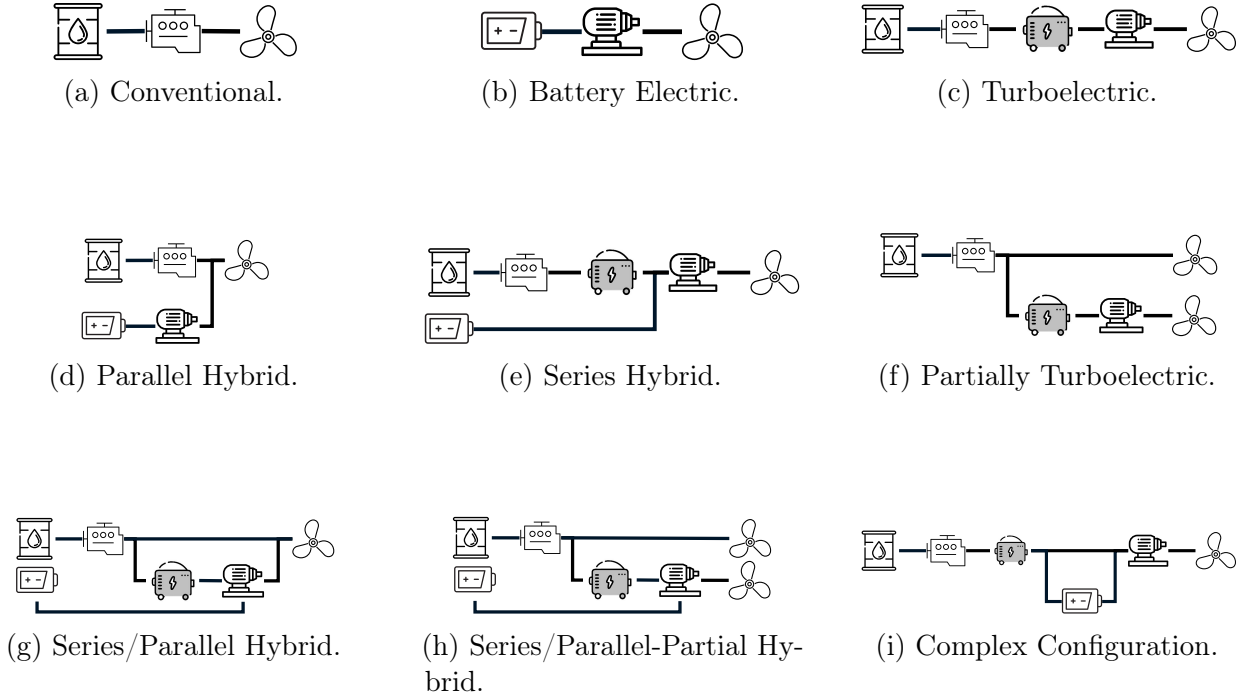


Figure 1.1: Examples of advanced propulsion systems to be integrated in a future aircraft concept.

The increased architectural complexity of an electrified propulsion system poses three challenges to state-of-the-art design methods/tools previously developed for analyzing aircraft with conventional propulsion systems. First, the increased interactions between the propulsion system and airframe demand novel analysis techniques to assess component- and system-level performance accurately. Existing aircraft design and analysis tools may be unable to accommodate the variety of propulsion system architectures in Fig. 1.1 (and others excluded from the figure), warranting further investigation to determine if enhanced tools and evaluation methods must be developed. Second, the introduction of additional components

in the propulsion system leads to additional failure modes that must be considered during design. Existing FAA certification requirements for transport aircraft (Part 25 regulations), particularly flight performance requirements, were developed in the 1960s prior to the advent of electrified propulsion systems [38]. Further exploration is necessary to assess whether the certification requirements must be generalized to accommodate aircraft leveraging electrified propulsion systems. Third, the myriad of electrified propulsion systems requires an engineer to identify the most promising propulsion system architecture (or a few candidates) that could be incorporated into the final aircraft design, and can be achieved using design optimization techniques. Further inquiry is necessary to explore the extent that system architecture optimization techniques have already been developed and applied to aircraft propulsion systems.

The literature review explores existing electrified aircraft designs that have been developed to reduce commercial aviation’s CO₂ emissions, followed by an exploration of the state-of-the-art methods available to support advanced aircraft design and optimal system architecture synthesis.

1.1 Nascent Advanced Aircraft Designs

Advanced aircraft designs have the potential to drastically reduce fuel burn if integrated into the commercial airline fleet. By integrating advanced technologies such as composite materials, higher aspect ratio wings, and engines with improved propulsive or thermal efficiencies, a single aisle aircraft could save approximately 20% fuel on a 6,000 km design mission range [39] while regional turboprop aircraft could save up to 30% fuel on 700 to 1,300 km design mission ranges [30]. Such fuel savings represents an entire generation of incremental technological improvements, requiring time to develop and demonstrate the technology and its airworthiness. However, advanced aircraft concepts and their propulsion systems have the potential to rapidly improve system-level performance, surpassing an entire generation of incremental technological improvements.

Multiple future aircraft concepts such as NASA’s Subsonic Single Aft Engine (SUSAN) [7], Elysian’s E9X [9, 40], and Electra.aero’s EL9 [41] have already been designed with electrified propulsion systems and have the potential to penetrate the commercial market. Many prior publications illustrate the high- and low-level design studies executed to design NASA’s SUSAN and Elysian’s E9X, and are described next. Minimal scholarly work is available on Electra.aero’s EL9.

1.1.1 NASA’s SUSAN

NASA’s Electrified Aircraft Propulsion research develops next-generation aircraft concepts for more efficient flight, and has been designing SUSAN to fly 4,630 km and 180 passengers (weighing 100 kg each) while cruising at Mach 0.775 [7]. Its series/parallel-partial hybrid propulsion architecture and advanced electrification technologies save 8.4% and 9.6% fuel on 1,390 and 1,850 km missions, respectively, relative to a Boeing 737-800 Max re-designed for the same 4,630 km design range [7].



Figure 1.2: NASA’s SUSAN aircraft concept [1].

SUSAN’s system and subsystem designs have been significant research thrusts at NASA for many years. Initial studies on SUSAN involved exploring four different airframe and propulsion architecture combinations [42], which included either counter-rotating propfans or distributed electric propulsors on the wing and counter-rotating propfans or a ducted turbofan aft of the fuselage. Ultimately, the aft turbofan with distributed electric propulsors was down-selected for further design because it exhibited the largest fuel burn savings on a 1,390 km economy mission and the greatest thrust specific fuel consumption reduction among the configurations evaluated. Further studies refined SUSAN’s design, as comprehensively summarized by Jansen et al. [43].

After selecting the system architecture, research efforts focused on the aft engine design and distributed propulsor placement. Mirhashemi et al. [44] designed the aft engine to be a ducted turbofan without any additional power-siphoning turbines beyond the low-pressure turbine. Natural gas was considered as an alternative to kerosene for powering the turbofan; its larger lower heating value relative to kerosene enabled the turbofan engine to be downsized

by 3%. Then, Machado et al. [45] performed detailed aerodynamic analyses to explore whether the distributed electric propulsors should be placed over the wing, under the wing, or aft of the wing’s trailing edge. They also began integrating the aft engine with the fuselage, utilizing shape optimization to ensure a smooth flow into the diffuser.

At the same time, other research explored optimal thermal management and electrical system designs. Heersema and Jansen [46] explored different thermal management system designs, including traditional liquid-air heat exchangers and oscillating heat pipes. Haglage et al. [47] explored different electrical system architectures to connect the generators and motors together. Ultimately, a 16 bus system was selected to limit over-sizing the components and provide a failure mitigation strategy rooted in the distributed electric propulsion system’s inherent redundancy.

Then, to further enhance SUSAN’s efficiency, advanced technologies were included in the design. Lynde et al. [48] reshaped the upper surface of SUSAN’s airfoils on the main wing to reduce the drag coefficient by 0.019. Machado et al. [49] compared the benefits of including distributed electric propulsors with mail-slot nacelles under each wing relative to a configuration with high-bypass turbofan engines. The distributed electric propulsors offered a 1.3% shaft power savings along with a 22% drag reduction for the same net thrust as the turbofan configuration.

The novelty of SUSAN’s propulsion system required additional efforts to ensure airworthiness and a path to certification. Denham and Jansen [50] explored the regulatory barriers to certifying SUSAN, especially those pertaining to the electrified propulsion system. Flight performance requirements, such as the ability to takeoff and climb with an inoperative turbofan engine, were also considered. As a result, a single-use backup battery was added to power the distributed electric propulsion system during a turbofan engine failure. Additionally, they cite that the inherent redundancy within SUSAN’s distributed electric propulsion system prevents any single-point propulsion system failures from occurring. In a follow-up study, Denham et al. [51] designed diversion mission profiles for SUSAN under different propulsion system failure modes such as an inoperative turbofan or electric propulsor during takeoff and cruise.

1.1.2 Elysian E9X

High-level studies were first performed on the Elysian E9X (hereinafter referred to as the “E9X”) by re-assessing the achievable range of battery electric aircraft [52] and performing a conceptual design study to estimate the vehicle’s size and environmental impact [9]. It

was found that electric aircraft may be able to fly between 500 and 1,200 km, a larger range than previously predicted in prior literature. Furthermore, a first-order analysis reveals that the aircraft’s lift-to-drag ratio and empty weight fraction are *independent* of the aircraft’s size [52], highlighting possible advantages of designing such aircraft for commercial transport. In the conceptual design study, the clean sheet aircraft is designed to carry 90 passengers for a 800 km range utilizing a low-wing, T-tail configuration with eight distributed propulsors across the wing. The initial aircraft’s maximum takeoff mass (MTOM) is approximately 76 tonnes; the battery weight is 46% of its MTOM. The E9X was later redesigned based on improved technology projections, resulting in a heavier aircraft (82.5 tonnes) that can only fly 750 km [40].

Since the electric aircraft does not store fuel in its wings, the designers elected to include batteries in the wings, minimizing the aircraft’s structural weight, particularly for minimizing the wing root bending moments [9]. To further minimize structural mass, an aero-elastic optimization was performed to vary the thickness of the wing skins and spars subject to buckling and aerodynamic instability constraints [53]. It was found that placing the battery in the wings instead of the fuselage reduces the wing structural mass by 32%, justifying the decision to do so in prior conceptual design studies.

Additional studies were also performed to quantify the environmental and noise impacts of flying an aircraft with a distributed electric propulsion system relative to a conventional turboprop aircraft [54]. Despite increases in the aircraft’s MTOM, distributed electric propulsion system with a propeller efficiency between 83 and 88% during cruise generated similar noise levels as conventional turboprop aircraft in the current regional fleet. Tradeoffs between reduced noise and aerodynamic efficiency were also identified, with greater noise reductions exhibited on propellers with more blades while modestly sacrificing aerodynamic performance.

1.2 Computational Tools for Advanced Aircraft Design

Designing electrified aircraft requires comprehensive tools capable of evaluating novel propulsion architectures and their impact on the overall design. Computational design tools play a pivotal role in this process, enabling engineers to explore diverse configurations and operational strategies efficiently.

Throughout the past decade, multiple aircraft design and analysis environments were developed for exploring both conventional, electrified, and unconventional aircraft designs.

However, the effectiveness of these tools often depends on their assumptions and capabilities, which may limit their applicability during the early stages of conceptual design when critical decisions about the configuration and its propulsion system are still being made. Many computational tools have been developed to design advanced aircraft concepts and explore the vast design space of electrified propulsion systems. Six recently developed computational frameworks were reviewed to assess each one’s strengths and identify areas that new capabilities are necessary, as well as extensions of prior ones. Table 1.1 summarizes their core capabilities and limitations.

Table 1.1: Features and capabilities of presently available aircraft design tools.

	SUAVE	LUCAS	Aviary	FAST-OAD	JPAD Modeller	E-PASS
Open-Source	✓	✗	✓	✓	✗	✗
Requires Component/ Subsystem Definitions	✓	✓	✓	✓	✓	✓
Interface with Optimization Tools	✓	✓	✓	✓	✓	✓
Evaluate any Propulsion System	✓	✗	✗	✗	✗	✓
Rapid (< 1-minute) Low-Fidelity Analysis	✗	✗	✗	✓	✓	✗

One of the long standing aircraft design tools available is SUAVE [55], an open-source, Python-based code developed at Stanford University that analyzes (un)conventional concepts using multi-fidelity physics-based models. These unconventional concepts are easily analyzed by SUAVE’s energy networks, allowing any propulsion system architecture to be modeled. Additionally, SUAVE includes weight build-ups for tube-and-wing, blended wing body, unmanned aerial, and electric vertical takeoff and landing vehicles. One of SUAVE’s distinguishing features is that it is built to be wrapped by optimizers for aircraft design using multidisciplinary design optimization. This is ideal for late conceptual design and early preliminary design once the aircraft configuration is down-selected and design refinement is underway. However, SUAVE’s lowest-fidelity aerodynamic analysis employs a vortex lattice method, which can be too computationally demanding and prohibit rapid design space exploration. For early design studies, a drag polar model involving empirical relations is more

desirable for its superior computational efficiency and simplicity.

Another aircraft design tool built upon SUAVE is LUCAS [56], which integrates a unified propulsion system model, boundary layer ingestion technology models, and advanced aerodynamic analysis models into the analysis. LUCAS wraps pyOptSparse [57] – an optimization framework that interfaces between any gradient-based or gradient-free optimization problems – around SUAVE. The wrapper allows LUCAS to easily run different aircraft-level optimizations, especially advanced aircraft concepts that include boundary layer ingestion or electrified propulsion systems. One feature that distinguishes LUCAS from other software is its unified propulsion system model, which can easily represent conventional, electrified, hybrid-electric, and fully-electric propulsion systems. This unified model relies on two electrification factors, namely the “source electrification factor”, which splits the power between kerosene fuel and batteries, and the “load electrification factor”, which splits the propulsive power between any gas turbine engines or electric motors [56]. However, the model only is valid for the previously mentioned propulsion architectures and cannot easily integrate other energy sources, such as hydrogen, into its model. Additionally, LUCAS is not open-source and cannot be accessed by the public.

NASA recently released Aviary [58], an open-source aircraft design program which unifies two legacy codes (FLOPS and GASP) under one software. Aviary is built upon OpenMDAO [3], an open-source framework built for tightly coupled multidisciplinary analysis and efficient derivative computation. Aviary is fully differentiated (derivatives from the analysis are computed analytically and available for use in an optimization framework), distinguishing itself from other codes. Its modular framework allows users to swap in their own analyses, particularly to replace Aviary’s empirical relations for higher-fidelity physics-based models. Also, depending on the design phase, the user may select an energy-based mission analysis or a more detailed dynamics model, which models the aircraft as a point mass or two degree-of-freedom system, respectively. One limitation of Aviary, however, is that its propulsion system models are governed by a maximum two control parameters. While this allows the user to control propulsion system components independently of each other, it only permits two degrees of freedom in the propulsion architecture, thus limiting those that can be analyzed.

Another recently developed aircraft sizing tool is FAST-OAD (Fixed-wing Aircraft Sizing Tool - Overall Aircraft Design) [59], an open-source, Python-based software developed by ONERA and ISAE-SUPAERO. Like Aviary, FAST-OAD is built upon OpenMDAO, which promotes modularity and provides the ability to integrate with an MDO environment for more detailed aircraft design. FAST-OAD features two different mission analysis options,

allowing the user to tradeoff between computational accuracy and efficiency. The default mission analysis assumes the aircraft is a point mass, using a time-step integration for all of its mission segments. However, a Breguet Range Equation-based analysis is also available. Despite the Breguet Range Equation’s usefulness for rapid aircraft analyses in early-phase conceptual design, it is only applicable for conventional aircraft. If an electrified aircraft design is selected, the Breguet Range Equation is no longer valid and other approximations are needed [60, 61]. This limits the number of advanced configurations that FAST-OAD can evaluate, though there are plans to include an electric propulsion discipline in future work.

Beyond Python-based tools, JPAD Modeller [62, 63] is a Java-based aircraft design tool that harnesses semi-empirical methods and statistical regressions to size an aircraft concept. JPAD Modeller has now been commercialized [64] and is not available as an open-source tool. One unique feature included is the capability to export a CAD model of the sized aircraft, making it useful for later design phases when further analyses or manufacturing are being considered. Additionally, Genetic Algorithm and Particle Swarm optimization algorithms are built into JPAD Modeller. However, these can be computationally expensive to run because they require thousands of function evaluations and do not rely on gradient-based techniques, making them converge more slowly [65].

The Electrified Propulsion Aircraft Sizing and Synthesis (E-PASS) tool is among the first conceptual aircraft design tools designed specifically for the sizing and system-level analysis of both conventional and electrified aircraft with a variety of propulsion architectures [29, 66, 67]. Its analysis modules include component-based weight estimations, propulsion performance, aerodynamics modeling using a drag polar, an energy-based mission analysis, and physics-based electrical component models (batteries, electric motors and generators, power converters, and transmission systems). One distinguishing factor in E-PASS is its matrix-based propulsion system framework, which utilizes Graph Theory to conveniently represent propulsion system architectures and operation using matrices. The other distinguishing factor is its propulsion system-agnostic mission analysis, which leverages the matrix-based propulsion system framework to use multiple energy sources during the mission. For example, an aircraft can fly with both a gas turbine engine powered by kerosene fuel and an electric motor powered by a battery simultaneously. This capability extends beyond legacy codes like FLOPS, which only permits one energy source to be used at any given time. However, E-PASS is primarily suited for late-phase conceptual design or preliminary design stages when a baseline configuration has been established. Its higher granularity and model fidelity make it less applicable for early-phase conceptual design, where flexibility and broad design

space exploration are typically prioritized.

1.3 Aircraft Performance and Propulsion System Analysis

While reviewing existing computational tools for advanced aircraft design, it was observed that not all tools can evaluate any propulsion system, supporting configuration-agnostic design practices. Thus, a deeper investigation in existing aircraft performance and propulsion system analysis methods are necessary to identify ones that are suitable for the conceptual design of future aircraft with advanced propulsion systems.

For conventional aircraft design, early-stage aircraft performance analysis relied upon the Breguet Range Equation by assuming a constant cruise segment and is a function of fuel weight reduction [68]; takeoff, climb, descent, and landing segments, as well as reserve missions, were not considered in the analysis. While adaptations to the Breguet Range Equation were developed for novel aircraft concepts – either for a single propulsion system configuration [60] or a more generic one [61] – electrified aircraft introduce new dynamics, such as hybridized power usage and dynamic power split strategies, rendering such models insufficient and necessitating more detailed analyses during conceptual design. Therefore, simulating the flight at a more granular level better approximates the benefits of flying electrically rather than estimating the aircraft’s mission as a constant cruise segment. This has led to both analytical and numerical optimizations of an electrified aircraft’s power management strategy [69–72], trajectory [73], or both [74]. In these studies, the propulsion system models were constructed by either employing an external high-fidelity tool [71, 72], developing elementary algebraic equations [69, 73], or leveraging actual engine test data [70].

While these approaches are reasonable for analyzing a single aircraft configuration, certain gaps emerge if they are to be used on other aircraft analyses. First, detailed analyses are often unnecessary during early conceptual design and are more effective upon down-selecting a system architecture. Second, algebraic relations necessitate reformulation for each propulsion system and is computationally inefficient, requiring new code to be developed for each configuration. Third, using engine test data may be reasonable, granted that the aircraft’s size does not significantly change, or the engine was specifically developed for an electrified aircraft concept. However, the airframe will become heavier due to the weight penalties imposed by integrating electrical components on the aircraft [17], requiring a larger engine.

Thus, there is a need to model electrified propulsion system architectures in a flexible and scalable manner to ensure that they can be seamlessly integrated into an aircraft sizing

software. Two modeling approaches are already present in the literature and require further exploration. The first approach involves defining a “hybridization factor” or “power split”, and then computing a set of analytical equations to represent the propulsion system’s operation. The second approach represents the propulsion system as a graph and utilizes matrix operations to simulate its operation.

1.3.1 Hybridization Factors

Previous studies have taken multiple approaches to defining hybridization factors, power splits, or similar quantities. The earliest notion of a power split for aerospace applications was explored by Lorenz et al. [75], and defines the energy- and power-based “degrees of hybridization” (DOH), which reflect the fraction of supplied energy/power relative to the total installed energy/power onboard the aircraft. Selecting an energy- and power-based DOH dictate the fraction of electrical energy stored onboard the aircraft and provided by the powertrain, respectively. For example, if both DOH are 0, then a conventional, turboshaft engine configuration exists. Conversely, a fully electric powertrain is defined by both DOH being 1. This methodology is useful because different propulsion systems are represented using a consistent set of quantities, regardless of how they are powered.

Isikveren et al. [76] later introduced two new dimensionless quantities for aircraft utilizing two energy sources. These quantities are used to demonstrate that there are more DOH combinations available in the design space than Lorenz et al. [75] previously predicted, and implies that they can change during flight.

In a design context, Cinar et al. [77] and Pornet et al. [78] define hybridization factors while sizing hybrid electric aircraft. Cinar et al. [77] defines a hybridization factor as the ratio of maximum electric motor power output to the sum of the maximum turboshaft engine and electric motor power output (consistent with the definition provided by Lorenz et al. [75]). Pornet et al. [78] uses a similar approach, but also defines individual power settings for each component’s operation during flight. Both works use analytical energy conservation equations to describe the propulsion system’s behavior. However, these expressions must be re-derived if the propulsion system architecture were to change, thus limiting their generality.

Other studies explored a more generic approach to represent multiple electrified propulsion architectures [79–81]. The propulsion architecture changes depending on the values of the two hybridization factors defined: 1) the ratio of battery power output to the total battery and fuel (or turboshaft engine) power output; and 2) the ratio of electric propulsor power output to the total electric and mechanical propulsor power output. Despite the improved generality,

the framework lumps the electric and mechanical propulsors together in the analysis, making it difficult to explicitly direct power to a specific propulsor. Such a capability is necessary for analyzing propulsion systems with redundant components for improved system safety.

There are two key strengths associated with using hybridization factors. First, the propulsion system’s operations can be encoded in a single linear system of equations [81], and can be solved using many available direct and iterative methods [82], allowing for a simple computational implementation. Second, the power management strategy is varied by changing up to two values, facilitating simple design space exploration and optimization studies. Such studies are critical during early phase conceptual design, especially while considering multiple electrified propulsion systems and power distribution strategies.

However, three main weaknesses emerge when using hybridization factors in propulsion system analysis techniques. First, any propulsion system architecture change requires a new set of algebraic relations to be defined [77, 80]. For a simple propulsion system involving few components, the relations can be re-written quickly. However, for more complex systems, the energy conservation equations may become cumbersome to derive manually. A similar weakness also exists in propulsion system analysis programs that require the user to manually define their propulsion system in a graphical user interface. Second, only two power distribution quantities are defined in the frameworks [75, 76, 79, 80], making them difficult to extend to redundant systems. The framework must be adapted to include additional variables for power distribution between redundant components, facilitating more granular control over the power flow paths. Third, some of the hybridization factors defined relate the battery power output to the total power output of the battery and the fuel [81] (or turboshaft engine [79, 80]). These parameters are intuitively difficult to select because the components operate along different power flow paths, obfuscating the useful power each one provides. These weaknesses can be circumvented using graph-based approaches, as demonstrated next.

1.3.2 Graph-based Methods

Graph-based approaches represent a propulsion system architecture as a directed graph, where the vertices and edges are represented by propulsion system components and their flow paths, respectively. For example, Lawhorn et al. [83] used graph-based models for representing an electrified propulsion system. This approach is scalable to propulsion architectures with many components, allowing redundant system architectures to be generated [84]. Despite this advantage, only uni-directional power flows are simulated, prohibiting battery recharging during flight.

The first formal graph-based propulsion system analysis framework was introduced by Cinar [66]. In this approach, propulsion system components were categorized into energy, power, and thrust sources. Then, two matrices were defined to represent connections between energy-power sources and power-thrust sources. This formulation provided a structured and scalable method for representing many propulsion architectures, including conventional, hybrid electric, and all electric ones.

Building on this foundation, subsequent work [8] expanded the framework by introducing additional matrices to define connections and interactions between: 1) thrust-power sources; 2) power-power sources; and 3) power-energy sources. These refinements enabled more detailed power flow simulations, including time-varying power splits, which allowed for the analysis of different operational scenarios and failure conditions [29], as well as power management optimization throughout a flight [67].

The framework developed by Cinar et al. [8] has three significant strengths. First, the energy conservation equations are implicitly represented in the matrix entries. This eliminates all manual derivations, making it more convenient to define new propulsion architectures and operational strategies. Furthermore, this method defines the propulsion system in a machine readable manner, rendering it useful for computational analysis. Second, all power splits are selected at the corresponding junctions between power-transmitting components, making graph-based methods more intuitive than prior approaches. This overcomes the limitations associated with hybridization factors, which were defined between components of a similar type rather than a common junction. Third, power splits are defined between every component connection, allowing for more granular power distribution strategies to be defined. This accommodates propulsion systems with redundant components, and is a significant enhancement from previous approaches that relied on hybridization factors.

However, the framework exhibits three limitations. First, the framework implicitly assumes that all thrust source outputs provide propulsive power, prohibiting power off-takes from being modeled. Second, the power splits defined are only used to propagate power from the thrust sources to the energy sources. Though this is suitable to compute the power required by each component, its use is limited, and cannot be used to describe how power flows from the energy sources to the thrust sources, a computation necessary for predicting the power available in a propulsion system. Third, the power flow simulation proposed is constrained by the requirement that only two power sources (such as turboshaft engines, electric generators, and electric motors) may be connected in series, requiring workarounds for certain propulsion architectures. This prohibits detailed modeling of series hybrid and turboelectric propulsion

architectures, which require three power sources to be arranged in series, as described by the schematics in Section 3.3.4 and Appendix A.1.

1.4 Generalized Aircraft Performance Requirements

Prior to entering service, airframe manufacturers must prove airworthiness by demonstrating that the configuration satisfies certification regulations for either “normal” or “transport” aircraft, codified in Part 23 and 25 regulations, respectively, by the United States’ Federal Aviation Administration (FAA) [85, 86] and European Union Aviation Safety Agency (EASA) ¹ [87, 88]. Aside from establishing airworthiness, the regulations act as constraints on the system- and subsystem-level designs.

For example, Fig. 1.3 illustrates the minimum takeoff path prescribed by FAA Part 25 regulations [2], which is divided into multiple segments with specific flight conditions and performance requirements. Airworthiness is ensured by demonstrating that the aircraft can takeoff and climb away from the runway under “one engine inoperative” (OEI) conditions upon reaching V_{EF} (engine failure speed) during the takeoff roll [89]. The FAA defines climb performance by establishing minimum climb gradients for each segment.

The climb gradients defined by the FAA are listed in Tab. 1.2 for 2-, 3-, and 4-engine aircraft [89], and have remained unchanged since the 1960s [38]. As defined by the Part 25.119 and 25.121d regulations, additional climb performance requirements are imposed during landing for “all engines operative” (AEO) and OEI conditions, ensuring that the aircraft can safely climb if a missed approach occurs.

Table 1.2: Climb performance requirements in FAA Part 25 regulations.

Requirement	2-Engine	3-Engine	4-Engine
25.111 (Takeoff Flight Path, OEI)	$\gamma \geq 1.2\%$	$\gamma \geq 1.5\%$	$\gamma \geq 1.7\%$
25.119 (Landing, AEO)	$\gamma \geq 3.2\%$	$\gamma \geq 3.2\%$	$\gamma \geq 3.2\%$
25.121a (First Segment Climb, OEI)	$\gamma \geq 0.0\%$	$\gamma \geq 0.3\%$	$\gamma \geq 0.5\%$
25.121b (Second Segment Climb, OEI)	$\gamma \geq 2.4\%$	$\gamma \geq 2.7\%$	$\gamma \geq 3.0\%$
25.121c (Final Climb, OEI)	$\gamma \geq 1.2\%$	$\gamma \geq 1.5\%$	$\gamma \geq 1.7\%$
25.121d (Landing, OEI)	$\gamma \geq 2.1\%$	$\gamma \geq 2.4\%$	$\gamma \geq 2.7\%$

¹Although FAA Part 23 and 25 regulations are broadly harmonized with EASA CS-23 and CS-25 regulations, respectively, differences exist and certification is authority-specific. The present work uses the FAA Part 25 climb gradient structure as a reference baseline for performance requirement generalization, and refers to them as “FAA regulations.”

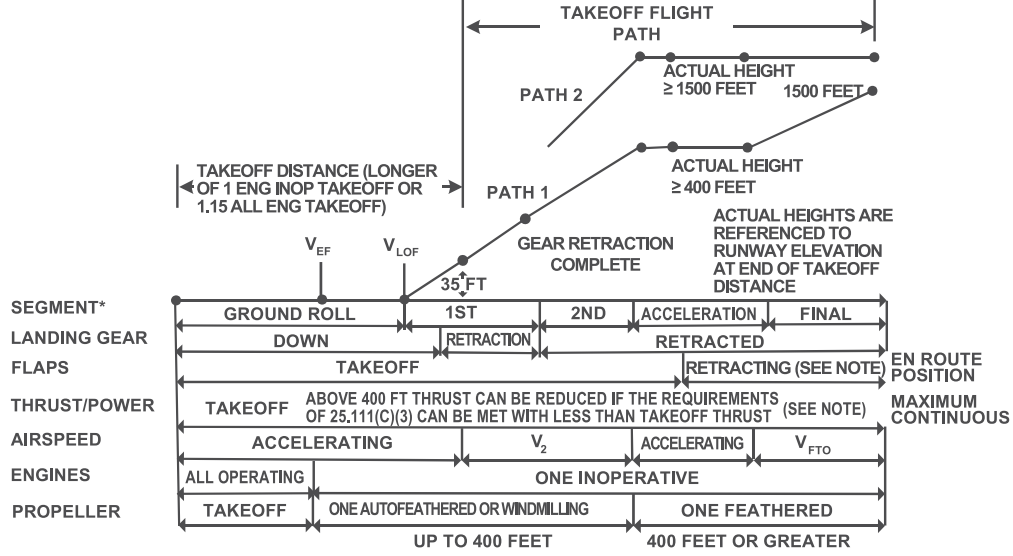


Figure 1.3: Takeoff segments prescribed by the FAA [2].

The OEI requirements in existing FAA regulations implicitly assume that a 2-, 3-, or 4-engine aircraft powered by gas turbine engines is being certified. This neglects advanced concepts, particularly distributed propulsion systems, which utilize more than four propulsors to generate thrust, such as NASA’s X-57 Maxwell (also known as SCEPTOR) [90] and SUBsonic Single Aft eNginE (SUSAN) [7] concepts. Furthermore, these concepts employ disparately sized propulsors, contradicting another implicit assumption that all propulsors are equally sized. A regulatory gap analysis [91] has already identified the lack of OEI requirements for distributed propulsion concepts, as well as a broader need to redefine OEI scenarios and nominal/off-nominal operating conditions.

Inspired by rendering distributed propulsion concepts airworthy, design and certification of these configurations has become an emergent research area, with a particular focus on engine inoperative conditions.

1.4.1 Existing Certification Regulation Adaptations

Prior research has focused on identifying the critical engine in an advanced propulsion system, modifying existing OEI conditions, and designing advanced concepts with distributed propulsion systems using current regulations.

Under OEI conditions, the “critical engine” is assumed to fail [89]. Prior research suggests identifying the critical engine as one that: 1) produces the largest thrust asymmetry upon

failing; 2) provides the largest amount of power at low altitudes; or 3) powers other propulsors in the propulsion system aside from itself [92]. For distributed propulsion systems, the propulsor closest to the wingtip generates the largest yaw moment, rendering it as the critical engine [93–95]. However, on NASA’s SUSAN concept [7], minimum propulsive power requirements are defined for aft turbofan engine *and* distributed propulsor failures during takeoff and cruise [51]. Although the aft turbofan engine produces the most power, the most outboard propulsor in the DEP system generates the largest yaw moment upon failure. Additional exploration is needed to identify SUSAN’s critical engine for certification.

Aside from selecting the critical engine, there have also been proposals to modify OEI conditions for future aircraft concepts. One approach involved designating larger, discrete propulsors as a “primary” propulsion system while considering the distributed propulsion system as a “secondary” one [96]. This fails to accommodate concepts that use distributed propulsion systems as their sole propulsive power source. Another approach assumed that any component failure in the propulsion system represents an engine inoperative condition [81, 97], capturing failure modes that could result in a loss of multiple propulsors. In this case, however, there may not be a “critical engine,” but rather a “critical loss of thrust” instead [98], which could involve multiple propulsors within one distributed propulsor array failing [99].

Since modified OEI regulations have not yet been adopted, two approaches were previously utilized to define climb requirements for distributed propulsion concepts. One option is to assume that the required climb gradients would match those for a 4-engine conventional aircraft [9, 97, 100], reflecting the most stringent requirements in Part 25 regulations. Another option was to develop an extrapolation function dependent on the number of propulsors [101]. However, the extrapolation function developed requires a steeper climb gradient as more engines are added. This may be unreasonable for distributed propulsion systems because climb performance degrades more gracefully as a result of many, smaller propulsors installed on the aircraft [102], possibly resulting in better off-nominal performance than their conventional counterparts.

OEI conditions are also utilized during aircraft sizing to ensure that the propulsion system provides sufficient power during an engine failure. Traditionally, each engine’s size is scaled by a correction factor of $n/(n-1)$, where n is the number of engines installed on the aircraft [103]. Examples of these scale factors for 2-, 3-, and 4-engine configurations are listed in Tab. 1.3, necessitating each propulsor to be oversized by 33 to 100%. However, this correction factor assumes that the engines are equally sized, which is invalid for advanced propulsion system architectures with disparately sized propulsors [7, 104].

Table 1.3: Engine inoperative correction factors for conventional aircraft configurations.

Number of Engines	Correction Factor
2	2.00
3	1.50
4	1.33

1.5 System Architecture Synthesis

System architecture synthesis includes the design of the system architecture topology (components and corresponding connections) and the components’ sizes. Most system architecture synthesis techniques perform a “top-down” approach, selecting an architecture topology and then sizing the components in a hierarchical design process [105]. The primary challenge associated with system architecture synthesis is managing the combinatorial design space imposed by such problems, possibly requiring tens of thousands of system architectures to be filtered and down-selected [106, 107]. To find the most promising system architectures, significant computational overhead is necessary to perform first-order, physics-based analyses on a vast number of candidate system architectures, possibly taking hours or days to perform such simulations, even with parallel computing capabilities [108]. Furthermore, such large-scale problems require optimal designs to be found automatically, with minimal manual pre- and post-processing efforts.

Multiple computational approaches are available to synthesize system architectures automatically, as described in Sections 1.5.1 through 1.5.6 next. Each approach either explores the entire design space exhaustively or searches through (a subset) of the design space strategically based on heuristics (user-defined rules to identify optimal designs).

1.5.1 Graph-Based Techniques

As previously introduced in Section 1.3.2, graphs can be used to represent propulsion system architectures by defining the vertices and edges to be the propulsion system components and power flow paths, respectively. For general system architecture synthesis studies, a graph’s vertices and edges can be more broadly abstracted as a component in the system architecture and energy/matter flows, respectively. Depending on the application, the graphs may be directed (as previously described), indicating that energy/matter flows must follow a specific path, or undirected, indicating that energy/matter flows may freely travel between connected

components in either direction. Although nearly all the literature reviewed in this section includes aircraft propulsion systems, automotive drivetrains, or electromechanical systems, graph-based approaches have also been utilized in the past for water/distribution [109–111], transit [112], and circuit [113] network designs. Two graph-based architecture synthesis techniques include architecture enumeration, performing a global search of candidate system architectures, and utilizing numerical optimization on a few *fixed* system architectures to select the optimal component sizes that maximize system-level performance.

1.5.1.1 Architecture Enumeration

During enumeration, all combinations of possible component arrangements are explored, followed by filtering out infeasible ones via constraints on the overall physical system design. The physical constraints are typically called “network structure constraints” [114], as the resultant system architecture is a network of components transmitting energy/matter between them. Examples of network structure constraints include limits on: 1) the number of components in the system architecture; 2) the number of connections to/from components; and 3) the number of paths within a system architecture. Additional constraints are necessary to ensure that all connections are physically compatible, ensuring that the output of one component matches the required input of another.

During enumeration, the design space is fully explored, providing insights about the best performing architectures at the expense of computational overhead to filter out physically intractable system architectures. However, methods to enumerate and filter architectures more rapidly have been developed, such as those utilizing graphs with “perfect matchings,” ensuring that all components have the correct number of input/output connections [107]. Enhancements were later developed to include additional network structure constraints, enabling simultaneous enumeration and filtering of infeasible designs [115]. Incorporating network structure constraints into the enumeration process significantly reduces the design space, returning hundreds of thousands of candidate architectures as opposed to millions and enabling vast searches of large-scale design spaces with reduced computational efforts. The aforementioned enumeration strategy is, however, limited in designing redundant system architectures, as the number of allowable input/output connections must be redefined for varying levels of redundancy. Furthermore, identifying the optimal system architecture still requires numerical analysis and remains computationally prohibitive, even with the reduced design space [107].

For non-redundant systems, combined architecture enumeration and filtering with network

structure constraints has recently been applied to design thermal management systems, identifying promising architectures that could be integrated into future aircraft concepts [106, 108, 116, 117]. After identifying feasible system architectures that satisfy the network structure constraints, the architectures are defined in physics-based modeling tools to predict system-level performance. While architecture enumeration for thermal management remains efficient, taking approximately one hour to generate 10,000 candidate system architectures, the physics-based analysis remains computationally expensive, requiring 20 hours to evaluate all system architectures. Of the 10,000 system architectures investigated, only 23 of them met all the performance/operational constraints defined [116].

Although first-order physics models are used during architecture evaluation and require minimal computational overhead, challenges emerge while sizing the components to meet performance/operational constraints. In prior studies, the component sizes are selected from a list of available ones, requiring combinatorial searches for each candidate system architecture [106, 108, 116]. While the architecture generation and filtering process can be performed rapidly with efficient methods [116], the combinatorial searches within the physics-based analysis remains to be time consuming, even if parallelized using high-performance computing. For example, during the design of a thermal management system 500,000 system architectures were enumerated in under one minute, but required 9.5 days to perform physics-based simulations on 32,000 of them, with only 5,500 running successfully (while utilizing parallel computing resources) [108]. An alternative architecture enumeration approach involves fixing the component sizes and performing a combinatorial search on their arrangement [117], but becomes computationally intractable for large-scale systems containing tens of components.

As an alternative to traditional graph theory approaches, Bond Graphs may also be utilized and are specifically useful for modeling power flows in a dynamical system [118, 119], such as hybrid vehicle drivetrains with planetary gears. Aside from enumerating the component connections within the system topology, an additional enumeration occurs assigning power transmission components as sun, ring, or carrier gears within the planetary gear drive(s). To prevent the combinatorial design space from growing too rapidly, only drivetrain configurations with up to two planetary gears were considered, resulting in up to 3,420 system architectures. To identify an optimal system architecture, a heuristic search was utilized, ultimately reducing the fuel burn by 8% relative to an automotive vehicle on the market already [118]. The heuristic search circumvents the need to perform a physics-based analysis on all enumerated configurations, making it less computationally expensive to find an optimum system architecture. However, if the heuristics are not rigorously developed, the

true globally optimal system architecture may not be identified.

Graph-based design methods have also been utilized for designing space system architectures. Rather than abstracting each component as a vertex and its energy/matter transfers as edges, the vertices represent different states/phases in a mission and the edges represent allowable mechanisms/maneuvers to transfer between those states [120]. One advantage of the method utilized is that multiple graphs may be linked together, indicating that different parts of the space vehicle may perform different functions throughout the mission, particularly during stage separations. For the synthesis of (electro)mechanical systems like aircraft propulsion systems and hybrid vehicle drivetrains, such an approach is out-of-scope, but is worthy of mentioning for completeness of the literature review.

1.5.1.2 Fixed System Architecture Optimization

Rather than utilizing architecture enumeration algorithms and combinatorial searches on the system architectures and their component sizes, respectively, subject matter experts may identify a few candidate system architectures and utilize numerical optimization to select the component sizes that maximizes system-level performance. To evaluate candidate system architectures efficiently, physics-based models are embedded in their graph-based representations, enabling detailed analyses while remaining scalable and computationally tractable for conservation-based complex system architectures [121], further lending itself to numerical optimization.

Since the system architecture is constrained to a few candidates, hierarchical numerical optimizations are performed to select the best system size and, optionally, the operational strategy to maximize system-level performance. Examples of such approaches include optimizing the component sizes and controller gains in a hybrid energy storage system [122], as well as the design of a thermal management system and control system trajectories under steady-state and transient analyses [123]. While prior enumeration approaches utilize Adjacency Matrices from Graph Theory, defining component connections in a vertex-centric representation, the dynamical systems previously described utilize Incidence Matrices, defining component connections in an edge-centric representation.

A similar graph-based approach is utilized to design both a cooling system and an electric vehicle powertrain. An edge-centric representation enables conversation-based system dynamics to be modeled, and additional “design matrices” permit changes to the system architecture being analyzed – a step towards combined architecture synthesis and component sizing without enumeration techniques [124]. Furthermore, the design variables permitting

architectural changes are modeled as RAMP functions and governed as continuous variables, enabling gradient-based optimization to select the optimum system architecture. However, part of the system architecture remains *fixed*. While this simplification eliminates the need for network structure constraints, the design space is implicitly constrained, limiting its broader applicability to generative design.

Overall, using expert knowledge to fix part or all of the system architecture enables numerical optimization to be performed, identifying optimal component sizes and/or operational/control strategies. Consequently, the globally optimal system architecture may not be found due to the implicit constraints on the overall system architecture.

1.5.1.3 Graph Grammars

Graph grammars provide rule-based methods to transform graphs from functional-based system architecture representations to physical-based system architecture representations, or convert between different physical system architectures [125], enabling various system architectures to be synthesized. Conversion between architectures typically occurs randomly, enabling broad design space exploration [126]. More recent system architecture synthesis methods utilize a rule-based methodology to enumerate a system architecture’s required behaviors and then possible components to satisfy those behaviors [127], even though graph grammars are not mentioned.

Depending on the graph grammar and architecture synthesis algorithms implemented, the search can be either exhaustive, exploring all possible system architectures, or random, exploring a small fraction of system architectures across the vast design space. Regardless of the approach utilized, physics-based models are still necessary to estimate system-level performance, and may require component sizes to be guessed/optimized outside of the graph grammar implementation.

1.5.2 Multidisciplinary Design Optimization

Multidisciplinary Design Optimization (MDO) approaches for system architecture synthesis involve solving a hierarchical optimization problem to select the system architecture, component sizes, and, optionally, the operational/control strategy. Hierarchical optimization problems can be solved utilizing either monolithic or distributed MDO architectures [65], returning the optimal design variables and states consistent with the prescribed governing equations of the system under study. In the context of system architecture optimization, the

design variables and system state variables are represented by the architectural variables governing the system architecture design and the sizing/control variables governing the component sizes and operation, respectively.

In a monolithic MDO architecture, the optimization problem is solved utilizing a coupled system of governing equations, eliminating the need for coupling variables and returning physically compatible system architectures if the optimization ends prematurely. An additional solver is embedded inside of the optimizer to solve for the state variables that are not controlled by the optimizer [65]. Therefore, monolithic MDO architectures provide a simpler implementation at the cost of additional computational time to solve for the state variables during each optimization iteration.

Monolithic MDO architectures for system architecture synthesis typically combine a stochastic optimizer to select a candidate system architecture for analysis and embed a gradient-based optimizer as the solver to size the components and operational/control strategy if desired. While optimizing a hybrid electric propulsion system architecture and its component sizes, a multi-objective genetic algorithm was utilized as the optimizer with a Sequential Least-Squares Quadratic Program (SLSQP) sizing the propulsion system components [128]. A generic hybrid electric propulsion system is also modeled and simulated in a Model-Based Systems Engineering tool, providing additional modularity and extensibility. However, additional modeling efforts are necessary prior to running an optimization and may not be reusable for other system architecture synthesis problems. Another study synthesizing gear trains utilized a heuristic search to select candidate system architectures while a gradient-based optimization identified each gear’s position in the overall system [129]. In the gear train synthesis problem, monotonicity is exploited in the heuristic search, as the gear train is optimized for minimum mass and inefficiency, suggesting that the fewest number of gears be included. Other problems, such as the propulsion system architecture optimization cannot take advantage of specific trends in the objective function, requiring more general approaches to select an optimal gear train.

In a distributed MDO architecture, the optimization problem is decomposed into smaller disciplinary optimization problems that can be solved in parallel, reducing the computational time required to solve the problem, but requires coupling variables between optimization problems to ensure a physically consistent solution. Additionally, each disciplinary optimization problem must be mathematically equivalent to the original optimization problem to guarantee convergence to the same solution as if it were solved using a monolithic MDO architecture [65].

One distributed MDO architecture is Analytical Target Cascading (ATC); disciplinary subproblems were solved for hybrid electric vehicle configuration design (powertrain system architecture), powertrain component sizing, and power management strategy via optimal control [130]. Decomposing the problem into its discrete variables (configuration design) and continuous variables (component sizing and power management strategy) simplifies the optimization process, particularly for mixed variable optimization, rather than performing a computationally expensive “all-in-one” approach. For simple and complex configurations, the overall optimization problem converged in 8 to 21 hours and 12 to 27 hours, respectively, making it a reasonable candidate for conceptual design studies.

While the previously described MDO architectures utilize a combination of stochastic and gradient-based optimizers, there is an increasingly large body of work on utilizing surrogate-based optimization to identify optimal system architectures. The surrogate-based optimization remains rooted in a graph-based representation of the system architecture, allowing different system architectures to be considered while maintaining compatibility between components interfacing with each other [131, 132]. Relative to stochastic optimization techniques, such as Genetic Algorithm, surrogate-based optimization problems with discrete and/or categorical variables performed similarly for problems with “hidden constraints” (resulting in a physics-based analysis failing for an unknown reason), but perform better if no such constraints are present [133]. In another study, a Bayesian Optimization technique was utilized to identify the optimal system architecture, but is only successful if the selected system architectures are feasible and converge, and required new methods to be developed for identifying viable system architectures [134].

Additionally, stochastic optimizers may be able to leverage the heuristics inherent to their functionality to avoid such regions of the design space with hidden constraints, ensuring that more feasible system architectures are evaluated during the optimization routine [135]. However, such algorithms include trial-and-error or random approaches to convert infeasible system architectures to feasible ones, and can require significant wall time to ensure all architectural constraints are satisfied [136].

Besides solving MDO problems, another adjacent branch of work involves dynamically defining and solving MDO problems for system architecture synthesis problems [137]. Depending on the system architecture being synthesized and disciplines required for evaluation, four MDO architectures may be developed: single static (develop a single MDO problem that is always solved); multi-static (develop multiple MDO problems and solve one depending on the system architecture); single dynamic (dynamically define the disciplines of an MDO

problem as a function of the system architecture); and on-demand (creating a unique MDO problem for each system architecture evaluated). For hybrid electric vehicle optimization, disciplinary analyses may remain the same, rendering a single static problem appropriate. For configurations with significant architecture changes (tube-and-wing vs BWB) where physics-based models may be different, a multi-static or single dynamic MDO problem may be more useful.

1.5.3 Constraint Programming

Constraint Programming is utilized to find a set of *feasible* design variables that satisfies a prescribed set of constraints. While a solution from a Constraint Program (or Constraint Satisfaction Problem) satisfies the system-level constraints, the *optimal* solution is not necessarily returned. In the context of system architecture synthesis, Constraint Programs can either identify feasible system architectures for further analysis [125] or select a feasible set of design variables for a given system architecture [126]. Depending on the solver used, the candidate system architectures returned may be non-isomorphic [125], ensuring that each one is unique and does not have similar component connections to another candidate system architecture. Thus, Constraint Programming is useful for finding feasible system architectures to evaluate further in a physics-based modeling tool but does not estimate system-level performance itself. Although system-level performance cannot be distinguished between the feasible system architectures identified, Constraint Programming provides a means of rapidly synthesizing baseline system architectures that could be analyzed or optimized further.

One advantage of a Constraint Program is its ability to rigorously enforce mathematical constraints, making it more attractive than enumerative or evolutionary-inspired optimization approaches that possibly explore infeasible system architectures [138]. However, the constraints defined for one system architecture may not be reusable, requiring redefinition for each analysis and limiting its scalability to broader system architecture synthesis problems.

Aside from Constraint Programs, a similar class of problems for finding *feasible* system architectures are Boolean Satisfiability Problems. Rather than finding real or integer design variables that satisfy all constraints, combinations of Boolean variables are identified to ensure all constraints are TRUE [139]. Since Boolean variables are utilized, the approach works well for identifying feasible system architectures and enables enumeration in a more mathematically rigorous manner than graph-based approaches. Consequently, feasible system architectures are synthesized with no knowledge of its system-level performance or required component sizes.

1.5.4 Branch-and-Bound Methods

Branch-and-Bound (BNB) Methods are commonly utilized to solve (mixed-)integer optimization problems, enabling a subset of the design variables to take integer values, and can be easily evaluated by commercial solvers such as Gurobi ² and CPLEX ³. Originally, the problem was developed for integer programming problems [140], providing a more efficient method to search a discrete design space than explicitly enumerating all possible combinations of design variables. BNB Methods have also been utilized to solve mixed-integer nonlinear programming (MINLP) optimization problems by decomposing the problem into two sub-problems; BNB methods identify the optimal discrete variables and a gradient-based optimizer selects the optimal continuous variables [141–143]. More recently, BNB has been combined with other stochastic optimization methods such as Simulated Annealing or Genetic Algorithm, serving as a method to exploit a specific region of the design space that appears to contain a promising solution [144].

In the BNB Method, a discrete design space is searched by “branching,” identifying candidate integer variables that could lead to an optimum solution. For binary variables, the branching approach constraints the design variables to be 0 or 1 [65]. For integer variables, the problem may first be relaxed by converting the integer variables into continuous ones and solving the problem via a continuous optimization method. For any design variables that do not take integer values, additional constraints are imposed on the problem requiring the design variable to be an integer value, and is re-solved using a continuous optimization method. The process continues until all integer design variables are assigned integer values [65]. Additional branching methods are available if only one or two of the integer values are allowed to be nonzero at any given time [145]. After identifying possible feasible solutions from the branches, they are “bounded,” ensuring that those with objective function values larger than the current global minimum are no longer searched [146].

An example of a systems architecting problem solved using integer optimization methods is a Redundancy Allocation Problem, and involves minimizing the cost of a system subject to reliability constraints [147]. For example, design spaces of 10^{13} and 10^{15} possible flight control system architectures were searched in 7 and 25 minutes, requiring 740 and 2,200 architectures to be evaluated, respectively [148]. The flight control system architecting problem highlights an advantage of BNB methods: efficiently searching a large combinatorial design space and providing rapid convergence to an optimum solution.

²<https://www.gurobi.com/>

³<https://www.ibm.com/products/ilog-cplex-optimization-studio>

One limitation, of BNB Methods is that the optimization problem must be relaxed on each branch, converting integer variables to continuous ones. On each branch, constraints are imposed to find the best integer value for each design variable [65]. In Redundancy Allocation Problems, the integer design variables represent discrete, numerical data – the number (or types) of components to be included in the system architecture. However, in graph-based systems architecting approaches, the binary variables represent if a component is included in the system architecture or not. Since any change to one variable results in a different system architecture, the conglomeration of all binary variables actually represents categorical data – the system architecture synthesized – hindering the effectiveness of branching methods that rely on numerical data. For categorical data, solving with a BNB Method would be similar to an exhaustive search, providing no computational benefit.

1.5.5 Integrated Multidisciplinary Design Optimization and Model-Based Systems Engineering Methods

Model-Based Systems Engineering (MBSE) has emerged as a digital means to track complex system design activities, maintain traceability between design decisions, and support requirement verification [149]. A centralized system model acts as a global repository, storing all information about the system architecture, its design requirements, and analyses performed [150], thus supporting systems design practices. A major advantage of MBSE is that a consistent set of parameters defines the system architecture – if a parameter or component is changed in one part of the model, it is modified across the entire system model. This ensures that all systems engineers use the same parameters, thus eliminating the need to manually update all design documents. However, MBSE currently lacks the capability to provide analytical and numerical methods [151], requiring connections to external analysis tools.

In contrast, MDO has been a longstanding facet of engineering design, employing computational models to simulate or optimize a system’s behavior and performance. Such models are used by disciplinary engineers to evaluate a component’s performance while interacting with other components in a subsystem. Furthermore, a component can be optimized for a specific objective and constraints, ensuring sufficient compatibility and performance once integrated into the overall system architecture. However, MDO tools only perform the analyses and are not intended for rapid reconfiguration and integration into different engineering workflows [152].

Though MBSE and MDO tools may seem disparate, their purposes are complementary.

MBSE excels at representing the system architecture in an understandable manner, capturing design requirements, and easily tracing design decisions throughout the design cycle. MDO excels at analyzing specific system architectures and identifying the optimal component design. Evidently, there is a synergy between these two tools, and combining their capabilities may help synthesize system architectures that rely on multidisciplinary teams (possibly distributed across multiple geographic locations) to design different parts of the system architecture. However, this becomes increasingly difficult as the size of the system architecture grows and more disciplinary engineers become involved [153].

Automated couplings between MBSE and MDO tools for system architecture synthesis have emerged in the literature, particularly as a result of the AGILE 3.0 and 4.0 projects funded by the European Union [149, 154, 155]. At first, these projects focused on integrating computational tools into MDO frameworks, and have since been broadened to include MBSE and MDO tool couplings. The tools are connected via a two-step process: 1) defining a data schema to facilitate information transfer; and 2) establishing (possibly automated) workflows to run analyses and pass information between the tools.

In the United States, the Air Force Research Laboratory’s (AFRL’s) REACT Project, initiated in 2019, aims to achieve similar goals with a stronger emphasis on coupling disparate computational tools for MDO while also maintaining modularity to integrate MBSE tools into a design workflow [156, 157]. The majority of REACT activities have been conducted internally and disseminated primarily within AFRL and its industrial partners. Boeing and NASA participated in technical interchange meetings with AFRL regarding REACT starting in 2022, while broader details of the program became available to the open literature through a 2026 publication [156]. Separately, one paper previously published by AFRL (not under the REACT project) introduced *Philote*, a communication standard to facilitate information transfer between heterogeneous MDO tools [158].

While existing literature demonstrating automated workflow capabilities predominantly focuses on generating mechanisms to facilitate data transfer and run computational analyses, there are two other notable works pertaining to data schema definition. The Common Parametric Aircraft Configuration Schema (CPACS) [159] was developed to establish a hierarchical definition of the aircraft and its subsystems. Although this definition is aerospace-specific, a similar data structure could be developed to represent a more general system architecture, its subsystems, and components. For MDO workflows, the Common MDO Workflow Schema (CMDOWS) [160] defines the optimization problem, design variables, constraints, and connections between disciplines.

Prior schema development established a foundation for developing interconnected workflows between MBSE and MDO tools. In the literature surveyed, the MBSE tool is commonly treated as a central repository, storing the data necessary for the workflow [151, 161–164]. However, other works utilize a central data schema to store the information [165, 166], with each tool accessing/updating the parameters individually. These two approaches are defined as “strong” and “weak” couplings, respectively [167], impacting the design and data transfer workflows.

1.5.5.1 Strong Coupling

A strong coupling centralizes the design process around MBSE, requiring both the system architecture and multidisciplinary workflows to be modeled. For example, a detailed MBSE model of an AGILE MDO system was developed by Ciampa et al. [161]. Another approach involved constructing Requirement-Functional-Logical-Physical (RFLP) frameworks for the system architecture (a “product RFLP”) and the workflow (a “process RFLP”) [168].

Other workflow models were developed using MBSE’s parametric or activity diagrams in conjunction with the system model [151, 163], permitting some analyses to be executed within the MBSE environment. However, it is also possible to interface external tools with MBSE, as done by Aiello et al. [162, 164]. They used Ttool [169] to extract information from a requirement diagram in MBSE and pass it to an MDO tool.

Modeling the MBSE-MDO workflow with a strong coupling has two key advantages. First, all product and design information resides in one, centralized location, ensuring that all engineers can access the most up-to-date data and use consistent values in all analyses, thus eliminating the need to transfer information between groups. Second, some analyses can be executed automatically using MBSE diagrams or by interfacing with external tools, further centralizing the workflow within MBSE and reducing the time to define and execute the workflow.

However, there are some weaknesses with this approach. First, the MBSE tool retains the “authoritative source of truth” (ASOT), requiring disciplinary engineers to access the latest design information by interfacing with MBSE. This contradicts the typical workflow paradigm in which MDO stores the subsystem- and component-level requirements and parameters. Furthermore, additional costs may be incurred for training all engineers in MBSE or purchasing enough software licenses to support the entire organization. Second, the MBSE models may be product- or workflow-specific, thus limiting its generality and applicability to other product/workflow designs within an organization. This undercuts the

previous advantage pertaining to the time reduction for defining and executing a workflow. Third, the MBSE model may become increasingly complex as more design disciplines are integrated into the analysis, limiting its scalability to more comprehensive design problems.

1.5.5.2 Weak Coupling

A weak coupling unifies the design process around a central data schema and emphasizes integrating all disciplinary models into a comprehensive analysis framework. Seamless data transfer is enabled by developing mappings between MBSE/MDO tools and a central data schema [165]. This approach has been exemplified in automated requirement verification frameworks [166] and broader system architecture design tools [170].

There are two key strengths to this approach. First, the data schema is accessible to all engineers without requiring any MBSE software, thus cutting costs to establish more fully integrated workflows within an engineering organization. Second, multiple data schema instances can be used to represent different system architecture configurations, allowing an organization to assess multiple designs simultaneously.

Despite the strengths, there are also two weaknesses. First, multiple versions of the data schema may exist and could lead to conflicting design changes between disciplinary teams within an organization. One potential solution is to store the data schema in a version-controlled environment, facilitating a pathway to reconcile such conflicts. Second, the data schema may become prohibitively large, limiting its effectiveness to clearly and quickly communicate data about a system architecture.

1.5.6 Alternative Approaches

There are multiple alternative approaches described in the literature that do not necessarily correspond to any of the high-level approaches previously described. A few are described here for completeness of the literature review.

Agent-based approaches involve defining different “agents” to configure and modify the system architecture, as well as manage the system architecting process [171]. Agents iteratively identify initial candidate designs and modify them, gradually filtering out sub-optimal ones. While this approach is similar to product design in disciplinary teams and mimics processes in evolutionary-based optimization algorithms, the approach is more heuristic-based, requiring rules to be developed for specific system architecture designs, thus limiting its applicability and reuse in other system architecting applications.

Another system architecture synthesis approach solves a gradient-based optimization problem and then uses a weighting metric to find similarly sized components from a catalog, enabling a more intelligent search of the discrete design space [172]. Rather than performing an exhaustive search of the combinatorial design space, the discrete design variables are converted to continuous ones, enabling gradient-based methods to be utilized to more quickly find an optimum. While the approach enables rapid optimization, it was demonstrated on a fixed system architecture and is likely less generalizable to system architectures with varying topologies.

Machine learning and artificial intelligence has also been used to synthesize advanced system architectures. One example includes identifying physically tractable thermal management system architectures that can be simulated in physics-based tools and predicting the ones that will perform optimally [173]. Using 5% of an available dataset, physically tractable thermal management system architectures can be identified with 97% accuracy, while 45% of the available data can be used to identify most of the optimally-performing thermal management system architectures.

Chapter 2

Literature Gaps and Research Objectives

The primary observations made from the literature review in Chapter 1 are further expanded upon, identifying critical gaps in the literature that must be addressed to advance state-of-the-art system architecture synthesis methods and configuration-agnostic computational tools for aircraft conceptual design and analysis. The identified literature gaps serve as a baseline for formulating the overarching research questions and hypotheses that drive the work in this thesis.

2.1 Literature Gaps

2.1.1 Nascent Advanced Aircraft Designs

Both NASA’s SUSAN and Elysian’s E9X leverage distributed electric propulsion systems to partially or fully propel the aircraft. While conceptual design studies have focused on exploring sensitivities of system-level performance with respect to key technological performance parameters and developing enabling technologies to improve system-level performance [42, 46, 47, 54], there has been minimal to no focus on exploring the safety-based ramifications of installing an advanced propulsion system on the airframe.

Prior system design studies on SUSAN assumed that redundancy in the distributed electric propulsion system renders the aircraft as “safe” to fly, particularly since single point failures were mitigated [50]. As a result, SUSAN’s subsystem design studies only considered performance and weight objectives without quantitatively analyzing system safety. However, such analysis is necessary for certification, along with identifying the propulsion system’s primary failure modes and conducting a reliability analysis to ensure that it meets the necessary system-level safety requirements.

Similarly, the conceptual design studies on the Elysian E9X account for one-engine inoperative conditions during sizing [9], but do not consider multi-propulsor failure modes. Additionally, there are no considerations of utilizing multiple batteries to power the distributed electric propulsion system, providing a level of redundancy in case a battery fire or thermal runaway occurs.

Due to the variety of failure modes enabled by advanced propulsion systems architectures, system safety analyses must be performed during conceptual design, capturing tradeoffs between system-level performance and safety. As redundant connections are included in the propulsion system architecture to satisfy the safety requirements, a weight penalty is incurred, hampering system-level performance [174]. Additionally, trade studies between various propulsion system failure modes can be conducted, indicating the sensitivity of an aircraft’s system-level performance with respect to its primary propulsion system failure mode.

2.1.2 Computational Tools for Advanced Aircraft Design

There are three limitations preventing existing computational tools from being easily applied to design future aircraft concepts with advanced propulsion systems. First, the tools require the aircraft’s geometry to be known prior to running the analysis, implying that design decisions have already been made about the configuration [55, 58, 66, 80]. Thus, the design space is implicitly constrained, possibly preventing the optimal system architecture from being identified. Furthermore, multiple configurations are explored during early-phase conceptual design and an aircraft’s geometry may not be fully known, limiting the effectiveness of component-based weight estimations and geometry-based aerodynamics analyses. A truly configuration-agnostic design tool must employ simplified aerodynamic performance and system-level weight estimations to predict an aircraft’s size and performance.

Second, very few tools accommodate a plethora of novel propulsion system designs and do so in a generalized computational framework. While some propulsion system modeling frameworks provide generalized hybrid-electric powertrains, the possible power flow paths remain fixed and are governed by different control variables depending on the model considered [58]. Additionally, some generalized hybrid-electric propulsion system models only permit kerosene fuel and batteries to serve as energy sources, failing to include other novel ones such as hydrogen [80]. Thus, a configuration-agnostic conceptual aircraft design tool must flexibly permit any type of energy source to be included in an aircraft’s propulsion system, enabling broad design space exploration studies without implicit constraints on the

propulsion system design.

Lastly, existing aircraft design tools that interface with high-fidelity MDO codes are not suitable for making initial design decisions on clean-sheet aircraft designs [55, 58]. Using detailed analysis and optimization modules during early-phase conceptual design prevents rapid identification of the most critical design parameters that will impact the final design and require significant computational resources. With many novel aircraft configurations and propulsion architectures being developed, there is a need for a conceptual aircraft design and analysis tool that can rapidly generate aircraft designs and allow for early-phase design space exploration while exploring different operational strategies for electrified aircraft concepts. While exploring different aircraft concepts, sensitivities of the configuration’s system-level performance with respect to key technological performance parameters and operational strategies may be estimated, identifying areas that more detailed analyses are necessary to reduce uncertainty in the overall aircraft design. Thus, an early-phase conceptual aircraft design tool must provide rapid analysis capabilities to identify promising aircraft configurations and corresponding operational strategies.

The three gaps identified suggest that there is a need for a parametric aircraft sizing tool with a generalized propulsion system analysis framework, enabling rapid configuration-agnostic advanced aircraft design. The parametric aircraft sizing tool must enable system-level analyses and assessments of the interdependencies between technologies integrated onboard the aircraft, serving to identify promising technologies and/or operational strategies that should be further analyzed and considered in the preliminary and detailed aircraft design phases. Additionally, the design tool must be able to interface with MDO frameworks, supporting system- and subsystem-level design optimization and eliminating uncertainty in the overall aircraft design for the key technological performance parameters that heavily impact system-level performance.

2.1.3 Propulsion System Analysis Frameworks

After reviewing current state-of-the-art propulsion system analysis frameworks, two key gaps were identified. First, there is a need for a flexible and robust propulsion system architecture framework that simply models propulsion systems with different components/arrangements, especially redundant systems that are critical to advanced aircraft propulsion systems, particularly distributed propulsion. Existing propulsion system analysis methods that define the energy conservation equations and power splits via a linear system of equations requires manual intervention, limiting its adoption and deployability in design optimization or systems

architecting frameworks [81]. Presently available graph-based methods have greater potential to define propulsion systems more generally, but are limited by the number of components that can be connected in series, limiting scalability and preventing propulsion system architectures three connected power transmitting components from being analyzed [8].

Second, a propulsion system analysis framework must provide connections to other aircraft subsystems, particularly as advanced propulsion systems necessitate tighter integration with the airframe. For example, electrified propulsion systems generate significantly more waste heat than conventional propulsion systems, requiring thermal management systems to be integrated into the system-level design. Additionally, power off-takes from the gas turbine engines are utilized to power the aircraft’s onboard navigation systems and amenities for passengers, such as lavatories, galleys, and entertainment systems. No existing propulsion system analysis frameworks accommodate embedded power off-take modeling [8, 83, 84], creating a significant gap in the literature.

Both of the identified gaps suggest that the field lacks a flexible and scalable propulsion system analysis framework capable of representing diverse electrified propulsion architectures and their operational modes within a unified structure. Thus, one must be developed to systematically define and encode a propulsion system architecture, its operational modes, and additional power flows. Additionally, power flows must be simulated rapidly, enabling integration within a configuration-agnostic conceptual aircraft design tool.

2.1.4 Generalized Flight Performance Requirements

The literature review exposed three critical gaps for designing advanced aircraft concepts with distributed propulsion systems, particularly in how their failure modes and corresponding flight performance requirements may be defined. First, there is uncertainty surrounding how OEI scenarios may be defined for distributed propulsion concepts, possibly encompassing multi-propulsor failures and additional failure modes at other power-transmitting components in the propulsion system architecture [96, 98, 99]. Flexible design methods are needed to explore multiple propulsion system failure modes and assess their implications on the resultant aircraft design.

Second, existing OEI performance requirements have not changed in over 60 years [38], and fail to accommodate future aircraft concepts leveraging distributed propulsion systems. The presently published climb performance requirements are defined as a function of the number of engines installed (a discrete variable) and implicitly assume that all propulsors are equally sized, prohibiting them from being broadly applied to future aircraft concepts with possibly

heterogeneous propulsion systems [89]. Prior studies designing aircraft with distributed propulsion systems utilize the published climb gradients from FAA Part 25 regulations for 4-engine aircraft or develop extrapolation functions based on the number of propulsors installed [9, 97, 100]. The former approach utilizes the climb performance requirements defined within the bounds of existing FAA requirements by imposing the most stringent performance requirements available, neglecting the failure mode severity while developing the projection to a climb performance requirement. The latter approach prescribes a larger climb gradient as the number of propulsors installed on the aircraft increases, expanding existing parameterizations for an aircraft with n propulsors, but requires the aircraft to perform more aggressively relative to conventional aircraft previously certified under FAA Part 25 regulations [101]. The strengths and shortcomings of both approaches could be combined to develop climb performance requirements in a configuration-agnostic manner while anchoring it within the bounds of the presently published published ones, enabling all transport aircraft to be certified under a unified set of regulations.

Third, novel approaches are needed to scale a propulsion system’s size for engine inoperative conditions. Previous methods assume each propulsor is equally sized, resulting in a correction factor that depends on the number of engines installed [103], but is not applicable to future aircraft concepts utilizing disparately sized propulsors. Again, configuration-agnostic methods are needed to size future aircraft concepts and their propulsion systems, independent of the number of propulsors installed on the airframe.

The identified shortcomings indicate that generalized flight performance requirements and engine sizing correction factors are necessary to accommodate future aircraft concepts with advanced propulsion systems and their corresponding failure modes. Additionally, the generalized flight performance requirements must be seamlessly integrated into a configuration-agnostic aircraft design tool, enabling sensitivities of the system-level performance with respect to different propulsion system failure modes to be predicted.

2.1.5 System Architecture Synthesis

Multiple gaps and shortcomings were identified with present state-of-the-art system architecture synthesis methods. First, graph enumeration and other combinatorial searches exhaustively explore expansive design spaces, enabling the globally optimal system architecture to be identified but at a significant computational cost. Since a vast majority of possible system architectures synthesized are physically intractable or fail to satisfy other architectural or operational constraints, computational resources are utilized inefficiently,

evaluating more infeasible system architectures than feasible ones [108, 116]. Additionally, combinatorial searches, particularly for component design, implicitly constrain the design space, preventing optimal system architectures from being synthesized. While efficient graph enumeration algorithms have been developed [107, 115], physics-based modeling is necessary to identify optimal system architectures, necessitating additional computational resources and slowing the design space exploration. Thus, there is a need to more strategically select and analyze candidate system architectures while broadly exploring the design space to identify optimally performing system architectures.

Second, MDO and numerical optimization can save computational resources by analyzing a limited number of system architectures, but may not converge on the globally optimal system architecture. BNB Methods are not effective for problems with categorical data, requiring stochastic optimization methods to be utilized. Since the design space may be multi-modal, heuristics are necessary to identify the optimal system architecture [129]. While the heuristics limit the number of system architectures analyzed, improving computational efficiency, they may not identify the globally optimal system architecture if its search algorithms are developed poorly. Additionally, most stochastic optimization methods are unable to accommodate constraints, requiring heuristics to filter out infeasible system architectures. For some system architecture synthesis studies, MDO was utilized to optimize a partially or fully fixed system architecture, optimizing only the component sizes and operational strategy [122–124]. Such simplifications implicitly constrain the design space, limiting the number of candidate system architectures that can be analyzed, again possibly preventing a globally optimal one from being identified by implicitly constraining the system architecture. However, multiple studies utilizing monolithic or distributed MDO frameworks show promise for identifying optimally performing system architectures, enabling both the system architecture, its component sizes, and, optionally, its operational strategy to be optimized hierarchically [128, 130].

Third, Constraint Programs and Boolean Satisfiability Problems identify *feasible* system architectures rather than *optimal* ones, requiring further analyses and optimization to be performed [125, 126, 138]. Furthermore, the constraints imposed on one system architecture design may not be easily scalable/transferable to larger/different ones, necessitating reformulation for each system architecture analyzed. Although implementation-dependent, some solvers filter out isomorphic architectures, eliminating near-identical topologies with different permutations of component connections [125]. Such topologies may perform differently at the system-level, as the physical connections between components may be longer/shorter and less/more efficient, respectively [174]. Therefore, physics-based analysis tools are needed to

explore all candidate system architectures, even if they are isomorphic to another system architecture, to identify the one that demonstrates optimal system-level performance while satisfying system-level safety constraints.

The three gaps identified indicate that there is a need for a unified architecture exploration and optimization environment to streamline the design space exploration process, enabling the combined search and optimization of candidate system architectures by solving a hierarchical MDO problem. Since utilizing evolutionary approaches may require a significant computational cost, similar to graph enumeration, generalized search heuristics must be developed to select candidate system architectures to evaluate and be implemented in an optimization algorithm that both strategically and randomly searches the design space while filtering out infeasible ones. Furthermore, some authors consider system architecture selection and optimal sizing to be an open challenge in the field [123], further justifying the need to fill such a gap.

2.1.6 Summary of Literature Gaps

The core gaps identified from the literature review are listed next, indicating the main technological advances necessary to improve state-of-the-art advanced aircraft design and propulsion system architecting:

- ***Nascent Advanced Aircraft Designs***: incorporate system safety analyses into conceptual design workflows.
- ***Computational Tools for Advanced Aircraft Design***: develop a parametric aircraft sizing tool that is capable of performing rapid, configuration-agnostic analyses for aircraft conceptual design while maintaining the ability to interface with an MDO framework.
- ***Propulsion System Analysis Frameworks***: develop a configuration-agnostic propulsion system analysis framework that remains flexible and scalable to propulsion systems and their respective power off-takes.
- ***Generalized Flight Performance Requirements***: develop generalized climb performance requirements parameterized by the propulsion system failure severity, accommodating both homogeneously- and heterogeneously-sized propulsion systems.

- ***System Architecture Synthesis***: develop a unified architecture exploration and optimization environment with problem-specific heuristics to strategically search the design space.

The five gaps described represent the core contributions of this thesis, and are more broadly addressed in the following research questions and hypotheses.

2.2 Research Questions and Hypotheses

2.2.1 Central Research Question

The overarching literature gaps reveal the need for enhancements to state-of-the-art configuration-agnostic design methods for future aircraft concepts with advanced propulsion systems, along with integrating system safety analyses into conceptual aircraft design workflows. Additional improvements to state-of-the-art propulsion system analysis methods are necessary, as well as generalized flight performance requirements that accommodate any aircraft concept and installed propulsion system architecture.

The central research question addresses the need for a parametric aircraft sizing tool with configuration-agnostic propulsion system and safety analyses to be integrated in a unified system architecture synthesis environment under both nominal and off-nominal performance considerations:

RQ 1.0: How can a unified conceptual aircraft design framework represent, evaluate, and synthesize advanced aircraft propulsion architectures while accounting for nominal aircraft performance, off-nominal propulsion system failure scenarios, and system-level safety constraints in a configuration-agnostic manner?

From the literature review, an MDO framework provides a reasonable tradeoff between identifying an optimal system architecture and the computational resources required to evaluate the candidate system architectures [128, 130]. While the globally optimal solution is not necessarily guaranteed, a stochastic optimization algorithm can efficiently traverse broad regions of the design space without having to evaluate all candidate system architectures to find an optimum [129]. Thus, a unified system architecture exploration and optimization environment with generalized heuristics to strategically search the design space is needed, corresponding to the gap identified in Section 2.1.6, and involves optimizing both the architecture topology (component arrangement/connections) *and* component sizes. Additionally,

no prior system architecture optimization studies in the literature included system safety constraints on the propulsion system architecture. The proposed MDO framework will include a system safety analysis, enabling redundant system architecture design.

Analyzing a broad design space of various propulsion system architectures requires a configuration-agnostic propulsion system analysis framework, corresponding to one of the gaps identified in Section 2.1.6. Based on the literature review, graph-based propulsion system analysis methods appear to be the most modular and scalable for supporting configuration-agnostic design [8, 83], enabling multiple system architectures to be represented within a compact mathematical abstraction. However, existing limitations require further enhancements to the state-of-the-art, particularly for eliminating implicit assumptions on the number of consecutive components arranged and incorporating power off-takes into the modeling framework [8], and will be addressed in a separate research question.

While the proposed MDO framework will include system safety constraints, off-nominal flight performance constraints must also be considered in the event of a propulsion system failure. However, as discovered in the literature review, off-nominal flight performance requirements have remained the same since the 1960s [38], and only accommodate conventional, homogeneously-sized propulsion systems with 2-, 3-, or 4-engines installed. Thus, flight performance requirements, particularly climb performance requirements under engine-inoperative conditions, must be parameterized by the propulsion system failure severity instead of the number of engines installed on the aircraft, addressing a gap identified in Section 2.1.6. One physical quantity that describes the combined aerodynamic and propulsion system performance is the specific excess power, and is impacted by the thrust power available and drag power that the aircraft must overcome [175], making it a useful quantity to generalize the propulsion system failure severity across any propulsion system architecture. Defining generalized flight performance requirements by the specific excess power merits further inquiry and is addressed in a separate research question.

Based on the observations from the literature review, Hypothesis 1.0 proposes the necessary elements in a computational framework to synthesize advanced propulsion system architectures in a configuration-agnostic manner while remaining competitive against other system architecture synthesis methods:

Hyp 1.0: A framework combining graph-based propulsion architecture analyses, off-nominal climb performance constraints based on specific excess power losses under critical failure modes, and joint optimization of the architecture topology and its corresponding power management strategy will identify architectures with a smaller Pareto Hypervolume

than baselines designed with a fixed architecture or stochastic optimization method with problem-agnostic heuristics.

To better answer Research Question 1.0 and test Hypothesis 1.0, three additional research questions are addressed, focused on the three critical methodological advancements necessary to develop a truly configuration-agnostic conceptual aircraft design framework. Improvements to graph-based propulsion system analyses, the development of generalized flight performance requirements, and MDO framework construction are addressed in Research Questions 2.0, 3.0, and 4.0, respectively:

RQ 2.0: How can propulsion system architectures be formally represented and analyzed to quantify system-level tradeoffs in a configuration-agnostic design environment?

RQ 3.0: How can the currently published FAA climb performance requirements for aircraft with discrete, homogeneously sized propulsors be generalized to accommodate propulsion systems with heterogeneously sized or distributed propulsors and their corresponding critical failure modes?

RQ 4.0: How can propulsion system architecture topologies and corresponding power management strategies be synthesized to optimize the system-level performance and safety of an aircraft concept?

Each additional research question is examined in greater detail, along with the hypotheses and relevant experiments in the respective chapters that they are addressed.

2.3 Thesis Outline

The remainder of the thesis is structured as follows, and is illustrated in Fig. 2.1, outlining the research questions, hypotheses, and experiments addressed in each chapter.

Chapter 3 addresses the former half of Research Question 2.0 by introducing a graph-based propulsion system architecture modeling and analysis framework. The chapter describes how concepts from Graph Theory are leveraged to encode propulsion system architectures and its operations in a formal, machine readable manner. A numerical algorithm is introduced to propagate power between components in a propulsion system architecture, accommodating bi-directional flows for conventional and novel operational modes. Lastly, an experiment validates the framework against a benchmark suite of advanced propulsion systems.

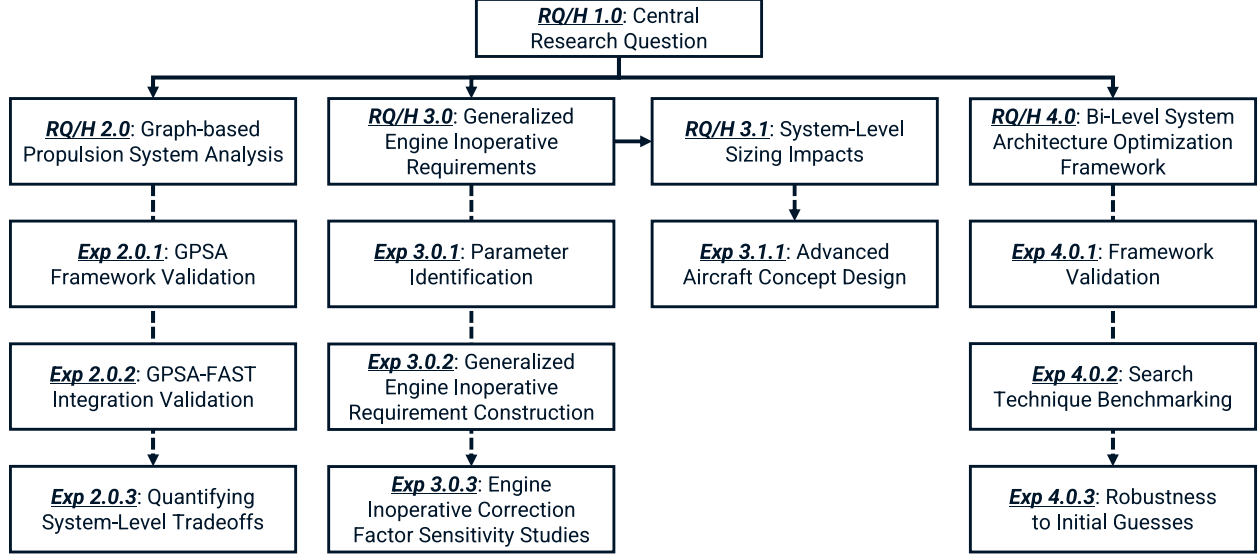


Figure 2.1: Thesis structure.

Then, Chapter 4 introduces a parametric aircraft sizing tool, serving as an enabler to quantify system-level tradeoffs in a configuration-agnostic environment, addressing the latter half of Research Question 2.0. The aircraft sizing and mission analysis methodologies are introduced, highlighting the integration of the graph-based propulsion system analysis framework into the computational workflow. Two experiments are then performed to: 1) validate the parametric aircraft sizing tool on both conventional and advanced aircraft configurations; and 2) quantify system-level performance tradeoffs on a regional jet sized with various advanced propulsion system architectures.

Research Question 3.0 is addressed in Chapter 5 and develops generalized flight performance requirements parameterized by the specific excess power loss during off-nominal flight conditions. Three experiments are performed to: 1) confirm that the specific excess power loss is an appropriate parameter to define a physically-consistent mapping to the flight performance requirements; 2) construct the mapping between the specific excess power and flight performance requirements; and 3) explore the sensitivity of the engine inoperative correction factor construction with respect to the parametric curve and constraints utilized. An additional research question is introduced to explore the system-level impacts of sizing an advanced aircraft concept with the generalized performance requirements relative to one sized with the presently published FAA requirements. A corresponding experiment sizes an aircraft with a distributed electric propulsion system under both the generalized flight performance

requirements constructed and presently published FAA climb performance requirements to answer the research question.

Using the previously described technical developments, Chapter 6 answers Research Question 4.0 and the contributions from Chapters 3 through 5 to establish an MDO framework for synthesizing advanced propulsion system architectures in a configuration-agnostic manner, describing the overarching optimization framework, heuristics developed, and optimization algorithms selected. Three experiments are performed to: 1) validate the MDO framework implementation against an already known performance-safety Pareto Frontier for an advanced aircraft configuration; 2) benchmark the heuristic search implemented in the optimizer against other global search algorithms; and 3) verify that the optimal system architecture selected is weakly sensitive to the initial guess provided.

Finally, Chapter 7 provides answers to the research questions, summarizes the work performed, and highlights the contributions made to the field. Recommendations for future work are also provided.

Chapter 3

Graph-based Propulsion System Analysis Framework

3.1 Introduction

The literature review revealed two critical gaps that prevent existing propulsion system analysis frameworks from performing configuration-agnostic analyses, a necessity for analyzing advanced propulsion system architectures. First, there is a need for a flexible and robust propulsion system architecture framework that simply models propulsion systems with different components/arrangements, especially redundant systems that are critical for certification, without any limitations on the number of components that can be arranged together. Second, a propulsion system analysis framework must provide connections to other aircraft subsystems, particularly as advanced propulsion systems become more tightly integrated with the airframe. Thus, a computational framework must be developed to systematically represent and analyze a propulsion system and its operational modes in a flexible and scalable manner, ensuring that a diverse set of advanced propulsion system architectures can be modeled within a unified framework.

3.1.1 Research Questions, Hypotheses, and Experiments

This chapter addresses Research Question 2.0 and focuses on developing a modeling framework to represent and analyze any propulsion system architecture independent of its component arrangement and power management strategy:

RQ 2.0: How can propulsion system architectures be formally represented and analyzed to quantify system-level tradeoffs in a configuration-agnostic design environment?

In the literature review, two prior approaches were identified for analyzing advanced propulsion systems: 1) defining power splits (or hybridization factors) at critical junctions in the propulsion system [81]; or 2) employing graph-based approaches to define the propulsion system [8, 83]. Graph-based approaches were determined to be superior to defining power splits because of their flexibility and versatility to describe multiple propulsion system architectures in a machine-readable manner, thus lending itself to support system architecture optimization studies in Chapter 6 of this thesis. Existing state-of-the-art methods developed in Ref. [8] proposed a graph-based propulsion system analysis framework but placed limitations on the number of components that can be arranged in series, implicitly limiting it to two power-transmitting components. Additionally, explicit power off-take models were excluded from the framework, and power flows are modeled only in the downstream direction (from the propeller to the fuel/battery), preventing upstream flows (from the fuel/battery to the propeller) from being easily analyzed. To overcome the aforementioned limitations, this chapter seeks to advance the state-of-the-art in configuration-agnostic propulsion system analyses by developing the Graph-based Propulsion System Analysis (GPSA) Framework.

Hypothesis 2.0 answers Research Question 2.0 by outlining the framework structure and provides benchmarks for comparing the constructed framework against existing ones in the literature:

Hyp 2.0: A graph-based representation with component constitutive models at each vertex and power transfer relationships along each edge will reproduce benchmark component power loads and overall energy conservation equations across representative conventional and electrified propulsion architectures within predefined accuracy thresholds while reducing topology-specific model setup effort and execution time relative to existing graph-based methods and propulsion architecture-specific analysis implementations.

To test Hypothesis 2.0, Experiment 2.0.1 validates the GPSA Framework’s ability to flexibly and robustly model both conventional and advanced propulsion systems. The benchmarks described in Hypothesis 2.0 are compared between the GPSA Framework and prior propulsion system analysis methods in the literature, demonstrating the GPSA Framework’s possible improvements in estimation accuracy and computational efficiency as well as manual efforts to define the propulsion system analysis problem.

Experiment 2.0.1: Validate the GPSA Framework’s ability to flexibly and robustly model conventional and advanced propulsion systems with reduced manual pre-processing,

component power load estimation error, and computational cost as existing state-of-the-art methods.

1. Select a representative benchmark suite of conventional and advanced propulsion systems, including conventional, parallel hybrid electric, series hybrid electric, series-parallel hybrid electric, distributed electric propulsion systems. Add a power off-take to one of the aforementioned propulsion system architectures as an additional test.
2. For each configuration in the benchmark:
 - (a) Define the respective component power loads and operational strategy to be simulated.
 - (b) Simulate propulsion system architecture operations utilizing the GPSA Framework and a prior state-of-the-art method.
 - (c) Compare the number of topology-specific user-inputs to define the propulsion system architecture and operational strategy under both methods to verify that less manual pre-processing is required with the GPSA Framework. Compare the component-level power load estimation error and wall time under both methods to verify reductions in both estimation error and computational cost under the GPSA Framework.

While Experiment 2.0.1 validates the GPSA Framework implementation and demonstrates its ability to represent and analyze any propulsion system architecture, it remains focused on subsystem-level design and analysis. To quantify the system-level tradeoffs between candidate propulsion system architectures, the GPSA Framework must be incorporated into a parametric aircraft sizing tool. Chapter 4 performs such an integration, demonstrating how system-level performance can be accurately represented and modeled for any aircraft concept using the GPSA Framework developed in this chapter. Additional experiments in Chapter 4 validate the integrated sizing tool and GPSA Framework implementation and demonstrate quantifying system-level tradeoffs while sizing a regional jet. Together, the answer to Research Question 2.0 lays a foundation for enabling the configuration-agnostic analyses necessary for the optimization framework constructed in Chapter 6.

3.1.2 Chapter Outline

The remainder of this chapter outlines the construction of the GPSA Framework and its validation against a representative benchmark suite of conventional and advanced propulsion

system architectures. First, the GPSA Framework is constructed in Section 3.2. Then, Experiment 2.0.1 is executed in Section 3.3 to validate the GPSA Framework. A benchmark suite of propulsion system architectures is introduced, along with the boundary conditions and power loads for each corresponding test. Then, the power flows are simulated and compared to the benchmark quantities against those from prior implementations in the literature. Lastly, Section 3.4 concludes the chapter, summarizes the key contributions, and reviews the findings from Experiment 2.0.1.

3.2 Graph-based Propulsion System Analysis (GPSA) Framework Construction

The GPSA Framework is now constructed based on its description in Hypothesis 2.0, providing a structured methodology for systematically classifying components, defining the propulsion system and its operation, and evaluating its performance. The GPSA Framework overcomes the three key limitations previously identified in the literature by: 1) including components that explicitly describe how power is expended, accommodating power off-takes; 2) introducing matrices to simulate both upstream and downstream power flows; and 3) developing a new power flow algorithm that allows an unlimited number of components to be connected in series.

A parallel hybrid (PHE) propulsion architecture, depicted in Fig. 3.1, is used to complement the methodology introduced and concretely illustrate the approach. Although the illustrative examples in this paper focus on electrified aircraft propulsion systems, the methodology is fully general and applicable to *any* propulsion system involving multiple (inter)connected components, such as those in cars, ships, and trains.

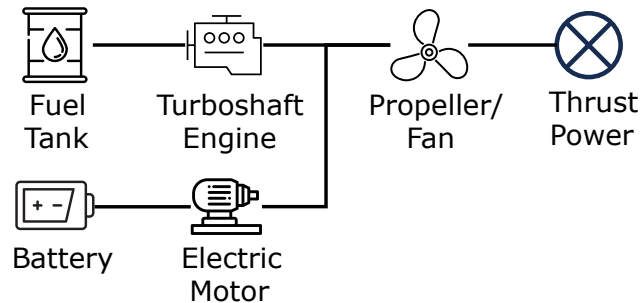


Figure 3.1: Parallel hybrid propulsion system architecture.

A four step methodology is proposed: Component Classification; Architecture Construction; Operation Definition; and Power Flow Simulation.

3.2.1 Component Classification

Components are classified into three types, depending on whether they supply, distribute, or expend power in the system:

1. *Sources*: store and supply power to other components in the system without inputs from entities inside the system boundary during its nominal operating mode. Examples of sources are kerosene fuel, batteries, or hydrogen.
2. *Transmitters*: convert or distribute power between components in the system. Examples of transmitters are turboshaft engines (chemical to mechanical power), electric motors (electrical to mechanical power), electric generators (mechanical to electrical power), and fans/propellers (mechanical power to thrust power).
3. *Sinks*: expend power to perform tasks outside of the system boundary, such as vehicle movement or subsystem operation. Examples of sinks include vehicle movement or subsystem operations.

Figure 3.2 illustrates these component types, along with the directions that power flows through them. Dotted arrows indicate that components may have multiple inputs or outputs.

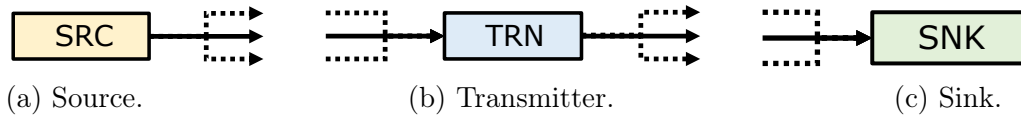


Figure 3.2: General component types in a propulsion system.

Regardless of the propulsion system architecture being modeled, each component has a requisite number of input and output power flows. Each source must provide power to at least one transmitter or sink (minimum of one output). Additionally, transmitters receive power from at least one source or transmitter and distribute power to at least one other transmitter or sink (minimum of one input and one output). Each sink receives power from either a source or transmitter (minimum of one input).

In the PHE propulsion architecture, there are two sources (Fuel and Battery [Batt], denoted as SRC_1 and SRC_2 , respectively), three transmitters (Turboshaft Engine [TSE], Electric Motor [EM], and Propeller [Prop], denoted as TRN_1 , TRN_2 , and TRN_3 , respectively), and one sink (Thrust Power [Thr], denoted as SNK_1). To generalize the analysis, the propulsion architecture is redrawn in Fig. 3.3, where components are labeled based on their functional classification.

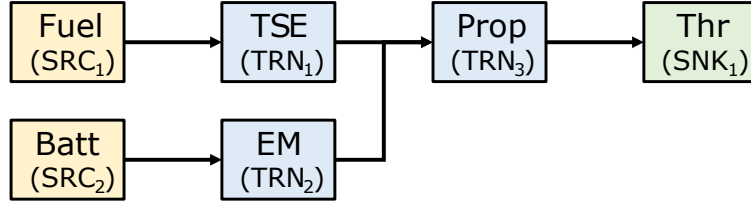


Figure 3.3: Parallel hybrid electric propulsion architecture schematic.

Aside from defining the components' role in the propulsion system and its interactions with other components, the classification also impacts propulsion system sizing. Sources and transmitters must be sized, while sinks only expend power and do not require sizing.

Including sinks in the framework permits various power expenditures to be modeled, overcoming one of the limitations previously discussed, enabling connections to other aircraft subsystems. Furthermore, the sinks can be connected to any source or transmitter in the propulsion architecture, enhancing the proposed framework's flexibility and generality.

After classifying the components, their interactions are systematically defined, as demonstrated next.

3.2.2 Architecture Construction

In general, a propulsion system architecture is a directed power flow between components. A useful mathematical concept to represent the flow of power between interconnected components is a graph, G . A graph is comprised of both vertices, $V(G)$, or points that power can flow between, and edges, $E(G)$, or a means for transporting power between two vertices. A propulsion system can be abstracted to a graph by representing its components (gas turbine engine, electric generator/motor, propeller, etc.) as vertices and its power transfer relationships (cables, mechanical couplings, etc.) as the edges. More specifically, a propulsion system is abstracted as a *directed graph* because there is a directionality to the power flow between components, as represented by the directed edges in Fig. 3.3.

Propulsion system component interactions are systematically encoded via an architecture matrix, A , representing the *direct power flow connections* between components, as defined in Eq. 3.1. In a physical context, each entry, A_{ij} , represents whether power can flow *directly* from Component i to Component j . The architecture matrix is a square matrix of size n , where n is the number of components in the propulsion architecture.

$$A_{ij} = \begin{cases} 1, & V_i \Rightarrow V_j \\ 0, & V_i \nRightarrow V_j \end{cases} \quad (3.1)$$

The architecture matrix is an abstraction of an adjacency matrix from graph theory, defining the connections between vertices in a graph [176], or more concretely, connections between propulsion system components. The sum of each row represents the “out-degree,” or number of components that power is output to. The sum of each column represents the “in-degree,” or number of components that power is received from.

Additionally, all propulsion system architectures are represented as directed acyclic graphs. This implies that there are no cycles, or closed loops [176], within the propulsion system, and further enforces the physical constraint requiring that power only flows between components in one direction at a time. Such a constraint also ensures that the power flow simulation algorithm introduced in Section 3.2.4 converges at all times.

Before constructing the architecture matrix for a PHE propulsion system, a generic architecture matrix is first provided in Eq. 3.2, illustrating all possible power flow connections within the system. For consistency in this thesis, all matrices adhere to the “best practice” of listing sources, then transmitters, and finally sinks in ascending numerical order. However, the framework remains general, allowing components to be listed in any order desired.

$$A = \begin{bmatrix} \text{Fuel} \rightarrow \text{Fuel} & \text{Fuel} \rightarrow \text{Batt} & \text{Fuel} \rightarrow \text{TSE} & \text{Fuel} \rightarrow \text{EM} & \text{Fuel} \rightarrow \text{Prop} & \text{Fuel} \rightarrow \text{Thr} \\ \text{Batt} \rightarrow \text{Fuel} & \text{Batt} \rightarrow \text{Batt} & \text{Batt} \rightarrow \text{TSE} & \text{Batt} \rightarrow \text{EM} & \text{Batt} \rightarrow \text{Prop} & \text{Batt} \rightarrow \text{Thr} \\ \text{TSE} \rightarrow \text{Fuel} & \text{TSE} \rightarrow \text{Batt} & \text{TSE} \rightarrow \text{TSE} & \text{TSE} \rightarrow \text{EM} & \text{TSE} \rightarrow \text{Prop} & \text{TSE} \rightarrow \text{Thr} \\ \text{EM} \rightarrow \text{Fuel} & \text{EM} \rightarrow \text{Batt} & \text{EM} \rightarrow \text{TSE} & \text{EM} \rightarrow \text{EM} & \text{EM} \rightarrow \text{Prop} & \text{EM} \rightarrow \text{Thr} \\ \text{Prop} \rightarrow \text{Fuel} & \text{Prop} \rightarrow \text{Batt} & \text{Prop} \rightarrow \text{TSE} & \text{Prop} \rightarrow \text{EM} & \text{Prop} \rightarrow \text{Prop} & \text{Prop} \rightarrow \text{Thr} \\ \text{Thr} \rightarrow \text{Fuel} & \text{Thr} \rightarrow \text{Batt} & \text{Thr} \rightarrow \text{TSE} & \text{Thr} \rightarrow \text{EM} & \text{Thr} \rightarrow \text{Prop} & \text{Thr} \rightarrow \text{Thr} \end{bmatrix} \quad (3.2)$$

The structure of the architecture matrix can be interpreted in two ways:

- **Row-wise Interpretation:** reading across a row identifies the components that the i^{th} one supplies power to. If A_{ij} is 1, then Component i supplies power to Component j .
- **Column-wise Interpretation:** reading along a column identifies the components that the j^{th} one receives power from. If A_{ij} is 1, then Component j receives power from Component i .

The PHE architecture is described by the following five power flow connections:

1. The fuel is directly connected to the turboshaft engine;
2. The battery is directly connected to the electric motor;
3. The turboshaft engine is directly connected to the propeller;
4. The electric motor is directly connected to the propeller; and
5. The propeller is directly connected to the thrust power.

Equation 3.3 reveals the architecture matrix constructed for the PHE propulsion system:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.3)$$

3.2.2.1 Component Classification Encoding

In addition to encoding direct power flow relationships, the architecture matrix provides a means of classifying components:

1. A *source* is identified by a nonzero row sum and zero column sum, indicating that it only supplies power to other components but does not receive power from any other component within the system boundary.

2. A *sink* is identified by a zero row sum and a nonzero column sum, indicating that it does not supply power to any other component within the system boundary but receives power from other components.
3. A *transmitter* has a nonzero row and column sum, meaning it both receives and supplies power.

These properties are mathematically expressed as:

$$\text{Source} \equiv \left\{ i : \sum_{k=1}^n A_{ik} > 0 \cap \sum_{k=1}^n A_{ki} = 0 \right\} \quad (3.4)$$

$$\text{Sink} \equiv \left\{ i : \sum_{k=1}^n A_{ik} = 0 \cap \sum_{k=1}^n A_{ki} > 0 \right\} \quad (3.5)$$

$$\text{Transmitter} \equiv \left\{ i : \sum_{k=1}^n A_{ik} > 0 \cap \sum_{k=1}^n A_{ki} > 0 \right\} \quad (3.6)$$

Applying the definitions from Eqs. 3.4 through 3.6, the components represented by the first two rows/columns of Eq. 3.3 are sources and the component represented by the last row/column is a sink. The remaining components, having both inputs and outputs, are classified as transmitters.

Utilizing Directed Graphs In the GPSA Framework, directed graphs were used to distinguish sources and sinks from each other, as described by Eqs. 3.4 and 3.5. The row and column sums represent the number of possible power flow paths out and into a component, respectively. If an undirected graph were used, the adjacency matrix would be symmetric, insinuating that bi-directional power flows exist between connected components. Additionally, the row and column sums for all sources and sinks would be greater than or equal to 1, indicating that they can both send and receive power. This contradicts the definitions of sources and sinks established previously. Furthermore, all transmitters would have row and column sums greater than or equal to 2 because they connect to at least two other components (one input and one output). Therefore, utilizing a directed graph makes it more convenient to identify components and their classification in the architecture matrix, as well as simulate the underlying physics in the propulsion system architecture.

Additionally, utilizing directed graphs enables simpler implementation into a Fault Tree Analysis (FTA) code introduced in Section 6, allowing the FTA to more easily traverse the system architecture to identify the combinations of component failures that could cause a system-level failure. If an undirected graph were to be used, the FTA would not be able to distinguish between sources and sinks, prohibiting a correct safety analysis from being conducted.

3.2.3 Operation Definition

Power flow within a propulsion system occurs in two distinct directions, as illustrated in Fig. 3.4. During operation in a physical environment, power flows from sources to sinks, defined as an “upstream flow.” From a computational perspective, the power required by the sinks is normally known, and the power that the sources must supply is unknown. For this case, the power flow must be reversed (from sink to source), defined as a “downstream flow.”

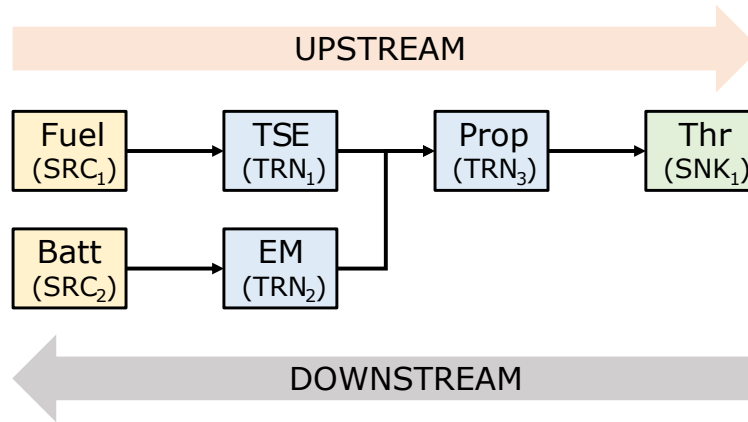


Figure 3.4: Flow direction definition on the PHE architecture.

The operational matrix provides a structured way to define the flow between components. Like the architecture matrix, the operational matrix is square and of size n . In the context of a propulsion system, the operational matrix transmits power between components, but it may also transmit current, torque, or other physical quantities.

Each row of the matrix encodes an energy conservation equation, defining how power from one component is distributed to the others. This approach more easily accommodates varying power distribution strategies and circumvents the need to derive the energy conservation equations manually for each propulsion system analyzed [80, 81]. Furthermore, all power

flows are represented in one matrix, preventing obfuscation of component contributions across multiple matrices [8].

Since there are two possible flow directions – upstream (source to sink) and downstream (sink to source) – a separate operational matrix and corresponding power splits are defined for each direction, further detailed in Sections 3.2.3.1 and 3.2.3.2. Additionally, losses along the power flow paths are represented by an efficiency matrix for each flow direction. These matrices are also square and of size n , and are described further in Section 3.2.3.4.

3.2.3.1 Upstream Flow

The upstream operational matrix, λ^u , defines the fraction of power available transferred from one component to its upstream (forward) connections, and is provided in Eq. 3.7. The definition of an upstream power split enhances the methodology that Cinar et al. [8] previously developed by formally defining a mechanism to represent upstream power flows.

$$\lambda_{ij}^u = \frac{P_{i \Rightarrow j}}{P_{av,i}} \quad (3.7)$$

Before assigning values to the upstream operational matrix, the architecture matrix is analyzed to determine which entries may be nonzero. Only entries corresponding to existing connections may have nonzero values, thus ensuring consistency with the physical system. If a connection exists between two components, then its entry in the upstream operational matrix may be nonzero. If a connection does not exist between two components, then its entry in the upstream operational matrix is guaranteed to be zero – if no connection exists, nothing can flow between them.

The available upstream power splits for a PHE architecture are provided in Eq. 3.8. The values assigned to these power splits are reviewed in Section 3.2.3.3.

$$\lambda^u = \begin{bmatrix} 0 & 0 & \lambda_{\text{Fuel,TSE}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_{\text{Batt,EM}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda_{\text{TSE,Prop}} & 0 \\ 0 & 0 & 0 & 0 & \lambda_{\text{EM,Prop}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda_{\text{Prop,Thr}} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.8)$$

3.2.3.2 Downstream Flow

The downstream operational matrix, λ^d , defines the fraction of a component's power required supplied from its downstream (backward) connections [8], as shown in Eq. 3.9:

$$\lambda_{ij}^d = \frac{P_{i \Rightarrow j}}{P_{\text{req},i}} \quad (3.9)$$

The architecture matrix inherently defines *upstream* connections and must be transposed to reverse the flow and identify the *downstream* connections, as shown in Eq. 3.10. The transposed architectural matrix represents the flow from sinks to sources. Due to the operational change, sinks act as sources (supply a power load) and sources act as sinks (receive a required power output).

$$A^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.10)$$

Using the transposed architecture matrix, any entry defining a connection represents a possible nonzero entry in the downstream operational matrix. Any zero in the transposed architecture matrix represents a zero entry in the downstream operational matrix – since there is no connection, nothing can flow between them.

For the PHE architecture, the available downstream power splits are provided in Eq. 3.11. The values assigned to these power splits are reviewed next.

$$\lambda^d = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda_{\text{TSE,Fuel}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda_{\text{EM,Batt}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda_{\text{Prop,TSE}} & \lambda_{\text{Prop,EM}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda_{\text{Thr,Prop}} & 0 & 0 \end{bmatrix} \quad (3.11)$$

3.2.3.3 Selecting Power Splits

Physical constraints on the propulsion system must be considered before selecting the power splits. A generic junction, illustrated in Fig. 3.5, is used to derive these constraints.

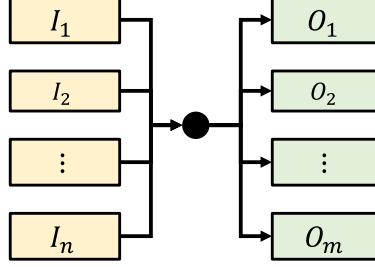


Figure 3.5: Generic junction with n sources and m sinks.

At the junction, each input component, I_i , can provide power to each output component, O_j . Assume the component efficiency at each output is η_j . To conserve energy across the entire junction:

$$I_1 + I_2 + \dots + I_n = \frac{O_1}{\eta_1} + \frac{O_2}{\eta_2} + \dots + \frac{O_m}{\eta_m} \quad (3.12)$$

The physical constraint governing power split selection is now derived for both upstream and downstream directions, highlighting two different interpretations of the same constraint.

Upstream Power Splits An energy conservation equation for each output component is defined in Eq. 3.13, accounting for possible contributions from each input component:

$$\begin{aligned} O_1 &= I_1 \lambda_{11} \eta_1 + I_2 \lambda_{21} \eta_1 + \dots + I_n \lambda_{n1} \eta_1 \\ O_2 &= I_1 \lambda_{12} \eta_2 + I_2 \lambda_{22} \eta_2 + \dots + I_n \lambda_{n2} \eta_2 \\ &\vdots \\ O_m &= I_1 \lambda_{1m} \eta_m + I_2 \lambda_{2m} \eta_m + \dots + I_n \lambda_{nm} \eta_m \end{aligned} \quad (3.13)$$

where λ_{ij} represents the upstream power splits defined in the upstream operational matrix. Dividing each equation by η_j , and then adding all equations together yields:

$$\frac{O_1}{\eta_1} + \dots + \frac{O_m}{\eta_m} = I_1 (\lambda_{11} + \dots + \lambda_{1m}) + \dots + I_n (\lambda_{n1} + \dots + \lambda_{nm}) \quad (3.14)$$

Both sources and transmitters connect to upstream components with nonzero power demands, meaning that $\sum_{j=1}^m (O_j/\eta_j) > 0$. Thus, to conserve energy and obey Eq. 3.12, the sums of power splits in Eq. 3.14 must be 1. In general,

$$\lambda_{i1} + \lambda_{i2} + \dots + \lambda_{im} = 1 \quad (3.15)$$

which requires that the sum of all power splits in a given row of the operational matrix be 1. This constraint requires each component to transmit all the power it receives to other components inside the system boundary, preventing power from accumulating within a component.

However, sinks only receive power and do not output it within the system boundary, meaning that $O_1 = \dots = O_m = 0$. Since the sink receives input power ($\sum_{i=1}^n I_i > 0$), the energy conservation equation is true if

$$\lambda_{i1} + \lambda_{i2} + \dots + \lambda_{im} = 0 \quad (3.16)$$

requiring the sum of all power splits in a given row of the operational matrix be 0.

Downstream Power Splits An energy conservation equation for each input component is defined in Eq. 3.17, accounting for possible power demands from each output component:

$$\begin{aligned} I_1 &= O_1\lambda_{11}/\eta_1 + O_2\lambda_{21}/\eta_2 + \dots + O_m\lambda_{m1}/\eta_m \\ I_2 &= O_1\lambda_{12}/\eta_1 + O_2\lambda_{22}/\eta_2 + \dots + O_m\lambda_{m2}/\eta_m \\ &\vdots \\ I_n &= O_1\lambda_{1n}/\eta_1 + O_2\lambda_{2n}/\eta_2 + \dots + O_m\lambda_{mn}/\eta_m \end{aligned} \quad (3.17)$$

where λ_{ij} now represents the downstream power splits defined in the downstream operational matrix. Adding these equations together yields:

$$I_1 + \dots + I_n = \frac{O_1}{\eta_1} (\lambda_{11} + \dots + \lambda_{1n}) + \dots + \frac{O_m}{\eta_m} (\lambda_{m1} + \dots + \lambda_{mn}) \quad (3.18)$$

In the downstream direction, both sinks and transmitters request nonzero power quantities, meaning that $\sum_{j=1}^m (O_j/\eta_j) > 0$. To conserve energy and obey Eq. 3.12, the sums of power splits in Eq. 3.18 must be 1, thus imposing the same constraint previously defined by Eq. 3.15 (replacing the subscript m with n to represent the n input components).

However, sources do not receive an input from within the system boundary ($I_1 = \dots = I_n = 0$). The energy conservation equation is only satisfied if the sums of the power splits are 0, thus imposing the same constraint previously defined by Eq. 3.16 (again, replacing the subscript m with n).

Power Splits for the PHE Configuration Using the constraints imposed by Eqs. 3.15 and 3.16, consider the upstream direction:

1. The fuel provides all of its power to the turboshaft engine;
2. The battery provides all of its power to the electric motor;
3. The turboshaft engine provides all of its power to the propeller;
4. The electric motor provides all of its power to the propeller; and
5. The propeller provides all of its power to the thrust power.

Since each component provides all of its power to a single upstream component, all power splits are 1 in this example, as illustrated in Eq. 3.19.

$$\lambda^u = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.19)$$

In the case of the PHE architecture, the upstream operational matrix in Eq. 3.19 is structurally identical to the architecture matrix in Eq. 3.3. However, this is not always the case. When one component distributes its power to multiple upstream ones, the corresponding row in the operational matrix contains multiple nonzero quantities with decimal values between 0 and 1, distinguishing it from the architecture matrix representation (that only has integer values of 0 and 1). Such behavior is observed in the series-parallel hybrid electric propulsion architecture in Section 3.3.1.4.

Next, consider the downstream direction:

1. The thrust power requires that all of its power be supplied by the propeller;

2. The propeller requires that part of its power be supplied by the turboshaft engine;
3. The propeller requires that part of its power be supplied by the electric motor;
4. The turboshaft engine requires that all of its power be supplied by the fuel; and
5. The electric motor requires that all of its power be supplied by the battery.

Equation 3.20 provides the downstream operational matrix for the PHE architecture. Since the propeller accepts power from both the turboshaft engine and electric motor, one of the power splits depends on the other to ensure that Eq. 3.15 is satisfied. Selecting these power splits has been the subject of many optimization problems in prior literature [29, 66, 77, 80].

$$\lambda^d = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda_{\text{Prop,TSE}} & \lambda_{\text{Prop,EM}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.20)$$

3.2.3.4 Component Efficiencies

The efficiency characteristics of a propulsion system influence power transmission and overall performance. These characteristics include component efficiencies, defining how effectively individual components convert and transfer power, and path efficiencies, accounting for losses along the power transmission pathways. Depending on the model fidelity used to analyze the propulsion system, it may be necessary to account for the components' efficiencies while simulating a power flow. In this case, an additional matrix is needed to account for the transmission losses associated with each component in the propulsion system.

To incorporate these effects into the propulsion system analysis, efficiency matrices are introduced. The upstream efficiency matrix, Π^u , and the downstream efficiency matrix, Π^d , modify power flow calculations by encoding the cumulative efficiency between connected components. The entries in these matrices are defined in Eqs. 3.21 and 3.22, respectively, as the product of the component (single subscript) and path (dual subscript) efficiencies:

$$\Pi_{ij}^u = (\eta_j) (\eta_{i \Rightarrow j}) \quad (3.21)$$

$$\Pi_{ij}^d = (\eta_i) (\eta_{i \Rightarrow j}) \quad (3.22)$$

Regardless of the power flow direction, the upstream component efficiency is always used in both matrices because power propagating through the system is subject to the efficiency characteristics of the component it passes through.

For components that are not directly connected (i.e., assigned a 0 in the architecture matrix), a value of 1 is assigned in the efficiency matrix. This does not imply a perfect efficiency, but serves as a computational convenience for the power flow simulation introduced in Section 3.2.4.

While efficiency matrices provide a structured mathematical representation of these losses, their specific implementation depends on the fidelity of the analysis. In simplified models, efficiency values may be precomputed and embedded in the matrices to streamline calculations, assuming they remain constant. In higher-fidelity methods, efficiency matrices may still be used, but their entries are dynamically updated based on system operating conditions. Instead of storing fixed efficiency values, these matrices contain time-dependent elements (that are governed by either empirical relations, physics-based models, or another computational model) that evolve with the system state, as done by Cinar [66].

Returning to the PHE configuration, its efficiency matrices are defined in Eqs. 3.23 and 3.24.

$$\Pi^u = \begin{bmatrix} 1 & 1 & \eta_{\text{TSE}}\eta_{\text{Fuel,TSE}} & & 1 & & 1 & & 1 \\ 1 & 1 & & 1 & \eta_{\text{EM}}\eta_{\text{Batt,EM}} & & 1 & & 1 \\ 1 & 1 & & 1 & 1 & \eta_{\text{Prop}}\eta_{\text{TSE,Prop}} & & & 1 \\ 1 & 1 & & 1 & 1 & \eta_{\text{Prop}}\eta_{\text{EM,Prop}} & & & 1 \\ 1 & 1 & & 1 & 1 & & 1 & \eta_{\text{Thr}}\eta_{\text{Prop,Thr}} & \\ 1 & 1 & & 1 & 1 & & 1 & & 1 \end{bmatrix} \quad (3.23)$$

$$\Pi^d = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \eta_{\text{TSE}}\eta_{\text{TSE,Fuel}} & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \eta_{\text{EM}}\eta_{\text{EM,Batt}} & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & \eta_{\text{Prop}}\eta_{\text{Prop,TSE}} & \eta_{\text{Prop}}\eta_{\text{Prop,EM}} & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & \eta_{\text{Thr}}\eta_{\text{Thr,Prop}} & 1 \end{bmatrix} \quad (3.24)$$

In this example, a simplified case is considered where all power transmission paths are assumed to be ideal. Specifically, it is assumed that there are no losses along the connections between components and that the sources and sinks operate with perfect efficiency. Under these assumptions, the efficiency matrices for the PHE propulsion architecture reduce to Eqs. 3.25 and 3.26.

$$\Pi^u = \begin{bmatrix} 1 & 1 & \eta_{\text{TSE}} & 1 & 1 & 1 \\ 1 & 1 & 1 & \eta_{\text{EM}} & 1 & 1 \\ 1 & 1 & 1 & 1 & \eta_{\text{Prop}} & 1 \\ 1 & 1 & 1 & 1 & \eta_{\text{Prop}} & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (3.25)$$

$$\Pi^d = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \eta_{\text{TSE}} & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \eta_{\text{EM}} & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & \eta_{\text{Prop}} & \eta_{\text{Prop}} & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (3.26)$$

3.2.3.5 Graph Theory Abstractions for the Operational/Efficiency Matrices

While the operational and efficiency matrices take the same structure as the architecture matrix (or its transpose), their values are represented by decimal numbers, as opposed to integers in the architecture matrix. In graph theory, such matrices are defined as *weighted adjacency matrices* [176]. However, there are key differences between the weighted adjacency matrix and the operational and efficiency matrices defined in the GPSA Framework.

In classical graph theory, the weighted adjacency matrix entries traditionally represent a “cost” to travel between two vertices via the corresponding edge [176]. Such problems lend themselves to optimization by finding the least expensive path to transport matter from one vertex to another. Depending on the modeling scenario, the weights may be placed on either the vertices or edges. For example, if a water distribution network were modeled by a graph, the distribution centers act as the vertices and the pipes transporting the water act as the edges. Each weight in the weighted adjacency matrix could represent the distance necessary to transport water from one distribution center to the other (the length of each pipe). Then, an optimization problem can be solved to minimize the distance that water must travel between two distribution centers. Alternatively, the weighted adjacency matrix could be the cost to process the water at each distribution center; the weighting is placed on a node (the cost to process at the distribution center) rather than the edge. In this case, an optimization would be solved to minimize the cost associated with processing the water while transporting it between two distribution centers.

In the GPSA Framework, the operational matrix places weights along the edges; each one represents the fraction of power transferred between components, reflecting a power management strategy, and is an abstraction beyond the traditional definition of a weighted adjacency matrix. The efficiency matrix, however, accounts for weights on both the vertices and edges. Each vertex weight represents the component efficiency and each edge weight represents the efficiency of the power transfer path (mechanical coupling, cables, etc.). Thus, each weight represents the overall efficiency of transferring power between components in the propulsion system.

In classical graph optimization problems, the weighted adjacency matrix values remain fixed and an optimum path is identified to minimize the cost of traversing the graph (transferring matter between two vertices) [176]. Power management optimization differs because the entries in the weighted adjacency matrix are varied during the optimization. This is a contrast to classical graph optimization problems because the weighted adjacency matrix entries are no longer fixed, and become the design variables. Additionally, power may be

transferred through all vertices in the graph, in contrast to classical graph optimization problems that select an optimum path to transfer matter.

A second difference between classical graph optimization and power management optimization problems is how the objective function is computed. Traditionally, the objective function in a classical graph optimization problem is directly computed as a total cost of the weights traversed in the path (their sum) [176]. However, the power management optimization is used to compute a system-level response such as fuel burn or energy expenditure. Rather than obtaining the objective function value directly from the graph, it must be propagated from the propulsion system (subsystem-level) analysis to the aircraft (system-level) analysis before obtaining a result, making it more computationally expensive and demanding.

3.2.4 Power Flow Simulation

The architecture, operational, and efficiency matrices defined in Sections 3.2.2 and 3.2.3 are now combined to simulate power flows. Matrix multiplication is used to iteratively compute the power passing through each component in the propulsion system. Power flows in the upstream and downstream direction are governed by Eq. 3.27; $\odot$ and $\oslash$ represent element-wise multiplication and division, respectively.

$$P^* = \begin{cases} \left(\lambda^u \odot (A + A^T) \odot \Pi^u \right)^T \times P, & \text{Upstream Direction} \\ \left(\lambda^d \odot (A + A^T) \oslash \Pi^d \right)^T \times P, & \text{Downstream Direction} \end{cases} \quad (3.27)$$

In both equations, the respective operational matrix is element-wise multiplied by the sum of the architecture matrix and its transpose, $(A + A^T)$. This step ensures that power only flows along paths where a physical connection exists in the underlying architecture, regardless of the operational direction. While seemingly a redundant safeguard, this is critical for simulating complex, bi-directional operational modes like battery charging, as a single operational matrix may contain both forward and backward power flows relative to the base architecture. Using $(A + A^T)$ ensures that all valid connections for a given mode are preserved during matrix multiplication.

Algorithm 1 defines a computational procedure utilizing a fixed point iteration to propagate power between components in a propulsion system. Both λ and Π may be either the upstream or downstream operational and efficiency matrices, respectively – no subscript is provided to prevent a loss of generality. The iteration allows for propulsion systems with differing path lengths to be easily analyzed under one framework, and is not restricted to a single set of

algebraic relations, an improvement over the methodology described in Refs. [77, 78].

Algorithm 1: Power flow propagation.

input : $A_{(n \times n)}$, $\lambda_{(n \times n)}$, $\Pi_{(n \times n)}$, $P_{(n \times 1)}$, and ϵ , a convergence tolerance
output : P

```

1  $P_{\text{old}} \leftarrow \mathbf{0}_{(n \times 1)}$ 
2 while  $\|P_{\text{old}} - P\|_2 > \epsilon$  do
3    $P^* \leftarrow f(A, \lambda, \Pi)$  (Eq. 3.27)
4    $P_{\text{old}} \leftarrow P$ 
5   for  $i = 1, \dots, n$  do
6     if  $|P_i^*| > \epsilon$  then
7        $P_i = P_i^*$ 
8     end
9   end
10 end

```

As previously mentioned in Section 3.2.2, all propulsion system architectures accommodated by this framework must be represented by a directed acyclic graph. This constraint on the propulsion system architecture and operation guarantees that power only flows in one direction between two components at any given time, and ensures that the fixed point iteration converges. If this constraint were not imposed, energy may not be conserved, returning an invalid simulation result.

The power flow algorithm assumes constant component efficiencies throughout the simulation. For conceptual design frameworks, this assumption simplifies the analysis, improving computational efficiency while sacrificing some solution accuracy. The algorithm can easily be modified to include time-dependent efficiencies for more detailed propulsion system models. This involves a higher-level fixed point iteration to converge on component-level efficiencies for a given power requirement, as described in Alg. 2.

Algorithm 2: Power flow simulation accounting for time-dependent efficiency.

input : $A_{(n \times n)}$, $\lambda_{(n \times n)}$, $\Pi_{(n \times n)}^{(0)}$, $P_{(n \times 1)}$, and ϵ , a convergence tolerance
output : P

```

1  $P_{\text{old}} \leftarrow \mathbf{0}_{(n \times 1)}$ 
2 while  $\|P_{\text{old}} - P\|_2 > \epsilon$  do
3    $P_{\text{old}} \leftarrow P$ 
4   Update  $P(A, \lambda, \Pi, P)$  using Alg. 1
5   Update  $\Pi(P)$  using propulsion system component models
6 end

```

3.2.4.1 Algorithm Demonstration

A downstream flow for the PHE propulsion architecture is simulated to serve as a demonstration of Alg. 1 and its core features, and consists of four steps. The demonstrated power flow is also a part of Experiment 2.0.1, as will be described in Section 3.3.1.2. For this example, perfect component and path efficiencies are assumed, guaranteeing that the power load at the thrust power and power off-take components match the power output by the fuel and battery, enabling a simpler verification of the GPSA Framework's behavior.

Step 1: Initialize First, a convergence tolerance, ϵ , is defined as 10^{-6} . Additionally, the architecture and downstream operational matrices from Eqs. 3.3 and 3.20, respectively, are input. Since perfect component and path efficiencies are assumed, then $\Pi^d = \mathbf{1}_{n \times n}$. From Eq. 3.20, two power splits must be defined: $\lambda_{\text{Prop,TSE}}^d$ and $\lambda_{\text{Prop,EM}}^d$. Assume that the power splits are:

$$\lambda_{\text{Prop,TSE}}^d = 0.82 \quad \text{and} \quad \lambda_{\text{Prop,EM}}^d = 0.18$$

At initialization, P only contains nonzero entries in the rows corresponding to the sinks. Assuming that the sink requires 100 kW, the initial power vector is:

$$P = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 100 \end{bmatrix}$$

Lastly, P_{old} is set to be a zero vector, which is updated in each iteration.

Note that there are two differences when initializing for an upstream power flow. First, the upstream operational and efficiency matrices are input instead of the downstream ones. Second, the initial power vector contains nonzero entries at the sources, indicating the power available at each one.

Step 2: Multiply While the 2-norm between P_{old} and P is greater than the convergence tolerance, power is propagated through the system. Initially, the two vectors are

$$P_{\text{old}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 100 \end{bmatrix}$$

indicating that power still needs to flow through the system.

Next, P^* is computed using Eq. 3.27 to advance the power in the propulsion architecture by one step. By assuming $\Pi^d = \mathbf{1}_{n \times n}$, the downstream flow equation becomes

$$P^* = \left(\lambda^d \odot (A + A^T) \right)^T \times P \quad (3.28)$$

After applying this equation to the initial power vector, the power has flowed from the thrust power to the propeller:

$$\begin{aligned} P^* &= \left(\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.82 & 0.18 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \odot \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \right)^T \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 100 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.82 & 0 \\ 0 & 0 & 0 & 0 & 0.18 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 100 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 100 \\ 0 \end{bmatrix} \end{aligned}$$

Step 3: Update After advancing the power through the propulsion architecture, the power vector from the previous iteration is stored:

$$P_{\text{old}} = P = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 100 \end{bmatrix} \quad (3.29)$$

Next, the subsequent **for**-loop ensures that the power requested at each component is updated correctly. Each element of P^* is checked to determine if it has received power (i.e., is greater than the tolerance defined). If the i^{th} element of P^* has received power, then the corresponding element of P is updated with that value. Otherwise, it retains its previous value to prevent premature zeroing before full propagation is achieved. This selective update process ensures that the power vector remains populated throughout the entire iteration. If this step were omitted, the final power distribution after convergence would incorrectly result in a zero vector, eliminating propagated power values.

After performing this check, the power vector is updated to:

$$P = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 100 \\ 100 \end{bmatrix} \quad (3.30)$$

Step 4: Iterate After updating P and P_{old} , the 2-norm between successive power distributions is still greater than the convergence tolerance, requiring the multiplication and update in Steps 2 and 3, respectively, to run again. This process continues until convergence is achieved. Table 3.1 lists the power requested at each component for all remaining iterations until convergence.

After transferring power to the propeller, it is split between the turboshaft engine and electric motor. Finally, in subsequent iterations, power is propagated from the turboshaft engine to the fuel and from the electric motor to the battery until convergence is achieved. Although power stops flowing through the system after the third iteration, the extra iteration is needed to ensure that the 2-norm between successive power distributions is less than the convergence tolerance.

Table 3.1: Downstream power flow in a PHE architecture

Iteration	Fuel	Batt	TSE	EM	Prop	Thr
0	0	0	0	0	0	100
1	0	0	0	0	100	100
2	0	0	82	18	100	100
3	82	18	82	18	100	100
4	82	18	82	18	100	100

The resultant power required at the fuel tank and battery is 82 and 18 kW, respectively, and will be confirmed in Test B within Experiment 2.0.1.

Required Iterations to Converge Since each iteration represents power transfer between successive components, the number of iterations to converge depends on the path length, or the number of edges traversed between the starting/ending vertices. Thus, for any propulsion architecture with maximum path length l , the power flow algorithm will converge in $l + 1$ iterations – l iterations to fully propagate power, and one extra to confirm convergence.

3.3 GPSA Framework Validation (Experiment 2.0.1)

The GPSA Framework constructed in Section 3.2 is now validated in Experiment 2.0.1. First, a benchmark suite of conventional and advanced propulsion systems are introduced, along with their initial power loads and power management strategies for each corresponding test. Then, the power flows are simulated using the GPSA Framework and prior propulsion system analysis methods in the literature. For each method, the number of topology-specific user inputs, component power load estimation error, and wall time to perform the simulation are recorded and compared against the others. Aside from validating the GPSA Framework, improvements in propulsion system analysis pre-processing and computational accuracy/efficiency are quantified.

3.3.1 Experimental Setup

The GPSA Framework constructed in Section 3.2 is now tested on six propulsion system architectures, as illustrated in Figs. 3.6 through 3.11. Aside from the five prescribed propulsion system architectures, one additional propulsion system architecture must be replicated with

an additional power off-take; the series hybrid propulsion architecture was selected to include the power off-take.

In each test, both downstream and upstream simulations are performed, leading to two experimental setups per propulsion system architecture studied. For all tests, perfect component and path efficiencies are assumed, resulting in the following efficiency matrices for all analyses:

$$\Pi^d = \Pi^u = \mathbf{1}_{n \times n} \quad (3.31)$$

with n representing the number of components in the propulsion system. The simplification ensures that the power required at the sink(s) matches the total power supplied by the source(s). Aside from simplifying the analysis, the assumption helps the reader to easily verify the power propagation process through each propulsion system architecture and power management strategy selected.

Additionally, two propulsion system analysis frameworks identified in the literature review are also considered: 1) a prior graph-based propulsion system analysis framework, developed in Ref. [8]; and 2) formulating the power flows as linear systems of equations and defining necessary hybridization factors, as done in Ref. [81].

For each simulation, the number of topology-specific user inputs, component-level power load estimation error, and wall time to perform the simulation were recorded. The number of topology-specific user inputs corresponds to the size of the matrices required to define the propulsion system architecture and its operational strategy. Since the efficiency matrix values are automatically set for the study, they are not included in the count of topology-specific user inputs; the only entries counted are in the architecture and operational matrix inputs, or the corresponding linear system. The component-level power load estimation error is measured at the sources for all downstream flows and sinks for all upstream flows, comparing the output component power loads to those described in the experimental setup within Section 3.3.1, and is computed via Eq. 3.32, with m representing the number of sources/sinks depending on the power flow simulation tested.

$$\text{Error} = \sum_{i=1}^m \left[100 \times \frac{|P_{i,\text{simulation}} - P_{i,\text{expected}}|}{P_{i,\text{expected}}} \right] \quad (3.32)$$

The wall time was measured by computing the average wall time to simulate the power flow across five repeated runs, allowing for small variations in times to execute floating point operations and access memory caches.

Each propulsion system architecture contains unique traits that make them useful for testing the GPSA Framework’s capabilities. Along with a description of the unique traits, the experimental conditions (initial power loads, power management strategy, and final power loads) are defined for each propulsion system architecture.

3.3.1.1 Test A: Conventional

The conventional architecture, depicted in Fig. 3.6, acts as a baseline before analyzing any advanced propulsion system architectures.

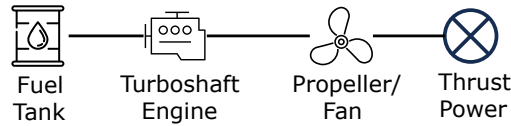


Figure 3.6: Conventional propulsion system architecture.

Downstream operations are simulated as follows, flowing power from the thrust power to the fuel tank:

Initialize: 100 kW power load at the thrust power.

Power Management Strategy: none, each component requests all their power from their downstream connection.

Expected Behavior: the fuel tank must provide 100 kW.

Upstream operations are simulated as follows, flowing power from the fuel tank to the thrust power:

Initialize: 100 kW of power available at the fuel tank.

Power Management Strategy: none, each component provides all their power to their upstream connection.

Expected Behavior: the thrust power provides 100 kW to propel the aircraft.

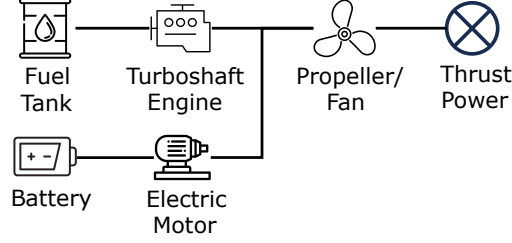


Figure 3.7: Parallel hybrid propulsion system architecture.

3.3.1.2 Test B: Parallel Hybrid

The parallel hybrid propulsion architecture, previously used in Section 3.2 while constructing the GPSA Framework and depicted again in Fig. 3.7, ensures that dual energy source configurations are accommodated.

Downstream operations are simulated as follows, flowing power from the thrust power to the fuel tank and battery:

Initialize: 100 kW power load at the thrust power.

Power Management Strategy: let the gas turbine engine and electric motor provide 82% and 18% of the propeller power, respectively.

Expected Behavior: the fuel tank and battery must provide 82 and 18 kW, respectively.

Upstream operations are simulated as follows, flowing power from the fuel tank and battery to the thrust power:

Initialize: 82 kW and 18 kW of power available at the fuel tank and battery, respectively.

Power Management Strategy: none, each component provides all their power to their upstream connection.

Expected Behavior: the thrust power provides 100 kW to propel the aircraft.

A charging operation is also performed in a separate upstream flow simulation, demonstrating a bi-directional power flow. Instead of the battery powering the electric motor, the propeller provides mechanical power to both the thrust power and electric motor operating in reverse as a generator to charge the battery [29]. Thus, charging operations are simulated as follows, flowing power from the fuel tank to the battery and thrust power:

Initialize: 100 kW at the fuel tank.

Power Management Strategy: the turboshaft engine provides 82% and 18% of its power thrust power and electric motor, respectively.

Expected Behavior: the thrust power provides 82 kW of power and the battery is charged with 18 kW.

3.3.1.3 Test C: Series Hybrid

The series hybrid propulsion architecture, depicted in Fig. 3.8, contains two different path lengths from the fuel tank and battery to the thrust power.

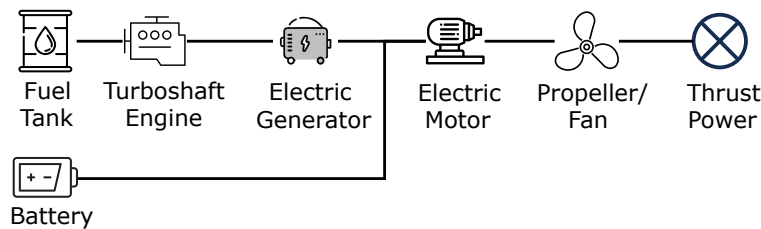


Figure 3.8: Series hybrid propulsion system architecture.

Downstream operations are simulated as follows, flowing power from the thrust power to the fuel tank and battery:

Initialize: 100 kW power load at the thrust power.

Power Management Strategy: let the electric generator and battery provide 82% and 18% of the electric motor power, respectively.

Expected Behavior: the fuel tank and battery must provide 82 and 18 kW, respectively.

Upstream operations are simulated as follows, flowing power from the fuel tank and battery to the thrust power:

Initialize: 82 kW and 18 kW available at the fuel tank and battery, respectively.

Power Management Strategy: none, all components provide power to their upstream connection.

Expected Behavior: the thrust power provides 100 kW to propel the aircraft.

3.3.1.4 Test D: Series-Parallel Hybrid

The series-parallel hybrid electric propulsion architecture, depicted in Fig. 3.9, splits power from the turboshaft engine to the electric generator and propeller, requiring non-trivial upstream power splits.

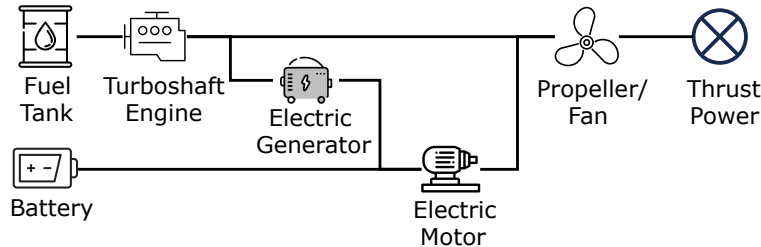


Figure 3.9: Series-Parallel hybrid propulsion system architecture.

Downstream operations are simulated as follows, flowing power from the thrust power component to the fuel tank and battery:

Initialize: 100 kW power load at the thrust power.

Power Management Strategy: let the turboshaft engine and electric motor provide 82% and 18% of the electric motor power, respectively. Let the electric generator and battery each provide 50% of the electric motor power, respectively.

Expected Behavior: the fuel tank and battery must provide 91 and 9 kW, respectively.

Upstream operations are simulated as follows, flowing power from the fuel tank and battery to the thrust power:

Initialize: 91 kW and 9 kW available at the fuel tank and battery, respectively.

Power Management Strategy: let the gas turbine engine provide 82 kW to the propeller and 9 kW to the electric generator. All other components provide all their power to their respective upstream connection.

Expected Behavior: the thrust power provides 100 kW to propel the aircraft.

3.3.1.5 Test E: Distributed Electric Propulsion

The distributed electric propulsion architecture, depicted in Fig. 3.10, also requires non-trivial upstream power splits and showcases a single energy source configuration that excludes a fuel tank.

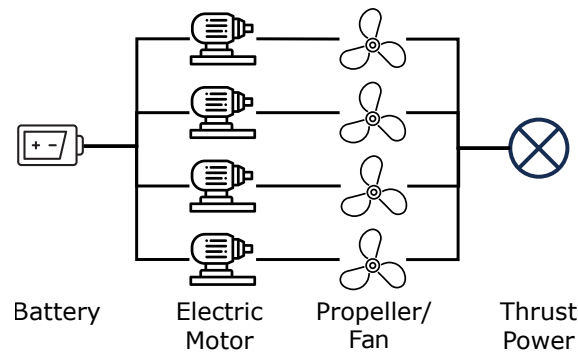


Figure 3.10: Distributed electric propulsion system architecture.

Downstream operations are simulated as follows, flowing power from the thrust power to the fuel tank and battery:

Initialize: 100 kW power load at the thrust power.

Power Management Strategy: split the power from the thrust power across all propellers equally.

Expected Behavior: the battery must provide 100 kW.

Upstream operations are simulated as follows, flowing power from the battery to the thrust power:

Initialize: 100 kW available at the battery.

Power Management Strategy: let the battery distribute its power to all electric motors equally.

Expected Behavior: the thrust power provides 100 kW to propel the aircraft.

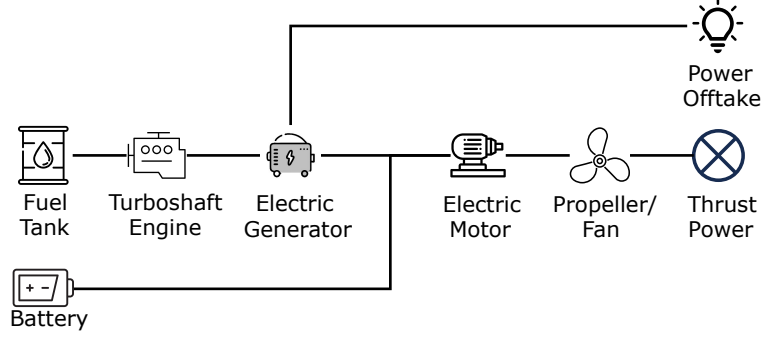


Figure 3.11: Series hybrid propulsion system with power off-take.

3.3.1.6 Test F: Series Hybrid with Power Off-take

Lastly, the series hybrid propulsion architecture with a power off-take, depicted in Fig. 3.11, utilizes two sinks: one to propel the aircraft and one to power its onboard systems.

Downstream operations are simulated as follows, flowing power from the thrust power and power off-take to the fuel tank and battery:

Initialize: 100 kW and 50 kW at the thrust power and power off-take, respectively.

Power Management Strategy: let the electric generator and battery provide 82% and 18% of the electric motor power, respectively. All power from the off-take is provided by the electric generator.

Expected Behavior: the fuel tank and battery must provide 132 and 18 kW, respectively.

Upstream operations are simulated as follows, flowing power from the fuel tank and battery to the thrust power and power off-take:

Initialize: 132 kW and 18 kW at the fuel tank and battery, respectively.

Power Management Strategy: let the electric 82 kW to the electric motor and 50 kW to the power off-take. The battery provides all its power to the electric motor.

Expected Behavior: the thrust power and power off-take provide 100 and 50 kW, respectively.

3.3.2 Test A: Conventional

A schematic of the conventional propulsion system is illustrated in Fig. 3.12.

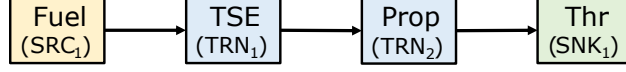


Figure 3.12: Conventional propulsion system architecture.

3.3.2.1 GPSA Framework

The architecture and upstream/downstream operational matrices are:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \lambda^u = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \lambda^d = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Recall that all matrices adhere to the “best practice” of listing sources, then transmitters, and finally sinks in ascending numerical order. For each matrix, 16 entries are required, resulting in 32 topology-specific entries for each downstream/upstream power flow simulation.

Downstream Flow As listed in Test A, the sink requires 100 kW. Since each component only has one connection, all downstream power splits are set to 1, indicating that 100% of each component’s power is requested from its downstream connection.

All iterations of Alg. 1 are listed in Tab. 3.2. Initially, the thrust power requires 100 kW, and propagates that request to the propeller. Next, the propeller requests the same amount of power from the gas turbine engine. Lastly, the gas turbine engine requests the necessary power from the fuel. An additional iteration confirms convergence. It took an average of 1.83×10^{-4} seconds to perform the simulation.

Table 3.2: Downstream power flow in a conventional architecture (all values reported in kW).

Iteration	Fuel	TSE	Prop	Thr
0	0	0	0	100
1	0	0	100	100
2	0	100	100	100
3	100	100	100	100
4	100	100	100	100

The resultant power required at the fuel tank is 100 kW, confirming the expected behavior for the downstream flow in Test A and corresponds to zero component-level estimation error.

Upstream Flow The power available at the fuel tank is 100 kW, respectively. The power management strategy requires each component to provide all their power to their upstream connection, as evidenced by all upstream power splits being set to 1.

Table 3.3 summarizes the iterations, with the fourth iteration confirming convergence after power was propagated through the propulsion system. The total thrust power available is 100 kW, matching the expected behavior outlined in the test and corresponds to zero component-level estimation error. It took an average of 6.84×10^{-4} seconds to perform the simulation.

Table 3.3: Upstream power flow in a conventional architecture (all values reported in kW).

Iteration	Fuel	TSE	Prop	Thr
0	100	0	0	0
1	100	100	0	0
2	100	100	100	0
3	100	100	100	100
4	100	100	100	100

3.3.2.2 Prior State-of-the-Art: Graph-Based Framework

In the framework, there is one energy source (fuel), one power source (gas turbine engine), and one thrust source (propeller). The corresponding architecture and operational matrices utilizing the approach described in Ref. [8] are:

$$B^{A,\text{PSES}} = \begin{bmatrix} 1 \end{bmatrix} \quad B^{A,\text{PSPS}} = \begin{bmatrix} 1 \end{bmatrix} \quad B^{A,\text{TSPS}} = \begin{bmatrix} 1 \end{bmatrix}$$

$$B^{\lambda,\text{PSES}} = \begin{bmatrix} 1 \end{bmatrix} \quad B^{\lambda,\text{PSPS}} = \begin{bmatrix} 1 \end{bmatrix} \quad B^{\lambda,\text{TSPS}} = \begin{bmatrix} 1 \end{bmatrix}$$

Additionally, there is a matrix for splitting the power over each thrust source, $B^{\lambda,\text{TS}\infty}$. Since there is only one thrust source, this term is set to 1¹. Thus, there are 7 topology-specific inputs that need to be provided, including the entry for $B^{\lambda,\text{TS}\infty}$.

¹Assume that $B^{\lambda,\text{TS}\infty} = \begin{bmatrix} 1 \end{bmatrix}$ unless otherwise specified in this chapter.

Table 3.4 lists the power loads after each corresponding matrix multiplication. All power loads are estimated correctly, corresponding to no component-level estimation error. The average time to compute the result was 1.19×10^{-4} seconds.

Table 3.4: Power flow using the framework developed in Ref. [8] (all values reported in kW).

Step	Fuel	TSE	Prop
(1) $TV \rightarrow TS$	0	0	100
(2) $TS \rightarrow PS^*$	0	100	100
(3) $PS^* \rightarrow PS$	0	100	100
(4) $PS \rightarrow ES$	100	100	100

3.3.2.3 Prior State-of-the-Art: Hybridization Factors

Downstream Operations The linear system defining downstream operations is:

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{\text{Fuel}} \\ P_{\text{GTE}} \\ P_{\text{Prop}} \\ P_{\text{Thr}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 100 \end{bmatrix}$$

The first three equations represent the energy conservation equations, flowing power between components. The final equation represents a boundary condition; the power load at the thrust power is 100 kW.

In this system of equations, there are 16 entries in the square matrix of the left-hand side and 4 entries in the column vector on the right-hand side, resulting in 20 total topology-specific inputs. After solving the linear system, the corresponding power loads are:

$$\begin{bmatrix} P_{\text{Fuel}} \\ P_{\text{TSE}} \\ P_{\text{Prop}} \\ P_{\text{Thr}} \end{bmatrix} = \begin{bmatrix} 100 \\ 100 \\ 100 \\ 100 \end{bmatrix} \quad (3.33)$$

The result matches the expected behavior, resulting in no component-level estimation error. It took an average of 4.39×10^{-5} seconds to compute the result.

Upstream Operations The linear system defining upstream operations is:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} P_{\text{Fuel}} \\ P_{\text{GTE}} \\ P_{\text{Prop}} \\ P_{\text{Thr}} \end{bmatrix} = \begin{bmatrix} 100 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.34)$$

The first equation represents a boundary condition; the power available from the fuel is 100 kW. The remaining three equations represent the energy conservation equations between connected components.

Again, 20 topology-specific entries are required to define the propulsion system architecture operation. Solving the linear system of equations returns the same result as demonstrated during Downstream Operations, and also corresponds to zero component-level estimation error. It took approximately 5.28×10^{-5} seconds to solve.

3.3.2.4 Summary of Benchmarking Metrics Recorded

The benchmarking metrics across all simulations and three propulsion system analysis methods are reported in Tab. 3.5.

Table 3.5: Benchmarking metrics for the conventional propulsion system.

Simulation	Method	Topology-Specific Inputs	Estimation Error (%)	Wall Time (s)
Upstream	GPSA Framework	32	0	6.84×10^{-4}
	Prior Graph-based	—	—	—
	Hybridization Factors	20	0	5.28×10^{-5}
Downstream	GPSA Framework	32	0	1.83×10^{-4}
	Prior Graph-based	7	0	1.01×10^{-4}
	Hybridization Factors	20	0	4.39×10^{-5}

3.3.3 Test B: Parallel Hybrid

A schematic of the parallel hybrid propulsion system is illustrated in Fig. 3.13.

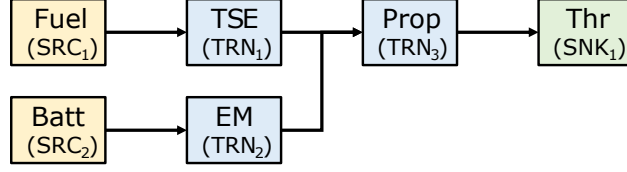


Figure 3.13: Parallel hybrid propulsion system architecture.

3.3.3.1 GPSA Framework

Recall that the architecture and upstream/downstream operational matrices were:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \lambda^u = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\lambda^d = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda_{\text{Prop,TSE}} & \lambda_{\text{Prop,EM}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

For each matrix, 36 topology-specific entries are defined per matrix, resulting in 72 total entries for each downstream/upstream power flow simulation.

Downstream Flow Recall that the downstream power flow was already demonstrated in Section 3.2.4. The required power load at the fuel tank and battery were 82 and 18 kW, respectively, verifying the behavior expected, resulting in zero error. It took an average of 3.57×10^{-5} seconds to perform the simulation.

Upstream Flow As described in Test B, the power available at the fuel and battery are 82 and 18 kW, respectively. The power management strategy requires each component to provide all their power to their upstream connection, as evidenced by all upstream power splits being set to 1 because each component transmits all of its power to a single upstream connection.

Table 3.6 summarizes the iterations, with the fourth iteration confirming convergence after power was propagated through the propulsion system. The combined power from the fuel and battery is propagated through the turboshaft engine, electric motor, and propeller, eventually delivering 100 kW to the thrust power. The 100 kW output matches the expected behavior outlined in the upstream operations simulation described in Test B, resulting in zero component-level estimation error. It took an average of 3.37×10^{-5} seconds to perform the simulation.

Table 3.6: Upstream power flow in a PHE architecture without battery charging (all values reported in kW).

Iteration	Fuel	Batt	TSE	EM	Prop	Thr
0	82	18	0	0	0	0
1	82	18	82	18	0	0
2	82	18	82	18	100	0
3	82	18	82	18	100	100
4	82	18	82	18	100	100

As observed from Tabs. 3.1 and 3.6, each iteration clearly illustrates the power propagation through each component in the parallel hybrid propulsion system architecture. Each source transmits its power to the sink in three steps: 1) from source to transmitter, 2) from one transmitter to another, and 3) from transmitter to sink.

Charging Operations Since power is directed from the fuel tank to the thrust power and battery, the upstream operational matrix must be reconfigured to simulate the bi-directional power flow:

$$\lambda^u = \begin{bmatrix} 0 & 0 & \lambda_{\text{Fuel,TSE}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda_{\text{TSE,Prop}} & 0 \\ 0 & \lambda_{\text{EM,Batt}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_{\text{Prop,EM}} & 0 & \lambda_{\text{Prop,Thr}} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.35)$$

Despite the structure of the upstream operational matrix changing, the number of topology-specific inputs remains the same.

The entire second row (corresponding to the battery) is 0 because it no longer powers any components and acts as a sink instead. Two new power splits are introduced to define the charging operation: $\lambda_{\text{EM,Batt}}$, governing the power flow from the electric motor to the battery; and $\lambda_{\text{TSE,EM}}$, representing the power flow from turboshaft engine to electric motor.

The new operational strategy requires a new set of power splits to be computed. The fuel, electric motor, and propeller all power single components, requiring their upstream power splits to be:

$$\lambda_{\text{Fuel,TSE}} = \lambda_{\text{EM,Batt}} = \lambda_{\text{TSE,Prop}} = 1 \quad (3.36)$$

Since the turboshaft engine transmits power to both the electric motor and propeller, two power splits must be defined. As prescribed by the test, the propeller must transfer 82% and 18% of its power to the thrust power and electric motor, respectively:

$$\lambda_{\text{Prop,EM}} = 0.18 \quad \text{and} \quad \lambda_{\text{Prop,Thr}} = 0.82 \quad (3.37)$$

Table 3.7 lists the iterations of Alg. 1. After transmitting all power from the fuel to the turboshaft engine, it is split between the electric motor and propeller. Then, the propeller continues to supply the thrust power, while the electric motor now receives power and transfers it to the battery. The resulting simulation provides 82 kW to the thrust power and 18 kW to the battery, resulting in zero component-level estimation error relative to the expected behavior from the test.

In this test, the battery acts as a sink instead of a source. From a component classification perspective, the battery is still a source because it stores power. However, it now receives inputs from an entity within the system boundary (like a sink). The definition of a source in

Table 3.7: Upstream power flow in a PHE architecture with battery charging (all values reported in kW).

Iteration	Fuel	Batt	TSE	EM	Prop	Thr
0	100	0	0	0	0	0
1	100	0	100	0	0	0
2	100	0	100	0	100	0
3	100	0	100	18	100	82
4	100	18	100	18	82	82
5	100	18	100	18	82	82

Section 3.2.1 clarifies that sources only supply power to other components “during its nominal operating mode.” Since battery charging is assumed to be an off-nominal operating mode, the battery can still be classified as a source without its definition being violated.

3.3.3.2 Prior State-of-the-Art: Graph-based Framework

Downstream Flow In the framework, there are two energy sources (fuel and battery), two power source (gas turbine engine and electric motor), and one thrust source (propeller). The corresponding architecture and operational matrices utilizing the approach described in Ref. [8] are:

$$\begin{aligned}
 B^{A,\text{PSES}} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & B^{A,\text{PSPS}} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & B^{A,\text{TSPS}} &= \begin{bmatrix} 1 & 1 \end{bmatrix} \\
 B^{\lambda,\text{PSES}} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & B^{\lambda,\text{PSPS}} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & B^{\lambda,\text{TSPS}} &= \begin{bmatrix} 0.82 & 0.18 \end{bmatrix}
 \end{aligned}$$

There are 21 topology-specific inputs that need to be provided, including the entry for $B^{\lambda,\text{TS}\infty}$.

Table 3.8 lists the power loads after each corresponding matrix multiplication. All power loads are estimated correctly, corresponding to no component-level estimation error. The average time to compute the result was 1.19×10^{-4} seconds.

Table 3.8: Power flow using the framework developed in Ref. [8] (all values reported in kW).

Step	Fuel	Battery	TSE	EM	Prop
(1) $TV \rightarrow TS$	0	0	0	0	100
(2) $TS \rightarrow PS^*$	0	0	82	18	100
(3) $PS^* \rightarrow PS$	0	0	82	18	100
(4) $PS \rightarrow ES$	82	18	82	18	100

Charging Operations The architecture matrices representing the charging operational mode are:

$$B^{A,TS\text{PS}} = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad B^{A,PS\text{PS}} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad B^{A,PS\text{ES}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Since thrust sources cannot output power to any power sources, the turboshaft engine is directly connected to the electric motor in the $B_{2,1}^{\lambda,PS\text{PS}}$ entry.

The charging process is completed in three steps [8, 67]. First, power is transmitted from the thrust source and any energy sources being charged to the driven power sources. A negative power split is used for the charging operation, highlighting that the flow direction was reversed. Second, the absolute value of each thrust/energy source contribution is added together to compute the power source demands. Lastly, the power propagation continues to the driving power sources and energy sources. The operational matrices defining the charging operation are:

$$B^{\lambda,TS\text{PS}} = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad B^{\lambda,PS\text{PS}} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad B^{\lambda,PS\text{ES}} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Since the turboshaft engine drives the electric motor, $B_{2,1}^{\lambda,PS\text{PS}} = 1$. To represent that the electric motor charges the battery, $B_{2,2}^{\lambda,PS\text{ES}} = -1$. Since the same number of components are included in the propulsion architecture, 21 topology-specific inputs are still needed, including the definition of $B^{\lambda,TS,\infty}$.

The power output for each component is listed in Tab. 3.9. Although the electric motor

operates in “reverse mode”, its power demand is positive. This is not intuitive and highlights a conceptual inconsistency while simulating charging operations. Despite the inconsistency, power is correctly propagated through the system, requiring the fuel to provide 100 kW. If the sign of the battery’s power output is ignored, then there is zero component-level estimation error. Given that the computational implementation is different and requires additional steps than the one proposed in Ref. [8], the computational speed was not recorded to ensure a fair comparison between the nominal state-of-the-art propulsion system analysis techniques.

Table 3.9: Power flow using the framework developed in Ref. [8] (all values reported in kW).

Step	Fuel	Batt	TSE	EM	Prop
(1) $TV \rightarrow TS$	0	0	0	0	82
(2) $TS \rightarrow PS^*$	0	0	82	0	82
(3) $ES \rightarrow PS^*$	0	0	82	18	82
(4) $PS^* \rightarrow PS$	0	0	100	18	82
(5) $PS \rightarrow ES$	100	-18	100	18	82

3.3.3.3 Prior State-of-the-Art: Hybridization Factors

Downstream Operations The linear system defining downstream operations is:

$$\begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & -1 & 0 \\ 0 & 0 & 0.18 & -0.82 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{\text{Fuel}} \\ P_{\text{Batt}} \\ P_{\text{GTE}} \\ P_{\text{EM}} \\ P_{\text{Prop}} \\ P_{\text{Thr}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 100 \end{bmatrix} \quad (3.38)$$

The first five equations represent the energy conservation equations, flowing power between components. The sixth equation represents the boundary condition; the power load at the thrust power is 100 kW.

In this system of equations, there are 36 entries in the square matrix of the left-hand side and 6 entries in the column vector on the right-hand side, resulting in 42 total topology-specific inputs. After solving the linear system, the corresponding power loads are:

$$\begin{bmatrix} P_{\text{Fuel}} \\ P_{\text{Batt}} \\ P_{\text{TSE}} \\ P_{\text{EM}} \\ P_{\text{Prop}} \\ P_{\text{Thr}} \end{bmatrix} = \begin{bmatrix} 82 \\ 18 \\ 82 \\ 18 \\ 100 \\ 100 \end{bmatrix} \quad (3.39)$$

The result matches the expected behavior, resulting in no component-level estimation error. It took 4.35×10^{-5} seconds to compute the result.

Upstream Operations The linear system defining upstream operations is:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} P_{\text{Fuel}} \\ P_{\text{Batt}} \\ P_{\text{GTE}} \\ P_{\text{EM}} \\ P_{\text{Prop}} \\ P_{\text{Thr}} \end{bmatrix} = \begin{bmatrix} 82 \\ 18 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.40)$$

The first two equations define the boundary conditions; the power available from the fuel and battery are 82 and 18 kW, respectively. The remaining equations represent the energy conservation equations between connected components.

Solving the linear system of equations returns the same result as demonstrated during Downstream Operations, and also corresponds to no component-level estimation error. It took approximately 3.95×10^{-5} seconds to solve.

Charging Operations For a parallel hybrid propulsion system defined in Ref. [81], the hybridization factor is defined as the fraction of battery power output relative to the total battery and fuel power output. Since the battery receives 18 kW and the fuel provides 100 kW, its value is

$$\Phi = \frac{P_{\text{Batt}}}{P_{\text{Fuel}} + P_{\text{Batt}}} = \frac{-18}{100 + (-18)} = -\frac{18}{82} = -0.2195 \quad (3.41)$$

In the linear system defined [81], there are six equations to solve for the power outputs at each component – four energy conservation equations, one hybridization factor definition, and one boundary condition (power available from the fuel). In this example, the energy conservation equations and hybridization factor definition remain the same, but the fuel power available acts as the new boundary condition.

After assuming perfect component efficiencies, the following linear system defines all propulsion system operations:

$$\begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ -0.2195 & -1.2195 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_{\text{Fuel}} \\ P_{\text{Batt}} \\ P_{\text{TSE}} \\ P_{\text{EM}} \\ P_{\text{Prop}} \\ P_{\text{Thr}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 100 \end{bmatrix}$$

There are still 36 entries in the square matrix and 6 on the right-hand side prescribing the boundary conditions, resulting in 42 total topology-specific entries. The solution to the linear system, representing the power output by each component, is:

$$\begin{bmatrix} P_{\text{Fuel}} \\ P_{\text{Batt}} \\ P_{\text{TSE}} \\ P_{\text{EM}} \\ P_{\text{Prop}} \\ P_{\text{Thr}} \end{bmatrix} = \begin{bmatrix} 100 \\ -18 \\ 100 \\ -18 \\ 82 \\ 82 \end{bmatrix}$$

The negative values in the result represent a reversed power flow relative to the prescribed flow paths. Since the battery was charging and the electric motors ran in “reverse mode”, both power outputs are negative. This numerical artifact quickly reveals which components are operating in “reverse mode.” If the signs of the returned power loads are ignored, then there is zero component-level estimation error.

In the linear system, the hybridization factor is -21.95%, despite that 18% of the fuel’s power is transferred to the battery. This is a result of the hybridization factor’s definition, which uses the fuel and battery power contributions in the denominator. Thus, it is difficult

to interpret the significance of its value in the linear system – the definition is less intuitive if an energy storage component receives power. It is easier to interpret the hybridization factor when the fuel and battery both output power.

The shortcomings presented could be remedied by redefining the hybridization factor or defining separate ones for upstream and downstream power propagation. The GPSA Framework uses the latter solution to ensure that the physical implications of each power split is intuitively understood.

3.3.3.4 Summary of Benchmarking Metrics Recorded

The benchmarking metrics across all simulations and three propulsion system analysis methods are reported in Tab. 3.10.

Table 3.10: Benchmarking metrics for the parallel hybrid propulsion system.

Simulation	Method	Topology-Specific Inputs	Estimation Error (%)	Wall Time (s)
Nominal Upstream	GPSA Framework	72	0	3.37×10^{-5}
	Prior Graph-based	–	–	–
	Hybridization Factors	42	0	3.95×10^{-5}
Nominal Downstream	GPSA Framework	72	0	2.26×10^{-5}
	Prior Graph-based	21	0	1.19×10^{-4}
	Hybridization Factors	42	0	4.35×10^{-5}
Charging	GPSA Framework	72	0	3.57×10^{-5}
	Prior Graph-based	21	0	4.42×10^{-6}
	Hybridization Factors	42	0	4.02×10^{-5}

Additionally, this example illustrates the advantages of redefining the power splits for bi-directional power flows. Using the prior frameworks [8, 66], a negative power split indicates power flow reversal between the battery and electric motor. However, the electric motor also operates in “reverse mode”, despite a positive power output. This convention lacks consistency across the power flow path, and is a result of adding the absolute value of the electric motor power to the propeller power required.

In the GPSA Framework, new power splits are defined, allowing for a simpler bi-directional power flow simulation while eliminating the mathematical inconsistencies previously presented.

The approach in the GPSA Framework is more robust because it permits bi-directional power flows in a propulsion architecture simulation without restricting the architecture to single-direction operation. Moreover, it establishes greater consistency in defining power splits. Regardless of the power flow direction, the sum of all power splits in any upstream (or downstream) operational matrix remains 1, making the method more intuitive, consistent, and easier to implement and check for errors.

3.3.4 Test C: Series Hybrid

A schematic of the series hybrid propulsion system is illustrated in Fig. 3.14.

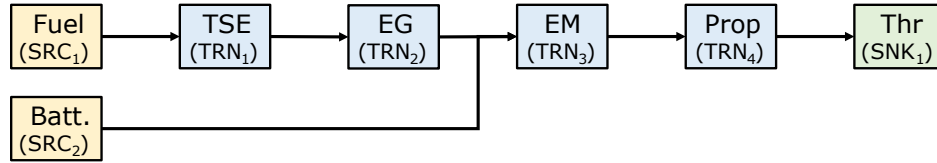


Figure 3.14: Series hybrid propulsion system architecture.

3.3.4.1 GPSA Framework

In Test C, a series hybrid propulsion system architecture is considered, as illustrated in Fig. 3.8. The architecture and upstream/downstream operational matrices are:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\lambda^u = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \lambda^d = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \lambda_{\text{EM,Batt}} & 0 & \lambda_{\text{EM,EG}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

In each matrix, 49 topology-specific entries are required, resulting in 98 total entries per upstream/downstream flow simulation.

In this configuration, power from the fuel is transmitted to the thrust power in five steps, while power from the battery only requires three steps to reach the thrust power. The path length difference demonstrates how the power distribution can vary along separate branches within the propulsion architecture.

Downstream Flow As listed in Test C, the sink requires 100 kW. In the series hybrid propulsion system architecture, one row of power splits must be specified, governing the power provided to the electric motor from the battery and electric generator (EG), namely $\lambda_{\text{EM,Batt}}$ and $\lambda_{\text{EM,EG}}$, respectively.

As described in the power management strategy for Test C, the electric generator and battery provide 82% and 18% of the electric motor power, respectively, requiring that:

$$\lambda_{\text{EM,EG}}^d = 0.82 \quad \text{and} \quad \lambda_{\text{EM,Batt}}^d = 0.18 \quad (3.42)$$

Notice that the sum of the power splits is 1, obeying the energy conservation equation implicitly described in each row of the operational matrix.

All iterations of Alg. 1 are listed in Tab. 3.11. Initially, the thrust power requires 100 kW, which is propagated to the propeller and then electric motor. Next, the electric motor splits its power between the electric generator and battery. Part of the power required reaches the battery immediately, while the remainder is still en-route from the electric generator to the fuel. Additional iterations are needed to fully propagate the power from the electric generator to the fuel, with one final iteration confirming convergence. It took an average of 2.49×10^{-5}

seconds to perform the simulation.

Table 3.11: Downstream power flow in a SHE architecture (all values reported in kW).

Iteration	Fuel	Batt	TSE	EG	EM	Prop	Thr
0	0	0	0	0	0	0	100
1	0	0	0	0	0	100	100
2	0	0	0	0	100	100	100
3	0	18	0	82	100	100	100
4	0	18	82	82	100	100	100
5	82	18	82	82	100	100	100
6	82	18	82	82	100	100	100

The resultant power required at the fuel tank and battery is 82 and 18 kW, respectively, confirming the expected behavior for the downstream flow in Test C and corresponding to zero component-level estimation error.

Aside: Convergence for Propulsion System Architectures with Different Path Lengths To better illustrate the power flow along the different paths in the series hybrid propulsion system, Tab. 3.12 lists the energy conservation equations solved in each iteration. Some equations are repeated to illustrate that they are used to propagate power from the thrust power to the fuel and battery.

Table 3.12: Equations solved in each downstream SHE analysis iteration.

Iteration	Thrust Power to Fuel	Thrust Power to Battery
1	$P_{\text{Prop}} = P_{\text{Thr}}$	$P_{\text{Prop}} = P_{\text{Thr}}$
2	$P_{\text{EM}} = P_{\text{Prop}}$	$P_{\text{EM}} = P_{\text{Prop}}$
3	$P_{\text{EG}} = P_{\text{EM}}\lambda_{\text{EM,EG}}$	$P_{\text{Batt}} = P_{\text{EM}}\lambda_{\text{EM,Batt}}$
4	$P_{\text{TSE}} = P_{\text{EG}}$	
5	$P_{\text{Fuel}} = P_{\text{TSE}}$	

The iterations highlight how differing path lengths affect the algorithm’s convergence. As observed, it takes five iterations to propagate the power from the sink to the fuel, but only three iterations from the sink to the battery. Since the maximum path length is five (fuel tank

to thrust power), the algorithm converges after six iterations. Previously, in Section 3.2.4.1, it was shown that the algorithm converges in $l + 1$ iterations, where l is the maximum path length in the propulsion architecture, confirming the claim previously made.

Upstream Flow As listed in Test C, the fuel tank and battery supply 82 kW and 18 kW, respectively. All power splits are set to 1 because each component supplies their power to a single upstream component, as described by the power management strategy.

All iterations of Alg. 1 are listed in Tab. 3.13. First, the fuel and battery send all their power to the turboshaft engine and electric motor, respectively. After that, the power from the battery reaches the thrust power, while the remaining power is only at the electric motor. Further iterations propagate the remaining power from the electric motor to the thrust power, receiving the 100 kW expected and matching expected behavior described in Section 3.3.1, resulting in zero component-level estimation error. It took an average of 3.03×10^{-5} seconds to perform the simulation.

Table 3.13: Upstream power flow in a SHE architecture (all values reported in kW).

Iteration	Fuel	Batt	TSE	EG	EM	Prop	Thr
0	82	18	0	0	0	0	0
1	82	18	82	0	18	0	0
2	82	18	82	82	18	18	0
3	82	18	82	82	100	18	18
4	82	18	82	82	100	100	18
5	82	18	82	82	100	100	100
6	82	18	82	82	100	100	100

As observed from the iterations, power from the battery was delivered quicker than that from the fuel, a behavior inherent to the architecture’s topology and numerical algorithm proposed.

3.3.4.2 Prior State-of-the-Art: Graph-based Framework

The previous state-of-the-art graph-based framework developed in Ref. [8] only accommodates downstream flows, as demonstrated here. There are two energy sources (fuel and battery), three power sources (turboshaft engine, electric generator, and electric motor), and one thrust

source (propeller). Using the same power splits defined in the downstream simulation, the architecture and operational matrices are:

$$\begin{aligned}
B^{A,\text{TSPS}} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} & B^{A,\text{PSPS}} &= \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} & B^{A,\text{PSES}} &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \\
B^{\lambda,\text{TSPS}} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} & B^{\lambda,\text{PSPS}} &= \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0.82 & 1 \end{bmatrix} & B^{\lambda,\text{PSES}} &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0.18 \end{bmatrix}
\end{aligned}$$

In the architecture and operational matrices, 18 topology-specific entries are required, along with an additional entry in the $B^{\lambda,\text{TS}\infty}$ matrix. Thus, 37 topology-specific entries are required.

The power required by the components after propagating power through them is listed in Tab. 3.14. It took an average of 3.02×10^{-6} seconds to perform the simulation. Select entries in the table are colored red to indicate where the framework returns an erroneous power load.

Table 3.14: Downstream power flow using the framework from Ref. [8] (all values reported in kW).

Step	Fuel	Batt	TSE	EG	EM	Prop
(1) $TV \rightarrow \text{TS}$	0	0	0	0	0	100
(2) $\text{TS} \rightarrow \text{PS}^*$	0	0	0	0	100	100
(3) $\text{PS}^* \rightarrow \text{PS}$	0	0	0	82	100	100
(4) $\text{PS} \rightarrow \text{ES}$	0	18	0	82	100	100

The first two steps transmit power to the electric motor. In the third step, power is transmitted from the electric motor to the electric generator, but no power is propagated to the turboshaft engine because it is not *directly* connected to the electric motor. As a result, the final step does not transfer power to the fuel, but the battery receives its power load. Since the power fails to reach the fuel and turboshaft engine, there is 100% component estimation error at each component, resulting in 200% total estimation error.

Two complications emerge while analyzing the SHE propulsion system. First, the example

highlighted the framework's limitation on arranging no more than two power sources in series. This limitation can be overcome by multiplying the PSPS matrix by the result from Step 3 an additional time, or as many times as necessary to transmit power through all the power sources (inspired by the fixed-point iteration in the GPSA Framework). Second, albeit cosmetic, the energy conservation equations are obfuscated by the presence of multiple matrices. Since the electric motor is connected to a battery (an energy source) and an electric generator (a power source), the energy conservation equation is distributed over the PSPS matrix (third row, second column) and the PSES matrix (third row, second column), making it more difficult to ensure that power correctly flows between the desired components. The GPSA Framework overcomes this by employing comprehensive architecture and operational matrices that account for all power flows between components in a unified manner.

3.3.4.3 Prior State-of-the-Art: Hybridization Factors

Downstream Operations The hybridization factor between the fuel and battery is:

$$\Phi = \frac{P_{\text{Batt}}}{P_{\text{Fuel}} + P_{\text{Batt}}} = \frac{18}{82 + 18} = \frac{18}{100} = 0.18$$

Thus, the linear system defining downstream operations is:

$$\begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0.18 & -0.82 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{\text{Fuel}} \\ P_{\text{Batt}} \\ P_{\text{GTE}} \\ P_{\text{EG}} \\ P_{\text{EM}} \\ P_{\text{Prop}} \\ P_{\text{Thr}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 100 \end{bmatrix}$$

The first five equations represent the energy conservation equations. The sixth equation represents the definition of the hybridization factor. The final equation represents the boundary condition; the thrust power requires 100 kW.

In this system of equations, there are 49 entries in the square matrix of the left-hand side and 7 entries in the column vector on the right-hand side, resulting in 56 total topology-specific inputs. After solving the linear system, the corresponding power loads are:

$$\begin{bmatrix} P_{\text{Fuel}} \\ P_{\text{Batt}} \\ P_{\text{TSE}} \\ P_{\text{EG}} \\ P_{\text{EM}} \\ P_{\text{Prop}} \\ P_{\text{Thr}} \end{bmatrix} = \begin{bmatrix} 82 \\ 18 \\ 82 \\ 82 \\ 100 \\ 100 \\ 100 \end{bmatrix}$$

The result matches the expected behavior, resulting in no component-level estimation error. It took 5.39×10^{-5} seconds to compute the result.

Upstream Operations The linear system defining upstream operations is:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} P_{\text{Fuel}} \\ P_{\text{Batt}} \\ P_{\text{GTE}} \\ P_{\text{EG}} \\ P_{\text{EM}} \\ P_{\text{Prop}} \\ P_{\text{Thr}} \end{bmatrix} = \begin{bmatrix} 82 \\ 18 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The first two equations represent boundary conditions; the fuel and battery can provide 82 and 18 kW, respectively. The remaining five equations represent the energy conservation equations between connected components.

Solving the linear system of equations returns the same result as demonstrated during Downstream Operations, and also corresponds to zero component-level estimation error. It took approximately 4.54×10^{-5} seconds to solve.

3.3.4.4 Summary of Benchmarking Metrics Recorded

The benchmarking metrics across all simulations and three propulsion system analysis methods are reported in Tab. 3.15.

Table 3.15: Benchmarking metrics for the series hybrid propulsion system.

Simulation	Method	Topology-Specific Inputs	Estimation Error (%)	Wall Time (s)
Upstream	GPSA Framework	98	0	3.03×10^{-5}
	Prior Graph-based	—	—	—
	Hybridization Factors	56	0	4.54×10^{-5}
Downstream	GPSA Framework	98	0	2.49×10^{-5}
	Prior Graph-based	39	200	3.02×10^{-6}
	Hybridization Factors	56	0	5.39×10^{-5}

3.3.5 Test D: Series/Parallel Hybrid Electric

3.3.5.1 GPSA Framework

A series-parallel hybrid electric propulsion architecture is provided in Fig. 3.15.

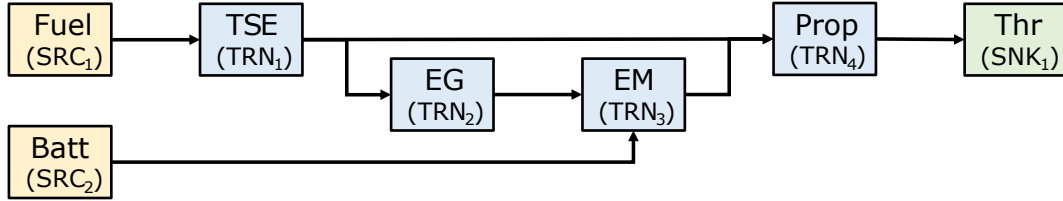


Figure 3.15: Series-Parallel hybrid electric propulsion architecture schematic.

Its architectural and upstream/downstream operational matrices are:

$$\begin{aligned}
 A &= \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
 \lambda^u &= \begin{bmatrix} 0 & 0 & 1 & & 0 & 0 & & 0 & 0 \\ 0 & 0 & 0 & & 0 & 1 & & 0 & 0 \\ 0 & 0 & 0 & \lambda_{\text{TSE,EG}} & 0 & \lambda_{\text{TSE,Prop}} & & 0 & \\ 0 & 0 & 0 & & 0 & 1 & & 0 & 0 \\ 0 & 0 & 0 & & 0 & 0 & & 1 & 0 \\ 0 & 0 & 0 & & 0 & 0 & & 0 & 1 \\ 0 & 0 & 0 & & 0 & 0 & & 0 & 0 \end{bmatrix} \\
 \lambda^d &= \begin{bmatrix} 0 & & 0 & & 0 & & 0 & 0 & 0 & 0 \\ 0 & & 0 & & 0 & & 0 & 0 & 0 & 0 \\ 1 & & 0 & & 0 & & 0 & 0 & 0 & 0 \\ 0 & & 0 & & 1 & & 0 & 0 & 0 & 0 \\ 0 & \lambda_{\text{EM,Batt}} & & & 0 & \lambda_{\text{EM,EG}} & & 0 & 0 & 0 \\ 0 & & 0 & \lambda_{\text{Prop,TSE}} & & 0 & \lambda_{\text{Prop,EM}} & 0 & 0 & 0 \\ 0 & & 0 & & 0 & & 0 & 1 & 0 & 0 \end{bmatrix}
 \end{aligned}$$

In each matrix, 49 topology-specific entries are required, resulting in 98 total entries per upstream/downstream power flow simulation.

Downstream Flow As listed in Test D, the sink requires 100 kW. Two sets of power splits must be specified. The first power split governs the power provided to the gas turbine engine and electric motor from the propeller, $\lambda_{\text{Prop,TSE}}$ and $\lambda_{\text{Prop,EM}}$, respectively. The second power

split governs the power provided to the electric motor from the battery and electric generator (EG), $\lambda_{\text{EM,Batt}}$ and $\lambda_{\text{EM,EG}}$, respectively.

As described in the power management strategy for Test D, the gas turbine engine and electric motor provide 82% and 18% of the electric motor power, respectively, requiring that:

$$\lambda_{\text{Prop,TSE}}^d = 0.82 \quad \text{and} \quad \lambda_{\text{Prop,EM}}^d = 0.18 \quad (3.43)$$

The power provided to the electric motor is split evenly between the battery and electric generator:

$$\lambda_{\text{EM,EG}}^d = 0.50 \quad \text{and} \quad \lambda_{\text{EM,Batt}}^d = 0.50 \quad (3.44)$$

All iterations of Alg. 1 are listed in Tab. 3.16. It took an average of 2.43×10^{-5} seconds to perform the simulation. The resultant power required at the fuel tank and battery is 91 and 9 kW, respectively, confirming the expected behavior for the downstream flow in Test D and corresponding to zero component-level estimation error.

Table 3.16: Downstream power flow in a SHE architecture (all values reported in kW).

Iteration	Fuel	Batt	TSE	EG	EM	Prop	Thr
0	0	0	0	0	0	0	100
1	0	0	0	0	0	100	100
2	0	0	82	0	18	100	100
3	82	9	82	9	18	100	100
4	82	9	91	9	18	100	100
5	91	9	91	9	18	100	100
6	91	9	91	9	18	100	100

Upstream Flow The fuel tank and battery supply 91 kW and 9 kW, respectively. For upstream operations, the power must be split from the gas turbine engine between the electric generator and propeller. According the the power management strategy prescribed in Test D:

$$\lambda_{\text{TSE,EG}}^u = \frac{9}{91} \quad \text{and} \quad \lambda_{\text{TSE,Prop}}^u = \frac{82}{91}$$

All iterations of Alg. 1 are listed in Tab. 3.17. First, the fuel and battery send all their power to the turboshaft engine and electric motor, respectively. Then, the turboshaft

engine power is split between the electric generator and propeller. The power from the fuel-turboshaft engine and battery-electric motor connections reach the thrust power first, followed by the remaining power that was transmitted from the turboshaft engine to the electric generator, electric motor, and propeller. Ultimately, 100 kW of power reaches the thrust power, confirming the expected behavior in Test D and resulting in zero component-level estimation error. It took an average of 2.77×10^{-5} seconds to perform the simulation.

Table 3.17: Upstream power flow in a series/parallel hybrid electric architecture (all values reported in kW).

Iteration	Fuel	Batt	TSE	EG	EM	Prop	Thr
0	91	9	0	0	0	0	0
1	91	9	91	0	9	0	0
2	91	9	91	9	9	91	0
3	91	9	91	9	18	91	91
4	91	9	91	9	18	100	91
5	91	9	91	9	18	100	100
6	91	9	91	9	18	100	100

3.3.5.2 Prior State-of-the-Art: Graph-Based Framework

There are two energy sources (fuel and battery), three power sources (turboshaft engine, electric generator, and electric motor), and one thrust source (propeller). The corresponding architecture and operational matrices utilizing the approach described in Ref. [8] are:

$$\begin{aligned}
B^{A, \text{PSES}} &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} & B^{A, \text{PSPS}} &= \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} & B^{A, \text{TSPS}} &= \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \\
B^{\lambda, \text{PSES}} &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0.5 \end{bmatrix} & B^{\lambda, \text{PSPS}} &= \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0.5 & 1 \end{bmatrix} & B^{\lambda, \text{TSPS}} &= \begin{bmatrix} 0.82 & 0 & 0.18 \end{bmatrix}
\end{aligned}$$

There are 18 topology-specific inputs in each set of architecture/operational matrices, resulting in 37 total inputs (including the entry for $B^{\lambda, \text{TS}\infty}$).

Table 3.18 lists the power loads after each corresponding matrix multiplication. The average time to compute the result was 2.68×10^{-6} seconds. Since three components are arranged in series, the required power loads are not estimated correctly, resulting in a lack of power being provided by the turboshaft engine. Again, this is a result of only two matrix multiplication steps propagating power across the power sources, preventing the electric generator power load from reaching the gas turbine engine. Thus, the component-level accuracy error is 9.89%.

Table 3.18: Power flow using the framework developed in Ref. [8] (all values reported in kW).

Step	Fuel	Battery	TSE	EG	EM	Prop
(1) $TV \rightarrow TS$	0	0	0	0	0	100
(2) $TS \rightarrow PS^*$	0	0	82	0	18	100
(3) $PS^* \rightarrow PS$	0	0	82	9	18	100
(4) $PS \rightarrow ES$	82	9	82	9	18	100

3.3.5.3 Prior State-of-the-Art: Hybridization Factors

In Ref. [81], the framework does not allow power to be siphoned from the gas turbine engine and returned back to the same shaft (via the electric generator, battery, and electric motor connections); power is always transmitted to a different shaft, as implied by the definition of the hybridization factor between two power shafts ($\varphi = P_1/(P_1 + P_2)$). The series/parallel hybrid electric propulsion system violates the implicit assumptions within the hybridization factor definition, preventing the propulsion system from being modeled accurately.

3.3.5.4 Summary of Benchmarking Metrics Recorded

The benchmarking metrics across all simulations and three propulsion system analysis methods are reported in Tab. 3.19.

Table 3.19: Benchmarking metrics for the series/parallel hybrid electric propulsion system.

Simulation	Method	Topology-Specific Inputs	Estimation Error (%)	Wall Time (s)
Upstream	GPSA Framework	98	0	2.77×10^{-5}
	Prior Graph-based	—	—	—
	Hybridization Factors	—	—	—
Downstream	GPSA Framework	98	0	2.43×10^{-5}
	Prior Graph-based	37	9.89	2.68×10^{-6}
	Hybridization Factors	—	—	—

3.3.6 Test E: Distributed Electric Propulsion

A distributed electric propulsion architecture is provided in Fig. 3.16.

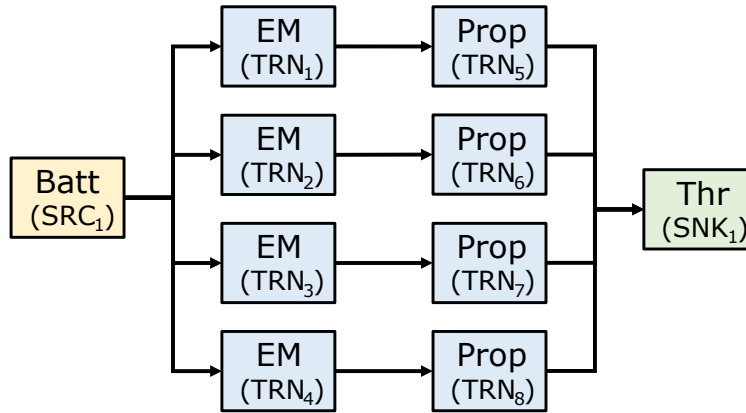


Figure 3.16: Distributed electric propulsion architecture schematic.

3.3.6.1 GPSA Framework

In the power management strategy defined within Test E, the power is split evenly between the thrust power and all propellers during downstream operations, and then split evenly between the battery and all electric motors during upstream operations. Since the power is split evenly, one power split can be defined in the downstream and upstream directions, $\lambda_{\text{Thr,Prop}}$ and $\lambda_{\text{Batt,EM}}$, respectively, and assigned to all corresponding connections. Thus, the

architecture and upstream/downstream operational matrices are:

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\lambda^u = \begin{bmatrix} 0 & \lambda_{\text{Batt,EM}} & \lambda_{\text{Batt,EM}} & \lambda_{\text{Batt,EM}} & \lambda_{\text{Batt,EM}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\lambda^d = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda_{\text{Thr,Prop}} & \lambda_{\text{Thr,Prop}} & \lambda_{\text{Thr,Prop}} & \lambda_{\text{Thr,Prop}} & 0 & 0 \end{bmatrix}$$

In each matrix, 100 topology-specific entries are required, resulting in 200 total entries per upstream/downstream power flow simulation.

Downstream Flow As listed in Test E, the sink requires 100 kW and power is split evenly across all four propellers, requiring that $\lambda_{\text{Thr,Prop}}^d = 0.25$ for each propeller.

All iterations of Alg. 1 are listed in Tab. 3.20. It took an average of 3.24×10^{-5} seconds to perform the simulation. After splitting the thrust power across all four propellers, it is propagated back to its respective electric motor. The overall power load is then aggregated at the battery, requiring 100 kW and matching the expected behavior outlined in Test E. Thus, there is no component-level estimation error.

Table 3.20: Downstream power flow in a DEP architecture (all values reported in kW).

Iteration	Batt	EM-1	EM-2	EM-3	EM-4	Prop-1	Prop-2	Prop-3	Prop-4	Thr
0	0	0	0	0	0	0	0	0	0	100
1	0	0	0	0	0	25	25	25	25	100
2	0	25	25	25	25	25	25	25	25	100
3	100	25	25	25	25	25	25	25	25	100
4	100	25	25	25	25	25	25	25	25	100

Upstream Flow The battery starts with 100 kW of power available. As prescribed by the power management strategy in Test E, the battery power is split evenly between all electric

motors, resulting in $\lambda_{\text{Batt,EM}}^u = 0.25$.

All iterations of Alg. 1 are listed in Tab. 3.21. First, the battery power is split evenly between all electric motors. Then, the power from electric motor power is propagated to its respective propeller before being aggregated at the thrust power. Ultimately, the thrust power receives 100 kW, matching the expected behavior defined in the test and resulting in zero component-level estimation error. It took an average of 4.45×10^{-5} seconds to perform the simulation.

Table 3.21: Upstream power flow in a DEP architecture (all values reported in kW).

Iteration	Batt	EM-1	EM-2	EM-3	EM-4	Prop-1	Prop-2	Prop-3	Prop-4	Thr
0	100	0	0	0	0	0	0	0	0	0
1	100	25	25	25	25	0	0	0	0	0
2	100	25	25	25	25	25	25	25	25	0
3	100	25	25	25	25	25	25	25	25	100
4	100	25	25	25	25	25	25	25	25	100

3.3.6.2 Prior State-of-the-Art: Graph-Based Framework

In the propulsion system, there is one energy source (battery), four power sources (electric motors), and four thrust sources (propellers). The corresponding architecture and operational matrices utilizing the approach described in Ref. [8] are:

$$B^{A,\text{PSES}} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad B^{A,\text{PSPS}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad B^{A,\text{TSPPS}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B^{\lambda,\text{PSES}} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad B^{\lambda,\text{PSPS}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad B^{\lambda,\text{TSPPS}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad B^{\lambda,\text{TS}\infty} = \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{bmatrix}^T$$

Since there are four thrust sources, $B^{\lambda, \text{TS}\infty}$ is no longer 1, and must evenly split the required power between all four propellers, resulting in a 1-by-4 matrix with each entry as 0.25. There are 76 total topology-specific entries required: 36 in the architecture matrices and 40 in the operational matrices.

Table 3.22 lists the power loads after each corresponding matrix multiplication. All power loads are estimated correctly, corresponding to zero component-level estimation error. The average time to compute the result was 1.01×10^{-4} seconds.

Table 3.22: Power flow using the framework developed in Ref. [8] (all values reported in kW).

Step	Battery	EM-1	EM-2	EM-3	EM-4	Prop-1	Prop-2	Prop-3	Prop-4
(1) $TV \rightarrow \text{TS}$	0	0	0	0	0	25	25	25	25
(2) $\text{TS} \rightarrow \text{PS}^*$	0	25	25	25	25	25	25	25	25
(3) $\text{PS}^* \rightarrow \text{PS}$	0	25	25	25	25	25	25	25	25
(4) $\text{PS} \rightarrow \text{ES}$	100	25	25	25	25	25	25	25	25

3.3.6.3 Prior State-of-the-Art: Hybridization Factors

Downstream Operations The linear system defining downstream operations is:

$$\begin{bmatrix}
 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\
 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & 1 \\
 0 & 0 & 0 & 0 & 0 & -0.75 & 0.25 & 0.25 & 0.25 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0.25 & -0.75 & 0.25 & 0.25 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0.25 & 0.25 & -0.75 & 0.25 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{bmatrix}
 \begin{bmatrix}
 P_{\text{Batt}} \\
 P_{\text{EM-1}} \\
 P_{\text{EM-2}} \\
 P_{\text{EM-3}} \\
 P_{\text{EM-4}} \\
 P_{\text{Prop-1}} \\
 P_{\text{Prop-2}} \\
 P_{\text{Prop-3}} \\
 P_{\text{Prop-4}} \\
 P_{\text{Thr}}
 \end{bmatrix}
 =
 \begin{bmatrix}
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 100
 \end{bmatrix}
 \quad (3.45)$$

The first six equations define the conservation of energy equations between interconnected components. The following three equations define three hybridization factors, splitting the

power between each propeller equally:

$$\varphi_i = \frac{P_{\text{Prop},i}}{\sum_{j=1}^4 P_{\text{Prop},j}}, i = \{1, 2, 3\}$$

In this case, the hybridization factor defines the power split between components on different shafts. However, its definition is modified to accommodate more than two components, exhibited by the summation within the denominator.

In this system of equations, there are 100 entries in the square matrix on the left-hand side and 10 entries in the column vector on the right-hand side, resulting in 110 total topology-specific inputs. After solving the linear system the corresponding power loads are:

$$\begin{bmatrix} P_{\text{Battery}} \\ P_{\text{EM-1}} \\ P_{\text{EM-2}} \\ P_{\text{EM-3}} \\ P_{\text{EM-4}} \\ P_{\text{Prop-1}} \\ P_{\text{Prop-2}} \\ P_{\text{Prop-3}} \\ P_{\text{Prop-4}} \\ P_{\text{Thr}} \end{bmatrix} = \begin{bmatrix} 100 \\ 25 \\ 25 \\ 25 \\ 25 \\ 25 \\ 25 \\ 25 \\ 25 \\ 100 \end{bmatrix} \quad (3.46)$$

The result matches the expected behavior, resulting in zero component-level estimation error. It took 5.70×10^{-5} seconds to compute the result.

Upstream Operations The linear system defining upstream operations is:

$$\begin{bmatrix}
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\
 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & 1 \\
 0 & -0.75 & 0.25 & 0.25 & 0.25 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0.25 & -0.75 & 0.25 & 0.25 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0.25 & 0.25 & -0.75 & 0.25 & 0 & 0 & 0 & 0 & 0
 \end{bmatrix}
 \begin{bmatrix}
 P_{\text{Batt}} \\
 P_{\text{EM-1}} \\
 P_{\text{EM-2}} \\
 P_{\text{EM-3}} \\
 P_{\text{EM-4}} \\
 P_{\text{Prop-1}} \\
 P_{\text{Prop-2}} \\
 P_{\text{Prop-3}} \\
 P_{\text{Prop-4}} \\
 P_{\text{Thr}}
 \end{bmatrix}
 =
 \begin{bmatrix}
 100 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0
 \end{bmatrix} \quad (3.47)$$

The first equation represents the boundary condition that the power available at the battery is 100 kW. The second through seventh equations define the conservation of energy equations for the distributed electric propulsion system. The final three equations define hybridization factors between the battery and electric motors in a similar fashion to those for the downstream power splits:

$$\varphi_i = \frac{P_{\text{EM},i}}{\sum_{j=1}^4 P_{\text{EM},j}}, i = \{1, 2, 3\}$$

Recall that the hybridization factor definition is modified to account for the fact that power is being split across four components.

Again, 110 topology-specific inputs are required to define the propulsion system operation. Solving the linear system of equations returns the same result as demonstrated during Downstream Operations, and also corresponds to zero component-level estimation error. It took approximately 5.69×10^{-5} seconds to solve.

3.3.6.4 Summary of Benchmarking Metrics Recorded

The benchmarking metrics across all simulations and three propulsion system analysis methods are reported in Tab. 3.23.

Table 3.23: Benchmarking metrics for the distributed electric propulsion system.

Simulation	Method	Topology-Specific Inputs	Estimation Error (%)	Wall Time (s)
Upstream	GPSA Framework	200	0	4.45×10^{-5}
	Prior Graph-based	–	–	–
	Hybridization Factors	110	0	5.69×10^{-5}
Downstream	GPSA Framework	200	0	3.24×10^{-5}
	Prior Graph-based	76	0	1.01×10^{-4}
	Hybridization Factors	110	0	5.70×10^{-5}

3.3.7 Test F: Multiple Sinks

A series hybrid propulsion architecture with power off-take is provided in Fig. 3.17. In this context, the power off-take is included to power the aircraft’s subsystems (e.g., navigation equipment, avionics, passenger entertainment systems) and in addition to the thrust power required for sustaining flight. Thus, the power off-take increases the power demand from the fuel, but the battery’s power required should remain the same as the SHE propulsion architecture previously analyzed in Test C.

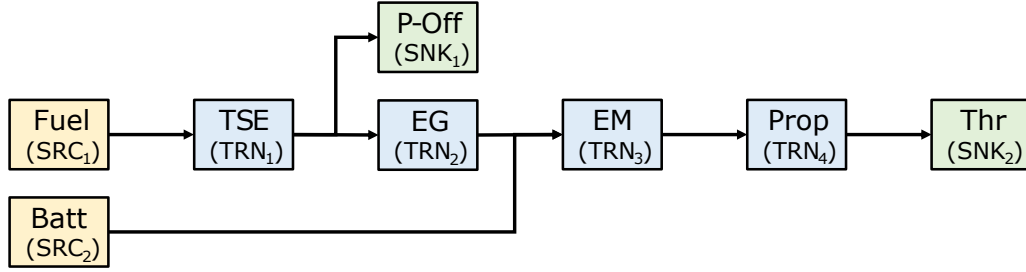


Figure 3.17: Series hybrid electric propulsion architecture with power off-take schematic.

3.3.7.1 GPSA Framework

The architecture and upstream/downstream operational matrices are:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \lambda^u = \begin{bmatrix} 0 & 0 & 1 & & 0 & 0 & 0 & & 0 & 0 \\ 0 & 0 & 0 & & 0 & 1 & 0 & & 0 & 0 \\ 0 & 0 & 0 & \lambda_{\text{TSE,EG}} & 0 & 0 & & \lambda_{\text{TSE,P-Off}} & 0 & \\ 0 & 0 & 0 & & 0 & 1 & 0 & & 0 & 0 \\ 0 & 0 & 0 & & 0 & 0 & 1 & & 0 & 0 \\ 0 & 0 & 0 & & 0 & 0 & 0 & & 0 & 1 \\ 0 & 0 & 0 & & 0 & 0 & 0 & & 0 & 0 \\ 0 & 0 & 0 & & 0 & 0 & 0 & & 0 & 0 \end{bmatrix}$$

$$\lambda^d = \begin{bmatrix} 0 & & 0 & 0 & & 0 & 0 & 0 & 0 & 0 \\ 0 & & 0 & 0 & & 0 & 0 & 0 & 0 & 0 \\ 1 & & 0 & 0 & & 0 & 0 & 0 & 0 & 0 \\ 0 & & 0 & 1 & & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda_{\text{EM,Batt}} & 0 & \lambda_{\text{EM,EG}} & & 0 & 0 & 0 & 0 & 0 \\ 0 & & 0 & 0 & & 0 & 1 & 0 & 0 & 0 \\ 0 & & 0 & 1 & & 0 & 0 & 0 & 0 & 0 \\ 0 & & 0 & 0 & & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

In each matrix, 64 topology-specific entries are required per matrix, resulting in 128 total entries required for each upstream/downstream power flow simulation.

Downstream Flow As provided in Test F, the thrust power and power off-take require 100 and 50 kW, respectively. The downstream operational matrix requires two power splits to be specified:

$$\lambda_{\text{EM,EG}}^d = 0.82 \quad \text{and} \quad \lambda_{\text{EM,Batt}}^d = 0.18$$

All iterations of Alg. 1 are listed in Tab. 3.24. The power from the power off-take is propagated to the fuel in two iterations. Due to the differing path lengths, the power load from the thrust power reaches the fuel after five iterations. While power is transferred from the thrust power to fuel, some power is split from the electric motor to the battery based on the selected power splits. The resultant power loads at the fuel tank and battery are 132 and 18 kW, respectively, matching the expected behavior from the test. Thus, there is zero component-level estimation error using the GPSA Framework. It took an average of 3.09×10^{-5} seconds to simulate the power flow.

Table 3.24: Downstream power flow with multiple sinks (all values reported in kW).

Iteration	Fuel	Batt	TSE	EG	EM	Prop	P-Off	Thr
0	0	0	0	0	0	0	50	100
1	0	0	50	0	0	100	50	100
2	50	0	50	0	100	100	50	100
3	50	18	50	82	100	100	50	100
4	50	18	132	82	100	100	50	100
5	132	18	132	82	100	100	50	100
6	132	18	132	82	100	100	50	100

Upstream Flow Test F initializes the simulation with 132 and 18 kW supplied by the fuel tank and battery, respectively. Since the turboshaft engine connects to two components, its power splits to the other components must be defined. The test prescribes that 50 kW be transferred to the power off-take and the remaining 82 kW be transferred to the electric generator:

$$\lambda_{\text{TSE,P-Off}}^u = \frac{50}{132} \quad \text{and} \quad \lambda_{\text{TSE,EG}}^u = \frac{82}{132}$$

All other upstream power splits are 1 because the remaining components only connect to one other component, as evidenced by the upstream operational matrix.

Table 3.25 illustrates the power flow propagation. Using Alg. 1, power first flows from the fuel and battery to the turboshaft engine and electric motor, respectively. After splitting the power at the turboshaft engine between the power off-take and electric generator, the flow to the power off-take is complete. The remaining iterations propagate power from the electric

generator to the thrust power until converging. The thrust power and power off-take receive 82 and 50 kW, respectively, aligning with the expected behavior prescribed in Test F and corresponding to zero component-level estimation error. It took an average of 2.48×10^{-5} seconds to simulate the power flow.

Table 3.25: Upstream power flow with multiple sinks (all values reported in kW).

Iteration	Fuel	Batt	TSE	EG	EM	Prop	P-Off.	Thr
0	132	18	0	0	0	0	0	0
1	132	18	132	0	18	0	0	0
2	132	18	132	82	18	18	50	0
3	132	18	132	82	100	18	50	18
4	132	18	132	82	100	100	50	18
5	132	18	132	82	100	100	50	100
6	132	18	132	82	100	100	50	100

3.3.7.2 Prior State-of-the-Art: Graph-based Framework

In the graph-based framework proposed by in Ref. [8], only energy, power, and thrust sources can be modeled in a propulsion system. If a power off-take were to be modeled, it could be included as an additional thrust source or power source. However, since the framework prevents more than two components from being arranged in series, the power off-take would need to be defined such that its power load can be propagated to the energy sources, possibly resulting in different power/thrust source classifications depending on the propulsion system architecture being analyzed. Therefore, the system cannot be modeled accurately.

3.3.7.3 Prior State-of-the-Art: Hybridization Factors

Downstream Operations A hybridization factor defines how the power is split between the fuel and battery:

$$\Phi = \frac{P_{\text{Batt}}}{P_{\text{Fuel}} + P_{\text{Batt}}} = \frac{18}{132 + 18} = \frac{18}{150}$$

The linear system defining downstream operations is:

$$\begin{bmatrix}
 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & -1 & 0 & 0 & 0 & -1 \\
 0 & -1 & 0 & -1 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
 \frac{18}{150} & \frac{18}{150} & -1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{bmatrix}
 \begin{bmatrix}
 P_{\text{Fuel}} \\
 P_{\text{Batt}} \\
 P_{\text{GTE}} \\
 P_{\text{EG}} \\
 P_{\text{EM}} \\
 P_{\text{Prop}} \\
 P_{\text{Thr}} \\
 P_{\text{P-Off}}
 \end{bmatrix}
 =
 \begin{bmatrix}
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 100 \\
 50
 \end{bmatrix}
 \quad (3.48)$$

The first five equations represent the energy conservation equations. The sixth equation defines the hybridization factor between the fuel and battery. The final two equations define the boundary conditions for the power flow simulation; the thrust power and power off-take require 100 and 50 kW, respectively.

In the system of equations, there are 64 entries in the square matrix on the left-hand side and 8 entries in the column vector on the right-hand side, resulting in 72 total topology-specific inputs. After solving the linear system, the corresponding power loads are:

$$\begin{bmatrix}
 P_{\text{Fuel}} \\
 P_{\text{Batt}} \\
 P_{\text{TSE}} \\
 P_{\text{EG}} \\
 P_{\text{EM}} \\
 P_{\text{Prop}} \\
 P_{\text{Thr}} \\
 P_{\text{P-Off}}
 \end{bmatrix}
 =
 \begin{bmatrix}
 132 \\
 18 \\
 132 \\
 82 \\
 100 \\
 100 \\
 100 \\
 50
 \end{bmatrix}
 \quad (3.49)$$

The result matches the expected behavior, corresponding to zero component-level estimation error. It took an average of 7.14×10^{-5} seconds to compute the result.

Upstream Operations In the upstream direction, a hybridization factor defines the fraction of power split from the gas turbine engine to the electric generator and power off-take:

$$\varphi = \frac{P_{\text{P-Off}}}{P_{\text{EG}} + P_{\text{P-Off}}} = \frac{50}{82 + 50} = \frac{50}{132}$$

The linear system defining upstream operations is:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \frac{50}{132} & 0 & 0 & 0 & \frac{50}{132} - 1 \end{bmatrix} \begin{bmatrix} P_{\text{Fuel}} \\ P_{\text{Batt}} \\ P_{\text{GTE}} \\ P_{\text{EG}} \\ P_{\text{EM}} \\ P_{\text{Prop}} \\ P_{\text{Thr}} \\ P_{\text{P-Off}} \end{bmatrix} = \begin{bmatrix} 132 \\ 18 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.50)$$

The first two equations represent boundary conditions; the power available from the fuel and battery is 132 and 18 kW, respectively. The following five equations represent the energy conservation equations between connected components. The final equation defines the hybridization factor between the power off-take and the electric generator.

Again, 72 topology-specific inputs are required to define the propulsion system architecture and its operations. Solving the linear system of equations returns the same result as demonstrated during Downstream Operations, and also corresponds to zero component-level estimation error. Additionally, it took an average of 4.68×10^{-5} seconds to solve the linear system.

3.3.7.4 Summary of Benchmarking Metrics Recorded

The benchmarking metrics across all simulations and three propulsion system analysis methods are reported in Tab. 3.26.

Table 3.26: Benchmarking metrics for the series hybrid propulsion system with power off-take.

Simulation	Method	Topology-Specific Inputs	Estimation Error (%)	Wall Time (s)
Upstream	GPSA Framework	128	0	2.48×10^{-5}
	Prior Graph-based	—	—	—
	Hybridization Factors	72		4.68×10^{-5}
Downstream	GPSA Framework	128	0	3.09×10^{-5}
	Prior Graph-based	—	—	—
	Hybridization Factors	72	0	7.14×10^{-5}

3.3.8 Conclusions

In Experiment 2.0.1, upstream and downstream power flows were simulated on six different propulsion system architectures to validate the GPSA Framework and benchmark it against existing state-of-the-art propulsion system analysis techniques. The six propulsion system architectures included:

1. **Conventional Propulsion Architecture**, defined by a single power flow path and provided an elementary baseline for modeling a propulsion system in the GPSA Framework;
2. **Parallel Hybrid Propulsion Architecture**, demonstrating that the GPSA Framework can model a propulsion system with two sources and, during power flow simulations, add/split power at a junction involving multiple power flow paths;
3. **Series Hybrid Propulsion Architecture**, demonstrating that the GPSA Framework can accurately model a power flow with differing path lengths;
4. **Series/Parallel Hybrid Propulsion Architecture**, demonstrating that the GPSA Framework can split power at different junctions in the upstream/downstream directions with intuitively defined power splits;
5. **Distributed Electric Propulsion Architecture**, demonstrating that the GPSA Framework can analyze propulsion system architectures with multiple, repeated components of the same type; and

6. Series Hybrid with Power Off-take, demonstrating that the GPSA Framework can model propulsion systems with multiple sinks.

In the experiment, accuracy was measured by the component-level estimation error relative to the baseline behavior described for each test in Section 3.3.1. Topology-specific model setup effort was measured by the total size of all matrices defined to simulate the power flows. Lastly, computational efficiency was measured by computing the average wall time to simulate the power flow across five repeated runs.

Overall, the GPSA Framework modeled all propulsion system architectures with equal or better accuracy than the two other state-of-the-art propulsion system analysis methods in the study [8, 81], returning zero component-level estimation error across all tests. The other state-of-the-art propulsion system analysis methods returned zero component-level estimation error for the conventional, parallel hybrid, and distributed electric propulsion architectures, but one or both methods failed to accurately model the power flow on the series hybrid architecture (with and without a power off-take) and the series/parallel hybrid architecture. The prior graph-based framework [8] was unable to accurately model the aforementioned propulsion system architectures because its power propagation algorithm only accommodates two power-transmitting components to be arranged in series, resulting in inaccurate power loads required for the series and series/parallel hybrid propulsion system architectures. Furthermore, its inability to explicitly model other power sinks hindered its ability to model the series hybrid propulsion system with a power off-take. The GPSA Framework overcomes both limitations by abstracting any propulsion system architecture as a graph and simulates power flows with a fixed point iteration, removing any limitations on the number of components arranged in series, as well as permitting multiple sinks to be modeled in a propulsion architecture.

Utilizing hybridization factors is valuable for the propulsion system architectures included in Ref. [81], but is not easily generalizable to others. For example, in the series/parallel hybrid propulsion system architecture, power is siphoned from the gas turbine engine to power an electric generator. Then, power is returned to the same shaft from the electric motor output, contradicting the implicit assumption in Ref. [81] that a power flow does not re-join the same shaft that it was previously split from.

Despite the ability to model more advanced propulsion system architectures accurately, the GPSA Framework required the most topology-specific model setup because n -by- n matrices define the propulsion system architecture and its corresponding operational strategy, with n representing the number of components in the propulsion system architecture. In the isolated

tests within Experiment 2.0.1, each propulsion system architecture was defined with separate architecture and upstream/downstream operational matrices. The matrices are typically sparse, containing few nonzero entries relative to their size.

However, the structure of the architecture and operational matrices in the GPSA Framework is more advantageous for generating different propulsion system architectures with the same set of components. For example, consider the series and series/parallel hybrid propulsion architectures in Tests C and D. While the component arrangements, power management strategy, and simulated behavior are different, the matrix sizes are the same. Thus, the matrices can be easily modified to generate different propulsion system architectures and power management strategies, rendering it useful for system architecture optimization.

Prior state-of-the-art methods required fewer topology-specific inputs than those required by the GPSA Framework. The prior graph-based implementation introduced in Ref. [8] required the least topology-specific model setup because multiple, smaller matrices define connections between energy, power, and thrust sources, eliminating the need to generate sparse matrices with many zero entries. Defining the energy conservation equations, hybridization factors, and boundary conditions in a linear system, as introduced in Ref. [81], also required less topology-specific model setup than the GPSA Framework because one sparse matrix and one column vector was defined to simulate the power flow.

Although not captured in Experiment 2.0.1, the GPSA Framework also provides an intuitive method for systematically defining propulsion system architectures and their corresponding power management strategies. Power splits are defined at each junction rather than defining hybridization factors that span different power flow paths in the propulsion system, as done in Ref. [81]. Furthermore, each row in the operational matrix represents an energy conservation equation, eliminating the need to manually define them. Additionally, encoding the power splits in one operational matrix better informs the engineer how power from one component flows to all others in the propulsion system, limiting obfuscations of the energy conservation equations across multiple matrices, as observed in the prior graph-based framework developed in Ref. [8] for series hybrid propulsion systems.

Neither the GPSA Framework nor the prior state-of-the-art propulsion system were more computationally efficient than the other. The wall times required to simulate a power flow are comparable and do not pose a significant computational burden. The power flow simulation in the GPSA Framework performs iterative matrix multiplication, resulting in approximately $\mathcal{O}(n^3)$ floating-point operations. Solving a linear system of equations, as done in Ref. [81], also requires approximately $\mathcal{O}(n^3)$ floating-point operations. In contrast, the

prior graph-based framework requires fewer floating-point operations because the matrices being multiplied are smaller. However, memory is accessed more frequently to retrieve and multiply the smaller matrices, possibly resulting in more/less computational time depending on the propulsion system architecture analyzed.

Therefore, based on the results presented, the GPSA Framework reproduces benchmark component power loads and overall energy conservation equations across representative electrified propulsion architectures within predefined accuracy thresholds while not necessarily reducing the topology-specific model setup effort and execution time relative to existing state-of-the-art propulsion system analysis implementations. Improved accuracy is attained by abstracting the propulsion system as a graph, accommodating any type of propulsion system architecture and power management strategy. Additionally, the structure of the architecture and operational matrices enables seamless adoption of the GPSA Framework to model any propulsion system architecture without the need to derive any energy conservation equations or assemble multiple matrices representing connections between different component types. However, the sparse matrices required to abstract the propulsion system as a graph may increase model setup effort and execution time. Although not captured in Experiment 2.0.1, but based on the author’s experience, the additional effort to define the sparse matrices typically outweighs the time required to derive the energy conservation equations and corresponding hybridization factors.

Overall, the GPSA Framework represents a significant advancement in the systematic analysis of electrified propulsion systems, providing a unified and adaptable methodology for a wide range of configurations and operational strategies that can also be adapted to other novel propulsion systems in the future. The success from Experiment 2.0.1 validates the GPSA Framework’s behavior at the subsystem-level, with a focus on formally representing and analyzing propulsion system architectures, and demonstrates its improved effectiveness to accurately and efficiently model advanced propulsion system architectures relative to prior state-of-the-art methods.

3.4 Chapter Summary

Prior limitations in existing propulsion system analysis frameworks led to Research Question 2.0, focusing on how propulsion system architectures can be formally represented and analyzed in a configuration-agnostic manner. Based on the literature review conducted, a graph-based framework was shown to be superior than other approaches that define hybridization factors

at critical junctions in the propulsion system. However, existing state-of-the-art graph-based propulsion system analysis frameworks imposed limitations on the number of components that can be arranged in series and excluded explicit power off-takes from being modeled. To this end, Hypothesis 2.0 proposed that propulsion system architectures can be formally represented and analyzed using a graph-based framework with component constitutive models at each vertex and power transfer relationships along the edges.

The GPSA Framework was then constructed to represent and analyze advanced propulsion system architectures in a configuration-agnostic manner with the capabilities described in Hypothesis 2.0. In the GPSA Framework, propulsion architectures are encoded via an architecture matrix (an abstract of an Adjacency Matrix from Graph Theory), delineating component connectivity, while complementary operational and efficiency matrices (abstractions of Weighted Adjacency Matrices from Graph Theory) quantify power distribution strategies and transmission losses, respectively. A fixed-point iteration algorithm was developed to simulate both uni- and bi-directional power flows, enabling the evaluation of complex operational modes such as in-flight battery charging. Furthermore, the GPSA Framework overcomes previous limitations on serial power-transmitting components and explicit power off-take modeling, representing an advancement in the state-of-the-art.

Hypothesis 2.0 was tested in Experiment 2.0.1, and involved validating the GPSA Framework against six different conventional and advanced propulsion system architectures. The propulsion system architectures were also evaluated using two prior state-of-the-art approaches: 1) a prior graph-based framework introduced in Ref. [8]; and 2) defining hybridization factors and assembling a linear system of equations to solve for the power loads at each component, introduced in Ref. [81]. For each propulsion system analysis method, the number of topology-specific user inputs, component-level estimation error, and wall time to perform the simulation was recorded. It was found that the GPSA Framework analyzes both upstream and downstream operations most accurately, as reflected in the zero component-level estimation error reported throughout Experiment 2.0.1. While the number of topology-specific user inputs is large due to the sparse matrix generated by the graph abstraction and results in possibly longer simulation times, the marginally larger computational overhead is a reasonable tradeoff for the improved accuracy and increased breadth of propulsion system architectures that can be modeled with the GPSA Framework. Additionally, though not captured in Experiment 2.0.1, defining a power flow simulation with the GPSA Framework is more intuitive by defining the power splits at each junction, circumventing the need to derive energy conservation equations for the propulsion system and overcoming the small penalty in

prescribing more topology-specific user inputs. The additional topology-specific user inputs required predominantly arise from defining a sparse matrix, and is not necessarily attributed to the complexity of the analysis technique.

Thus, the GPSA Framework, with its component constitutive models at each vertex and power transfer relationships along the edges, accurately models conventional and advanced propulsion systems, emphasizing its usefulness in configuration-agnostic design. Beyond the subsystem-level analysis developed through the GPSA Framework, optimal propulsion system architecture synthesis requires further integration at the system-level, necessitating an aircraft analysis to predict system-level performance and quantify tradeoffs between propulsion system architectures. Such needs are fulfilled in Chapter 4, focusing on integrating the GPSA Framework into a parametric aircraft sizing tool.

Chapter 4

Parametric Aircraft Sizing Tool

4.1 Introduction

The literature review revealed that existing aircraft analysis tools require component/subsystem definitions, including geometries, preventing rapid analysis for early-phase conceptual design. Such information may not be known if disparate, high-level aircraft concepts – such as tube-and-wing or blended wing body configurations – are being considered. Furthermore, there are limited open-source aircraft sizing tools to size an aircraft in a configuration-agnostic manner and accommodate any type of propulsion system. Thus, a parametric aircraft sizing tool is needed to size an aircraft concept at a high-level with minimal information about the configuration. To provide a configuration-agnostic propulsion system analysis, the GPSA Framework developed in Chapter 3 must be integrated into the tool.

4.1.1 Research Questions, Hypotheses, and Experiments

This chapter continues exploring Research Question and Hypothesis 2.0 in the context of aircraft design and analysis for estimating system-level performance and quantifying tradeoffs between the system architectures analyzed. In Chapter 3, the GPSA Framework was developed to analyze any propulsion system architecture, focusing on a subsystem-level analysis only. Experiment 2.0.1 verified the GPSA Framework’s behavior and confirmed that it can accurately represent and analyze any propulsion architecture in a systematic manner, answering the first half of Research Question 2.0. However, modeling the propulsion system only fails to consider the interactions with other systems in the aircraft and its overall performance, requiring the GPSA Framework to be integrated into a parametric aircraft sizing tool. Together, the GPSA Framework and parametric aircraft sizing tool can predict an aircraft’s system-level performance, accounting for both the propulsion system design and

its impacts on the overall aircraft design, enabling the latter half of Research Question 2.0 to be answered. First, the GPSA Framework is integrated into the Future Aircraft Sizing Tool (FAST), a MATLAB-based parametric aircraft sizing tool that enables design space exploration and system architecture optimization studies [177]. Then, Experiment 2.0.2 validates that system-level performance can be accurately estimated for both conventional and advanced aircraft concepts, providing a configuration-agnostic analysis for the aircraft (system-level) and its propulsion system (subsystem-level).

Experiment 2.0.2: Verify that the GPSA Framework integration into FAST enables system-level performance to be accurately estimated for both conventional and advanced aircraft architectures under the same top-level aircraft requirements, technology levels, and design mission profile.

1. Define a representative benchmark suite including both conventional and advanced aircraft concepts.
2. Analyze each aircraft architecture in FAST, ensuring that the predicted block fuel and maximum takeoff weight are within a 3% relative error to those published in the literature.

In Experiment 2.0.2, conventional and advanced aircraft concepts are modeled by matching the top-level aircraft requirements and technology levels available in the literature. Although the mission profile may not be fully known, one mimicking nominal flight operations can be developed, consisting of multiple climb phases, a cruise phase, multiple descent phases, and the necessary reserve missions (a 100 nautical-mile diversion and 30 minute loiter). Then, the system-level performance parameters of the sized aircraft (MTOW and block fuel) can be compared to those in the literature, serving as a means to validate the computational procedure implemented in FAST. Since the analyses in FAST leverage first-order physics- and data-based models, there may be modest system-level performance estimation error propagated through the analysis. Therefore, a modeling accuracy threshold of 3% error was considered acceptable for an early-phase conceptual aircraft design tool. The aircraft model can also be tuned to minimize the MTOW and block fuel estimation error, ensuring that modeling error resides within the 3% threshold prescribed.

Upon successfully completing Experiment 2.0.2, the GPSA Framework is validated within the context of an aircraft sizing code and ensures its ability to estimate the system-level performance of conventional and advanced aircraft concepts. The validated code also provides

a baseline framework for the physics-based analysis embedded within the system architecture optimization framework described in Chapter 6.

After validating that the FAST-GPSA Framework integration accurately models conventional and advanced aircraft concepts, an additional experiment is performed to demonstrate its ability to quantify system-level performance tradeoffs, answering the latter half of Research Question 2.0. The demonstration is described in Experiment 2.0.3.

Experiment 2.0.3: Demonstrate that the GPSA Framework integration into FAST enables system-level performance tradeoffs to be quantified for aircraft designed under the same top-level performance requirements and mission profile with varying propulsion system architectures.

1. Select a representative conventional aircraft to be analyzed in FAST.
2. Define a suite of advanced propulsion system architectures and corresponding power management strategies to be integrated into the representative aircraft.
3. For each propulsion system architecture and corresponding power management strategy, size the aircraft in FAST.
4. For all aircraft architectures analyzed, compare the system-level performance parameters such as fuel burn, maximum takeoff weight, energy expenditure, etc. to quantify system-level performance tradeoffs.

In Experiment 2.0.3, the conventional aircraft utilized in Experiment 2.0.2 (an already validated model) will be remodeled with the same top-level aircraft requirements and technology levels but with different advanced propulsion system architectures installed. The differences in system-level performance can be attributed to the architectural design decisions made since both the GPSA Framework and FAST were validated to accurately model conventional and advanced propulsion systems and aircraft concepts, respectively. Thus, the effects of installing different advanced propulsion system architectures on a known aircraft concept can be quantified, enabling system-level tradeoffs to be identified.

4.1.2 Chapter Outline

The remainder of the chapter is as follows. First, FAST’s aircraft sizing and performance analyses are introduced in Section 4.2, illustrating the methods utilized and critical assumptions made in the underlying analysis. Then, Experiment 2.0.2 is performed in Section 4.3,

validating FAST’s ability to accurately model both conventional and advanced aircraft concepts. After validating FAST, Experiment 2.0.3 is performed in Section 4.4 to quantify system-level tradeoffs, further exhibiting FAST’s ability to estimate system-level performance in a configuration-agnostic manner. System-level performance parameters are recorded from the sizing study, enabling tradeoffs to be identified between the multiple electrified propulsion system architectures and power management strategies considered. Lastly, Section 4.5 concludes the chapter, summarizes the key contributions, and reviews the findings from Experiments 2.0.2 and 2.0.3.

4.2 Aircraft Sizing Tool Modules

In this section, FAST’s core features are described, with an emphasis on its aircraft sizing and performance capabilities. At large, FAST’s core capabilities and distinguishing features include:

1. A rapid aircraft sizing environment to facilitate design space exploration and sensitivity studies for conventional and electrified aircraft (including the ability to add other component-level models for concepts involving hydrogen or other novel fuels), with designs usually converging in less than one-minute;
2. An energy-based mission analysis, requiring no assumptions about the aircraft’s propulsion architecture, thus opening the possibility to compare aircraft concepts with different configurations (tube-and-wing, blended wing body, etc.) and propulsion architectures at a conceptual design-level;
3. The FAST Aerobase [178], a database of over 450 conventional, kerosene-powered aircraft to drive regressions for predicting unknown aircraft performance parameters and propulsion system sizing estimates [179];
4. Simplified turbofan and turboprop models combined with historical data regressions for accurate engine performance analysis and fuel burn estimates [177]; and
5. A visualization toolbox to render drawings of an aircraft concept both inside and outside the aircraft analysis [180].

With the aforementioned features, FAST enables the rapid analysis and comparison of aircraft concepts with different configurations (both at an aircraft-level and a propulsion system-level) during conceptual design.

Since FAST was developed collaboratively between multiple researchers in the Integrated Design of Efficient Aerospace Systems Lab, some of its distinguishing features were created by other contributors. The first two distinguishing features – the overall rapid sizing environment and the energy-based mission analysis – were primarily developed by the author; his collaborators developed the data-based analyses, propulsion system models for engine on-design analysis, and visualization capabilities (as indicated by the citations in the last three bullet points). This thesis presents an in-depth description of the aircraft sizing and performance models because these modules directly connect to configuration-agnostic analyses at the system- and subsystem-levels. The collaborators that developed the other aforementioned capabilities may elaborate more deeply on those contributions in their own manuscripts; such contributions are not claimed to be the author’s.

4.2.1 Computational Procedure

FAST’s core capabilities and analysis modules are outlined in the Design Structure Matrix (DSM) illustrated in Fig. 4.1. Each gray cell represents a specific analysis module within FAST, with each one containing its own numerical methods and implicit assumptions. Each blue cell represents a collection of functions grouped together to perform a specific portion of the overall aircraft analysis. Information fed between modules is indicated by the gray boxes in the DSM. For example, in the DSM’s first row and second column, the gray box represents data transfer from the USER INPUTS module to the INITIALIZATION module.

There are three main analysis modules in FAST: 1) an INITIALIZATION module that processes the user inputs, identifies any unknown values, and estimates the unknown values using regressions from historical data; 2) an AIRCRAFT SIZING module that sizes the airframe and propulsion system, subject to *fixed* point performance parameters (wing loading and thrust-to-weight or power-to weight ratio); and 3) a MISSION ANALYSIS module that calculates the mission performance and the energy required to fly the prescribed mission. FAST’s modular framework construction enables the user to swap any existing models in FAST with more/less detailed ones that they construct to support their design studies.

Two file inputs are required to analyze a configuration in FAST: 1) Aircraft Specification File; and 2) Mission Profile Specification File. The Aircraft Specification File lists all known design and performance parameters that describe the configuration and prescribes run settings (options) on how FAST should analyze the configuration. It also defines the propulsion system architecture and its operational strategies. The Mission Profile Specification File describes the flight phases and trajectory flown while analyzing the configuration, and includes either

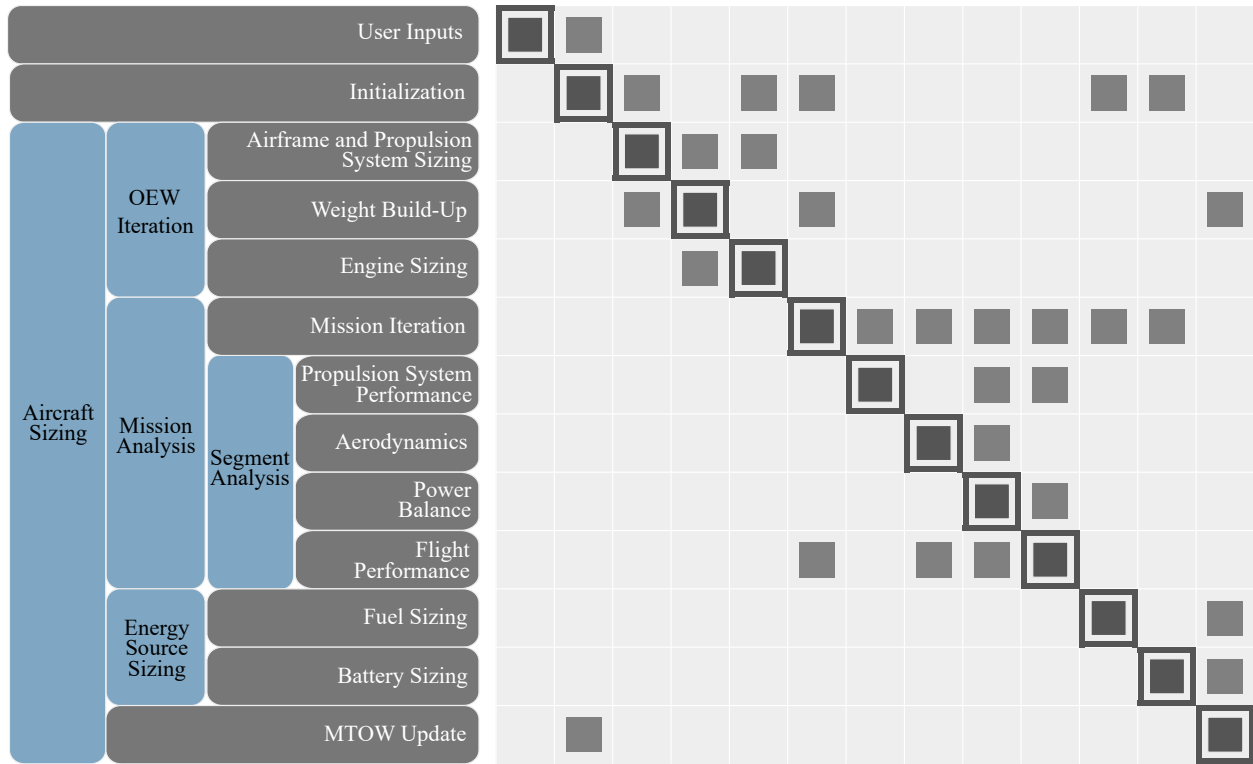


Figure 4.1: FAST’s overarching computational procedure, generated using OpenMDAO [3] (FAST, however, is not implemented in OpenMDAO).

distance- or time-based targets, dictating the aircraft’s required range or endurance. Each mission profile is comprised of takeoff, climb, cruise, descent, and landing segments. Since the work in this thesis focuses on the physics-based modeling within the Aircraft Sizing and Mission Analysis modules, refer to Ref. [177] for more information about creating the input files to run FAST and its INITIALIZATION module capabilities.

FAST’s overarching computational procedure is governed by three fixed point iterations to: 1) size the overall aircraft; 2) size the airframe and propulsion system; and 3) execute the mission analysis. A high-level summary is provided here, with the later subsections focusing on the physics- or data-based models developed. To initialize the iteration, a guess for the aircraft’s MTOW is input (either by the user or the data-based models embedded in FAST). In the overall aircraft sizing iteration (outlined by the data transfer from the MTOW UPDATE module to the INITIALIZATION module), the aircraft’s MTOW changes as a function of the energy required to fly the design mission, thus impacting the fuel/battery (or other energy source) weight carried onboard the aircraft.

Once the aircraft's MTOW is known, data is transferred to the OEW ITERATION, sizing the airframe and propulsion system. The airframe is sized using a Gaussian Process Regression [179], and is a function of the wing loading, MTOW, and other parameters. The propulsion system components are sized by simulating a downstream flow utilizing the GPSA Framework, propagating the sea level static (SLS) thrust or power through each component based on the given power management strategy. Once the power required by each component is computed, additional data-based models predict each component's weight. Any gas turbine engines are also sized using a physics-based model governed by thermodynamics; refer to GAS TURBINE MODELING section in Ref. [177]. After predicting each component's weight, the aircraft's MTOW is recomputed. The airframe and propulsion system components are sized iteratively until converging on an MTOW (the relative difference in MTOW output by successive iterations is less than a prescribed tolerance).

After the airframe and propulsion system are sized, data is transferred to the the MISSION ANALYSIS module and the aircraft is flown on the design mission prescribed within the Mission Profile Specification File. Each mission segment is simulated using the SEGMENT ANALYSIS functionalities, predicting the distance/time flown and the energy required to fly the segment. After the mission is flown, the cruise segment length is modified such that the cumulative distance/time flown matches the prescribed target in the Mission Profile Specification file. The process repeats until the mission analysis converges on the distance- or time-based target prescribed (the absolute difference between the cumulative distance/time flown and the prescribed target is less than a prescribed tolerance), returning the energy required to fly the mission.

Based on the energy requirements from the MISSION ANALYSIS modules, data is transferred to the ENERGY SOURCE SIZING module to size the fuel and the battery. After that, the aircraft's MTOW is updated in the MTOW UPDATE module, feeding back into the overall aircraft sizing iteration previously described. The aforementioned process continues iteratively until the aircraft's MTOW has converged (the relative difference in MTOW output by successive iterations is less than a prescribed tolerance).

Next, the modeling methodology and assumptions within the AIRCRAFT SIZING and MISSION ANALYSIS modules are described in greater detail.

4.2.2 Aircraft Sizing

As illustrated in Fig. 4.1, the AIRCRAFT SIZING module predicts the airframe, propulsion system, and energy source (fuel, battery, etc.) weights. The OEW ITERATION module sizes

the airframe and propulsion system while the ENERGY SOURCE SIZING module sizes the energy sources. Detailed descriptions of the computational models developed to size the components are described next.

4.2.2.1 Airframe and Propulsion System Sizing

The airframe and propulsion system are sized using the point parameters (wing loading and thrust-to-weight or power-to-weight ratio), either prescribed in the Aircraft Specification file by the user or using the data-based models embedded in FAST. One embedded assumption in FAST is that the point performance parameters *remain fixed* during sizing.

Airframe Sizing In the airframe sizing, both the wing area and airframe weight is predicted. The wing area is computed using Eq. 4.1, and is simply the quotient of MTOW and the wing loading.

$$S = \text{MTOW} / (W/S) \quad (4.1)$$

In FAST, an “airframe weight” is predicted instead of an “operating empty weight” (OEW) because the OEW accounts for the propulsion system. However, FAST’s data-based models are driven by a historical aircraft database containing only conventional aircraft powered by gas turbine engines, and do not include electric generators/motors and cables that must be included in electrified aircraft concepts. Therefore, to accurately predict an OEW, the airframe weight is predicted from the historical aircraft database (by subtracting the propulsion system and crew weights from the OEW), and then the propulsion system and crew weights are added in FAST. Additional electric generator/motor and cable weight models are described later in this section.

The airframe weight is estimated differently depending on the aircraft class [177]. For turbofan-powered aircraft, the data-based models predict the airframe weight as a function of the wing area, SLS thrust, EIS year, and MTOW. For turboprop-powered aircraft, the airframe weight is a linear function of MTOW.

Propulsion System Sizing The propulsion system is sized by first computing the SLS thrust or power, depending on whether the aircraft is powered by turbofan or turboprop engines, as described by Eqs. 4.2 and 4.3, respectively.

$$T_{\text{SLS}} = \left(\frac{T}{W} \right) \text{MTOW} \quad (4.2)$$

$$P_{\text{SLS}} = \left(\frac{P}{W} \right) \text{MTOW} \quad (4.3)$$

The SLS thrust/power is then propagated through the propulsion system using the downstream power flow simulation detailed in Alg. 1 from Chapter 3. The power flow is enabled by the user prescribing the architecture and downstream operational/efficiency matrices and power management strategy within the Aircraft Specification File. After simulating the downstream power flow, the power load at each component is known, enabling each one to be sized with greater granularity.

The turbofan/turboprop engines are sized using a gas turbine engine model to predict its thrust-specific fuel consumption, number of compressor/turbine stages, and other key performance parameters for such engines. The turbofan/turboprop engine weight is predicted by a data-based model, and is a function of the engine's maximum SLS thrust/power. Refer to Ref. [177] for more information. The electric generators/motors and cables are sized utilizing component-specific models, described next.

Electric Machine Sizing Electric machines (both generators and motors) are sized based prior data listing the specific power (or power-to-weight ratio) and rated power of such components [4]. The data spans electric machines with power ratings from 10^1 to 10^3 kW, with most data residing within a regime of 10^2 to 10^3 kW. Since the sizing module must predict the electric machine weight, the weight of each electric machine recorded in the data was estimated by dividing its rated power by its power-to-weight ratio using Eq. 4.4. The resulting relationship between rated power and electric machine weight is plotted in Fig. 4.2.

$$W = P_{\text{rated}} / (P/W) \quad (4.4)$$

Then, a curve was fit to the data to predict the electric machine weight as a function of its rated power. Equation 4.5 reveals the selected curve fit, returning an R^2 value of 0.9992, indicating a strong correlation and that the variance in the data is well-captured by the model.

$$W_{\text{EM}} = 0.2118 (P_{\text{rated}})^{1.0332} \quad (4.5)$$

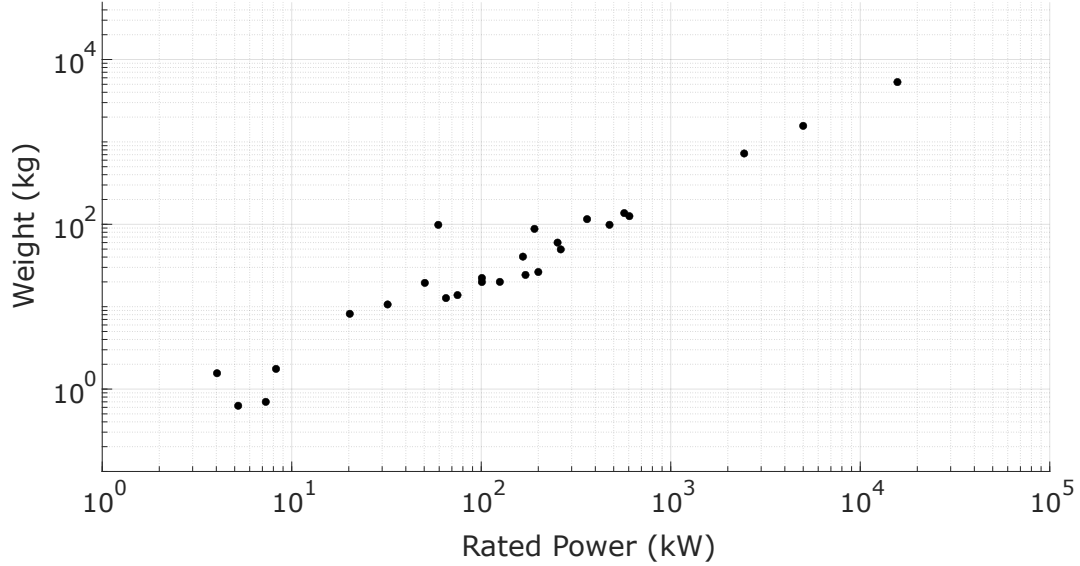


Figure 4.2: Electric machine sizing data from Ref. [4].

Additionally, some aircraft sizing studies in FAST explore integrated aircraft designs with future technologies that may perform better than existing ones. In such cases, the user may desire to swap out the data-based model in Eq. 4.5 with a fixed power-to-weight ratio that exceeds those captured by the present data. Then, the electric machine weight is simply computed by dividing its rated power by the prescribed power-to-weight ratio, as previously demonstrated in Eq. 4.4.

Cable Sizing All cables are sized using key performance parameters including the cable length, power output, and weight-to-power ratio, as described by Eq. 4.6. Recall that the power output is computed apriori from the downstream power flow simulation while sizing the components in the propulsion system. The cable weight-to-power ratio is defined by the user.

$$W_{\text{cable}} = (W/P)_{\text{cable}} \times P_{\text{out}} \times l \quad (4.6)$$

For a known aircraft concept, the cable length can be provided from either known values or by digitizing a schematic of the aircraft's propulsion system and adding all the cable lengths together. Alternatively, if such data is not available, the user may guess a cable length to be input into the analysis. In such cases, it may be useful to design an aircraft with multiple cable lengths to understand the sensitivity of the system-level responses with

respect to the assumed values.

Weight Build-Up Using all the airframe and propulsion system weights previously computed, the weight build-up in Eq. 4.7 estimates the aircraft’s MTOW. The airframe (W_{Airframe}), engine (W_{Eng}), electric machine (W_{EM} , includes electric generators and motors), and cables (W_{Cable}) are predicted using the procedures described in this section. The crew and passenger weights (W_{Crew} and W_{Pax} , respectively) are either input by the user or predicted using FAST’s data-based models. The fuel and battery weights are either initial guesses to begin the aircraft sizing process, or are extracted from the ENERGY SOURCE SIZING module output.

$$\text{MTOW} = W_{\text{Airframe}} + W_{\text{Eng}} + W_{\text{EM}} + W_{\text{Cable}} + W_{\text{Crew}} + W_{\text{Pax}} + W_{\text{Fuel}} + W_{\text{Battery}} \quad (4.7)$$

The resultant MTOW is computed iteratively within the OEW ITERATION module until convergence is achieved.

4.2.2.2 Energy Source Sizing

The ENERGY SOURCE SIZING module computes the weight of the required fuel, battery, or other energy source as a function of the energy required to fly the design mission from the MISSION ANALYSIS module. Methods to predict the fuel and battery weight are described next. The fuel weight is estimated using an energy-based approach. The battery size can be estimated either using an energy-based approach, only considering the energy required to fly the mission, or a energy- and power-based approach, considering the aforementioned factor as well as ensuring that the battery discharging rate (C-rate) remains below a maximum allowable value.

Fuel Sizing Fuel is defined as a substance that is pumped into a storage tank on the aircraft, such as kerosene or hydrogen. The fuel weight is estimated by dividing the cumulative energy expenditure during the design mission, E_{fuel} , by its gravimetric specific energy, e_{fuel} , as listed in Eq. 4.8.

$$W_{\text{fuel}} = \frac{E_{\text{fuel}}}{e_{\text{fuel}}} \quad (4.8)$$

Currently, fuel sizing only considers the energy expenditure and neglects the weight of the fuel tanks. While this is a reasonable estimate for kerosene-powered aircraft, it may not

be reasonable for hydrogen-powered aircraft, which require significantly heavier fuel tanks due to the pressurized cryogenic liquid, insulation, and necessary structure to support the pressure vessel from any buckling loads [24, 25]. To circumvent this issue, the airframe weight computed during in the OEW ITERATION module can be multiplied by a scale factor greater than unity to indicate that larger fuel tanks are necessary, thus making the airframe heavier. This scale factor could be a constant, or depend on the fuel weight output from this module.

Battery Sizing, Energy-Based Approach During early-phase conceptual design, the user may want to size the battery only as a function of the energy required to fly the design mission, simplifying the analysis by neglecting how the cells are arranged (number in series/parallel). In this case, the new battery weight is computed using Eq. 4.9, and solely depends on the energy expended during the design mission (E_{Batt}) and the battery’s pack-level gravimetric specific energy (e_{Batt}).

$$W_{\text{Batt}} = E_{\text{Batt}}/e_{\text{Batt}} \quad (4.9)$$

The rudimentary result provided from the analysis is typically an underestimation of the actual battery weight because the integrated energy- and power-based approach (described next) also requires the battery state of charge (SOC) to remain above 20% at the end of the design mission, as permanent damage could occur if it is discharged further [66]. Therefore, the battery must carry additional energy than it would if it were sized only under energy considerations, resulting in a heavier weight.

Battery Sizing, Integrated Approach The integrated battery sizing approach accounts for both energy- and power-based considerations in the analysis. In this detailed model, a battery pack is built by arranging Lithium-ion cells in series (to obtain a desired system voltage) and in parallel (to obtain a desired power output). FAST assumes that each battery cell has a maximum capacity (Q_{max}) and maximum voltage (V_{max}) of 2.6 Ah and 3.6 V, respectively. FAST’s modular computational framework enables users to substitute the default Lithium-ion battery cell model with an alternative battery chemistry, provided the input/output structure remains consistent with FAST’s current functions. Two checks are performed to size the battery for energy and power considerations.

The first check ensures that the the battery’s final SOC remains at or above the 20% threshold previously described. Depending on the mission flown, the SOC change before reaching the 20% threshold (ΔSOC) is computed to estimate the energy that must be

added/removed from the battery. Then, Eq. 4.10 uses the existing battery capacity (Q_{Batt}) and the SOC change to recompute the required number of cells in parallel [66]. To ensure that the result remains as an integer quantity, the brackets $\lceil$ and $\rceil$, are used represent the ceiling function, rounding the result to the next largest integer.

$$N_{\text{Par},\text{SOC}} = \left\lceil \frac{Q_{\text{Batt}} + (\Delta\text{SOC}) Q_{\text{max}} N_{\text{Par}}}{Q_{\text{max}}} \right\rceil \quad (4.10)$$

The second check ensures that the maximum discharge rate (C-rate) is not exceeded, as defined in Eq. 4.11. FAST assumes a maximum C-rate of 5, but this may be modified by the user.

$$\text{C-rate} = \frac{I}{Q_{\text{max}} N_{\text{Par}}} \quad (4.11)$$

If the maximum C-rate ($\overline{\text{C-rate}}$) is exceeded, additional cells are added in parallel to ensure that the battery does not discharge too rapidly, as described by Eq. 4.12.

$$N_{\text{Par},\text{C-rate}} = \left\lceil \frac{\max(\text{C-rate})}{\overline{\text{C-rate}}} \right\rceil N_{\text{Par}} \quad (4.12)$$

Using the values from Eqs. 4.10 and 4.12, the battery is resized using Eq. 4.13 by selecting the number of cells in parallel based on the more demanding sizing requirement (the maximum number required to satisfy both conditions). The battery's weight is then computed, as shown in Eq. 4.14, and is also a function of the number of cells in series (N_{Ser}).

$$N_{\text{Par,new}} = \max(N_{\text{Par},\text{SOC}}, N_{\text{Par},\text{C-rate}}) \quad (4.13)$$

$$W_{\text{Batt,new}} = V_{\text{max}} Q_{\text{max}} N_{\text{Ser}} N_{\text{Par,new}} / e_{\text{Batt}} \quad (4.14)$$

4.2.3 Mission Analysis

The MISSION ANALYSIS module flies the design mission and predicts flight performance, particularly the energy required to fly, and informs the ENERGY SOURCE SIZING module for sizing the fuel and battery, as previously discussed in Section 4.2.2.2. Its computational procedure is illustrated in Fig. 4.3.

The mission analysis consists of two functionalities: 1) the mission iteration; and 2) the energy-based segment analysis, abbreviated as the “segment analysis” hereinafter. The mission iteration performs a fixed-point iteration over the distance/time of the cruise segment

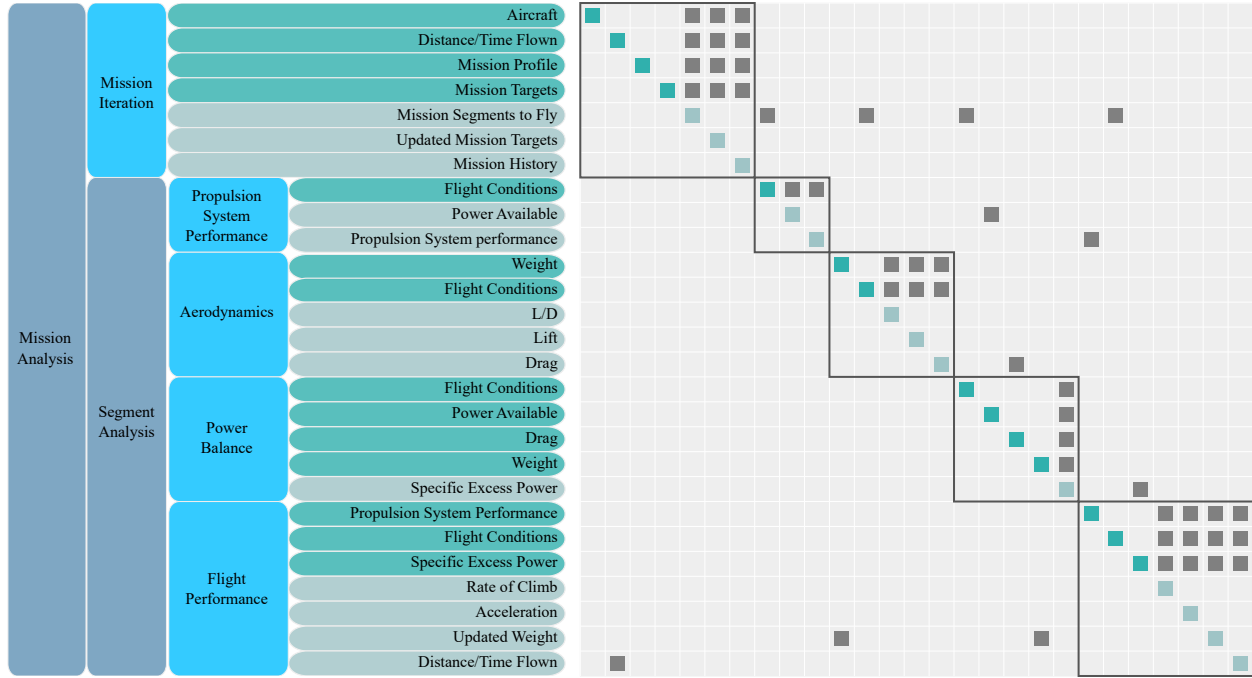


Figure 4.3: FAST's mission analysis module.

allocated to each mission target. For this paper, the mission iteration is not discussed any further because it is not central to estimating the energy required to fly the mission; it is merely a numerical method for ensuring that the target distance/time is flown.

In the segment analysis, the aircraft flies the segments prescribed by the user (takeoff, climb, cruise, descent, and landing) in the Mission Profile Specification file. Each segment is analyzed iteratively by predicting the propulsion system and aerodynamic performance, followed by a power balance to estimate the excess power available to climb or accelerate. A flight performance model then predicts the acceleration and rate of climb before computing the total power required (and hence, energy) to fly the segment. The power required is then input into another propulsion system analysis to predict the fuel burn and, if necessary, model battery discharging. After the fuel burn is subtracted from the aircraft's weight, the iteration repeats until converging on the power required to fly the mission segment (the relative difference between the power required in successive iterations is within a prescribed tolerance).

The remaining sections explore the modeling approaches within each of the four segment analysis functionalities, along with other modeling features available that are used for later work in Chapters 5 and 6. Section 4.2.3.1 reviews all underlying assumptions in FAST's

mission analysis. Then, Section 4.2.3.2 through 4.2.3.5 describes the physics-based modeling in each of the four segment analysis functionalities.

4.2.3.1 Segment Analysis Assumptions

The segment analysis is rooted in one overarching assumption – the aircraft is assumed to be a point mass, neglecting all rigid body dynamics. This assumption enables FAST’s segment analysis to evaluate any aircraft configuration with any propulsion architecture. Thanks to the GPSA Framework introduced in Chapter 3 and its ability to model any propulsion system architecture, FAST’s propulsion system analysis (and overarching mission analysis) are enhancements to the state-of-the-art over previous work that developed propulsion architecture-dependent aircraft performance equations [61, 181, 182].

The remaining assumptions reviews are segment-specific to either the takeoff, climb, cruise, descent, or landing segments modeled in FAST. Table 4.1 lists the assumptions made in the base-level analysis.

Table 4.1: Assumptions made in FAST’s base-level segment analysis. Higher fidelity codes can be embedded into FAST to override the listed assumptions.

Assumption	Takeoff	Climb	Cruise	Descent	Landing
Fixed Time	✓	✗	✗	✗	✓
Constant Acceleration	✓	✗	✗	✗	✓
Constant Lift-to-Drag Ratio	✗	✓	✓	✓	✗
Linearly Spaced Altitudes and Airspeeds	✓	✓	✓	✓	✓
Reverse Thrust	✗	✗	✗	✗	✓

The “Fixed Time” assumption fixes the takeoff and landing segments to durations of 60 and 30 seconds, respectively, under a “Constant Acceleration.” Such assumptions allow the aircraft’s interactions with the ground, such as rolling friction and the impact of “ground effect” on the induced drag [175], to be neglected. The simplification also eliminates the need to integrate the governing equations of motion, thus avoiding costly numerical integration schemes and reducing the computational overhead. Additionally, in the landing segment, “Reverse Thrust” is assumed to be modeled as 30% of the total power available from the aircraft’s propulsion system.

In the climb, cruise, and descent segments, a constant lift-to-drag ratio eliminates the need for a drag polar, limiting computational overhead. Additionally, this assumption enables the

segment analysis to be applied to both conventional and unconventional aircraft, maintaining a configuration-agnostic analysis, rather than parameterizing aircraft performance by the geometry (which may not be known during early-phase conceptual design). For some studies in Chapter 5, a more detailed drag polar model is necessary for simulating engine inoperative conditions, and such capabilities have been integrated into FAST for that purpose. If desired, the user could use the detailed drag polar model (discussed later) in their aircraft analysis, assuming that the configuration’s geometry is known.

In all mission segments, the initial/final altitudes and airspeeds are prescribed in the Mission Profile Specification File. To predict the power required to fly, each segment is discretized by multiple control points, enabling the governing equations to be evaluated at multiple points in the segment besides the initial/final ones. The altitude and airspeed at each control point is linearly interpolated based on the boundary conditions prescribed by the user. Additionally, in each of the discretized segments, the aircraft’s acceleration is assumed to be constant, circumventing the need to integrate any equations of motion. As a result, the flight performance analysis computes the time required to fly between each control point rather than numerically integrating to find the altitude and airspeed at each successive control point, saving computational overhead.

4.2.3.2 Propulsion System Performance

The propulsion system performance module estimates the propulsion system’s power available as a function of altitude. The power available is computed via a two-step process.

First, the transmitters (as described in Chapter 3, any component that transmits or distributes power to others, such as gas turbine engines, electric machines, propellers/fans, etc.) at the downstream end of the propulsion system are identified, and their power available is computed. If they are gas turbine engines, then their power available is computed using Eq. 4.15 or 4.16 depending on whether the aircraft being analyzed is turbofan- or turboprop-powered, respectively [175]. For turbofan-powered aircraft, the lapse rate is assumed to be equal to the density ratio at the aircraft’s given altitude. For turboprop-powered aircraft, the lapse rate is a power of the density ratio, typically less than unity [175], and is prescribed by the user (m).

$$TV_{\infty, \text{turbofan}} = \left(T_{\text{SLS}} \frac{\rho}{\rho_{\text{SLS}}} \right) V_{\infty} \quad (4.15)$$

$$TV_{\infty, \text{turboprop/piston}} = P_{\text{SLS}} \left(\frac{\rho}{\rho_{\text{SLS}}} \right)^m \quad (4.16)$$

Alternatively, if the most downstream components are electric machines, their power available does not lapse with altitude; their power available is assumed to be equal to their SLS power that they were sized for in Section 4.2.2.1.

Second, the power available at the most downstream components is input into the GPSA Framework to propagate power upstream and estimate the aircraft’s total power available. The architecture and upstream operational/efficiency matrices and corresponding power management strategy are stored in the Aircraft Specification File, enabling the power available to be predicted using the power flow simulation provided in Section 3.2.4.

4.2.3.3 Aerodynamic Performance

The aerodynamic performance module predicts the lift-to-drag ratio during flight. FAST primarily achieves this by assuming a constant lift-to-drag ratio during each climb, cruise, and descent segment, reducing the computational overhead and circumventing the need to know the aircraft’s geometry before executing an analysis. If the geometry is known, FAST also offers an empirical aerodynamic performance model adapted from NASA’s Aviary code [58] – an open-source, Python-based aircraft sizing code built upon OpenMDAO [3] – that is a function of the aircraft’s geometry. Refer to Ref. [58] for more information about the empirical aerodynamic analysis.

Using the empirical aerodynamic performance model provided in Aviary [58], windmilling and trim drag models were built on top, enabling a detailed aerodynamic performance of aircraft under engine inoperative scenarios, and is the focus for the remainder of this section. The outputs from the windmilling and trim drag models ultimately act as increments on the parasite drag coefficient.

Windmilling Drag Windmilling drag was modeled from Walsh and Fletcher’s *Gas Turbine Performance* textbook [5] for windmilling turbofan engines (Chart 10.5, free windmilling curve), and parameterizes the drag increment by the engine’s specific thrust. The windmilling drag model assumes that the fan continues to spin freely after failing (hence, the “free

windmilling”) and is not locked in place. This assumption results in a higher theoretical drag coefficient than if the fan were to be locked in place after failing.

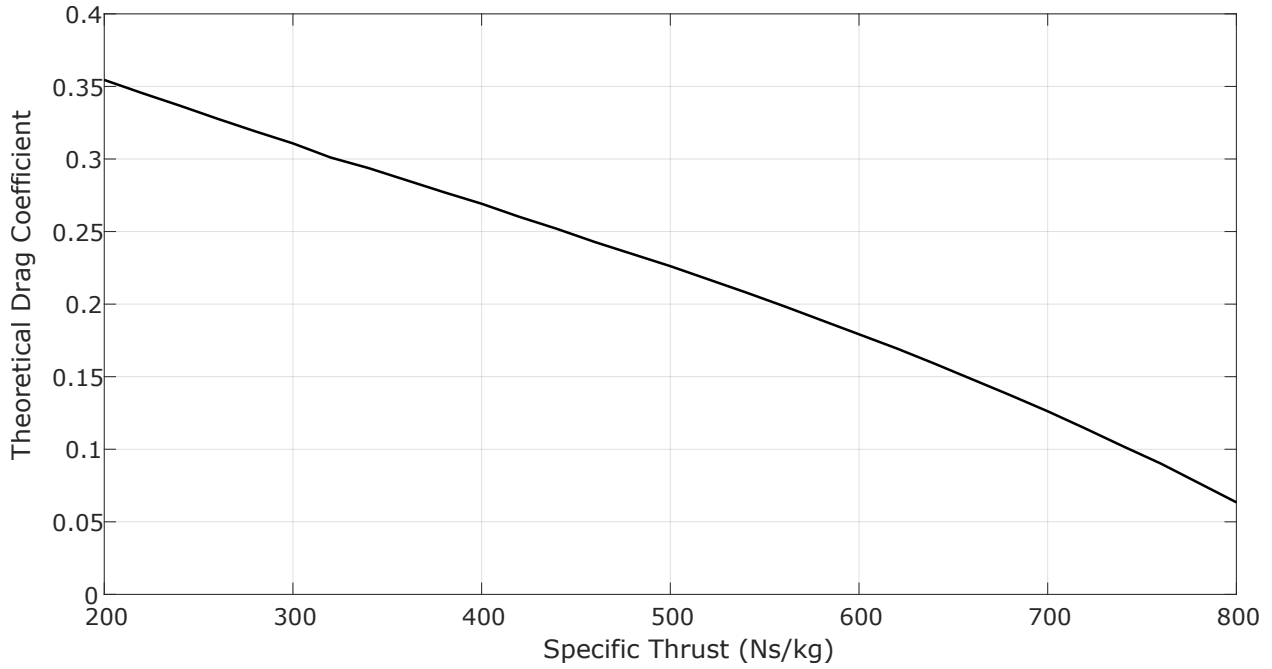


Figure 4.4: Theoretical drag coefficient for a windmilling engine, reproduced from Ref. [5].

To predict the windmilling drag, the theoretical drag coefficient ($\hat{C}_{D_{\text{Windmill}}}$) is first estimated using Fig. 4.4. Then, the drag coefficient correction is estimated from Fig. 4.5 and *subtracted* from the theoretical drag coefficient, resulting in the actual drag coefficient for the windmilling engine ($C_{D_{\text{Windmill}}}$). After computing the actual drag coefficient, the windmilling drag is computed using Eq. 4.17, where S_{engine} represents the inlet area of the turbofan engine.

$$D_{\text{Windmill}} = C_{D_{\text{Windmill}}} \times q \times S_{\text{engine}} \quad (4.17)$$

The windmilling drag can also be modeled for a ducted fan within a distributed propulsor array by introducing a calibration factor (k_{Windmill}) to scale up/down the estimated windmilling drag. Given that the model applies to turbofan engines, substantial calibration factors may be needed to reasonably estimate the windmilling drag for a ducted fan. For example, to match the windmilling drag of one ducted fan on NASA’s SUSAN aircraft [92], the calibration factor was approximately 3.17, indicating that the windmilling drag for a ducted fan is significantly under-predicted from the present model.

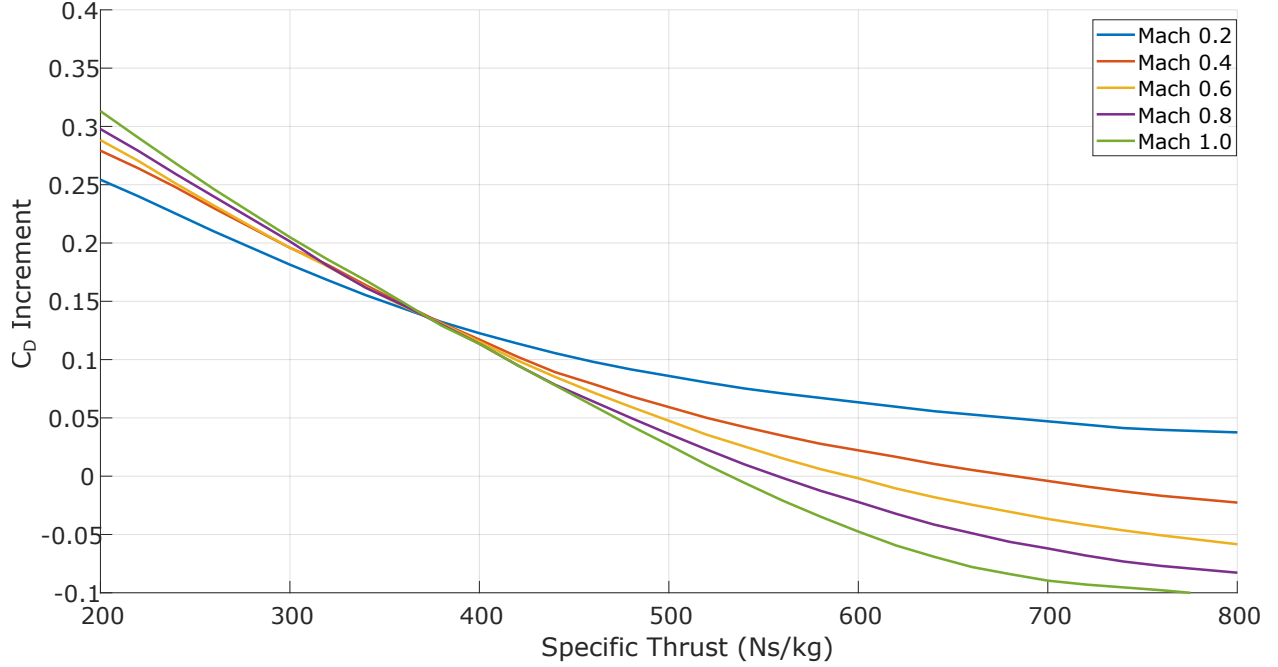


Figure 4.5: Drag coefficient correction, reproduced from Ref. [5].

Trim Drag The trim drag model is an elementary lateral stability model to predict the parasite drag coefficient increment corresponding to the necessary rudder input to maintain lateral stability during one engine inoperative conditions. The model assumes that one engine has failed (regardless of whether it is a turbofan/turboprop engine, ducted fan, etc.) and the symmetrically-positioned engine on the other side of the wing remains fully operational. As a result, the rudder must produce a yaw moment to overcome the windmilling drag from the inoperative engine and the thrust available from the remaining operative engine.

First, the yaw moment required from the rudder is computed in Eq. 4.18, considering both the thrust from the remaining operative engine, T , and the windmilling drag from the inoperative engine, D_{Windmill} , multiplied by the moment arm between the (in)operative engines and the fuselage centerline, r .

$$N_v = (T + D_{\text{Windmill}}) r \quad (4.18)$$

Next, Eq. 4.19 computes the yaw moment coefficient, and is a function of the dynamic pressure, wing area, and wingspan.

$$C_n = N_v / (q \times S_w \times b_w) \quad (4.19)$$

Then, the lift curve slope of a two-dimensional airfoil, $C_{l_{\alpha_v}}$, on the vertical tail was estimated as 90% of that of a thin airfoil (2π /rad). Using this information, the lift curve slope for the vertical tail (converted to units of /deg) was predicted in Eq. 4.20.

$$C_{L_{\alpha_v}} = \frac{C_{l_{\alpha_v}}}{1 + \frac{C_{l_{\alpha_v}}}{\pi \mathcal{AR}_{ve_v}}} \times \frac{\pi}{180} \quad (4.20)$$

After that, the flapped area fraction was computed as the ratio of the rudder area with respect to the vertical tail area, S_r/S_v , and input into a digitized version of Figure 2.21 in Nelson's *Flight Stability and Automatic Control* [6] (reproduced in Fig. 4.6), returning the “flap effectiveness parameter,” τ .

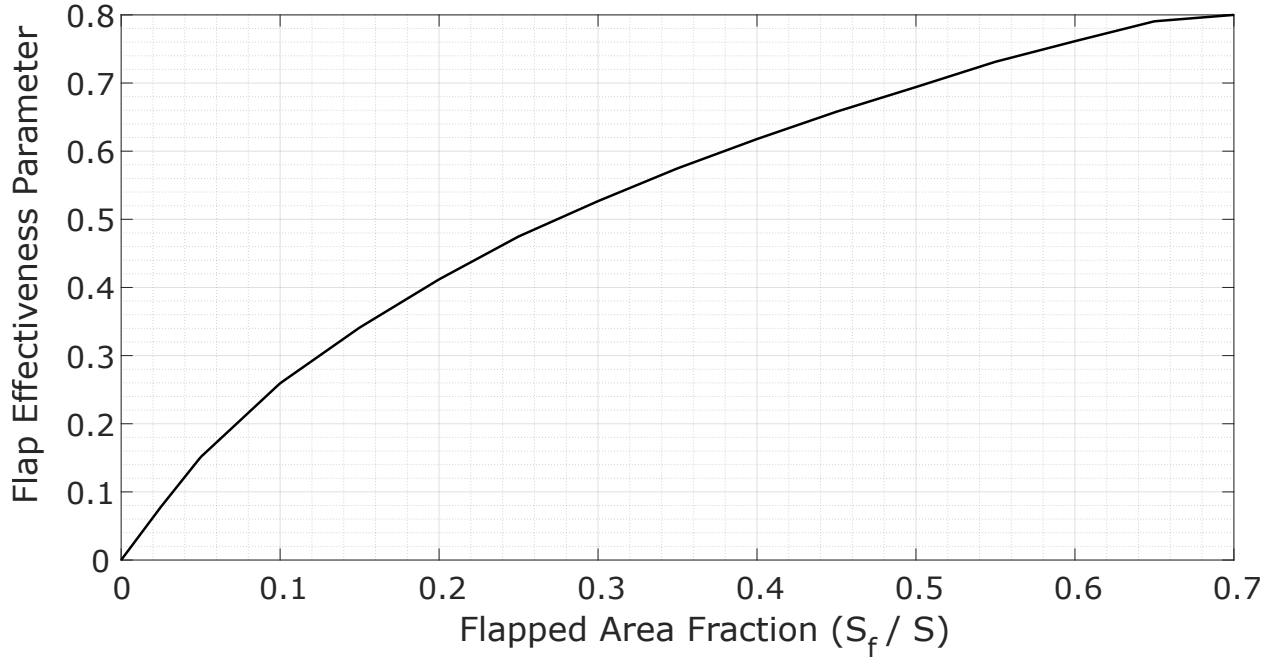


Figure 4.6: Flap effectiveness parameter, reproduced from Ref. [6].

With this information, the yaw moment coefficient for the rudder deflection was computed in Eq. 4.21, assuming that the dynamic pressure on the vertical tail is equal to that on the wing ($\eta = 1$). Note that l_v is the moment arm between the positions of the wing and vertical tail mean aerodynamic chords.

$$C_{n_{\delta r}} = \eta \frac{l_v S_v}{S_w b_w} C_{L_{\alpha_v}} \tau \quad (4.21)$$

Lastly, the rudder deflection was computed using Eq. 4.22. The result was input into Eq. 12.37 from Raymer’s *Aircraft Design: A Conceptual Approach* [183] to predict the parasite drag increment from deflecting the rudder (Eq. 4.23).

$$\delta r = C_n / C_{n_{\delta r}} \quad (4.22)$$

$$\Delta C_{D_0} = 0.0023 \delta r (b_r / b_w) \quad (4.23)$$

The resulting parasite drag coefficient increment is added to the respective value in the empirical drag model described at the beginning of this section.

4.2.3.4 Power Balance

The power balance utilizes information from the propulsion and aerodynamic performance modules to predict the excess power available to climb or accelerate. Equation 4.24 computes the specific excess power (P_s) [175], representing the difference between the thrust power available and drag power that must be overcome at the present flight conditions, normalized by the aircraft’s weight.

$$P_s = (TV_\infty - DV_\infty) / W \quad (4.24)$$

For the flight simulation to be physically tractable, the specific excess power must be nonnegative at all times. Otherwise, the aircraft does not have enough power to overcome drag and must either slow down or descend to a lower altitude. Although FAST will not change the trajectory to ensure that the specific excess power remains nonnegative, the warning is thrown to the user, indicating that they may need to modify the design mission profile to attain physically tractable results.

4.2.3.5 Flight Performance

The flight performance module first computes the duration of each segment, and then returns the aircraft’s rate of climb and acceleration. Using the computed flight performance data, the power required to fly is computed and the energy expenditures are predicted.

Rate of Climb and Acceleration Before estimating the rate of climb and acceleration, the time to fly between control points in each segment is computed in one of two ways, and depends on the trajectory and constraints prescribed by the user.

The first way is to compute the time to fly as a function of the aircraft's energy height and specific excess power, implicitly assuming that the trajectory is prescribed by the user without any limitations on the rate of climb. For this case, the time to fly is the quotient of the energy height – described in Eq. 4.25 based on the given flight conditions – and the specific excess power, as listed in Eq. 4.26.

$$H_e = h + \frac{(V_\infty)^2}{2g} \quad (4.25)$$

$$\Delta t = \left| \frac{\Delta H_e}{P_s} \right| \quad (4.26)$$

The second way is to compute the time to fly is based on a user-prescribed rate of climb for the given segment, implicitly assuming that the aircraft has enough specific excess power throughout the mission to reach its target speed at the end of the segment. In such cases, the time to fly is computed using Eq. 4.27, and is based on the altitude difference between control points in the trajectory and the prescribed rate of climb.

$$\Delta t = \frac{\Delta h}{(dh/dt)} \quad (4.27)$$

In this case, the power utilized to match the prescribed rate of climb may prevent the aircraft from accelerating to the target airspeed at the successive control points. Therefore, the maximum acceleration is computed using Eq. 4.28 to check whether or not the acceleration required exceeds the maximum realizable one.

$$\left(\frac{dV}{dt} \right)_{\max} = \frac{(P_s - dh/dt) g}{V_\infty} \quad (4.28)$$

If the acceleration ever exceeds the maximum realizable one, the airspeed profile is recomputed at the remaining control points in the trajectory, assuming that the maximum realizable acceleration is utilized for the remainder of the flight segment.

After computing the time to fly, and modifying the trajectory (if needed), the rate-of-climb and acceleration are updated using Eqs. 4.29 and 4.30, respectively.

$$\frac{dh}{dt} = \frac{\Delta h}{\Delta t} \quad (4.29)$$

$$\frac{dV_\infty}{dt} = \frac{\Delta V_\infty}{\Delta t} \quad (4.30)$$

Power/Energy Required Using the drag power predicted from the aerodynamic performance module and the rate of climb and acceleration predicted from the flight performance module, the power required to fly at each control point is predicted using Eq. 4.31.

$$P_{\text{req}} = DV_\infty + W \frac{dh}{dt} + mV_\infty \frac{dV_\infty}{dt} \quad (4.31)$$

The three terms on the right-hand side represent the drag power to overcome, the power required to climb, and the power required to accelerate, respectively.

The power required to fly is then input into another propulsion system analysis to compute the power required at each component in the propulsion system, and is accomplished by simulating a downstream power flow using the GPSA Framework using the architecture and downstream/operational efficiency matrices. After propagating the required power to each component by using Alg. 1 from Section 3.2.4, component-specific analyses are utilized to predict the fuel flow and/or battery discharging dynamics. For any gas turbine engine, the fuel flow is predicted using an empirical equation based on a database of existing turbofan engines; refer to Ref. [177] (Off-Design Engine Modeling section) for more information. For any electric motor boosted by a battery, a discharging dynamics model developed by Tremblay and Dessaint [184] was utilized.

The fuel flow or battery power requirements are then converted to a fuel or energy expenditure, respectively, by multiplying the fuel flow or power required by the time to fly, as described by Eq. 4.32. The resulting energy requirements are utilized in the ENERGY SOURCE SIZING module, as previously described in Section 4.2.2.

$$E_{\text{req, SRC}}^{(i)} = (P_{\text{req, SRC}})^{(i)} \Delta t \quad (4.32)$$

4.2.4 Segment Analysis Algorithm

The physics-based models introduced throughout Section 4.2.3 are synthesized into one, overarching segment analysis algorithm, as detailed in Alg. 3, summarizing the mission analysis module implemented in FAST. The resulting mission analysis module is configuration-agnostic,

enabling any aircraft configuration with any propulsion system to be rapidly analyzed in FAST. The ability to analyze advanced aircraft concepts in a configuration-agnostic manner also supports the system architecture optimization studies explored in Chapter 6 of this thesis.

4.3 Aircraft Sizing Tool Validation (Experiment 2.0.2)

After developing FAST, Experiment 2.0.2 is performed to validate the integrated FAST-GPSA Frameworks, confirming that both conventional and advanced aircraft concepts and propulsion systems are modeled accurately, paving the way for more electrified propulsion system architectures to be studied in Experiment 2.0.3. First, a representative benchmark suite of conventional and advanced aircraft concepts are defined. Then, models of the aircraft in the benchmark suite are generated and analyzed in FAST. The sized aircraft’s system-level outputs from the analysis are compared to those available in the literature to validate FAST and its computational implementation.

4.3.1 Experimental Setup

The baseline conventional aircraft selected to validate the methodology presented was the Embraer ERJ175LR, a turbofan-powered regional jet designed to fly a 2,150 nmi design mission and carry 78 passengers. Data to model the ERJ175LR was gathered from a publicly available Type Certificate Data Sheet [185] and airport planning manual [186]. The baseline advanced aircraft selected was NASA’s SUBsonic Single Aft eNginE (SUSAN) concept [7, 42, 187], a single aisle aircraft designed to fly a 2,500 nmi design mission and carry 180 passengers using a series/parallel-partial hybrid propulsion system. The key performance parameters utilized for each aircraft design are listed in Tab. 4.2. The design mission profile for ERJ175LR and SUSAN [51] are provided in Tabs. B.1 and B.2 within Appendix B. Since no explicit design mission profile is publicly available for the ERJ175LR, one was approximated based on its expected behavior during airline operations, including a constant equivalent airspeed climb to its cruise altitude.

Algorithm 3: Energy-based segment analysis.

input : Initial/Final h and V_∞ , number of control points, n , and dh/dt , if prescribed
output : Power/Energy Required (P_{req} and $E_{\text{req, SRC}}$, respectively)

```
1  $h \leftarrow \text{linspace}(h_i, h_f, n)$ 
2  $V_\infty \leftarrow \text{linspace}(V_{\infty_i}, V_{\infty_f}, n)$ 
3  $P_{\text{req, old}} \leftarrow \mathbf{0}_{n \times 1}$ 
4  $P_{\text{req}} \leftarrow \mathbf{1}_{n \times 1}$ 
5 while  $\|P_{\text{req}} - P_{\text{req, old}}\|_2 > \epsilon$  do
6    $H_e \leftarrow h + (V_\infty)^2 / (2g)$  (Eq. 4.25)
7   if Aircraft is Turbofan-Powered then
8      $TV_\infty \leftarrow T_{\text{SLS}}(\rho/\rho_{\text{SLS}}) V_\infty$  (Eq. 4.15)
9   end
10  if Aircraft is Turboprop-Powered then
11     $TV_\infty \leftarrow P_{\text{SLS}}(\rho/\rho_{\text{SLS}})^m$  (Eq. 4.16)
12  end
13  Use Alg. 3.27 to compute  $TV_\infty$  at each component
14   $L \leftarrow mg \cos(\alpha)$ 
15  Compute  $L/D$  from desired aerodynamic performance module
16   $D \leftarrow L / (L/D)$ 
17   $P_s \leftarrow (TV_\infty - DV_\infty) / W$  (Eq. 4.24)
18  if  $dh/dt$  is prescribed then
19     $\Delta t \leftarrow \Delta h / (dh/dt)$  (Eq. 4.27)
20     $(dV_\infty/dt)_{\text{max}} \leftarrow (P_s - [dh/dt]) g / V_\infty$  (Eq. 4.28)
21    if any  $(dV_\infty/dt) > (dV_\infty/dt)_{\text{max}}$  then
22       $(dV_\infty/dt) \leftarrow (dV_\infty/dt)_{\text{max}}$ 
23       $V_\infty^{(i)} \leftarrow V_\infty^{(i-1)} + (dV_\infty/dt)^{(i-1)} \Delta t^{(i-1)}$ 
24    end
25  end
26  if  $dh/dt$  is not prescribed then
27     $\Delta t \leftarrow |\Delta H_e / P_s|$  (Eq. 4.26)
28     $(dh/dt) \leftarrow \Delta h / \Delta t$  (Eq. 4.29)
29    if any  $(dh/dt) > (dh/dt)_{\text{max}}$  then
30       $(dh/dt) \leftarrow (dh/dt)_{\text{max}}$ 
31       $\Delta t \leftarrow \Delta h / (dh/dt)$  (Eq. 4.27)
32    end
33  end
34   $(dh/dt) \leftarrow \Delta h / \Delta t$  (Eq. 4.29)
35   $(dV_\infty/dt) \leftarrow \Delta V_\infty / \Delta t$  (Eq. 4.30)
36   $P_{\text{req}} \leftarrow DV_\infty + W (dh/dt) + mV_\infty (dV_\infty/dt)$  (Eq. 4.31)
37  Use Alg. 3.27 to compute  $P_{\text{req}}$  at each component
38   $E_{\text{req, SRC}} = (P_{\text{req, SRC}}) \Delta t$  (Eq. 4.32)
39   $P_{\text{req, old}} \leftarrow P_{\text{req}}$ 
40 end
```

Table 4.2: Key performance and design parameters for modeling the ERJ175LR and SUSAN concepts.

Parameter	Embraer ERJ175LR	NASA SUSAN
Passengers	78	180
Class	Turbofan	Turbofan
Entry into Service	2005	2035
Design Range [km]	3,982	4,630
Cruise Speed [Mach]	0.78	0.775
Cruise Altitude [m]	10,668	10,668
Takeoff Speed [m/s]	69.45	84.88
Maximum Rate of Climb [m/min]	685.8	685.8
Climb/Descent L/D	10.98	15.76
Cruise L/D	15.20	19.70
Wing Loading [kg/m ²]	534	634
MTOW [kg]	37,890	86,586
Block Fuel [kg]	9,428	13,762
Operational Empty Weight [kg]	21,500	54,862
Engine	CF34-8E5	LEAP-1B
Design Thust [kN]	64.54	130.41
Cruise SFC [mg/Ns]	19.26	12.52
Thrust-Weight Ratio	0.3393	0.2980

4.3.2 Results: Aircraft Sizes

Using the aircraft specifications provided in Tab. 4.2 and the mission profiles in Appendix B, both aircraft models were calibrated by adjusting four scale factors – airframe weight, fuel flow, cruise lift-to-drag ratio, and climb/descent lift-to-drag ratio – until its MTOW and block fuel those provided in the literature within 3% error, corresponding to the accuracy threshold prescribed in Experiment 2.0.2. Key sizing and performance parameters for each baseline aircraft are presented in Tab. 4.3, with the percent difference relative to the values presented in literature. SUSAN’s engine thrust and TSFC are compared to a CFM International LEAP-1B engine. Despite that SUSAN is powered by a single aft turbofan, Chau et al. [7] modeled SUSAN’s engine after the LEAP-1B engine because it is the same turbofan engine installed on the Boeing 737 MAX, the aircraft that SUSAN is modeled after.

Table 4.3: Baseline aircraft sizing and performance parameters

Configuration	Embraer ERJ175LR	NASA SUSAN
MTOW (kg)	38,620 (+1.93%)	86,577 (-0.01%)
Block Fuel (kg)	9,425 (-0.03%)	13,766 (+0.03%)
Engine Thrust (kN)	61.7 (-4.34%)	84.2 (-35.43%)
Cruise SFC (mg/Ns)	19.93 (+3.48%)	11.8 (-5.75%)
EG Power (MW)	—	4.1 (-18.00%)
EM Power (MW)	—	0.8 (+0.00%)

Based on the results in Tab. 4.3, FAST can reasonably predict both system- and subsystem-level performance for both conventional and advanced aircraft configurations. At the system-level, with calibration, FAST is able to predict a block fuel that is nearly identical to that in the literature for each aircraft configuration. SUSAN’s MTOW is also accurately estimated, almost matching the value from literature exactly. The ERJ175LR’s MTOW is modeled within 2% accuracy, an acceptable level of error for a conceptual design tool when little may be known about the configuration. At the subsystem-level, FAST also reasonably predicts the propulsion system performance. For the ERJ175LR, the engine thrust is slightly under-predicted, resulting in a smaller engine installed on the aircraft. Consequently, the cruise SFC increases because the smaller turbofan engine operates less efficiently than a larger one.

On SUSAN, the engine size comparison is slightly skewed because the turbofan engine produces mechanical power to spin the turbofan engine and also power four electric generators. On the Boeing 737 MAX that SUSAN was designed after, two LEAP-1B engines provide approximately 260.8 kN of thrust. In the literature, the turbofan engine produces 35% of SUSAN’s SLS thrust while the distributed electric propulsion system produces the remaining 65% [7]. Assuming that SUSAN’s propulsion system produces the same 260.8 kN of thrust, then the turbofan engine must produce approximately 91.3 kN of thrust. Based on this

value, FAST only under-predicted the engine thrust by 7.78%, a much smaller level of error than the 35.43% reported by comparing SUSAN’s engine directly to the LEAP-1B engine. Thus, FAST still predicts the turbofan engine size with reasonable accuracy for an advanced propulsion system architecture.

4.3.3 Conclusions

In Experiment 2.0.2, the conventional and advanced aircraft concepts were modeled in FAST and calibrated to ensure that the output system-level performance parameters matched those from the literature. By embedding the GPSA Framework into FAST’s propulsion system analysis methods, it was hypothesized that system-level performance could be accurately estimated for both conventional and advanced aircraft concepts. FAST’s ability to accurately predict system-level performance of such aircraft was tested by modeling both the Embraer ERJ175LR, a conventional aircraft concept, and NASA’s SUSAN aircraft, an advanced aircraft concept. With little model calibration, both the MTOW and block fuel estimated by FAST were within 2% error of the corresponding values published in literature, meeting the prescribed accuracy threshold within Experiment 2.0.2.

Based on the results from Experiment 2.0.2, integrating the GPSA Framework into FAST exhibited promising capabilities to model both conventional and advanced aircraft concepts, regardless of the propulsion system installed. The configuration-agnostic propulsion system analysis offered by the GPSA Framework accurately predicts the power loads in each propulsion system component while FAST estimates the aircraft’s corresponding flight performance. The ability to accurately model both system- and subsystem-level performance emphasizes FAST’s utility for broader system architecture optimization studies performed in Chapter 6.

With Experiments 2.0.1 and 2.0.2 completed, both the GPSA Framework and FAST have been demonstrated to analyze advanced system architectures in a configuration-agnostic manner. Next, Experiment 2.0.3 demonstrates that the GPSA Framework can systematically represent and analyze propulsion system architectures and predict an aircraft’s system-level performance in a design space exploration study. Rather than defining different architecture, operational, and efficiency matrices for each propulsion system architecture, as done in the prior experiments, Experiment 2.0.3 fixes the size of the aforementioned matrices and only modifies their values, effectively changing the component connections to attain different propulsion system architectures and corresponding power management strategies.

4.4 Electrified Aircraft Design Space Exploration (Experiment 2.0.3)

The successful completion of Experiment 2.0.2 demonstrated that, after integrating the GPSA Framework into FAST, both conventional and advanced aircraft configurations were accurately modeled within approximately 2% modeling error at the system-level. In the prior experiment, both aircraft were defined in FAST utilizing differently sized architecture, operational, and efficiency matrices, each representing their own propulsion system architecture. Rather than redefining the propulsion system architecture manually, a script can be wrapped outside of FAST dynamically to modify the architecture, operational, and efficiency matrices input, effectively generating different propulsion system architectures and corresponding power management strategies. Within such a script, the matrices may be updated from a list of preset propulsion architectures (prescribed by the user) or in a more automated fashion from another analysis module (such as an optimization code), enabling large-scale parametric trade studies.

Therefore, Experiment 2.0.3 serves as an initial test to verify that FAST and the integrated GPSA Framework can perform such trade studies, enabling quantification of system-level performance tradeoffs by systematically representing and analyzing multiple propulsion systems using architecture, operational, and efficiency matrices of a fixed size. By keeping the matrix size constant, propulsion system components may be active or inactive, indicating their inclusion or exclusion in the propulsion system architecture, respectively, making the process amenable to parametric trade studies, demonstrated in this chapter, or numerical optimization, demonstrated in Chapter 6.

4.4.1 Experimental Setup

To test Hypothesis 2.0 and the GPSA Framework’s capabilities to analyze multiple propulsion system architectures, a design space exploration study is performed utilizing the ERJ175LR model that was utilized in Experiment 2.0.2. Rather than sizing the aircraft with its conventional propulsion system, the ERJ175LR is sized with a variety of electrified propulsion systems, each resulting in a distinct aircraft design.

Different propulsion system architectures are developed by changing the values in the architecture matrix, followed by corresponding changes to the operational and efficiency matrices. In each propulsion system architecture, up to seven components are allowed, corresponding to the respective rows/columns of the architecture matrices in Fig. 4.7: 1) fuel;

2) battery; 3) turboshaft engine; 4) electric generator; 5) electric motor; 6) propeller/fan; and 7) thrust power.

Four different propulsion system architectures and corresponding power management strategies are tested:

1. Parallel hybrid propulsion architecture from Fig. 3.1 in Chapter 3 with a 5% power split between the propeller and electric motor on takeoff,
2. Series hybrid propulsion architecture from Fig. 3.8 in Chapter 3 with a 5% power split between electric motor and electric generator on takeoff,
3. Series-Parallel hybrid propulsion architecture from Fig. 3.9 from Chapter 3 with a 5% power split between the propeller and electric motor during takeoff and climb, and 50% and 10% power splits between the electric motor and battery during takeoff and climb, respectively, and
4. Turboelectric propulsion architecture from Appendix A.1.

For convenience and simplicity, the architecture matrices for all four propulsion system architectures are displayed in Fig. 4.7, with each one representing the necessary components and connections for one propulsor. Since the ERJ175LR is powered by two turbofan engines, the respective matrices were modified in FAST with additional turboshaft engines, electric motors/generators and propellers where appropriate.

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(a) Parallel hybrid.

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(b) Series hybrid.

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(c) Series/Parallel-partial hybrid.

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(d) Turboelectric.

Figure 4.7: Propulsion system architectures examined in the electrified aircraft design study for Experiment 2.0.3. Each row/column corresponds to: Fuel; Battery; Turboshaft Engine; Electric Generator; Electric Motor; Propeller; and Thrust Power Sink.

Throughout Experiment 2.0.3, the following technology assumptions were made:

1. Battery Gravimetric Specific Energy: 250 Wh/kg
2. Electric Generator and Motor Power-Weight Ratio: 10 kW/kg

Additionally, to better explore the tradeoffs between propulsion system architectures, the efficiency matrices no longer assume perfect component and path efficiencies, previously represented as $\mathbf{1}_{n \times n}$ in Experiment 2.0.1, but are now populated with component efficiencies of 96% for all electric generators and motors and 80% for all propellers. Turboshaft engine efficiencies are computed separately in FAST's engine off-design analysis [177] and do not need to be set prior to the analysis.

While convenient for the initial GPSA Framework verification in Experiment 2.0.1, assuming perfect component efficiencies suggests that all components function similarly and exhibit no losses, resulting in undersized propulsion system components and adversely impacting the results of the design space exploration study. Such an assumption unfairly favors turboshaft engines that normally operate at 40% to 50% overall efficiencies, much lower than the 96% efficiencies available in electric generators/motors. Thus, the design space exploration could favor conventional propulsion system architectures with turboshaft engines instead of electrified ones, resulting in a misleading design space exploration study.

4.4.2 Results: System- and Subsystem-Level Performance

Upon installing the respective electrified propulsion system architectures on the ERJ175LR and re-sizing each configuration in FAST while flying the prescribed mission profile with the corresponding power management strategy, both system- and subsystem-level performance and sizing parameters were recorded, as listed in Tab. 4.4. The percent differences recorded are measured with respect to the baseline ERJ175LR sized in Section 4.3.

Table 4.4: Embraer ERJ175LR variant sizing and performance parameters.

Propulsion Architecture	Parallel Hybrid	Series Hybrid	Series- Parallel Hybrid	Turboelectric
MTOW (kg)	37,499 (-2.9%)	83,365 (+115.8%)	43,511 (+12.6%)	82,654 (+113.9%)
Block Fuel (kg)	9,123 (-2.9%)	22,011 (+134.2%)	9,750 (+3.8%)	21,351 (+127.2%)
Engine Thrust (kN)	56.9 (-7.8%)	137.2 (+122.4%)	70.6 (+14.3%)	143.2 (+132.1%)
Cruise SFC (mg/Ns)	19.4 (-0.8%)	13.4 (-31.1%)	18.0 (-8.0%)	13.3 (-32.1%)
EG Power (MW)	—	9.15	2.39	9.55
EM Power (MW)	0.21	9.25	2.41	9.17

The results presented in Tab. 4.4 suggest that the parallel hybrid configuration minimizes fuel burn relative to both the conventional and other electric propulsion systems installed on the ERJ175LR. The power boost provided by the battery and electric motor enables the turboshaft engine to be downsized, reducing the weight onboard the aircraft. Despite the smaller engine size, it is operated at a higher throttle setting, slightly reducing the cruise TSFC. Thus, the system-level benefits span both the turboshaft engine’s size and operations during the design mission. Further system-level benefits could be reaped by varying the takeoff power split and added a climb power split to explore the design space and identify the operational strategy that further minimizes fuel burn. Despite the promising results, the propulsion system analysis excluded a gearbox between the turboshaft engine, electric motor,

and propeller, possibly adding a significant weight penalty to the overall system if this were considered.

In the series hybrid and turboelectric configurations, the engine TSFC decreases by over 30% due to the larger engine size, but requires more fuel as a result of adding the electric components. Each electric generator/motor weighs between 915 to 955 kg, creating a significant weight penalty and requiring the aircraft size to more than double relative to the conventional configuration. Thus, arranging the electric generator and motor in series requires them to output nearly the same power as the turboshaft engine to spin the propeller, necessitating larger component sizes than those in the parallel hybrid configuration. While the rotational motion between the turboshaft engine and propeller are decoupled, allowing each one to operate at its own optimal conditions, the larger electric generator and motor prevent any system-level benefits from being attained.

4.4.3 Conclusions

The results found in Experiment 2.0.3 demonstrate that the GPSA Framework integrated into FAST enables system-level tradeoffs to be quantified between aircraft configurations with disparate propulsion system architectures. Though the series hybrid, series-parallel hybrid, and turboelectric configurations used the same components, the propulsion system connections and corresponding operational strategies were different. Rather than manually redefining the connections and power flows between components, the architecture, operational, and efficiency matrices embedded in the GPSA Framework were modified accordingly. The demonstrated approach is particularly useful during early conceptual design when little is known about the configuration, few design decisions have been made, and multiple candidate system architectures are being explored.

In summary, integrating the GPSA Framework with FAST, a parametric aircraft sizing tool, enabled both system-level performance (block fuel, MTOW) and subsystem-level performance (engine size, cruise SFC, and electric machine sizes) to be assessed by sizing an Embraer ERJ175LR with multiple candidate electrified propulsion system architectures. The electrified aircraft design space exploration study showcased the GPSA Framework's ability to analyze multiple propulsion system architectures using architecture, operational, and efficiency matrices of a fixed size, underscoring its usefulness in future automated design space exploration or system architecture optimization studies.

4.5 Chapter Summary

Research Question and Hypothesis 2.0 focused on formally representing and analyzing propulsion system architectures and their operations in a configuration-agnostic manner, enabling system-level tradeoffs to be quantified between an aircraft modeled with the same top-level aircraft requirements and technology levels but disparate propulsion system architectures. While the GPSA Framework captured subsystem-level performance, integration with a parametric aircraft sizing tool was necessary to predict system-level performance. In Chapter 4, the GPSA Framework was integrated with FAST, the Future Aircraft Sizing Tool, to predict system-level performance of an aircraft concept with any propulsion system installed.

Additionally, the literature review on existing parametric aircraft design tools indicated that component/subsystem definitions are needed to perform the analysis, requiring configuration-specific geometries to be known apriori, preventing multiple system architectures from being explored during early-phase conceptual design. Therefore, aside from serving as a foundation for integrating the GPSA Framework into an aircraft sizing code, FAST also provides advancements to the state-of-the-art in configuration-agnostic aircraft analysis, distinguishing itself with three novel capabilities. First, FAST provides propulsion system-agnostic design flexibility, accommodating a wide range of propulsion architectures, enabling comparative analyses between aircraft concepts with different propulsion system architectures, including hybrid-electric, fully-electric, and conventional designs. Second, FAST's energy-based segment analysis assumes that the aircraft is a point mass, enabling a flexible analysis across diverse mission profiles, making it well-suited for evaluating configurations that require rapid analysis at a more granular level than existing Breguet Range Equation-level analyses for electrified aircraft concepts. Lastly, the simplifying assumptions described in Section 4.2.3 enable a rapid, integrated aircraft sizing and performance analysis, typically converging on a design in under one minute. The rapid analysis facilitates iterative design space exploration and optimization, even for complex propulsion system architectures. Thus, FAST's capabilities make it an effective tool for design space exploration and sensitivity analysis, addressing the need for flexible, propulsion architecture-independent tools that support emerging technologies in aviation. Additional contributions to advance the state-of-the-art were made in FAST, but cannot be claimed by the author and are not reviewed in this thesis because they were developed by other collaborators.

To test the integrated GPSA Framework within FAST, Experiment 2.0.2 was designed to confirm that both system- and subsystem-level performance can be assessed for known conventional and advanced aircraft configurations. In Experiment 2.0.2, the ERJ175LR,

a conventional aircraft, and NASA’s SUSAN concept, an advanced aircraft configuration with a series/parallel-partial hybrid propulsion system architecture, were analyzed in FAST. With minimal calibration, both the aircraft’s system- and subsystem-level performance were accurately modeled, demonstrating FAST’s ability to analyze both conventional and advanced aircraft configurations. System and subsystem-level performance were modeled within 2% error, a reasonable error threshold for a conceptual design tool, particularly if little is known about the aircraft configuration. Thus, the GPSA Framework was successfully integrated into FAST, demonstrating its potential to predict system-level performance in early-phase conceptual aircraft design studies. More broadly, the promising results from the study continued to corroborate Hypothesis 2.0, suggesting that representing a propulsion system in a graph-based manner with component constitutive models and power transfer relationships was sufficient to predict both system- and subsystem-level performance.

After the integrated FAST-GPSA Framework implementation was tested in Experiment 2.0.2, confirming that any aircraft and propulsion system architecture can be analyzed within FAST, an additional study was needed to assess the GPSA Framework’s use for quantifying system-level tradeoffs. To this end, Experiment 2.0.3 tested how multiple propulsion system architectures could be represented and analyzed within a fixed matrix size; changes in the architecture matrix warranted corresponding changes in the operational and efficiency matrices, defining a new propulsion system architecture and power management strategy. Aside from quantifying system-level tradeoffs, testing the aforementioned approach also assessed the GPSA Framework’s ability to support large-scale parametric trade studies or system architecture optimization.

The GPSA Framework’s ability to represent and analyze multiple propulsion system architectures with a fixed matrix size was tested by designing four electrified propulsion system architectures, installing them on the ERJ175LR, and resizing each configuration in FAST, resulting in a distinct aircraft size for each configuration analyzed. The resulting analysis demonstrated that aircraft with disparate propulsion architectures can be easily defined and rapidly analyzed with the GPSA Framework embedded in FAST, once again confirming that the capabilities described in Hypothesis 2.0 are sufficient for representing and analyzing propulsion system architectures at both system- and subsystem-levels. From the design study, it was found that the parallel hybrid propulsion system architecture demonstrated system-level benefits over the conventional and other electrified propulsion system architectures, primarily due to the downsized turbofan engine (system-level weight savings) and its more efficient operations (reduced SFC, and hence, fuel burn) during cruise.

The results from the design study are not evidence that the parallel hybrid propulsion system architecture is the best electrified propulsion system architecture candidate, but is an artifact of the technology levels and modeling assumptions utilized in the study.

Overall, the GPSA Framework was successfully integrated with FAST, demonstrating how subsystem-level architecture design studies can impact system-level performance during conceptual aircraft design. The results from Experiments 2.0.1 through 2.0.3 from Chapters 3 and 4 confirm that a graph-based representation with component constitutive models at each vertex and power transfer relationships along each edge will reproduce benchmark component power loads and overall energy conservation equations across representative conventional and electrified propulsion architectures within predefined accuracy thresholds. The work presented in this chapter provided a foundation for future large-scale parametric trade studies or system architecture optimization, explored further in Chapter 6 of this thesis.

Chapter 5

Generalized Engine Inoperative Performance Requirements

5.1 Introduction

Traditionally, improved system-level performance of future aircraft concepts focuses on incremental technology developments. The literature review indicated that an entire generation of technological improvements, spanning approximately 15 to 20 years, can reduce fuel burn by up to 20 or 30% for turbofan- or turboprop-powered aircraft, respectively [30, 39]. Besides technological developments, certification requirements also impact system-level performance.

The application of FAA Part 25 and EASA CS-25 regulations to advanced aircraft concept certification faces multiple obstacles, particularly for engine inoperative conditions¹. The FAA climb performance regulations were developed in the early 1960s [38], and only considered aircraft powered by two, three, or four gas turbine engines, making direct application to distributed and heterogeneous propulsion systems ambiguous. Additionally, novel approaches are needed to scale an propulsion system’s size under engine inoperative conditions, necessitating an advancement from traditional scale factors used in existing aircraft design literature and textbooks that assume all propulsors are homogeneously sized [103].

Outdated, poorly applicable requirements could limit system-level performance of future aircraft concepts and their advanced propulsion systems. Thus, there is a need to develop generalized engine inoperative performance requirements and propulsion system sizing heuristics that are applicable to any conventional or advanced aircraft propulsion system, overcoming limitations on their current definition as a function of the number of propulsors installed.

¹Although FAA Part 25 regulations are broadly harmonized with EASA CS-25 regulations, differences exist and certification is authority-specific. The present work uses the FAA Part 25 climb gradient structure as a reference baseline for performance requirement generalization, and refers to them as “FAA regulations.”

Small adjustments made to the certification requirements and propulsion system sizing heuristics have the possibility to widen the design space, possibly resulting in improved system-level performance [188]. Such system-level performance improvements could be nominally equivalent to multiple years of incremental technological developments, underscoring the need to broaden the applicability of the presently published certification requirements for future aircraft concepts.

5.1.1 Research Questions, Hypotheses, and Experiments

To support advanced aircraft and propulsion system design, generalized engine inoperative performance requirements must be developed in a configuration-agnostic manner, accommodating advanced and distributed propulsion systems, including those with disparately sized propulsors. Research Question 3.0 investigates how configuration-agnostic flight performance requirements can be defined to accommodate all propulsion system architectures:

RQ 3.0: How can the currently published FAA climb performance requirements for aircraft with discrete, homogeneously sized propulsors be generalized to accommodate propulsion systems with heterogeneously sized or distributed propulsors and their corresponding critical failure modes?

In the literature review, two prior approaches were identified for selecting engine inoperative performance requirements for advanced aircraft concepts leveraging distributed propulsion systems: 1) utilize the existing FAA Part 25 requirements for 4-engine aircraft; or 2) develop an extrapolation function to predict the climb requirements for an aircraft with n propulsors installed on the aircraft. The former approach defines the climb performance requirements within those already published by the FAA and imposes the most stringent ones available, neglecting the failure mode severity. The latter approach prescribes a larger climb gradient as the number of propulsors installed on the aircraft increases, expanding existing parameterizations for aircraft with n propulsors, but results in required climb gradients that are steeper than those published under FAA Part 25 regulations. The strengths and shortcomings of both approaches can be combined to develop engine inoperative performance requirements in a configuration-agnostic manner.

Rather than defining the climb performance requirements by the number of propulsors installed on the aircraft, the propulsion system failure severity can be better characterized by the specific excess power lost during a propulsion system failure, indicating the aircraft's degraded performance capabilities, while remaining anchored to those presently published by

the FAA. The specific excess power, described in Eq. 5.1, may be a more useful parameter than the number of propulsors installed because of its ability to capture the system-level impacts of a propulsion system failure; the thrust power available (TV) decreases due to the engine failure and the drag power (DV) increases due to the windmilling drag from the engine failure and the trim drag to maintain directional stability. As a result, the climb performance requirements are prescribed by a continuous variable rather than the number of propulsors installed on the aircraft (a discrete variable), projecting the propulsion system failure severity to a climb performance requirement.

$$P_s = \frac{TV - DV}{W} \quad (5.1)$$

Based on the aforementioned considerations, Hypothesis 3.0 answers Research Question 3.0 by describing a methodology to construct configuration-agnostic flight performance requirements for any propulsion system failure severity:

Hyp 3.0: A monotonic, bounded mapping between the specific excess power loss and the required climb gradient will reproduce the climb gradient requirements implied by the published regulations for conventional aircraft utilizing discrete, homogeneously sized propulsors while extending the same flight performance logic to accommodate aircraft concepts utilizing propulsion systems with heterogeneously sized or distributed propulsors.

Hypothesis 3.0 can be tested in two parts. First, an experiment is necessary to confirm that the specific excess power loss is a suitable parameter to establish a monotonic, bounded mapping to the required climb gradient under one-engine inoperative conditions. Then, generalized climb performance requirements can be constructed based on the suitable parameterization identified, ensuring that the published climb gradients for conventional aircraft are reproduced.

Parameter suitability can be assessed by performing a test to compare the relationship between the flight performance degradation under one-engine inoperative conditions to possible parameters describing the propulsion system failure severity. In addition to the specific excess power loss, the installed power available lost during engine inoperative conditions is also considered, highlighting that the propulsion system failure severity may be impacted by *only* the power lost during the failure, allowing other system-level effects such as windmilling and trim drag to be neglected. Experiment 3.0.1 compares the suitability of these two parameters

by exploring the relationship of each one against the climb performance degradation during one-engine inoperative conditions.

Experiment 3.0.1: Verify that the specific excess power loss is a sufficient state variable to enable a physically consistent mapping between the propulsion system failure severity and off-nominal climb performance.

1. Simulate all-engine operative (AEO) and one-engine inoperative (OEI) conditions on a benchmark suite of conventional aircraft under a diversion mission profile at various takeoff and climb speeds. Record the power available, specific excess power, and maximum attainable climb gradients across the climb from takeoff to cruise altitudes.
2. Compute the installed power available, specific excess power, and climb gradient loss fractions across the respective AEO and OEI conditions simulated.
3. Plot the climb gradient loss fraction against the installed power available and specific excess power loss fractions. Compare the R^2 value and RMSE between the two plots.

In Experiment 3.0.1, both AEO and OEI simulations are performed on a benchmark suite of conventional aircraft flown on a diversion mission profile. After performing the simulations, the specific excess power, installed power available, and climb gradient loss fractions are computed, indicating the normalized propulsive and climb performance degradations due to the engine failure. The normalized flight performance degradations encapsulated in the loss fractions enable comparison across the entire benchmark suite of conventional aircraft, regardless of the aircraft size or number of propulsors installed. The mapping with the strongest correlation and/or smallest error is then selected as the suitable parameter to generate a monotonic, bounded mapping to the required climb gradients. It is expected that the specific excess power loss is a more suitable parameter due to its ability to capture the both aerodynamic and propulsive effects of a propulsion system failure within one quantity.

After confirming that the specific excess power loss is a suitable parameter for mapping the propulsion system failure severity to the required climb gradients, the mappings are generated. As stated in Hypothesis 3.0, the mapping must reproduce the climb gradient requirements implied by the existing published regulations. Therefore, the mappings must be validated against a subset of the representative benchmark aircraft by predicting the required climb gradient for a given specific excess power loss, maintaining requirement consistency for previously certified aircraft while extending to future aircraft concept designs with advanced

propulsion systems. The overall mapping construction and validation procedure is outlined in Experiment 3.0.2.

Experiment 3.0.2: Construct a mapping between the specific excess power losses and FAA-published climb performance requirements and validate its ability to accurately estimate the FAA-published climb performance requirements for a benchmark suite of conventional aircraft.

1. Split the available specific excess power loss and corresponding FAA-published climb performance requirements into training and validation data at an 80-20% ratio across the benchmark suite of conventional aircraft.
2. For each FAA climb performance requirement (25.111, 25.121[a]-[d]):
 - (a) Utilize the training data to construct a monotonic, bounded mapping between the specific excess power losses and FAA-published climb performance requirements.
 - (b) Use the validation data to predict the climb performance requirements from the constructed mapping as a function of the specific excess power losses. Compute the absolute and relative error between the predicted climb performance requirements and those published by the FAA for each conventional aircraft.

The mappings constructed in Experiment 3.0.2 act as proxy flight performance requirements for designing advanced aircraft concepts. Although the generalized flight performance requirements are useful for providing first-order aircraft sizing estimates, they are not substitutes for demonstrating compliance under FAA Part 25 regulations. For advanced aircraft concepts in the near future, compliance will likely be demonstrated under FAA “Special Conditions” [189], requiring each airframe manufacturer to work directly with the FAA to determine the corresponding primary propulsion system failure mode and flight performance requirements.

In addition to the proxy flight performance requirements, a correction factor is needed to scale the propulsion system size as the primary failure mode severity increases. Traditional $n/(n - 1)$ correction factors used to size conventional aircraft implicitly assume that all propulsors are sized homogeneously, limiting their applicability to aircraft with heterogeneously-sized propulsors. Thus, a sizing correction factor must be constructed as a function of the propulsion system failure severity, fitting the specific excess power loss to the traditional $n/(n - 1)$ correction factors, enabling direct application to advanced propulsion systems. Intuitively, the propulsion system size correction factor should increase as

the propulsion system failure severity increases, suggesting that a monotonically increasing function is appropriate. Since multiple parametric curves could be utilized to construct the correction factor, a sensitivity study is necessary to identify the preferred curve for defining the engine inoperative correction factor.

Additionally, the maximum correction factor returned under the traditional $n/(n - 1)$ correction factor is 2 (corresponding to the correction factor for 2-engine aircraft), requiring the propulsion system to double in size. For the engine inoperative correction factor construction, a constraint must be placed on the maximum allowable correction factor to be returned, ensuring that the propulsion system is not significantly over-sized relative to its conventional counterparts. Such a value may be chosen heuristically, requiring experimentation to find a suitable one.

Both the parametric curve and maximum allowable correction factor are varied in Experiment 3.0.3, serving as a sensitivity study to explore possible engine inoperative correction factor constructions.

Experiment 3.0.3: Identify the sensitivity of the engine inoperative correction factor curve with respect to the curve parameterization and maximum allowable engine inoperative correction factor.

1. Define a set of maximum allowable engine inoperative correction factors.
2. Define a set of candidate parametric curves to fit a mapping of the specific excess power loss to traditional one-engine inoperative sizing correction factors.
3. For all combinations of maximum allowable engine inoperative correction factors and candidate curves, perform a constrained optimization to minimize the RMSE of the curve fit between the specific excess power loss and traditional one-engine inoperative sizing correction factors without exceeding the maximum allowable engine inoperative correction factor.
4. For each candidate curve, plot the RMSE as a function of the maximum allowable engine inoperative correction factor.

After completing Experiments 3.0.1 through 3.0.3, both the proxy climb performance requirements and engine inoperative correction factor will be parameterized by the specific excess power, rendering them useful for configuration-agnostic aircraft design. Such constructions can then act as conceptual design heuristics to size an advanced aircraft concept

under various propulsion system failure modes. The resultant system- and subsystem-level performance can be compared across configurations and against the same concept sized under the existing FAA climb performance requirements with a traditional engine inoperative correction factor. The comparison between configurations merits exploring an additional research question:

Research Question 3.1: What are the system-level impacts of sizing an advanced aircraft concept with the generalized climb performance requirements and engine inoperative correction factor relative to one sized under FAA-published requirements and traditional propulsion system scaling factors for the same design mission?

System-level performance will be heavily impacted by the propulsion system failure severity because it governs the proxy climb performance requirements and engine inoperative correction factor constructed in Experiments 3.0.2 and 3.0.3, respectively. During aircraft sizing, a generic climb performance requirement, as described in Eq. 5.2, indicates that the propulsion system size is impacted by the correction factor (κ) and climb gradient (γ) [103]. The equation’s form indicates that the engine inoperative correction factor scales the propulsion system sizing requirements, and will likely be the more dominant term.

$$T/W = \kappa \left[f(W/S, C_{L_{\max}}, C_{D_0}, AR, e, \dots) + \gamma \right] \quad (5.2)$$

Based on the observation from Eq. 5.2 and knowing that the engine inoperative correction factor must return larger values for more severe propulsion system failures, Hypothesis 3.1 predicts the resultant aircraft size as a function of its propulsion system failure severity:

Hypothesis 3.1: For an advanced aircraft concept with a propulsion system failure mode that produces a smaller specific excess power loss than a conventional aircraft under a legacy one-engine inoperative failure mode, sizing under the generalized climb performance requirements and engine inoperative correction factor will reduce over-sizing and improve system-level performance while maintaining the required off-nominal climb gradient for the corresponding failure mode.

Hypothesis 3.1 can be tested by sizing an advanced aircraft concept under the existing FAA Part 25 requirements with a traditional engine inoperative correction factor and the configuration-agnostic climb performance requirements and engine inoperative correction factor. By fixing the technology levels, top-level aircraft requirements, and design mission

profile, the aircraft concept sized under FAA Part 25 requirements can be compared to those sized under the configuration-agnostic climb performance requirements. Experiment 3.1.1 outlines the steps to compare the system-level performance of an advanced aircraft concept sized under the given sets of requirements and engine inoperative correction factors.

Experiment 3.1.1: Quantify the system-level impacts of an advanced aircraft concept sized under varying propulsion system failure modes relative to one sized under FAA-published climb performance requirements and traditional propulsion system scaling factors.

1. Select a representative advanced aircraft to model and prescribe multiple propulsion system failure modes of varying severity, resulting in different specific excess power losses.
2. Predict the baseline system-level responses by sizing the advanced aircraft concept under the existing FAA-published climb performance requirements by generating a constraint diagram, selecting a design point, and sizing the aircraft in FAST for the selected design point.
3. Repeat the same sizing process for each novel propulsion system failure mode but utilizing the generalized performance requirements and engine inoperative correction factor instead of the FAA-published climb performance requirements and traditional propulsion system scaling factors.
4. Compare the total energy expenditure, maximum takeoff weight, propulsion system component sizes and time-to-climb for the configurations sized under the generalized climb performance requirements relative to that sized under the FAA-published climb performance requirements.

The results from Experiment 3.1.1 identify the system-level impacts of sizing an aircraft using the proposed conceptual design heuristics, providing a comparison against an advanced aircraft sized under the presently published FAA regulations. Recall that the design heuristics are not intended to be used as a means for certification, but can provide first-order sizing estimates to identify possible propulsion system failure modes worthy of certification under the FAA’s “Special Conditions” [189] in the future.

Together, Research Questions 3.0 and 3.1 provide a foundation for generating configuration-agnostic climb performance requirements and assessing the system-level impacts of sizing an advanced aircraft with such requirements.

5.1.2 Chapter Outline

The remainder of this chapter is structured as follows. First, the fleet-level operational analysis is performed in Section 5.2 to compute the AEO and OEI flight performance across all benchmark conventional aircraft. Then, Experiment 3.0.1 is performed to confirm that the specific excess power loss is a suitable parameter for constructing a monotonic, bounded mapping to the required climb gradient. Next, Section 5.4 constructs and tests the required climb gradient mappings as part of Experiment 3.0.2, validating that they accurately estimate the climb gradients implied by the published regulations for the benchmark conventional aircraft. After that, Experiment 3.0.3 is performed in Section 5.5, identifying the sensitivity of the engine inoperative correction factor construction with respect to its curve parameterization and maximum allowable correction factor.

Using the constructed climb performance requirements and engine inoperative correction factor, Section 5.6 executes Experiment 3.1.1, revealing the system-level impacts of sizing an aircraft under the proposed generalized requirements relative to those sized under the existing FAA Part 25 regulations. Lastly, Section 5.7 concludes the chapter, summarizes the key contributions, and reviews the findings from all experiments under Research Questions 3.0 and 3.1.

5.2 Parameter Suitability Test (Experiment 3.0.1)

5.2.1 Experimental Setup

This section describes the fleet-level operations simulations performed to identify the specific excess power and installed power available lost during engine inoperative conditions, as well as its corresponding climb performance degradation. In Section 5.2.1.1, the benchmark aircraft suite is defined, listing all the transport aircraft simulated in the operational analysis. Then, the diversion mission profile is defined as well as the aerodynamic performance corrections for engine inoperative conditions in Sections 5.2.1.2 and 5.2.1.3, respectively. Lastly, the specific excess power, installed power available, and climb gradient reduction fractions are defined in Section 5.2.1.4 for further use in Experiment 3.0.1.

5.2.1.1 Benchmark Aircraft Suite Definition and Modeling

Fleet-level simulations were performed on 156 conventional aircraft concepts previously certified under the FAA’s Part 25 (transport aircraft) regulations. The aircraft examined

span regional, narrowbody, and widebody configurations manufactured by Airbus, Boeing, and Embraer, and serve both the regional, domestic, and international markets. Table 5.1 lists the aircraft configurations and number of variants accounted for during the analysis ². Across the 156 aircraft analyzed, 118 aircraft had two engines installed, 4 aircraft had three engines installed, and 34 aircraft had four engines installed.

Table 5.1: Aircraft configurations and variants considered during the operational analysis.

Aircraft	Number of Variants
Airbus A300	15
Airbus A310	10
Airbus A320	46
Airbus A380	3
Boeing 707	1
Boeing 717	2
Boeing 737	14
Boeing 747	30
Boeing 757	3
Boeing 767	7
Boeing 777	4
Boeing 787	3
Embraer E135	2
Embraer E145	2
Embraer E170	4
Embraer E190	6
Lockheed Martin L-1011	4

All aircraft were modeled in FAST, described in Chapter 4, leveraging data from its Aerobase [190], a publicly-available database with over 400 aircraft previously certified by the FAA or EASA. The Aerobase primarily contains system-level performance parameters and weights such as maximum takeoff weight, fuel weight, and cruise speed. FAST is also an ideal tool for executing the simulations because its energy-based mission analysis assumes that the aircraft is a point mass, enabling a rapid analysis to accommodate the 156 aircraft considered in this study. Additionally, its graph-based propulsion analysis framework [191] and segment-based power management strategy definition allow individual engines to be powered on/off, thus enabling the engine inoperative simulations necessary for this work.

²The row for the Airbus A320 includes Airbus A318, A319, and A321 variants, as well as other “new engine option” (NEO) variants.

5.2.1.2 Diversion Mission Profile

Each aircraft was flown twice on a 370 km diversion mission illustrated in Fig. 5.1 – once under AEO conditions and once under OEI conditions, failing one engine after takeoff. During an engine failure, the aircraft will likely not continue to its destination airport and divert to a nearby one, thus justifying the shorter mission profile.

The takeoff and climb speeds, V_{Tko} and V_{Cib} , respectively, were varied to represent multiple operational strategies during departure, outlining an operational envelope for evaluating climb performance. The climb speed was computed by incrementing the takeoff speed with a fixed value, ΔV . The takeoff speed and increments are listed in Tab. 5.2. Additional boundary conditions for each mission profile segment are provided in C.1.

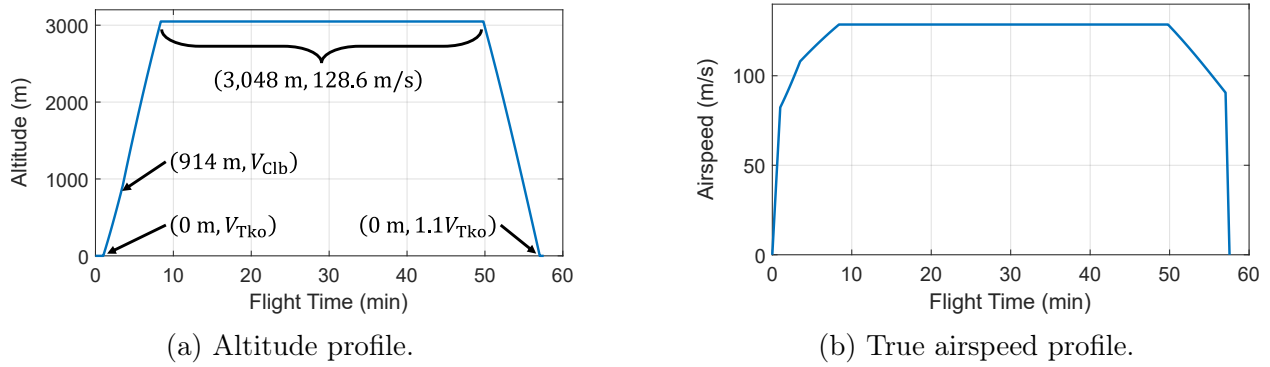


Figure 5.1: Mission profile for OEI diversion.

Table 5.2: Takeoff speeds and increments (both in m/s) simulated under OEI conditions. Note that $V_{Cib} = V_{Tko} + \Delta V$.

Speed	Lower Bound	Upper Bound	Step Size
V_{Tko}	66.88	102.89	5.14
ΔV	0.00	25.72	5.14

5.2.1.3 Aerodynamic Correction for Engine Inoperative Conditions

Previously in Chapter 4, FAST’s detailed aerodynamic model was introduced as an adaptation of an empirical drag polar model from Aviary, NASA’s recently developed open-source aircraft sizing tool for preliminary design [58], and heavily relies on the aircraft’s geometry. Due to

the lack of geometric data available for all 156 aircraft analyzed, FAST’s simpler aerodynamic model – assuming a constant lift-to-drag ratio during each flight segment – was instead utilized. To scale the constant lift-to-drag ratios for OEI conditions, the AEO lift-to-drag ratio was multiplied by a correction factor, and is dependent upon the flight conditions simulated and number of engines installed on the aircraft. The correction factor accounts for both windmilling and trim drag considerations, as previously described in Section 4.2.3.3.

The correction factors were obtained by performing a detailed aerodynamic analysis on six notional aircraft models – Airbus A300, Airbus A310, Airbus A320, Airbus A380, Boeing 777, and Embraer E175 – in FAST under both AEO and OEI conditions. For these six configurations only, detailed geometric data was extracted from publicly available CAD files [192, 193], enabling utilization of FAST’s detailed aerodynamic model. After simulating the models under AEO and OEI conditions, the aerodynamic performance fraction, defined as $(L/D_{\text{OEI}}) / (L/D_{\text{AEO}})$, was computed for both configurations. At each flight condition, the correction factor was estimated as the following:

1. 2-Engine Aircraft: the mean aerodynamic performance fraction across the Airbus A300, Airbus A310, Airbus A320, Boeing 777, and Embraer E175 models analyzed.
2. 3-Engine Aircraft: the average aerodynamic performance fraction estimated for both 2- and 4-engine aircraft.
3. 4-Engine Aircraft: the aerodynamic performance fraction from the Airbus A380 model analyzed.

All correction factors are listed in Appendix C.2, nominally reflecting a 20% decrease in the lift-to-drag ratio.

5.2.1.4 Propulsion and Flight Performance Parameters Computed

The propulsion system performance was recorded utilizing two different parameters: 1) the installed power available (TV); and 2) the specific excess power, previously defined in Eq. 5.1. The installed power available represents the maximum propulsion system power output for a given flight condition. The specific excess power represents the power remaining after overcoming the drag power (DV), and is normalized by the aircraft’s weight. Climb performance was computed by predicting the maximum available climb gradient, computed in Eq. 5.3.

$$\gamma = \tan \left[\arcsin \left(\dot{h}/V_{\infty} \right) \right] \quad (5.3)$$

To enable comparison of the aforementioned quantities across different aircraft sizes, their relative difference is computed with respect to nominal flight operations, serving as non-dimensional quantities to describe the propulsion system failure severity and its system-level impacts. Equations 5.4 through 5.6 define the specific excess power loss fraction (abbreviated as “SEPLF”), installed power available loss fraction (abbreviated as “IPALF”), and the climb gradient loss fraction (abbreviated as “CGLF”), respectively, representing a reduction in a given flight/propulsion system performance quantity.

$$P_{s_{\text{loss}}} = \frac{P_{s_{\text{AEO}}} - P_{s_{\text{OEI}}}}{P_{s_{\text{AEO}}}} = 1 - \frac{P_{s_{\text{OEI}}}}{P_{s_{\text{AEO}}}} \quad (5.4)$$

$$TV_{\text{loss}} = \frac{TV_{\text{AEO}} - TV_{\text{OEI}}}{TV_{\text{AEO}}} = 1 - \frac{TV_{\text{OEI}}}{TV_{\text{AEO}}} \quad (5.5)$$

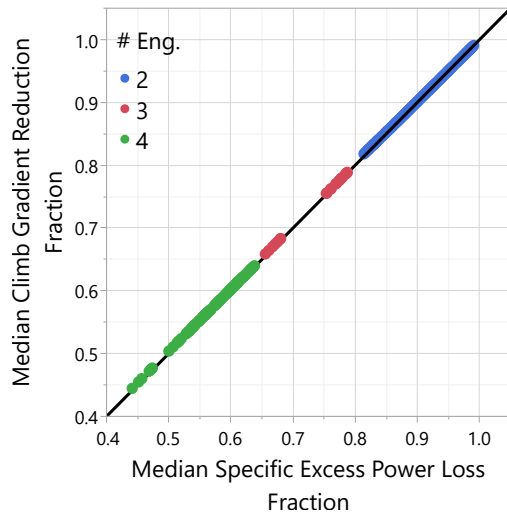
$$\gamma_{\text{loss}} = \frac{\gamma_{\text{AEO}} - \gamma_{\text{OEI}}}{\gamma_{\text{AEO}}} = 1 - \frac{\gamma_{\text{OEI}}}{\gamma_{\text{AEO}}} \quad (5.6)$$

All of the fractions defined remain bounded between 0 and 1, indicating that the respective quantities during off-nominal operations never exceed those during nominal operations.

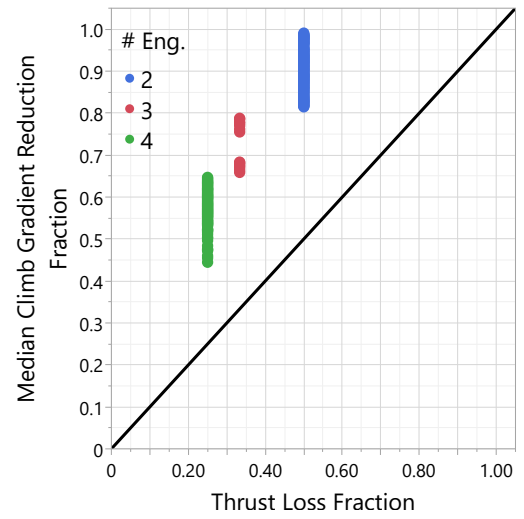
Since the SEPLF and CGLF values will vary based on the present flight conditions, the minimum, median, and maximum loss fractions were computed, indicating the lower, nominal, and upper bounds of such losses. The median loss fractions are primarily utilized for predicting flight performance degradation under OEI conditions, while the minimum/maximum ones are employed to assess the broader implications of the estimations made within this chapter. For the remainder of this chapter, assume that the median loss fractions are utilized unless otherwise specified. The IPALF, however, remains constant throughout the entire off-nominal flight simulation, and does not require minimum, median, and maximum loss fraction to be computed.

5.2.2 Results

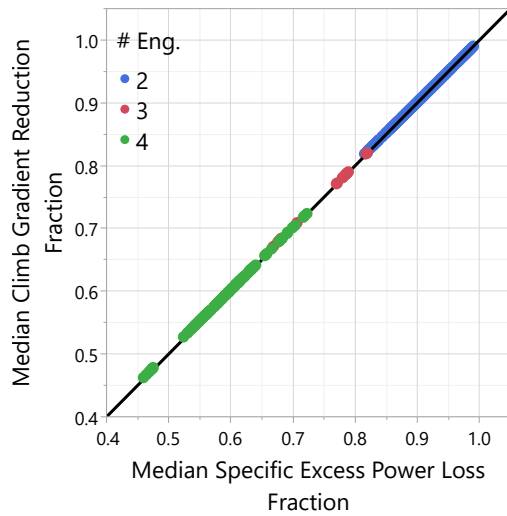
The relationships between the CGLF and the SEPLF/IPALF are plotted in Fig. 5.2 for all aircraft operations simulations within the operational envelope prescribed; the data in each plot spans all takeoff speeds and one ΔV value.



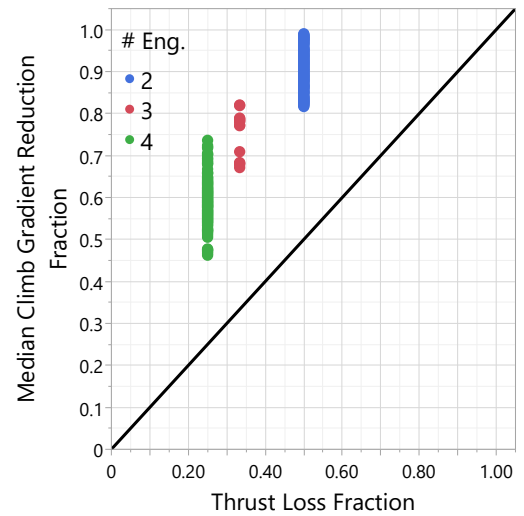
(a) SEPLF, $\Delta V = 0.00$ m/s.



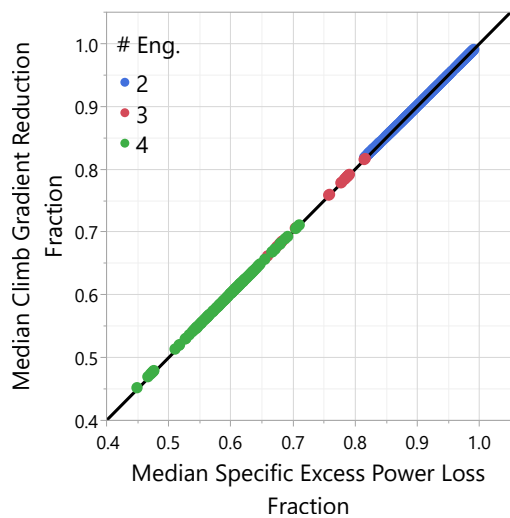
(b) IPALF, $\Delta V = 0.00$ m/s.



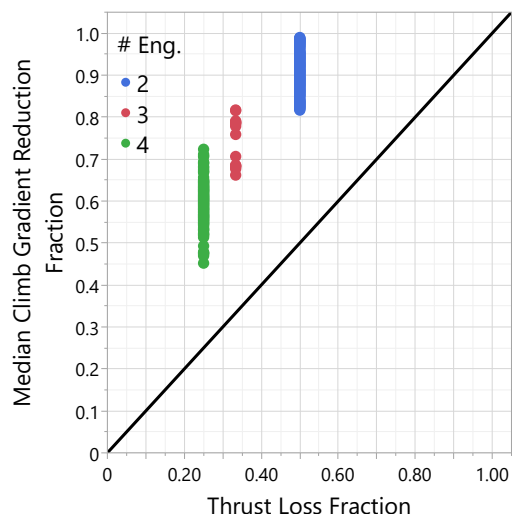
(c) SEPLF, $\Delta V = 5.14$ m/s.



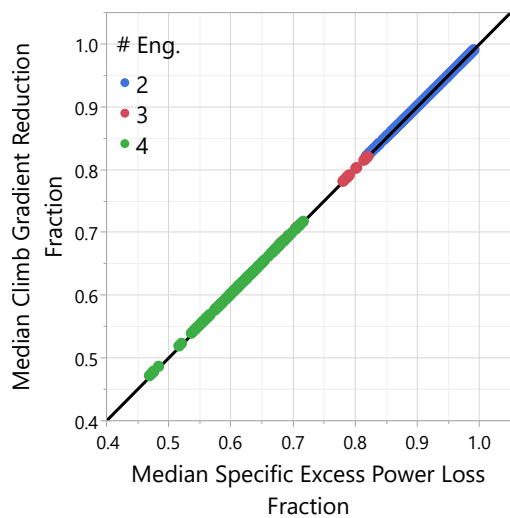
(d) IPALF, $\Delta V = 5.14$ m/s.



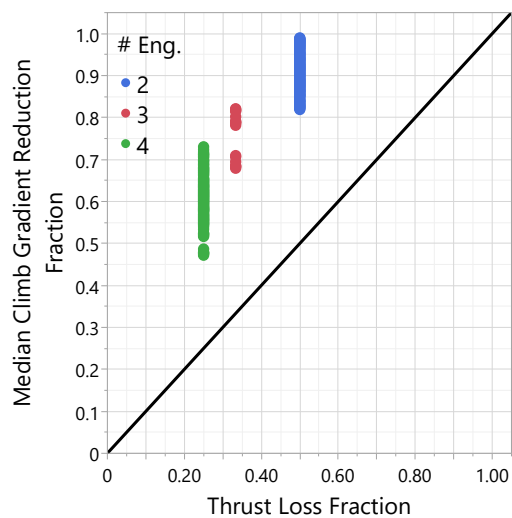
(e) SEPLF, $\Delta V = 10.29$ m/s.



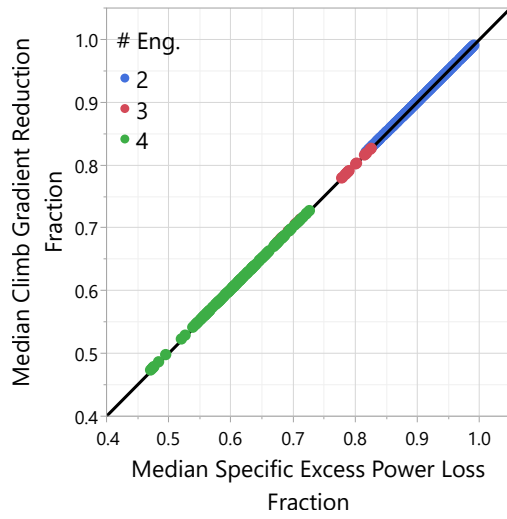
(f) IPALF, $\Delta V = 10.29$ m/s.



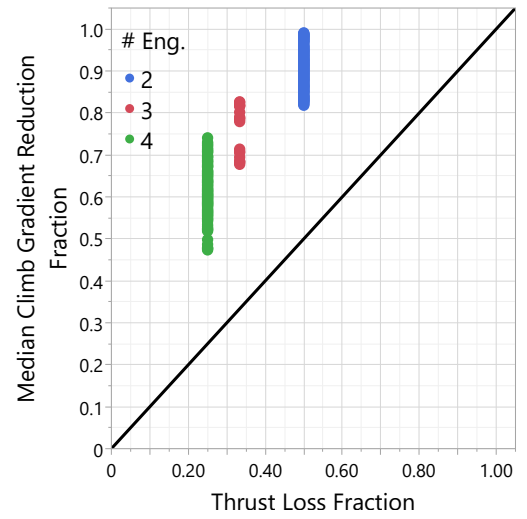
(g) SEPLF, $\Delta V = 15.43$ m/s.



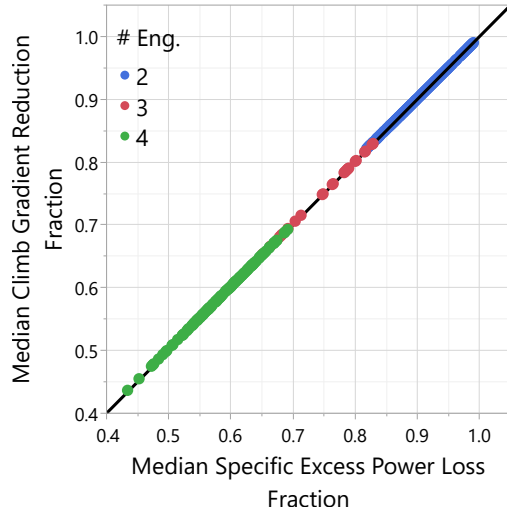
(h) IPALF, $\Delta V = 15.43$ m/s.



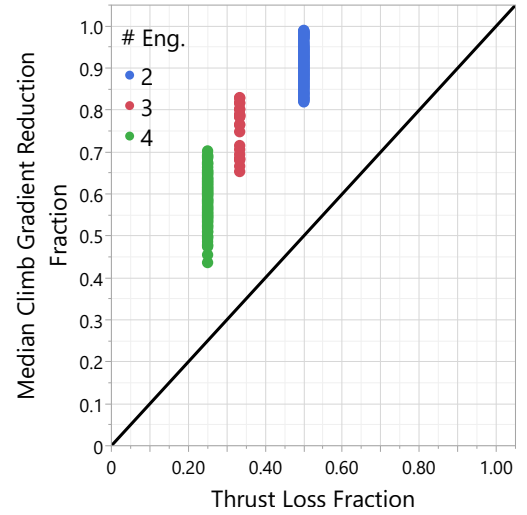
(i) SEPLF, $\Delta V = 20.58$ m/s.



(j) IPALF, $\Delta V = 20.58$ m/s.



(k) SEPLF, $\Delta V = 25.72$ m/s.



(l) IPALF, $\Delta V = 25.72$ m/s.

Figure 5.2: Climb performance impacts due to a propulsion system failure parameterized by the SEPLF or IPALF. The black diagonal line is a reference line indicating $CGLF = SEPLF$ or $CGLF = IPALF$, respectively.

In the plots, the IPALF are clustered based on the number of engines installed on the aircraft, returning different climb performance degradations for the same IPALF. As a result, the relationship between the CGLF and IPALF remains ambiguous. In contrast, the SEPLF are spread across the entire domain, resulting in a distinguishable CGLF for each SEPLF reported. Additionally, the SEPLF appears to be approximately equal to the CGLF, resulting in a one-to-one mapping between the propulsion system failure severity and climb performance degradation.

To assess the relationship between the two parameterizations in a quantifiable manner, a linear fit was generated, relating the SEPLF/IPALF to the CGLF recorded in each flight simulation. The resulting R^2 and RMSE values are listed in Tab. 5.3 for each ΔV analyzed.

Table 5.3: Statistical performance metrics recorded to assess parameter suitability.

ΔV (m/s)	R^2 (SEPLF)	R^2 (IPALF)	RMSE (SEPLF)	RMSE (IPALF)
0.00	0.9999	0.9126	0.0008	0.0462
5.14	0.9999	0.8961	0.0007	0.0481
10.29	0.9999	0.8991	0.0007	0.0477
15.43	0.9999	0.8883	0.0007	0.0482
20.58	0.9999	0.8823	0.0006	0.0487
25.72	0.9999	0.9005	0.0006	0.0484

Overall, the relationship between the SEPLF and CGLF is much stronger than that of the IPALF and CGLF. There is a strong correlation between the SEPLF and CGLF, as indicated by the high R^2 value. While the R^2 value for the IPALF-CGLF relation is approximately 0.90, it indicates a weaker correlation than that of the SEPLF-CGLF relation. Additionally, the RMSE for the SEPLF-CGLF relation is two orders of magnitude smaller than the IPALF-CGLF relation.

5.2.3 Conclusions

In Experiment 3.0.1, a relationship between the climb performance degradation and propulsion system severity during engine inoperative conditions was sought by exploring different propulsion system severity parameterizations. The climb performance degradation was quantified by the CGLF, while the propulsion system failure severity was quantified by the SEPLF and IPALF. Such parameters were estimated by simulating both AEO and OEI conditions on a diversion mission profile in FAST for a benchmark suite of 156 conventional

aircraft previously certified under FAA Part 25 requirements.

The SEPLF parameterization provided a monotonic mapping between the propulsion system failure severity and climb performance degradation. Furthermore, the SEPLF was approximately equal to the CGLF across the entire operational envelope, possibly suggesting weak sensitivity to the takeoff/climb speeds flown. In contrast, the IPALF failed to produce unique CGLF values for a given propulsion system failure severity. Beyond the ability to produce unique CGLF values for a given propulsion system failure severity, the parameterization utilizing the SEPLF returned higher R^2 values and lower RMSE values after performing a linear fit.

Thus, the SEPLF parameterization better captures the system-level impacts of the propulsion system failure severity by considering both the installed power available lost during the failure and the resultant aerodynamic performance penalties due to windmilling and trim drag. For Experiments 3.0.2 and 3.0.3, the SEPLF will serve as the independent variable to construct the generalized climb performance requirements and engine inoperative correction factor, respectively.

5.3 Examining the Relationship between the SEPLF and CGLF

In Experiment 3.0.1, there was a strong correlation between the SEPLF and CGLF, indicating that the two quantities were nearly equal across the entire operational envelope simulated. A deeper exploration of the aforementioned relationship was explored across the entire range of SEPLF and CGLF returned from the operational analysis.

As observed in Fig. 5.2, the SEPLF closely approximates the climb performance reduction from a given propulsion system failure. For a propulsion system failure on a 2-, 3-, and 4-engine aircraft, the SEPLF nominally spans between 80 to 100%, 65 to 85%, and 45 to 75%, respectively, also reflecting a similar degradation in climb performance. At smaller ΔV values (0.00 and 5.14 m/s), the points deviate the most from the reference line (indicating that $\text{CGLF} = \text{SEPLF}$), but still demonstrate a strong direct relationship. As ΔV increases, the deviation from the reference line decreases, indicating that the SEPLF better approximates the CGLF.

Based on the data observed, the SEPLF is approximately equal to the CGLF across the entire operational envelope considered, suggesting weak sensitivity with respect to the operational strategy. Further investigation is needed to assess if such a claim spans all

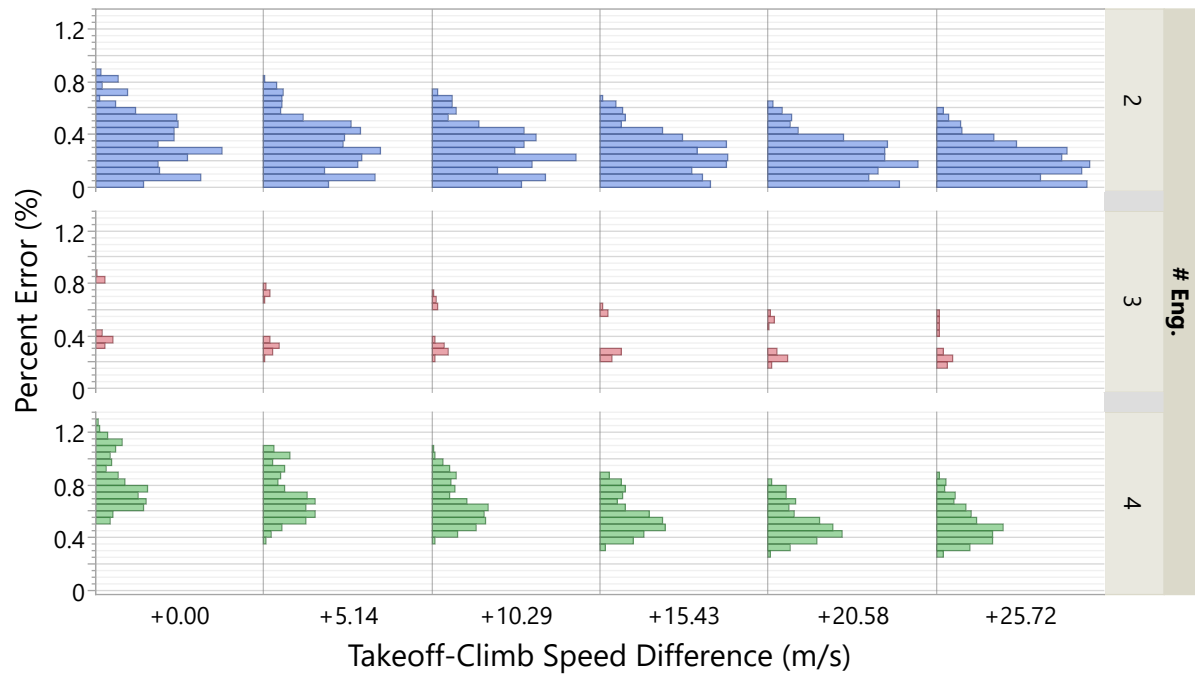
SEPLF and CGLF recorded from the operational analysis, and can be tested by computing the percent error between the minimum, median, and maximum SEPLFs and CGLFs, as indicated by Eq. 5.7. Histograms of the percent error for the minimum, median, and maximum SEPLFs/CGLFs are plotted in Fig. 5.3.

$$\text{Err} = 100 \times \frac{|\text{CGLF} - \text{SEPLF}|}{\text{SEPLF}} \quad (5.7)$$

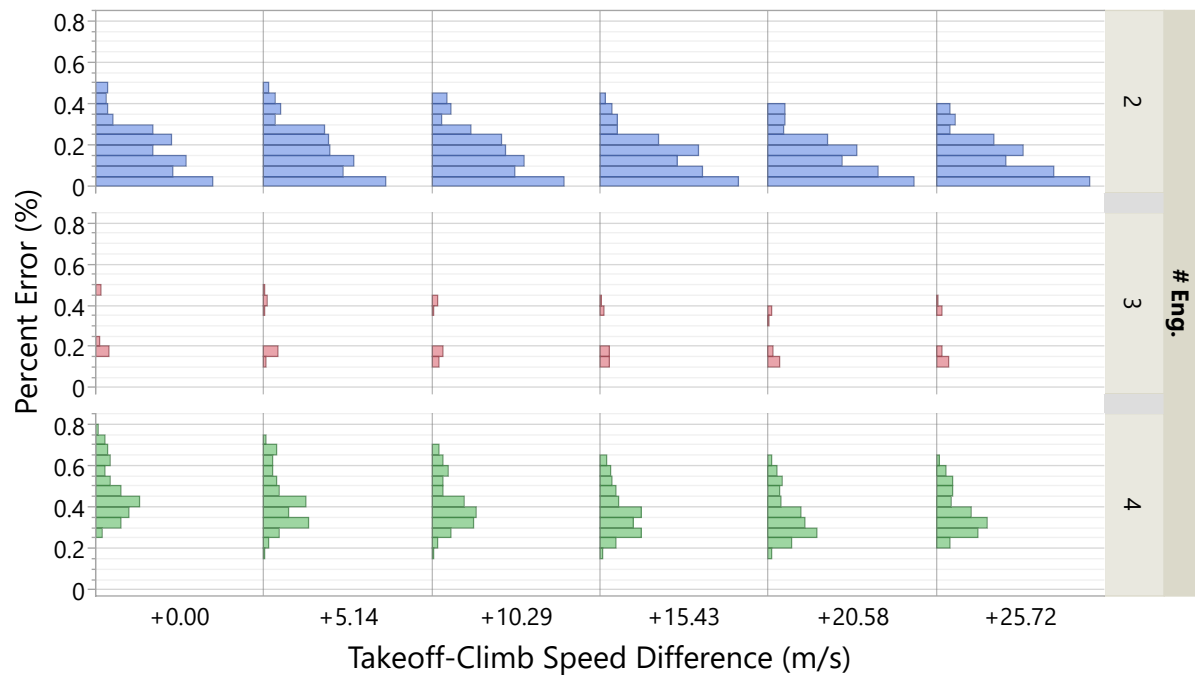
For the all the SEPLFs and CGLFs examined, the percent error is no larger than 1.30%. Across the operational envelope, the largest percent error is exhibited by 4-engine configurations at the minimum SEPLF and CGLF values recorded during the operational analysis. As ΔV increases, the percent error decreases, particularly for 4-engine aircraft, corroborating the previous observation that the approximation between the SEPLF and CGLF approaches an equality for larger ΔV . Therefore, for the given operational strategies reported – takeoff speeds between 66.88 and 102.89 m/s and climb speeds up to 25.72 m/s faster than the takeoff speed – the SEPLF is approximately equal to the CGLF.

To confirm the claim that the SEPLF and CGLF are approximately equal, the percent error was computed across *all* operating points in the mission profile during both climb segments. Figure 5.4 plots a cumulative distribution function of the percent error across these operating points, spanning 84,919 samples; additional statistics are presented in Tab. 5.4. A percent error less than or equal to 1% is demonstrated for over 95% of the data samples considered, with a maximum percent error of 1.53%. The vast majority of samples considered demonstrate less than 0.5% percent error, indicating a strong relationship between the SEPLF and CGLF across all takeoff and climb speeds evaluated. Both observations confirm that the SEPLF and CGLF are approximately equal within the operational envelope prescribed in Tab. 5.2, demonstrating the SEPLFs versatility to capture both propulsion system performance and climb performance degradations across a variety of operational strategies.

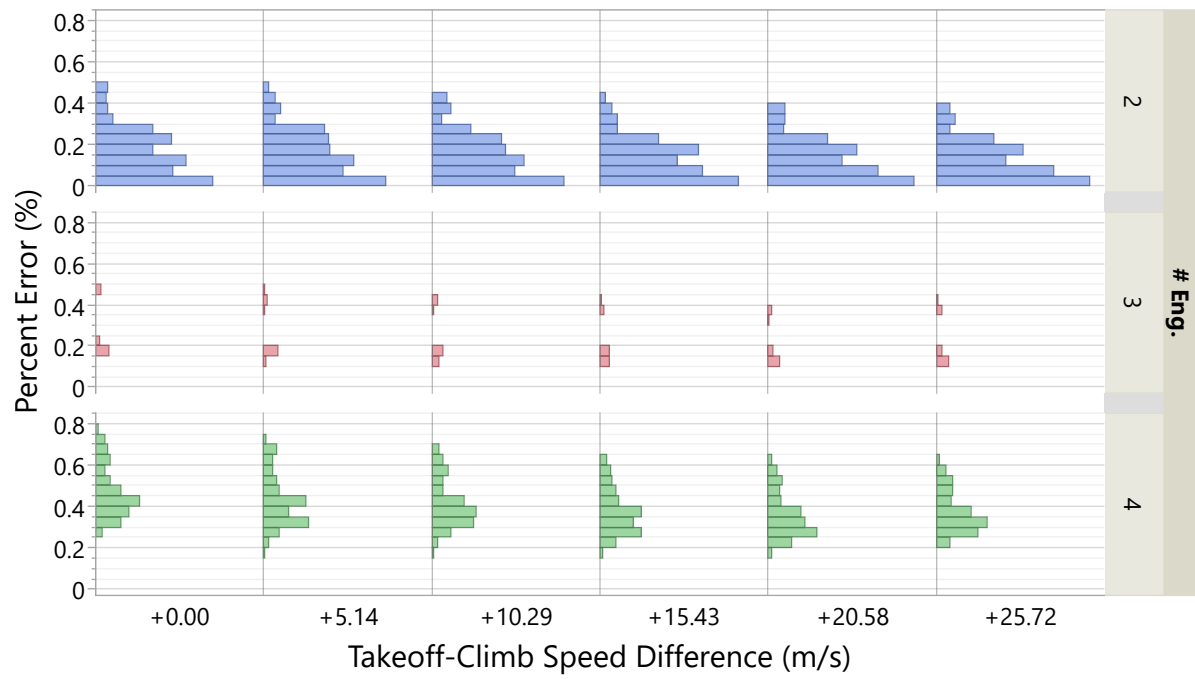
Therefore, the SEPLF and corresponding CGLF is weakly sensitive to the operational strategies included in the analysis, indicating that the SEPLF is approximately equal to the CGLF across the prescribed operational envelope in Tab. 5.2. Although approximate equivalence between the SEPLF and CGLF was demonstrated using empirical data from the aircraft operational analysis, particularly for the median SEPLF and CGLF, it can also be demonstrated mathematically, further justifying the observations made.



(a) Minimum SEPLF and CGLF.



(b) Median SEPLF and CGLF.



(c) Maximum SEPLF and CGLF.

Figure 5.3: Percent error for the predicted CGLF for a given SEPLF.

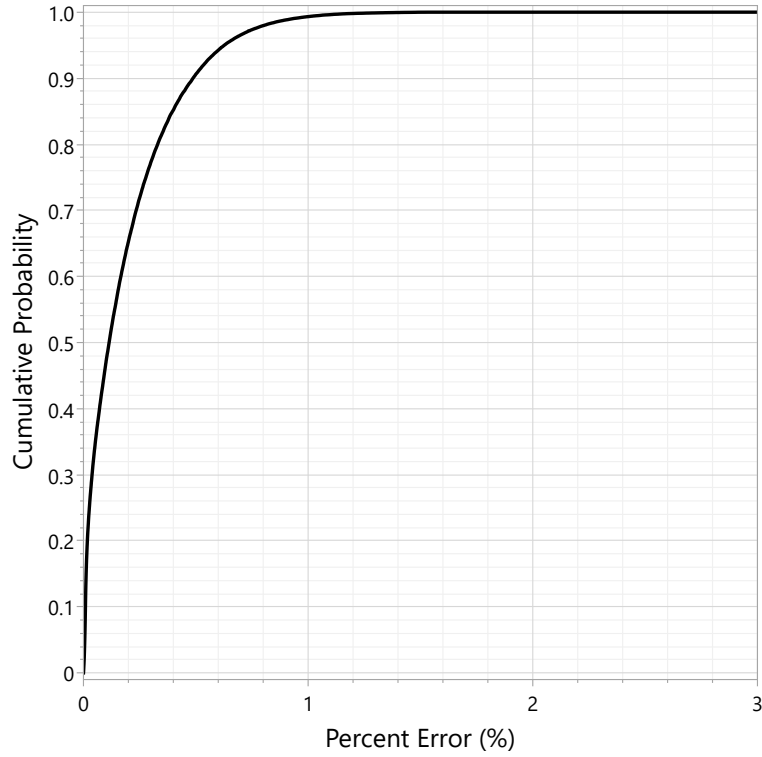


Figure 5.4: Cumulative distribution function of percent error between the SEPLF and CGLF for *all* operating points during climb.

Table 5.4: Sample statistics across *all* operating points during climb.

Statistic	Value (%)
Mean	0.1889
5 th Percentile	0.0058
10 th Percentile	0.0088
25 th Percentile	0.0262
Median (50 th Percentile)	0.1142
75 th Percentile	0.2780
90 th Percentile	0.4873
95 th Percentile	0.6271

5.3.1 Mathematical Justification of Approximate Equivalence between the SEPLF and CGLF

Before justifying the approximate equivalence of the SEPLF and CGLF, an alternative form of the specific excess power is defined in Eq. 5.8, highlighting that the excess power is used to either climb or accelerate [175].

$$P_s = \dot{h} + \frac{V_\infty \dot{V}_\infty}{g} \quad (5.8)$$

Using the alternative definition of the specific excess power defined in Eq. 5.8, normalize the expression by the airspeed, as shown in in Eq. 5.9. The terms on the right hand side of Eq. 5.9 represent the normalized fraction of specific excess power allocated to climbing and accelerating, and are labeled as the “normalized climb fraction” and “normalized acceleration fraction,” respectively. The normalized climb fraction matches the argument required predict the climb gradient in Eq. 5.3, relating it mathematically to the specific excess power. For convenience, Eq. 5.3 is restated in Eq. 5.10 to illustrate the equivalence between the normalized climb fraction and the argument used to compute the climb gradient.

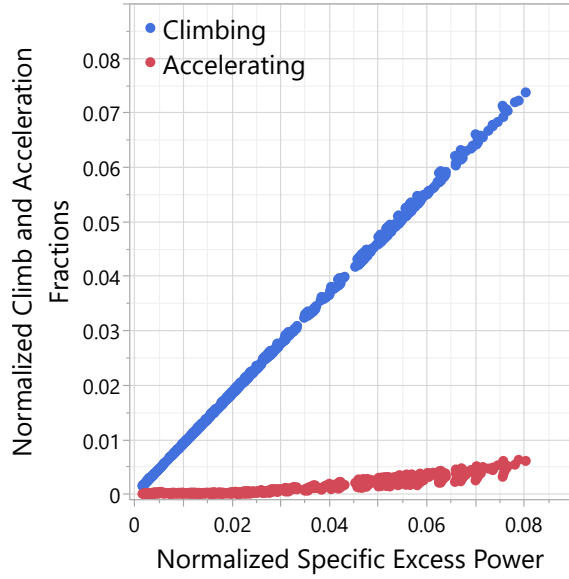
$$P_s/V_\infty = \dot{h}/V_\infty + \dot{V}_\infty/g \quad (5.9)$$

$$\gamma = \tan \left[\arcsin \left(\dot{h}/V_\infty \right) \right] \quad (5.10)$$

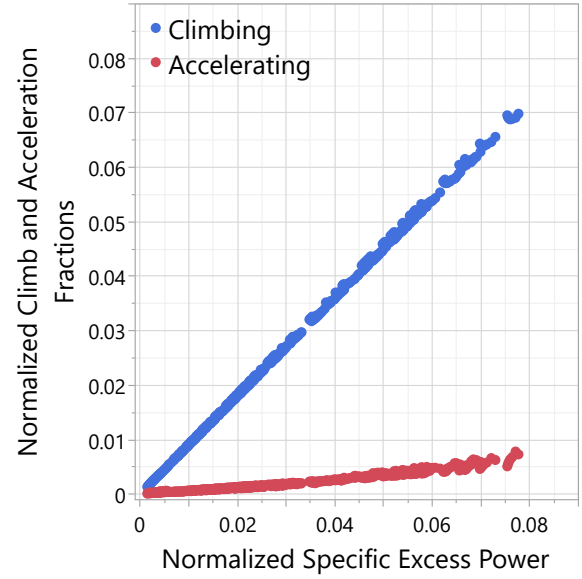
Using data from the fleet-level simulations performed during Experiment 3.0.1 in Section 5.2, the normalized climb and acceleration fractions are plotted in Fig. 5.5, with each subplot corresponding to a different ΔV . In some cases, the median SEPLF and CGLF computed corresponded to a flight condition during the second climb segment while the aircraft climbed between 914 and 3,048 m and accelerated to its diversion cruise speed of 128.6 m/s, explaining why some normalized acceleration fractions are nonzero in the case that $\Delta V = 0.00$ m/s.

As ΔV increases, the normalized climb fraction decreases while the normalized acceleration fraction increases, indicating that more specific excess power is utilized to accelerate. However, regardless of the ΔV examined, the normalized climb fraction is always larger than the normalized acceleration fraction. The relative impact of the both contributions to the normalized specific excess power is analyzed in Tab. 5.5, listing the normalized climb/acceleration fractions as well as the ratio between them for the smallest/largest normalized specific

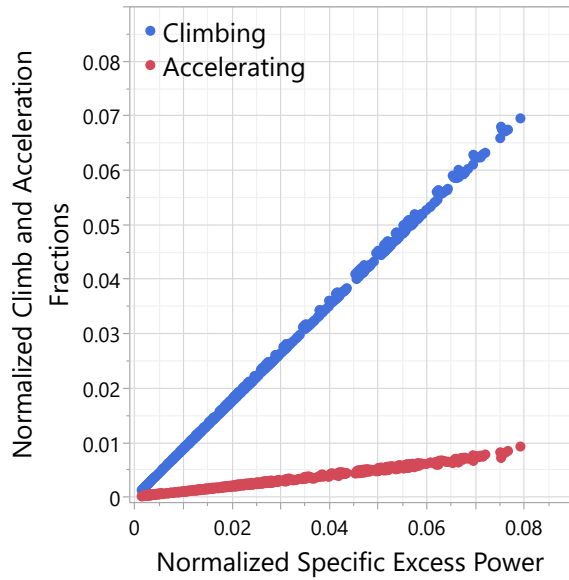
excess power values (representative of the varied operational strategies considered).



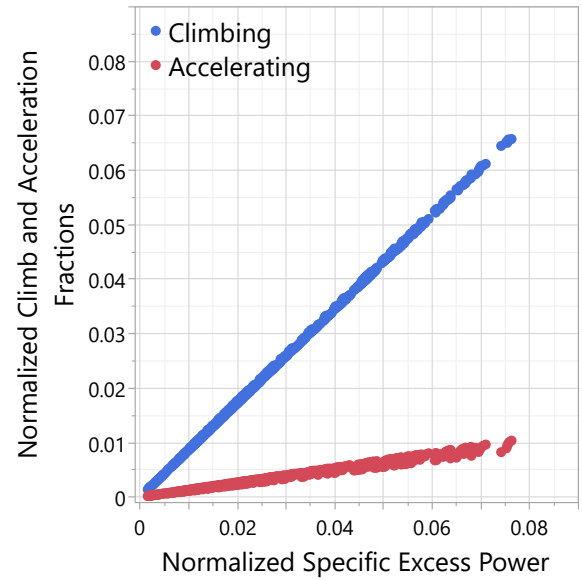
(a) $\Delta V = 0.00$ m/s.



(b) $\Delta V = 5.14$ m/s.



(c) $\Delta V = 10.29$ m/s.



(d) $\Delta V = 15.43$ m/s.

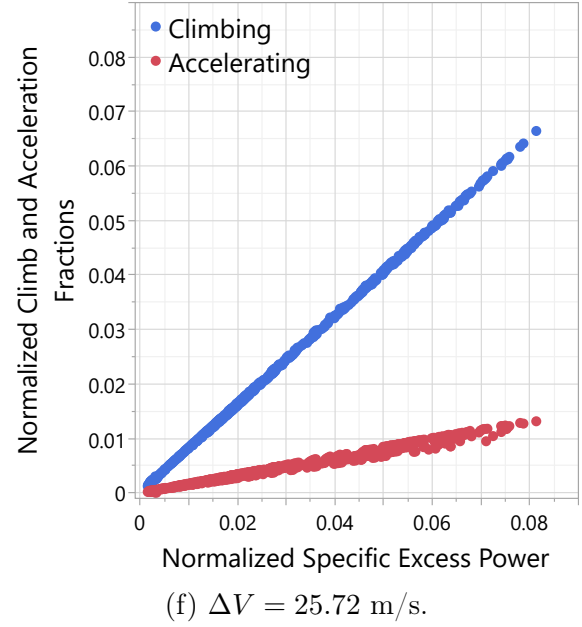
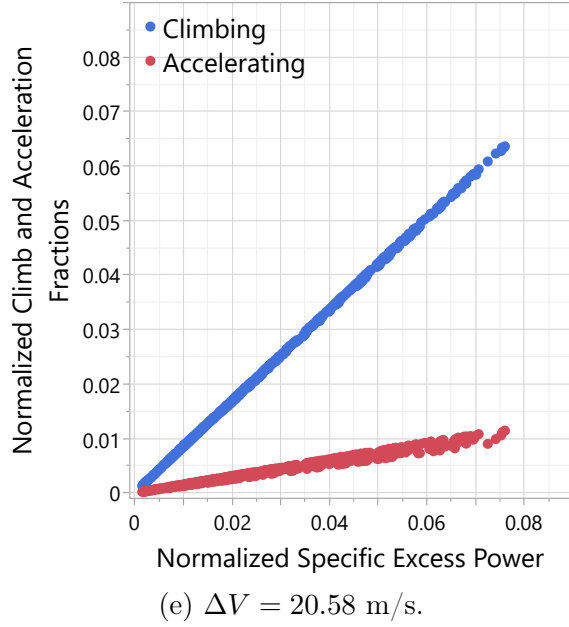


Figure 5.5: Normalized specific excess power components allocated for climbing (plotted as circles) and accelerating (plotted as crosses) for different takeoff and climb speeds. The takeoff speed is represented by the point's color.

Table 5.5: Relative climb/acceleration fraction contributions.

ΔV (m/s)	Norm. Climb Fraction		Norm. Acceleration Fraction		Climb-Acceleration Fraction Ratio	
	Min.	Max.	Min.	Max.	Min.	Max.
0.00	0.0016	0.0738	0.0001	0.0061	16.00	12.10
5.14	0.0014	0.0698	0.0001	0.0073	14.00	9.56
10.29	0.0013	0.0695	0.0002	0.0093	6.50	7.47
15.43	0.0014	0.0657	0.0002	0.0104	7.00	6.32
20.58	0.0013	0.0636	0.0002	0.0114	6.50	5.58
25.72	0.0012	0.0664	0.0002	0.0131	6.00	5.07

At the largest ΔV values assessed in the operational envelope, the normalized climb fraction always remains at least 5- to 6-times larger than the acceleration fraction, and increases as ΔV decreases. The observations suggest that, across all operational strategies, the vast majority of specific excess power is utilized for climbing. Therefore, it is assumed that the normalized acceleration fraction is negligible relative to the normalized climb fraction,

implying that the normalized specific excess power is approximately equal to the normalized climb fraction, as shown in Eq. 5.11.

$$P_s/V_\infty \approx \dot{h}/V_\infty \quad (5.11)$$

Substituting this approximation into Eq. 5.3 reveals an approximate relationship between the specific excess power and climb gradient in Eq. 5.12:

$$\gamma \approx \tan \left[\arcsin (P_s/V_\infty) \right] \quad (5.12)$$

The normalized specific excess power recorded in Fig. 5.9 is no larger than 0.08, remaining close to zero. Using a small angle approximation, Eq. 5.12 is linearized about zero to simplify the relationship between the climb gradient and specific excess power, resulting in the linear relationship described in Eq. 5.13.

$$\gamma \approx P_s/V_\infty \quad (5.13)$$

As a result of the small angle approximation, the maximum achievable climb gradient is linearly proportional to the specific excess power. More broadly, an approximate direct relationship between the propulsion system performance and overall flight performance has been established.

To connect the approximation to the propulsion system failure severity and flight performance degradation under off-nominal conditions, the CGLF defined in Eq. 5.6 is rewritten utilizing the approximation established in Eq. 5.13.

$$\begin{aligned} \gamma_{\text{loss}} &\approx 1 - \frac{P_{s_{\text{OEI}}}/V_\infty}{P_{s_{\text{AEO}}}/V_\infty} \\ &\approx 1 - \frac{P_{s_{\text{OEI}}}}{P_{s_{\text{AEO}}}} \\ &\approx P_{s_{\text{loss}}} \end{aligned} \quad (5.14)$$

The right-hand side of Eq. 5.14 represents SEPLF, demonstrating nominal equivalence between the SEPLF and CGLF, further supporting the results found from Section 5.2. Thus, the aircraft's degraded climb performance can be nominally predicted by the propulsion system failure severity within the operational envelope proposed in Tab. 5.2.

5.4 Generalized Performance Requirement Construction (Experiment 3.0.2)

Generalized flight performance requirements are constructed using the SEPLF parameterization identified in Experiment 3.0.1 from Section 5.2. First, a mapping between the SEPLF to the existing FAA climb performance requirements for 2-, 3-, and 4-engine aircraft is generated in Section 5.4.1. Then, the generalized requirements are constructed in Section 5.4.2 by fitting a curve to the previous mapping generated. Only 80% of the available data serves as training data to generate the mapping. The remaining 20% of the data serves as validation data to confirm that the generalized climb performance requirement constructions return the published FAA flight performance requirements for conventional aircraft, as performed in Section 5.4.3.

Since the SEPLF exhibited weak sensitivity with respect to the operational strategies simulated, all analyses utilize the SEPLF corresponding to a ΔV of 25.72 m/s, and was chosen because the percent error between the SEPLF and CGLF was the lowest among all ΔV values considered in Section 5.2. Given the strong correlation between the SEPLF and CGLF, it is expected that repeating the climb performance requiring constructions under a different ΔV would minimally impact the results.

5.4.1 Fleet-Level SEPLF Trends

The SEPLF distribution at a ΔV of 25.72 m/s is plotted in Fig. 5.6 for all training data utilized (80% of the available data in the dataset). Additional SEPLF plots for the other ΔV are provided in Appendix C.3.

All SEPLFs for 2-engine aircraft are between 80 and 100%, with the majority between 85 and 95%. Due to the limited data available for 3-engine aircraft, the SEPLFs are scattered between 65 and 80%, though the majority trend towards the upper bound of the SEPLFs reported. The SEPLFs for 4-engine aircraft vary between 40 and 70%, with the majority between 55 and 65%.

The data also reveals that the SEPLF decreases as more engines are installed on the airframe, and is a result of both propulsive, aerodynamic, and stability/control considerations. First, configurations with more engines installed have a larger fraction of their thrust power remaining after an engine failure. For a 2-engine aircraft, 50% of the thrust power remains while 66% and 75% of the thrust power remains for 3- and 4-engine configurations, respectively. Second, configurations with fewer engines have larger fan diameters, resulting in

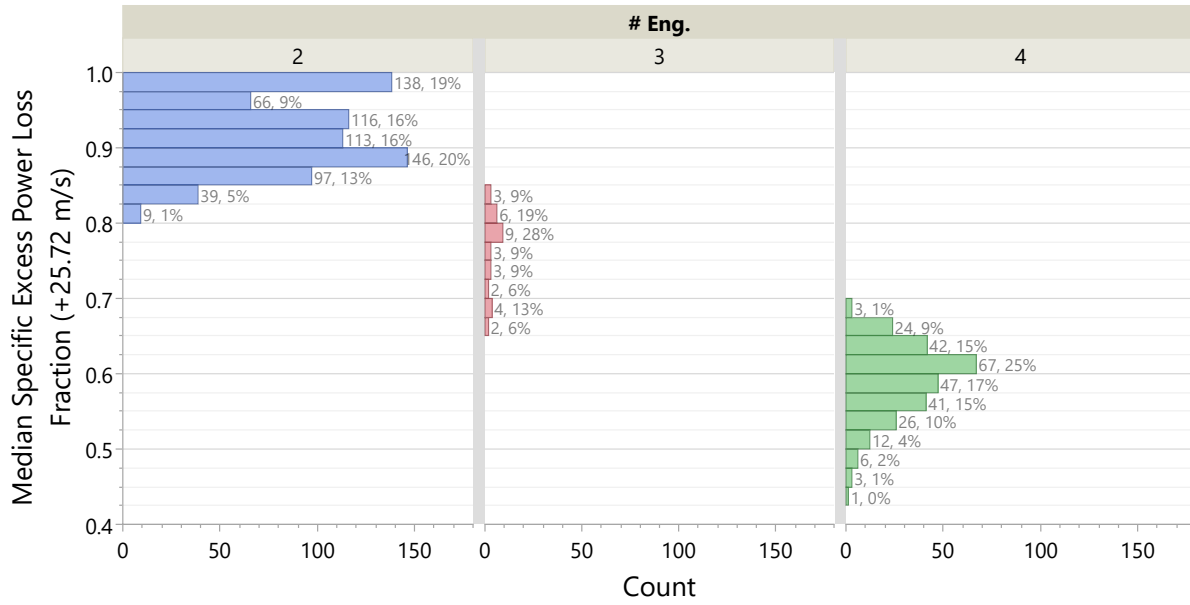


Figure 5.6: SEP loss fraction distribution, $\Delta V = 25.72$ m/s.

more windmilling drag when the engine fails. The yaw moment induced by the increased windmilling drag requires a larger rudder deflection to maintain directional stability, further increasing the trim drag and, hence, the SEPLF.

Percentiles from each distribution, provided in Tab. 5.6 for $\Delta V = 25.72$ m/s, highlight the range of SEPLFs exhibited under engine inoperative conditions for aircraft previously certified under FAA Part 25 regulations. Additional tables listing the distributional anchors for the remaining ΔV values are available in Appendix C.3.

Table 5.6: Selected quantities across the SEPLF distributions for $\Delta V = 25.72$ m/s.

Percentile	Engines Installed		
	2	3	4
10	0.8534	0.6798	0.5317
25	0.8779	0.7384	0.5638
50	0.9155	0.7845	0.5991
75	0.9571	0.8004	0.6271
90	0.9824	0.8272	0.6522

Using the data from Tab. 5.6, the median (50th percentile) SEPLF for 2-, 3-, and 4-engine aircraft are mapped to their corresponding FAA climb performance requirements, as listed

in Tab. 5.7. The mapping re-parameterizes the FAA climb performance requirements by the SEPLF (an analogue for propulsion system failure severity) rather than the number of engines installed on the aircraft, serving as a basis for constructing the generalized climb performance requirements.

Table 5.7: Median SEPLFs from the operational analysis mapped to the corresponding FAA Part 25 regulations.

Requirement	Median SEPLF		
	91.55%	78.45%	59.91%
25.111 (Takeoff Flight Path, OEI)	$\gamma \geq 1.2\%$	$\gamma \geq 1.5\%$	$\gamma \geq 1.7\%$
25.119 (Landing, AEO)	$\gamma \geq 3.2\%$	$\gamma \geq 3.2\%$	$\gamma \geq 3.2\%$
25.121a (First Segment Climb, OEI)	$\gamma \geq 0.0\%$	$\gamma \geq 0.3\%$	$\gamma \geq 0.5\%$
25.121b (Second Segment Climb, OEI)	$\gamma \geq 2.4\%$	$\gamma \geq 2.7\%$	$\gamma \geq 3.0\%$
25.121c (Final Climb, OEI)	$\gamma \geq 1.2\%$	$\gamma \geq 1.5\%$	$\gamma \geq 1.7\%$
25.121d (Landing, OEI)	$\gamma \geq 2.1\%$	$\gamma \geq 2.4\%$	$\gamma \geq 2.7\%$

5.4.2 Constructing Generalized Engine Inoperative Performance Requirements

Using the mapping outlined in Tab. 5.7, a Sigmoid Curve was fit to the training data for each FAA Part 25 climb requirement. A Sigmoid Curve is a common equation for predicting technological performance; its general equation is defined in Eq. 5.15. The Sigmoid Curve’s ability to exhibit asymptotic behavior is also advantageous, ensuring that the climb requirements remain anchored to those published by the FAA, minimally deviating from them for configurations with extremely large/small specific excess power losses during an engine inoperative scenario. Such behavior is also critical for guaranteeing that the climb performance requirements for future aircraft concepts are similar to those for conventional aircraft, particularly towards SEPLFs of 0 to 40% (reflective of regimes for distributed propulsion system failures).

$$f(x) = \frac{a}{1 + \exp[-b(x - c)]} + d \quad (5.15)$$

The generalized Sigmoid Curve from Eq. 5.15 is then re-parameterized by the SEPLF and outputs a nominal required climb gradient, as shown in Eq. 5.16. The coefficients a and d affect the curve’s upper and lower asymptotes while b and c affect the curve’s steepness

and inflection point.

$$\hat{\gamma}(P_{s_{\text{loss}}}) = \frac{a}{1 + \exp[-b(P_{s_{\text{loss}}} - c)]} + d \quad (5.16)$$

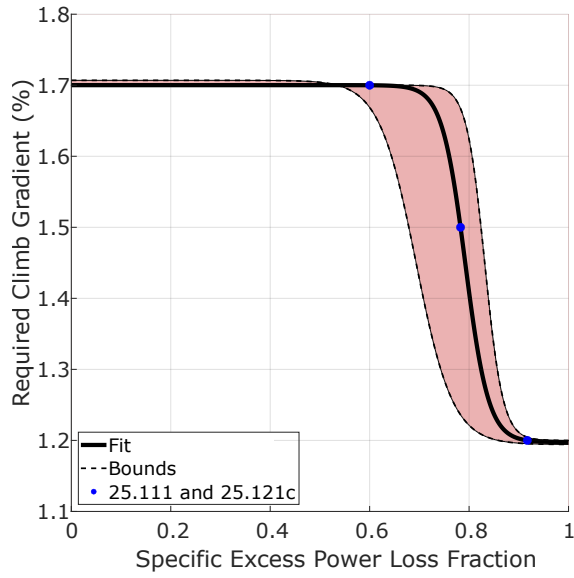
The Sigmoid Curve was fit to the data by solving the optimization problem in Eq. 5.17; the root mean square error (RMSE) is minimized with respect to the mapped FAA Part 25 climb performance requirements listed in Tab. 5.7 by varying the four coefficients governing the Sigmoid Curve shape. To ensure that the Sigmoid Curves exhibit asymptotic behavior for the extreme SEPLF values, the beginning and ending slopes ($\hat{\gamma}'(0)$ and $\hat{\gamma}'(1)$, respectively) were constrained to be within $\pm 10^{-2}$.

$$\begin{aligned} \text{Minimize: } \text{RMSE} &= \sqrt{\frac{\sum_{i=1}^n (\gamma_i - \hat{\gamma}_i)^2}{n}} \\ \text{With Respect to: } &\text{Sigmoid Curve Parameters } a, b, c, \text{ and } d \\ \text{Subject to: } &-10^{-2} \leq \hat{\gamma}'(0) \leq +10^{-2} \\ &-10^{-2} \leq \hat{\gamma}'(1) \leq +10^{-2} \end{aligned} \quad (5.17)$$

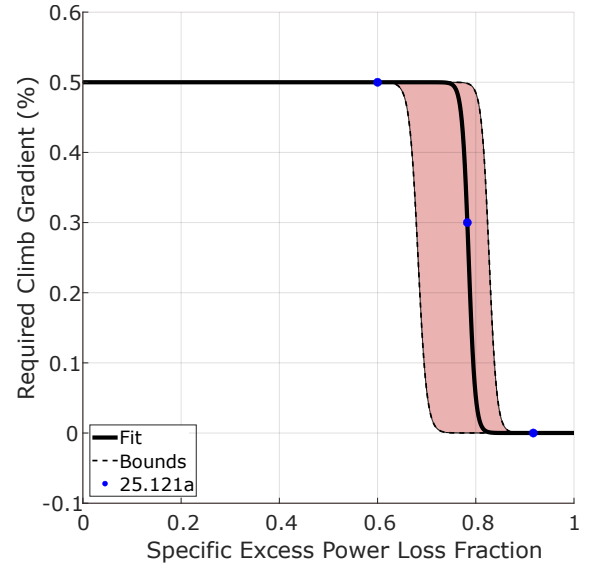
The climb performance requirement under FAA Part 25.121a necessitates the climb gradient to be greater than 0% for 2-engine aircraft. To ensure that a nonnegative climb gradient is prescribed across all SEPLFs, an additional constraint was imposed, requiring that $\hat{\gamma}(1) \geq 0$.

The optimized Sigmoid Curves are plotted in Fig. 5.7; their coefficients are corresponding RMSE values are listed in Tab. 5.8. Uncertainty bounds are also plotted by constructing Sigmoid Curves using data from the 10th and 90th SEPLF percentiles listed in Tab. 5.6, as indicated by the red shaded region. No Sigmoid Curve was constructed for FAA Part 25.119 requirements because the minimum climb gradient for all configurations is 3.2% for all 2-, 3-, and 4-engine aircraft already, enabling its universal application to conventional and future aircraft concepts already.

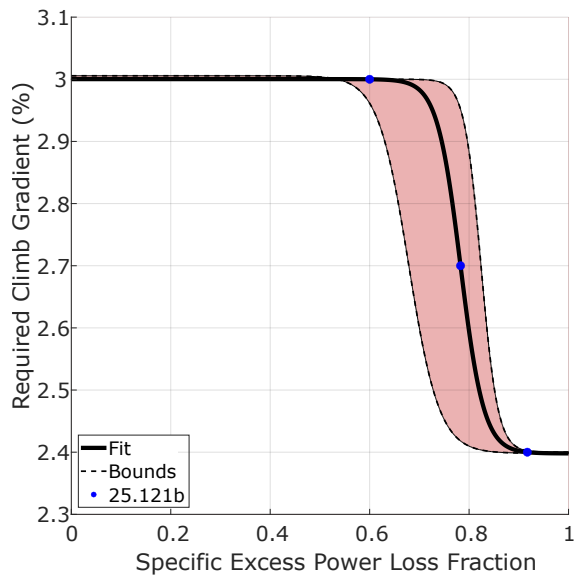
The Sigmoid Curves illustrated in Fig. 5.7 capture the FAA Part 25 regulations while spanning all propulsion system failure severities. The constrained derivatives at the endpoints prevent the climb gradients from becoming either too ambitious or conservative relative to the existing requirements, guaranteeing that conventional and future aircraft concepts experiencing propulsion system failures with similar SEPLFs will be sized for similar climb performance requirements. In each fit, the uncertainty region is larger under the Sigmoid Curve than it is above, suggesting that the SEPLF distributions are skewed towards larger values, as observed in Fig. 5.6, particularly for 3- and 4-engine aircraft.



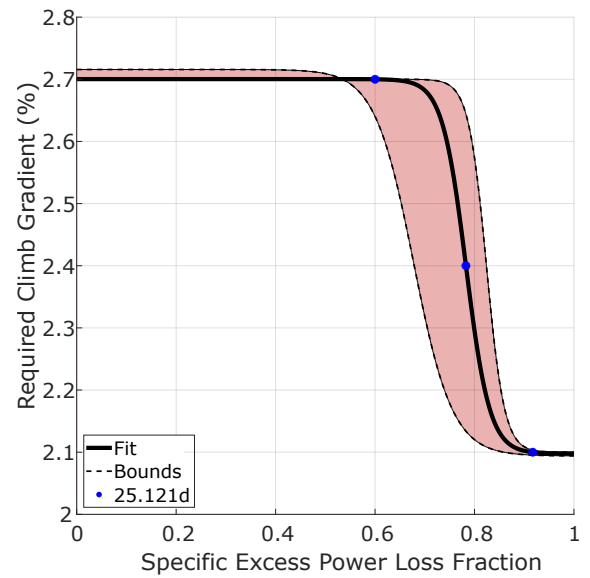
(a) Part 25.111 and 25.121c.



(b) Part 25.121a.



(c) Part 25.121b.



(d) Part 25.121d.

Figure 5.7: Sigmoid Curves constructed to represent FAA Part 25 climb performance requirements.

Although the curves smoothly transition between the mapping generated, the curve modeling FAA Part 25.121a in Fig. 5.7b is steeper due to the additional constraint requiring

Table 5.8: Optimized Sigmoid Curve parameters for all FAA Part 25 performance requirements.

FAA Regulation	RMSE	a	b	c	d
25.111 and 25.121c	1.95×10^{-8}	0.5035	-41.39	0.7945	1.1967
25.121a	4.00×10^{-5}	0.4999	-157.22	0.7871	0.0001
25.121b	1.50×10^{-8}	0.6024	-42.91	0.7847	2.3978
25.121d	1.69×10^{-8}	0.6025	-41.69	0.7847	2.0977

$\hat{\gamma}(1) \geq 0$, resulting in a stiffer optimization problem and an RMSE that is three-orders of magnitude larger than the other curve fits. Despite the poorer RMSE, the curve fit still reasonably and accurately models the climb performance requirements.

All Sigmoid Curves are constructed with asymptotic regions spanning SEPLFs between 0 and 59.91% (upper asymptote) and 91.55 and 100% (lower asymptote), aligning with the climb performance requirements for 4- and 2-engine aircraft, respectively. Such an artifact of the Sigmoid Curve construction reveals that, with the constraints on the slopes at the endpoints, the 4- and 2-engine climb performance requirements may be utilized for a majority of the aircraft designs considered. In other words, the Sigmoid Curve constructions almost act as step-functions, quickly transitioning between the required climb gradients for 2- and 4-engine aircraft. More broadly, the generalized requirement constructions suggest that each climb requirement could be defined utilizing the upper asymptote (corresponding to a 4-engine climb performance requirement) for SEPLFs between 0 and 78.45% and the lower asymptote (corresponding to a 2-engine climb performance requirement) for SEPLFs between 78.45 and 100%. Future work should explore the ramifications of this consideration and how the size of 3-engine aircraft would change if only the upper/lower asymptotes were utilized to define the performance requirements.

The climb gradients returned by the Sigmoid Curve constructions are also consistent with prior assumptions made in the literature while designing advanced aircraft. Previously, the FAA’s 4-engine climb performance requirements were utilized to design aircraft with distributed propulsion systems [9, 97, 100]. Assuming that a distributed propulsion system failure involves only one or two propulsors, the SEPLF is likely to be less than 60%, aligning with the upper asymptote on the Sigmoid Curves and corresponding to the FAA’s 4-engine climb requirements. Since the generalized climb performance requirements are bounded by the FAA certification regulations, they provide a consistency-based heuristic to predict the

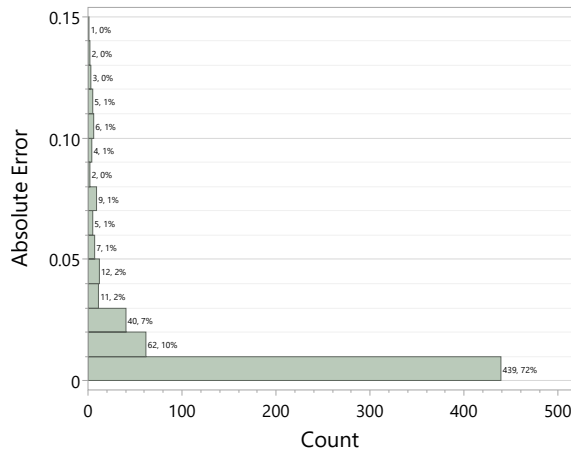
climb performance requirements for aircraft with advanced propulsion system architectures under failure modes that do not map to a discrete engine count.

5.4.3 Generalized and FAA Climb Performance Requirement Comparison

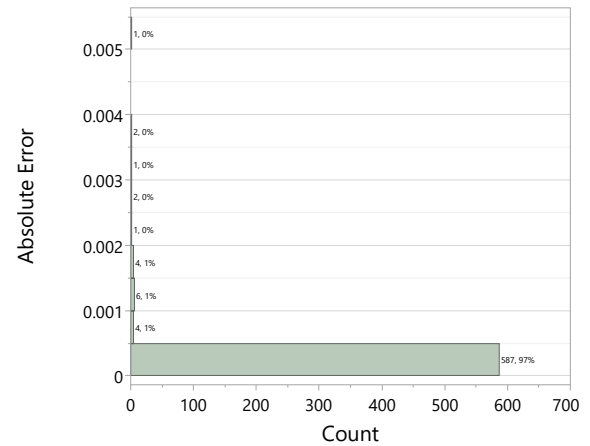
After constructing the generalized performance requirements, a test was performed to demonstrate nominal equivalence between them and those presently published by the FAA. In the test, the SEPLFs from the validation data were used to estimate the required climb gradients using the constructions from Section 5.4.2. Then, the absolute difference between the generalized and FAA climb performance requirements was computed, as described in Eq. 5.18. Ideally, the required climb gradients minimally deviate from each other, maintaining consistency for previously certified aircraft.

$$\text{Error} = |\hat{\gamma} - \gamma_{\text{FAA}}| \quad (5.18)$$

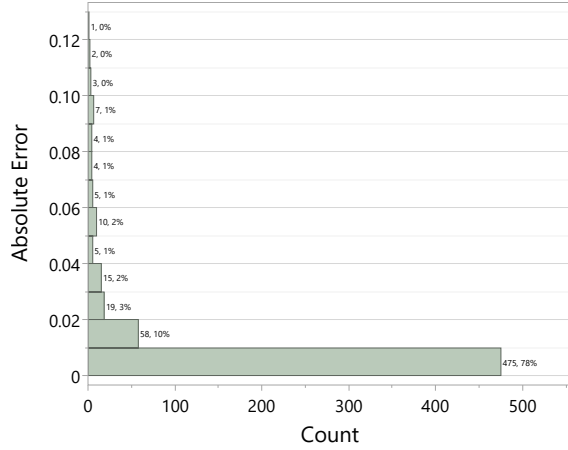
Figure 5.8 plots histograms of the absolute error for all 2-engine aircraft examined in the operational analysis. Additional histograms for 3- and 4-engine aircraft are provided in Figures C.2 and C.3 within Appendix C.4.



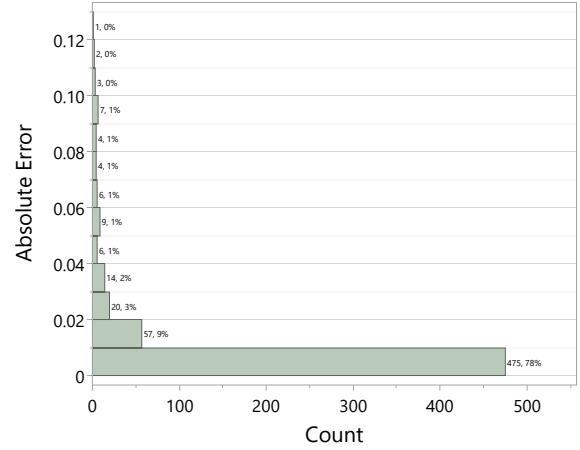
(a) 25.111 and 25.121c.



(b) 25.121a.



(c) 25.121b.



(d) 25.121d.

Figure 5.8: Absolute error between FAA and proposed flight performance requirements for 2-engine aircraft.

At a minimum, 72% of the climb gradients are altered by no more than 0.01% for 2-engine aircraft, indicating reasonable consistency between the generalized and existing FAA requirements. Such consistency is maintained because the required climb gradients slightly increase for SEPLFs smaller than the median and marginally decrease for SEPLFs larger than the median. Similar behavior is observed in Fig. C.3 (see Appendix C.4) for 4-engine aircraft; the largest climb gradient alterations are approximately 0.002%, 5-times smaller than the largest climb gradient alteration for 2-engine aircraft. Such behavior occurs because the median SEPLF for 4-engine aircraft lies on the upper asymptote of each Sigmoid Curve. In contrast, the median SEPLF for 2-engine aircraft near the lower asymptote of the Sigmoid Curve, remaining closer to the regions of significant curvature change, explaining the larger absolute error relative to those for 4-engine aircraft.

However, for 3-engine aircraft, the climb gradient alterations are as large as 0.3%, as illustrated in Fig. C.2 (see Appendix C.4), indicating possible inconsistencies between the generalized and FAA climb performance requirements. The discrepancy is a result of the median SEPLF for 3-engine aircraft residing in the regions of the Sigmoid Curve with the steepest slopes, resulting in significant climb performance requirement differences for small SEPLF perturbations. As a consequence, existing 3-engine aircraft may be over- or under-designed for the generalized climb performance requirements. Despite the error observed, the results are less concerning because all 3-engine examined in this study entered into service

between 1963 and 1975, and are no longer frequently manufactured or operated.

In summary, the results of the test performed suggests that the proposed performance requirements are nominally equivalent to those already defined in FAA Part 25 regulations for 2- and 4-engine aircraft, and can be useful for designing future aircraft with both conventional and advanced propulsion systems.

5.4.4 Conclusions

Generalized climb performance requirements are needed to define off-nominal flight performance requirements in a configuration-agnostic manner, supporting advanced aircraft design, as proposed by Research Question 3.0. Hypothesis 3.0 supposed that a monotonic, bounded mapping between the specific excess power loss and the required climb gradient will reproduce the climb gradient requirements implied by the published regulations for conventional aircraft utilizing discrete, homogeneously sized propulsors. Previously, Experiment 3.0.1 indicated that the SEPLF is a suitable parameter for constructing the generalized flight performance requirements. Then, in Experiment 3.0.2, the generalized flight performance requirements were constructed using the aforementioned parameterization.

Prior to constructing the performance requirements, the data utilized in Experiment 3.0.1 was divided into training/validation data at an 80/20% split. The training data was mapped to the FAA climb performance requirements for 2-, 3-, and 4-engine conventional aircraft, defining a one-to-one relationship between the two quantities and re-parameterizing the climb performance requirements as a function of the SEPLF. Sigmoid Curves were then fit to the data mapping by minimizing the RMSE subject to constraints on the slopes at the endpoints of the curve, defining the climb performance requirements as a function of the SEPLF.

Overall, the Sigmoid Curves well fit the data, resulting in optimized RMSE values on the order of 10^{-8} for each requirement. However, the fit for the FAA Part 25.121a climb performance requirement exhibited an RMSE on the order of 10^{-5} because an additional constraint was required to ensure that the required climb gradient remained nonnegative, stiffening the optimization problem. In addition to the Sigmoid Curves generated, Sigmoid Curves using the 10th and 90th percentile SEPLFs were also constructed, illustrating possible bounds on the climb performance requirements for a given SEPLF. The majority of the SEPLFs reside within the Sigmoid Curves' upper/lower asymptotes, highlighting a minimal change in the climb requirement if a different SEPLF were to be used, particularly for SEPLFs between 0-59.91% and 91.55-100%.

After constructing the Sigmoid Curves, a back-test was performed to ensure that the

generalized climb performance requirements were nominally equivalent to those presently defined by the FAA. For all 2- and 4-engine aircraft examined in the fleet-level operational analysis, the absolute error remained less than 0.15% and 0.015%, respectively, indicating consistency between the two sets of requirements. For 3-engine aircraft, however, the absolute error may be as large as 0.3%, underscoring some inconsistency between the generalized and FAA climb performance requirements; this is a result of the median SEPLF for 3-engine aircraft residing in the region of the Sigmoid Curve with the largest slopes and curvature changes. However, climb performance requirements for 3-engine aircraft are less concerning due to their waning presence in the global airline fleet.

Overall, Experiment 3.0.2 successfully demonstrated that generalized climb performance requirements can be constructed utilizing a monotonic, bounded mapping as a function of the specific excess power loss during off-nominal operational scenarios. The generalized climb performance requirements constructed will be used later in Experiment 3.1.1 to design an advanced aircraft concept with a distributed propulsion system, enabling climb performance requirements to be defined for different primary propulsion system failure modes.

5.5 Generalized Engine Inoperative Correction Factor Construction (Experiment 3.0.3)

While Experiment 3.0.2 constructed generalized climb performance requirements for advanced aircraft concepts, Experiment 3.0.3 repeats a similar process for constructing a generalized “engine inoperative correction factor”, defined as $\hat{\kappa}$. Together, both constructions can be seamlessly integrated into an aircraft sizing code, enabling advanced aircraft concepts to be sized.

5.5.1 Propulsion System Failure Mapping to a Correction Factor

Prior to constructing an engine inoperative correction factor, a mapping must be generated between the SEPLF and traditional $n/(n - 1)$ engine inoperative correction factors used in current aircraft design literature [103]. The median SEPLFs identified from Experiment 3.0.2 are again utilized to map the propulsion system failure severity to existing engine inoperative correction factors previously listed in Tab. 1.3, representing the correction factor of $n/(n - 1)$ traditionally used, with n representing the number of engines installed on the aircraft. The mapping generated is similar to that of Experiment 3.0.2, parameterizing the

engine inoperative correction factor based on the SEPLF, as defined in Tab. 5.9. The data indicates that the engine inoperative correction factor increases as the propulsion system failure severity increases. Intuitively, the observation makes physical sense; if more propulsors fail, the other ones must be sized larger to generate the necessary power to divert to an airport and land safely.

Table 5.9: Engine inoperative correction factors as a function of the SEPLF.

Median SEPLF	Correction Factor
91.55%	2.00
78.45%	1.50
59.91%	1.33

5.5.2 Constructing an Engine Inoperative Correction Factor

Utilizing the mapping generated, a curve can be fit to the data, again minimizing the RMSE. In Experiment 3.0.2, constraints were applied to ensure that the climb performance requirements remained anchored within those presently published by the FAA. For constructing the engine inoperative correction factor in Experiment 3.0.3, three constraints are needed to ensure physical consistency between the mapping and the curve fit:

1. Return a value of 1 for a SEPLF of 0, suggesting that the propulsion system need not be enlarged if no specific excess power is lost;
2. Be monotonically increasing, implying that the propulsion system must become larger as more specific excess power is lost; and
3. Correction factors should not be excessively large for configurations with SEPLFs that approach 1, preventing existing twin-engine aircraft from being over-sized relative to their current design points.

The first two constraints are enforced mathematically by requiring $\hat{\kappa}(0) \geq 1$ and $d\hat{\kappa}/dP_{s_{\text{loss}}} \geq 0$ for all SEPLFs. The third constraint must be enforced heuristically by defining an upper bound on the correction factor's value, and requires experimentation to identify a reasonable upper bound.

Equation 5.19 defines an optimization problem to minimize each curve fit's RMSE subject to the three constraints previously described. To enforce that the curve monotonically

increases, a nonnegative first derivative is checked at 101 linearly spaced points in the domain between SEPLFs of 0 and 1.

$$\begin{aligned}
&\text{Minimize: } \text{RMSE} = \sqrt{\frac{\sum_{i=1}^n [(\kappa_i - \hat{\kappa}_i)^2]}{n}} \\
&\text{With Respect to: } \text{Curve Parameters } \mathbf{c} \\
&\text{Subject to: } \hat{\kappa}(0) = 1 \\
&\quad d\hat{\kappa}/dP_{s_{\text{loss}}} \geq 0 \\
&\quad \hat{\kappa}(1) \leq \hat{\kappa}_{\text{max}}
\end{aligned} \tag{5.19}$$

5.5.3 Experimental Setup

In Experiment 3.0.3, multiple parametric curves and maximum allowable correction factors are considered for constructing the engine inoperative correction factor. The parametric curves considered and their general equations are listed in Tab. 5.10. All coefficients, c_i , are optimized while minimizing the RMSE in Eq. 5.19.

Table 5.10: Curves fits tested while constructing the engine inoperative correction factor.

Curve	General Equation
Linear	$c_1x + c_2$
Quadratic	$c_1x^2 + c_2x + c_3$
Cubic	$c_1x^3 + c_2x^2 + c_3x + c_4$
Exponential	$c_1 \exp \left[\frac{c_2 (x - c_3)}{c_4} \right] + c_4$
Tangent	$c_1 \tan \left[\frac{c_2 (x - c_3)}{c_4} \right] + c_4$
Hyperbolic Sine	$c_1 \sinh \left[\frac{c_2 (x - c_3)}{c_4} \right] + c_4$

Two maximum allowable engine inoperative correction factors are considered: 1) 2.25, a value 12.5% larger than the largest traditional engine inoperative correction factor; and 2) 2.05, a value reflective of making minimal changes to the current range of engine inoperative correction factors available.

5.5.4 Results: Upper Maximum Allowable Correction Factor

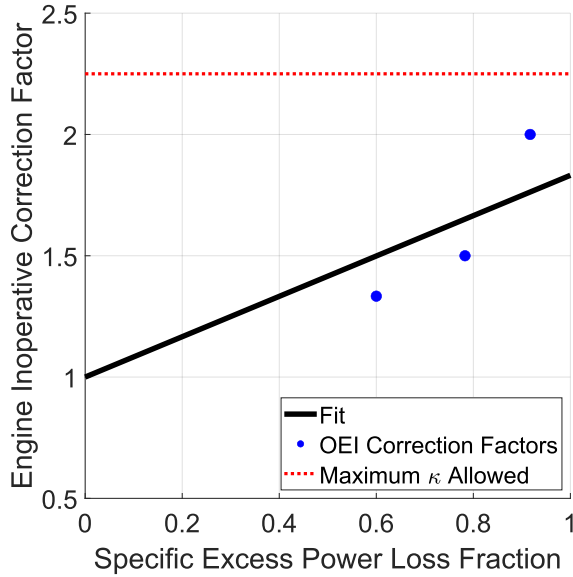
The optimized RMSEs for each parametric curve under a maximum allowable engine inoperative correction factor of 2.25 are listed in Tab. 5.11. All candidate curves return an optimized RMSE on the order of 10^{-1} or 10^{-2} , suggesting that there are many suitable candidates for

defining the engine inoperative correction factor. All optimized curves are plotted in Fig. 5.9 to further examine each engine inoperative correction factor construction.

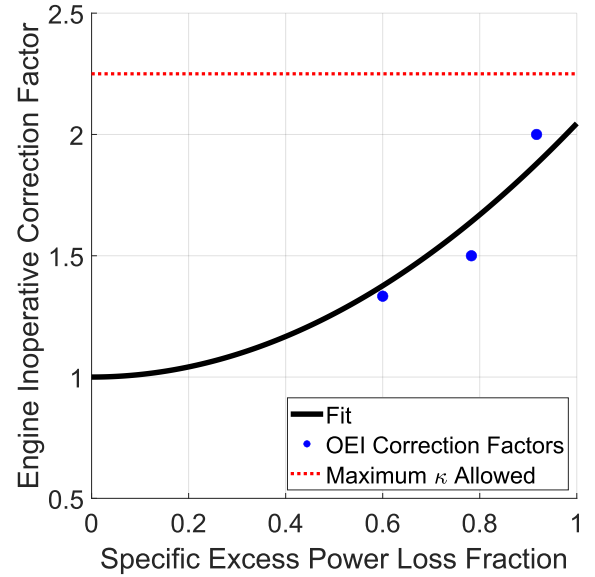
Table 5.11: Engine inoperative curve constructions, $\hat{\kappa}_{\max} = 2.25$.

Curve	RMSE
Linear	1.885×10^{-1}
Quadratic	1.098×10^{-1}
Cubic	5.735×10^{-2}
Exponential	6.357×10^{-2}
Tangent	7.045×10^{-2}
Hyperbolic Sine	6.257×10^{-2}

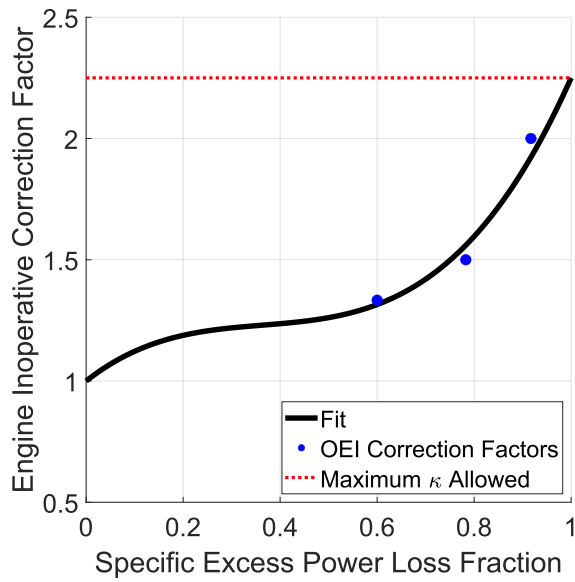
First, the cubic, exponential, tangent, and hyperbolic sine curves are considered. For SEPLFs greater than 70%, each curve's slope is significantly larger than those of the linear and quadratic ones, suggesting that the propulsion system size increases rapidly for increasingly severe propulsion system failures. Given that tube-and-wing aircraft designs with turbofan engines are technologically mature, the propulsion system size should not significantly grow for a slight increase in the propulsion system failure severity, rendering these four curves less desirable.



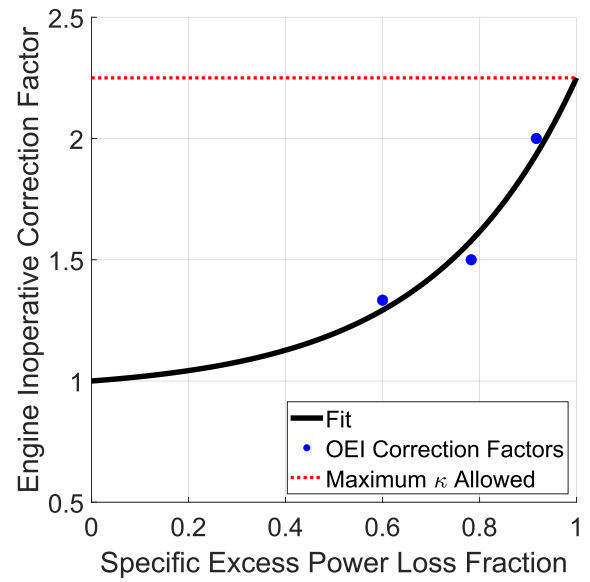
(a) Linear function.



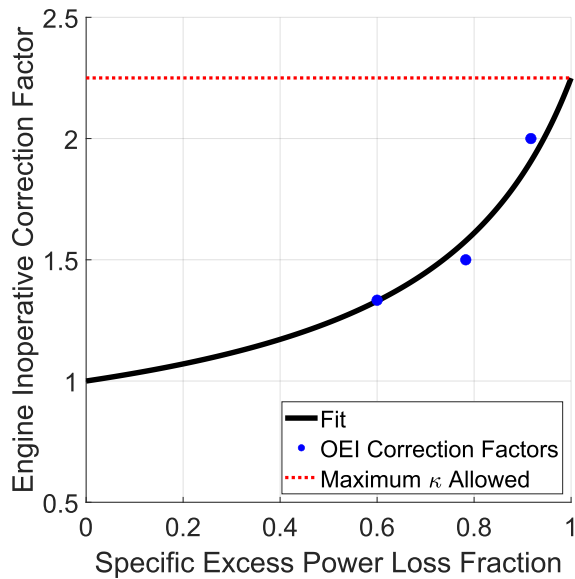
(b) Quadratic function.



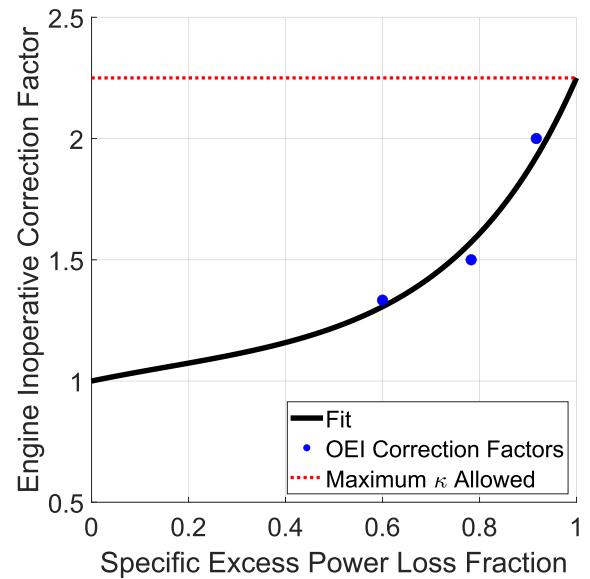
(c) Cubic function.



(d) Exponential function.



(e) Tangent function.



(f) Hyperbolic sine function.

Figure 5.9: Candidate engine inoperative correction factor curve fits.

There are multiple distinguishing features between the linear and quadratic curves. The linear function exhibits the shallowest slope at larger SEPLFs, but fails to reach a correction factor of 2, resulting in aircraft being under-sized relative to those that are presently operated.

The quadratic function, however, minimally surpasses a correction factor of 2 for SEPLFs greater than 95%, rendering the constraint on $\hat{\kappa}(1)$ to be inactive. As a result, the propulsion system size would not significantly increase relative to existing ones installed on aircraft presently in the global fleet.

While both curves over-predict the correction factors for 3- and 4-engine aircraft and under-predict that for 2-engine aircraft, the optimized quadratic curve returns a lower RMSE than the linear curve. Thus, for the given maximum allowable engine correction factor, the quadratic curve was selected as the preferred one to be utilized for defining an engine inoperative correction factor. Equation 5.20 defines the engine inoperative correction factor with its optimized parameters:

$$\hat{\kappa}(P_{s_{\text{loss}}}) = 1.045 (P_{s_{\text{loss}}})^2 + 1.000 \quad (5.20)$$

5.5.5 Results: Lower Maximum Allowable Correction Factor

The same curve fit outlined in Eq. 5.19 is performed for all candidate curves listed in Tab. 5.10, but with a maximum correction factor of 2.05. Table 5.12 lists the updated RMSE from each curve fit, with plots of each curve fit in Fig. C.4 within Appendix C.5.

Table 5.12: Engine inoperative curve constructions, $\hat{\kappa}_{\text{max}} = 2.05$.

Curve	RMSE
Linear	1.885×10^{-1}
Quadratic	1.098×10^{-1}
Cubic	9.875×10^{-2}
Exponential	1.027×10^{-1}
Tangent	1.090×10^{-1}
Hyperbolic Sine	1.027×10^{-1}

As the maximum allowable correction factor decreases, the RMSE of the cubic, exponential, tangent, and hyperbolic sine curves all increase and begin to approach that of the quadratic function. As observed in Fig. 5.9, the four aforementioned curves all over-predicted the correction factor for 3-engine aircraft but under-predicted that for 2- and 4-engine aircraft. By imposing a smaller maximum engine inoperative correction factor, the exponential, tangent, and hyperbolic sine functions all over-predict the engine inoperative correction factor for 3- and 4-engine aircraft but under-predict that for 2-engine aircraft, exhibiting similar behavior

to the quadratic function.

The convergent behavior towards that of the quadratic curve suggests that it remains as the preferred candidate to represent the engine inoperative correction factor, particularly if the maximum engine inoperative correction factor were to decrease further. Thus, the engine inoperative correction factor defined in Eq. 5.20 remains as the preferred curve even after testing with a different maximum allowable correction factor.

5.5.6 Sensitivity Study

To summarize the results in Sections 5.5.4 and C.5, a sensitivity study is performed to analyze the sensitivity of the RMSE to the parametric curves and maximum allowable correction factors illustrated. Figure 5.10 plots the RMSE for each parametric curve fit under a varying maximum allowable engine inoperative correction factor.

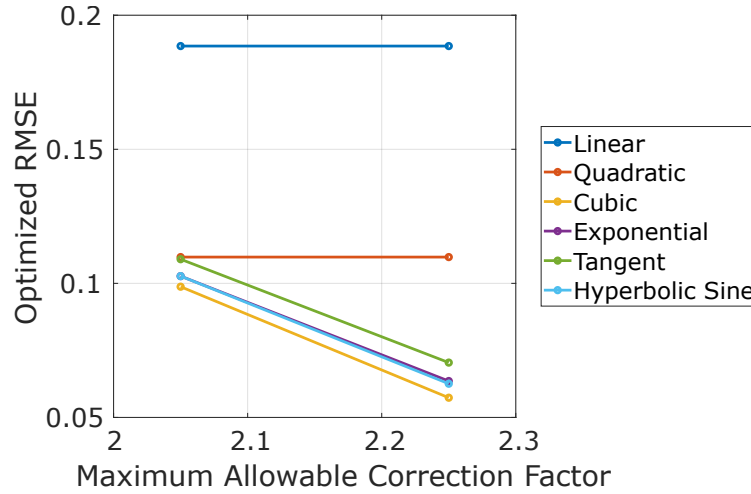


Figure 5.10: Sensitivity of RMSE with respect to the parametric curves and maximum allowable engine inoperative correction factor.

As observed in Sections 5.5.4 and C.5, the RMSE for each parametric curve either decreases or remains constant for an increasing maximum allowable engine inoperative factor. The RMSE of the linear and quadratic curves do not change for the selected maximum allowable correction factors, indicating weak sensitivity.

However, the RMSE of the remaining curves decreases, indicating a stronger fit with the traditional $n/(n - 1)$ engine inoperative correction factors. Thus, there is a tradeoff between the optimized RMSE values and the practical/physical implications of the curve fit. Although

the RMSE decreases, the curve fit more closely matches the traditional engine inoperative correction factors, resulting in larger correction factors for more severe propulsion system failures. The increasing correction factor requires the propulsion system size to increase more than a conventional one, resulting in an over-sized propulsion system and likely degraded system-level performance.

The results from the study suggest that the quadratic curve remains the preferred curve for defining the engine inoperative correction factor. While its RMSE is reasonably low, the curve fit does not change under the different maximum allowable correction factors utilized in Experiment 3.0.3. Since the RMSE does not change as the maximum allowable correction factor varies, the curve fit is robust, remaining closely tied to the traditional $n/(n-1)$ engine inoperative correction factors, but does not return large correction factors that would cause the propulsion system to be over-sized.

5.5.7 Conclusions

In Experiment 3.0.3, a generalized engine inoperative correction factor was constructed by fitting multiple candidate parametric curves to a mapping of the SEPLF and traditional $n/(n-1)$ correction factors used in conventional aircraft sizing. Aside from the multiple candidate curves considered, multiple maximum allowable engine inoperative correction factors were imposed as constraints in the curve fit, assessing each curve fit's robustness to the different constraints imposed.

For both maximum allowable engine inoperative correction factors of 2.25 and 2.05, the linear and quadratic curves returned the same RMSE because the constraint on the maximum allowable correction factor remained inactive. However, for the cubic, exponential, tangent, and hyperbolic sine curves, the RMSE worsened as the maximum allowable correction factor decreased, suggesting that the maximum allowable correction factor heavily impacts the RMSE. For a constant maximum allowable correction factor, the quadratic curve was selected because of its low RMSE and ability to return engine inoperative correction factors comparable to the traditional ones for severe propulsion system failures. While the cubic, exponential, tangent, and hyperbolic sine curves exhibited lower RMSE values than the quadratic function selected, the required propulsion system size increased rapidly for more severe propulsion system failures, making them less desirable. Results from the sensitivity study further corroborate this claim; as the maximum allowable correction factor increased, the corresponding curves' RMSE decreased, resulting in a closer fit to the traditional $n/(n-1)$ correction factors and an over-prediction of the engine inoperative correction factor for severe

propulsion system failures (approximately 2.25, 12.5% greater than the largest traditional one).

In the future, additional studies could be performed by further varying the maximum allowable factor heuristically, or by including it as a design variable in the optimization problem. The former enables further exploration of the design space while the latter could serve as a useful validation study confirming the robustness of the quadratic curve with respect to different maximum allowable correction factors.

Together, the results from Experiments 3.0.2 and 3.0.3 can be integrated into an aircraft sizing code, enabling advanced aircraft concepts to be sized for different primary propulsion system failure modes. Experiment 3.1.1 performs such a test for an aircraft with a distributed propulsion system.

5.6 Advanced Aircraft Sizing Study (Experiment 3.1.1)

The generalized climb requirements and engine inoperative correction factor established in Sections 5.4 and 5.5, respectively, are now applied to size a notional Elysian E9X (hereinafter referred to as the “E9X”), a battery-electric aircraft that carries 90 passengers and utilizes a distributed propulsion system with eight propulsors [9, 40]. First, the notional E9X is modeled in FAST to identify the SEPLFs associated with multiple novel engine inoperative scenarios. Next, separate constraint diagrams for each engine inoperative scenario are generated using FAA Part 25 regulations and the generalized climb performance requirements from Section 5.4, enabling a comparison between the optimum sizing points. Lastly, the selected sizing points are utilized to resize the notional E9X. The system-level impacts observed highlight the importance of generalizing the performance requirements for future aircraft concepts with advanced propulsion systems.

In Experiment 3.1.1, the traditional “one engine inoperative” condition utilized on 2-4 turbofan-/turboprop-powered aircraft is being extrapolated to an aircraft with a distributed propulsion system. Rather than simulating traditional “one engine inoperative” conditions, use cases of 1-6 propulsor failures are considered. The sizing results presented in this section are derived from the generalized climb performance requirement and engine inoperative correction factor constructions presented in Sections 5.4 and 5.5. Such constructions are conceptual design heuristics grounded in the fleet behavior of previously certified aircraft, and are only intended to provide a first-order sizing estimate rather than a means to certify a future aircraft concept.

Rather than prescribing compliance, the generalized climb performance requirements and engine inoperative correction factors are intended to help the airframe manufacturer explore the design space and predict the sensitivity of the aircraft’s system-level performance with respect to multiple candidate primary propulsion system failure modes. Given that the presently published FAA Part 25 requirements were developed only for conventional aircraft powered by 2-4 turbofan/turboprop engines, it is envisioned that certification authorities would consider multi-propulsor failures as a “special condition,” pertaining to only the aircraft seeking certification; the multi-propulsor failures considered under the special conditions would be driven by the airframe manufacturer’s system safety assessments.

5.6.1 Aircraft Modeling

A notional E9X model was developed in FAST [177], serving as a baseline for identifying the SEPLFs under multiple novel engine inoperative scenarios. All parameters used to model the notional E9X are included in Tab. C.9 in Appendix C.6, along with all modeling assumptions and approximations made. The boundary conditions for an assumed design mission profile are defined in Tab. C.10 in Appendix C.6, as no detailed one was defined in the literature. To limit the battery size, the E9X flies its reserve mission utilizing a “Reserve Energy System” comprised of a fuel-powered turbogenerator, producing electrical power for the distributed propulsion system. However, the weight breakdowns in Refs. [9, 40] fail to provide exact weight contributions of such systems, obfuscating their impact on their overall contribution to the maximum takeoff weight (MTOW) and system performance. As a result, the analysis in this work was simplified by excluding the reserve mission from the flight simulation and assuming that the Reserve Energy System weight is accounted for in the aircraft’s operational empty weight.

The notional E9X was calibrated by defining an “airframe calibration factor” (included in Tab. C.9 in Appendix C.6) to tune the model until its MTOW and battery weight closely matched those published in the literature. Although a battery weight was not explicitly reported, Fig. 3 in Ref. [9] estimates that the battery weight is 46% of MTOW, enabling model validation. The minimal error in MTOW and less than 3% error on the battery mass is a reasonable level of error for a conceptual design study, indicating that model calibration was successful.

After calibrating the notional E9X, novel engine inoperative scenarios were defined for one to six propulsor failures, assuming that they all simultaneously failed immediately after takeoff. The failure modes demonstrated in this experiment are for illustrative purposes, showcasing

Table 5.13: Model calibration for the notional E9X. All masses are in units of kg.

Mass	Literature	FAST Model	Percent Error (%)
MTOW	76,000	76,047	+0.06
Battery	34,960	35,985	+2.93

the impacts of system-level performance as a result of the changing requirements. In reality, four or more propulsors failing on the aircraft is highly unlikely, and would probably not result from an arbitrary combination of propulsors failing. Rather, the failure would likely occur due to a common failure mode or external event, such as a battery pack fire, severed electrical cables, or collision with foreign object debris. Propulsors in the distributed propulsion system could also be shut down to mitigate the effects a flight control system failure, providing differential thrust. Since the E9X has only been designed conceptually [9, 40], there are no details about the electrical subsystem design and the mechanisms to ensure that there are no single point of failure or common-mode failures; the resulting subsystem architecture would likely be similar to NASA’s SCEPTOR Distributed Propulsion Flight Demonstrator that uses multiple battery packs to power the electric propulsors [90].

The aforementioned propulsion system failures were simulated on the generic diversion mission profile previously outlined in Tab. C.1 as part of Appendix C.1. For this case study, two changes were made to the mission profile:

1. The climb speed was set equal to the takeoff speed ($\Delta V = 0$ m/s).
2. The diversion cruise speed was reduced to Mach 0.40, as defined in the literature [9].

The SEPLFs were computed for each engine inoperative scenario, as outlined in Tab. 5.14. Due to the lack of publicly available geometric data available for the notional E9X, a detailed aerodynamic model could not be used, meaning that windmilling and trim drag were not accounted for. As a result, the SEPLFs are an underestimate relative to their actual value if engine inoperative conditions were to be simulated in a physical environment.

Relative to AEO conditions, the SEPLF increments by approximately 16% for each propulsor failure. Compared to an engine inoperative scenario on an average 4-engine aircraft, the SEPLF for one engine inoperative on the notional E9X is approximately 3.75-times smaller, highlighting that distributed propulsion systems facilitate a more graceful flight performance degradation during such failures. The SEPLFs are then used to construct constraint diagrams under both the FAA and generalized climb performance requirements.

Table 5.14: SEPLFs for each novel engine inoperative scenario.

Propulsors Failed	Specific Excess Power (m/s)	Specific Excess Power Loss Fraction
0	11.6531	—
1	9.7607	0.1624
2	7.8698	0.3247
3	5.9820	0.4867
4	4.0972	0.6484
5	2.2156	0.8099
6	0.3370	0.9711

5.6.2 Constraint Diagram Generation and Sizing Point Comparison

First, a notional E9X constraint diagram is generated and validated against one published in the literature. Private communications with the authors of Ref. [9] confirmed that the propulsion system size was corrected for engine inoperative conditions (using a traditional $n/(n - 1)$ correction factor for an 8-engine aircraft) and MTOW during the sizing process. Second, the constraint diagram is regenerated using the proposed performance requirements and corresponding correction factors. For each of the constraint diagrams generated, the selected design point is utilized to resize the E9X in FAST, enabling system-level performance and electric motor sizes to be compared.

5.6.2.1 Notional E9X Constraint Diagram and Validation

A constraint diagram for the notional E9X was generated using FAA Part 25 climb performance requirements for a 4-engine aircraft and accounted for the aforementioned correction factors. In addition to the climb constraints, the authors of Ref. [9] also imposed an AEO climb gradient to ensure that the E9X had enough power to climb after takeoff. All parameters to generate the constraint diagram are listed in Tab. C.11 within Appendix C.6, including the aerodynamic performance modeling methodology and assumptions. The resulting constraint diagram is plotted in Fig. 5.11, with an optimum sizing point corresponding to a wing loading of 5 kN/m² and power loading of 0.067 N/W, matching the values obtained in prior literature [9].

The constraint diagram was validated by utilizing an online digitizer ³ to capture the power loading values corresponding to each constraint at wing loadings of 4 and 6 kN/m².

³<https://apps.automeris.io/wpd4/>

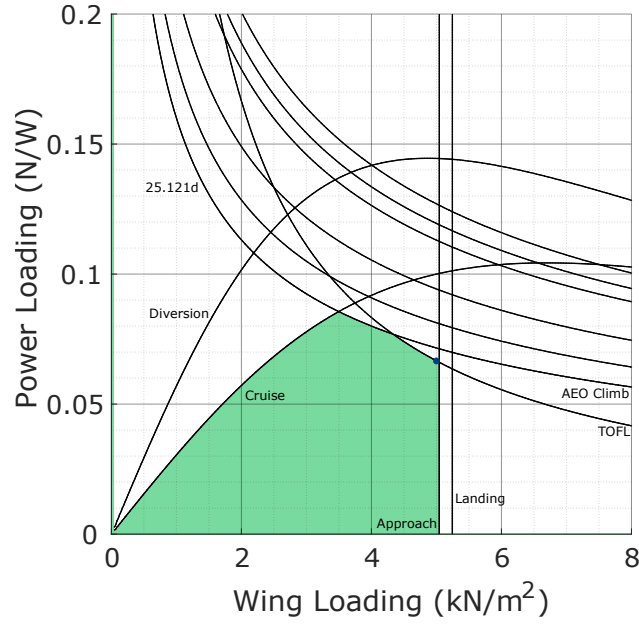


Figure 5.11: Baseline constraint diagram to match literature. The feasible region is shaded in green.

Table 5.15 lists the digitized values and the absolute/relative error for a wing loading of 4 kN/m². Table C.12 in Appendix C.6 reports the absolute/relative error for a wing loading of 6 kN/m². The approach and landing field length constraints are not included in the tables because their values only depend on the wing loading, and were matched exactly.

Table 5.15: Constraint diagram comparison at a 4 kN/m² wing loading.

Constraint	Model	Literature	Absolute Error	Relative Error
FAA 25.121a	0.1438	0.1672	-0.0234	-13.995%
FAA 25.121b	0.1278	0.1302	-0.0024	-1.843%
FAA 25.121c	0.1353	0.1472	-0.0119	-8.084%
FAA 25.121d	0.0918	0.0979	-0.0061	-6.231%
FAA 25.119	0.1065	0.1050	+0.0015	+1.429%
AEO Climb	0.0808	0.0811	-0.0003	-0.370%
Cruise	0.0930	0.0958	-0.0028	-2.923%
Diversion	0.1438	0.1354	+0.0084	+6.204%
Takeoff Field Length	0.0843	0.0906	-0.0063	-6.954%

The error reported during validation is attributed to two key modeling differences. First,

the authors in Ref. [9] model the takeoff field length using a semi-empirical method from Torenbeek [194], dividing the takeoff field length into rolling and airborne distances, and requires knowledge of the second-segment climb speed (V_2 , previously shown in Fig. 1.3). In this work, the takeoff field length is simply approximated using the takeoff parameter provided by Roskam [103]. Second, due to the lack of publicly available aerodynamic data, the cruise and diversion constraints were scaled by a factor of $1/\sigma^a$, where σ is the density ratio at the respective altitude and a is a user-selected parameter to tune the constraint; this factor typically accounts for gas turbine engines lapsing. Although electric motors do not lapse at altitude, tuning the constraints in such a manner enables the modeling approach to be seamlessly adapted for concepts utilizing gas turbine engines or turbogenerators in the future, particularly if the E9X’s Reserve Energy System is modeled in greater detail. Given that the aerodynamic parameters were estimated using first-order approximations and that little geometric information is provided about the E9X, the level of modeling error present is deemed acceptable, though it could be improved in future work.

5.6.2.2 Notional E9X Sizing with the Generalized Requirements

To fully assess the system-level impacts of sizing an aircraft for different propulsion system failure severities, constraint diagrams are now regenerated utilizing the generalized climb performance requirements and engine inoperative correction factor. The resultant constraint diagrams are plotted in Fig. 5.12 for 1- and 6-engine inoperative conditions; additional constraint diagrams for the other engine inoperative conditions are in Fig. C.5 within Appendix C.6.4. Again, given that the windmilling and trim drag were not computed due to the lack of geometric data publicly available, the SEPLFs are underestimated, resulting in an optimistically sized aircraft. If the windmilling and trim drag were modeled, the selected power loadings from the constraint diagram would be smaller.

As the propulsion system failure severity increases, the required power loading monotonically decreases, resulting in tighter constraints that shift towards the bottom-left corner of the constraint diagram. At first, the observed behavior may seem to contradict that observed in Fig. 5.7; as the SEPLF (propulsion system failure severity) increases, the climb performance requirements are loosened, requiring a shallower climb gradient and enabling the propulsion system to be downsized. However, for increasing SEPLFs, the generalized engine inoperative correction factor increases, requiring the propulsion system size to grow. Thus, there is an interaction between the generalized requirements and engine inoperative correction factor constructed – increasing the propulsion system failure severity decreases the required climb

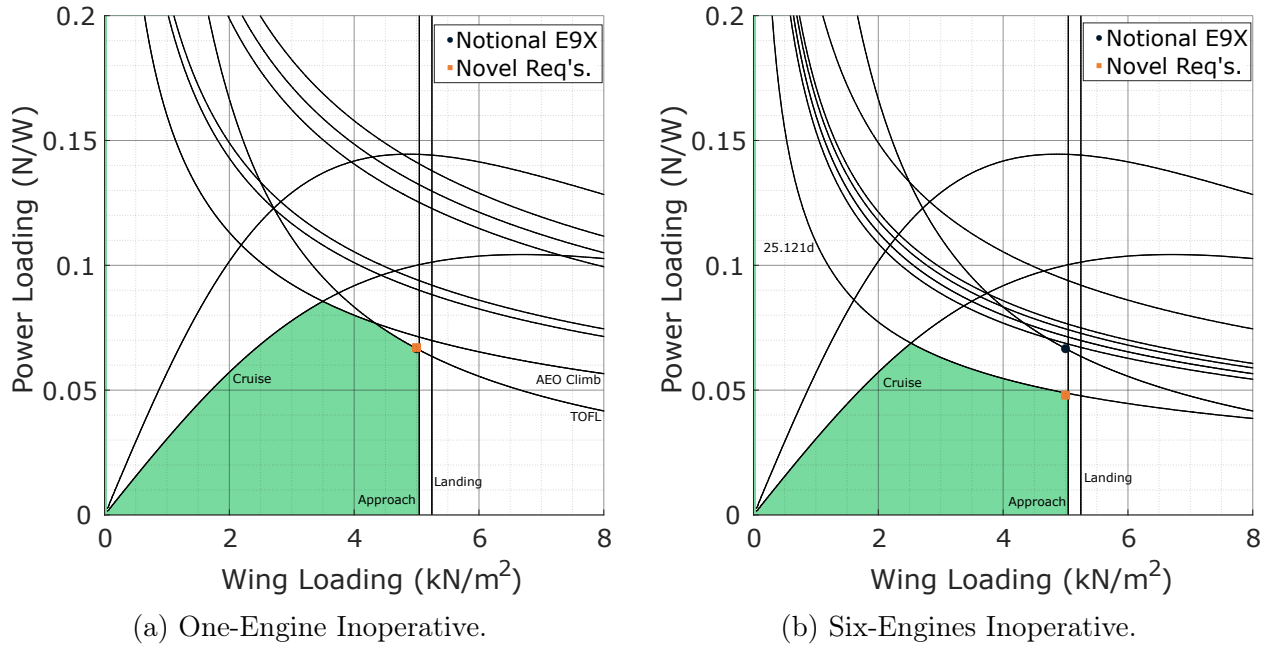


Figure 5.12: Constraint diagrams using the proposed climb requirements and engine inoperative correction factor. The feasible region is shaded in green.

gradients but increases the engine inoperative correction factor. As observed in Fig. 5.12, the constraints are tightened for an increasing propulsion system failure severity, shifting the contours towards the bottom-left of the constraint diagram and revealing that the engine inoperative correction factor's behavior dominates the interaction.

For the 1-, 2-, and 3-engine inoperative scenarios (constraint diagrams also included in Appendix C.6), the Part 25 climb constraints are loosened relative to those constraining the notional E9X, shifting them towards the upper-right corner of the constraint diagrams. However, the takeoff field length (TOFL) and AEO climb constraints remain active, preventing a design point with a larger power loading from being selected. If these constraints were to be neglected, the power loading could be increased, possibly resulting in improved system-level performance, such as reduced MTOW or battery energy stored on the aircraft. Further investigation would be necessary to confirm that such system-level performance improvements are observed.

For the 4-, 5-, and 6-engine inoperative scenarios (constraint diagrams also included in Appendix C.6), the Part 25 climb constraints are tightened relative to those constraining the notional E9X, shifting them towards the lower-left corner of the constraint diagrams.

As a result, the FAA Part 25.121d constraint becomes active while the TOFL and AEO climb constraints become inactive, decreasing the required power loading and increasing the propulsion system size.

Using the newly identified design points for all novel engine inoperative scenarios, the notional E9X was resized in FAST. Table 5.16 reports the system-level performance parameters and electric motor size relative to the notional E9X. Since the windmilling and trim drag were not modeled, the system-level benefits reported are optimistic, and would be worse if such aerodynamic considerations were accounted for.

Table 5.16: Performance parameters and weights for the redesigned E9Xs. Percentages listed are the relative difference with respect to the notional E9X.

Configuration	Power Loading (N/W)	MTOW (kg)	Battery Weight (kg)	Battery Energy (MWh)	Motor Weight (kg)	Motor Power (MW)	Time to Climb (min)
Notional E9X	0.067	76,047	35,985	12.95	322	1.61	19.47
1-Engine Inop.	0.067 (+0.00%)	76,047 (+0.00%)	35,985 (+0.00%)	12.95 (+0.00%)	322 (+0.00%)	1.61 (+0.00%)	19.47 (+0.00%)
2-Engines Inop.	0.067 (+0.00%)	76,047 (+0.00%)	35,985 (+0.00%)	12.95 (+0.00%)	322 (+0.00%)	1.61 (+0.00%)	19.47 (+0.00%)
3-Engines Inop.	0.067 (+0.00%)	76,047 (+0.00%)	35,985 (+0.00%)	12.95 (+0.00%)	322 (+0.00%)	1.61 (+0.00%)	19.47 (+0.00%)
4-Engines Inop.	0.064 (-3.90%)	77,139 (+1.44%)	36,596 (+1.70%)	13.17 (+1.66%)	340 (+5.65%)	1.70 (+5.65%)	18.16 (-6.71%)
5-Engines Inop.	0.056 (-15.92%)	83,171 (+9.37%)	39,897 (+10.87%)	14.36 (+10.85%)	419 (+30.20%)	2.09 (+29.89%)	14.65 (-24.74%)
6-Engines Inop.	0.048 (-27.93%)	88,777 (+16.74%)	42,726 (+18.73%)	15.38 (+18.72%)	521 (+61.89%)	2.61 (+62.20%)	12.83 (-34.09%)

Since no new sizing point was identified for the E9X resized under 1-, 2-, or 3-engine inoperative scenarios, no system-level benefits are attained. However, for the E9X resized under 4-, 5-, and 6-engine inoperative scenarios, system-level performance worsens as a result of the smaller power loadings selected from the constraint diagram. The optimal power loading identified from the constraint diagrams was between 0.048 and 0.064 N/W, resulting in a 4 to 28% decrease relative to the notional E9X, resulting in the electric motors' size increasing by 5 to 60%. The power loading decrease is observed because the engine inoperative correction factor corresponding to the aforementioned propulsion system failures is larger than that for the notional E9X aircraft (8/(8-1), reflecting a "traditional" engine inoperative correction factor). As a result of the increased propulsion system size, the aircraft's MTOW increases, requiring more battery energy to be stored onboard the aircraft. The only system-level performance benefit from increasing the propulsion system size is that it takes less time to climb to cruise altitude.

More generally, if the selected propulsion system failure mode results in a SEPLF corresponding to an engine inoperative correction factor greater than the traditional $n/(n - 1)$ correction factor, then the aircraft’s propulsion system size increases and system-level benefits will not be observed. Conversely, system-level benefits may be observed if the selected propulsion system failure mode results in a SEPLF corresponding to an engine inoperative correction factor less than the traditional $n/(n - 1)$ correction factor. Thus, the failure mode chosen by the engineer significantly impacts the resultant aircraft size. Based on this observation, the failure mode may become a critical design variable during aircraft conceptual design, particularly for concepts leveraging advanced and/or distributed propulsion systems.

5.6.2.3 Neglecting TOFL and AEO Climb Constraints

The results presented in Fig. 5.12 and Tab. 5.16 revealed that system-level benefits were not observed if the notional E9X was resized for 1-, 2-, or 3-engine inoperative conditions because the TOFL and AEO Climb constraints remained active. For experimentation purposes only, the constraint diagrams for 1-, 2-, and 3-engine inoperative conditions were re-considered by neglecting the TOFL and AEO Climb constraints. If this were to occur, the Part 25.121d constraint becomes active, mimicking the constraint diagram’s observed behavior while resizing the notional E9X for 4-, 5-, and 6-engine inoperative conditions. As a result of neglecting these two constraints, the feasible region increases and larger power loadings are admitted, enabling the propulsion system size to decrease. The larger power loadings make it possible to realize system-level benefits such as an MTOW reduction or decreased battery energy stored onboard.

Table 5.17 lists the system-level performance parameters and electric motor size for the resized notional E9X under 1-, 2-, and 3-engine inoperative conditions while neglecting the TOFL and AEO Climb constraints. After neglecting the TOFL and AEO climb constraints, the corresponding power loadings selected for 1-, 2-, and 3-engine inoperative scenarios are 9 to 35% larger than that of the notional E9X, enabling the propulsion system to be downsized. Despite downsizing the propulsion system, system-level benefits are only observed in the configuration re-sized for 3-engine inoperative conditions, resulting in a 1.70% MTOW reduction and a 1.62% battery energy savings. System-level benefits are not realized for the configurations sized under 1- and 2-engine inoperative conditions because the time to climb to cruise altitude becomes prohibitively large, requiring additional battery energy to climb for an extended period of time.

Table 5.17: Performance parameters and weights for the redesigned E9Xs. Percentages listed are the relative difference with respect to the notional E9X.

Configuration	Power Loading (N/W)	MTOW (kg)	Battery Weight (kg)	Battery Energy (MWh)	Motor Weight (kg)	Motor Power (MW)	Time to Climb (min)
Notional E9X	0.067	76,047	35,985	12.95	322	1.61	19.47
1-Engine Inop.	0.090 (+35.14%)	83,293 (+9.53%)	41,234 (+14.59%)	14.84 (+14.55%)	261 (-18.90%)	1.30 (-19.21%)	37.95 (+94.96%)
2-Engines Inop.	0.083 (+24.62%)	76,673 (+0.82%)	36,929 (+2.62%)	13.29 (+2.59%)	260 (-19.21%)	1.30 (-19.21%)	30.59 (+57.15%)
3-Engines Inop.	0.073 (+9.61%)	74,756 (-1.70%)	35,388 (-1.66%)	12.74 (-1.62%)	289 (-10.20%)	1.44 (-10.51%)	23.12 (+18.77%)

Although primarily illustrative due to the neglected TOFL and AEO Climb constraints, the propulsion system failure mode still impacts the overall system-level performance, but is not strictly improved. If the SEPLF causes the propulsion system size to be significantly downsized, then climb performance during nominal (AEO) operations is degraded, resulting in climb performance penalties. These observations further emphasize the significance of the failure mode chosen by the engineer and its impact on the overall aircraft size and system-level performance during conceptual design.

In the cases that system-level performance benefits are not observed, an additional design space exploration or optimization study can be executed to identify the power loading that maximizes system-level performance. Rather than selecting the highest possible power loading in the feasible region, a smaller one can be selected, balancing the tradeoffs between improved system-level performance (downsized propulsion system) and reduced nominal climb performance.

5.7 Chapter Summary

Future aircraft concepts leveraging distributed propulsion are key enablers for more efficient flight, but face major certification gaps prior to entering service. One of these gaps is the lack of applicability of engine inoperative performance requirements, which have only been defined in FAA Part 25 regulations for conventional aircraft utilizing two, three, or four gas turbine engines. Further challenges emerge for advanced aircraft concepts utilizing disparately sized propulsors, as the present regulations implicitly assume that all propulsors are homogeneously sized.

As a result of the aforementioned gaps, Research Question 3.0 sought to construct

generalized climb performance requirements and an engine inoperative correction factor in a configuration-agnostic manner, enabling broad application to distributed and heterogeneously-sized propulsion systems. To answer Research Question 3.0, Hypothesis 3.0 claimed that a monotonic, bounded mapping between the specific excess power loss and the required climb gradient will reproduce the climb gradient requirements implied by the published regulations for conventional aircraft utilizing discrete, homogeneously sized propulsors while extending the same flight performance logic to accommodate aircraft concepts utilizing propulsion systems with heterogeneously sized or distributed propulsors. Experiments 3.0.1 and 3.0.2 identified a suitable parameterization and constructed generalized flight performance requirements, respectively, while Experiment 3.0.3 constructed an engine inoperative correction factor under the same parameterization.

Experiment 3.0.1 found that the SEPLF was a suitable parameter for describing system-level performance under engine inoperative conditions because of its ability to aggregate both the propulsive and aerodynamic impacts of an propulsion system failure within one quantity. Then, in Experiment 3.0.2, generalized climb performance requirements were constructed as a function of the SEPLF, a proxy for the propulsion system failure severity, overcoming limitations on existing FAA Part 25 regulations defined by the number of engines installed on the airframe. The generalized climb requirements were constructed by fitting Sigmoid Curves to a mapping between the median SEPLF across a benchmark suite of conventional aircraft and the corresponding climb performance requirements in the FAA Part 25 regulations. The resulting Sigmoid Curves mimicked step functions, suggesting that future climb performance requirements could be governed by an upper/lower limit depending on the propulsion system failure severity. The constructed climb performance requirements were also pivotal for validating prior assumptions in the literature – in the absence of generalized climb performance requirements, it is reasonable to size an aircraft with a distributed propulsion system using the FAA’s 4-engine climb performance requirements.

To ensure that the generalized climb performance requirements reproduced the published climb gradients for conventional aircraft concepts, a back-test was performed to predict the required climb gradients for a set of validation data. The back-test demonstrated that the generalized climb performance requirements for most 2- and 4-engine aircraft change by no more than 0.01% relative to those defined in FAA Part 25 regulations. The climb gradients were reproduced successfully because the Sigmoid Curve slopes were constrained to be within a tolerance of zero at the endpoints, guaranteeing that the generalized flight performance requirements remained bounded by those defined in the FAA Part 25 regulations for extremely

large/small SEPLFs.

In Experiment 3.0.3, an engine inoperative correction factor was constructed by fitting a curve to a mapping of the SEPLF and traditional $n/(n - 1)$ engine inoperative correction factors. Since multiple parametric curves could have defined the engine inoperative correction factor, a sensitivity study was executed to identify the curve that fit the data well while remaining robust to different maximum allowable correction factor values. The quadratic curve fit was found to satisfy the aforementioned criteria because of its low RMSE and robustness to define engine inoperative correction factors that remain close to the traditional $n/(n - 1)$ ones without over-sizing the propulsion system across multiple maximum allowable correction factor values.

The results from Experiments 3.0.1 through 3.0.3 provided a baseline framework for constructing generalized engine inoperative performance requirements and an engine inoperative correction factor. Then, Research Question 3.1 explored how the size of an advanced aircraft concept would be impacted if the generalized performance requirements were utilized instead of the FAA-published ones. Hypothesis 3.1 claimed that for an advanced aircraft concept with a propulsion system failure mode that produces a smaller specific excess power loss than a conventional aircraft under a legacy one-engine inoperative failure mode, sizing under the generalized climb performance requirements and engine inoperative correction factor will reduce over-sizing and improve system-level performance while maintaining the required off-nominal climb gradient for the corresponding failure mode.

Hypothesis 3.1 was tested by sizing a notional Elysian E9X under multiple novel propulsion system failure modes. Due to the TOFL and AEO climb constraints remaining active for 1-, 2-, and 3-engine inoperative scenarios, no system-level benefits were observed. For the configurations designed under 4-, 5-, and 6-engine inoperative scenarios, system-level performance penalties were exhibited due to the increased propulsion system size. In the aforementioned scenarios, the power loading decreased by approximately 4 to 28%, resulting in an MTOW increase between 1 to 17% and required 1 to 19% more battery energy to be stored onboard the aircraft. System-level performance penalties were observed because the SEPLF considered returned an engine inoperative correction factor larger than one output using a traditional $n/(n - 1)$ correction factor.

If the TOFL and AEO Climb constraints are neglected in the analysis, then the power loading increases by 9 to 35% for the configurations sized under 1-, 2-, and 3-engine inoperative conditions. However, system-level benefits are only observed for the configuration sized under 3-engine inoperative conditions; a 1.70% MTOW savings is realized, reducing the required

battery energy to be stored onboard the aircraft by 1.62%. For the other configurations, system-level performance benefits are not attained because the propulsion system is significantly downsized and degrades nominal climb performance (during AEO conditions), requiring more time to climb to cruise altitude and, consequently, additional battery energy.

Thus, for an advanced aircraft concept with a propulsion system failure mode that produces a smaller SEPLF than a conventional aircraft under a legacy one-engine inoperative failure mode, sizing under the generalized climb performance requirements and engine inoperative correction factor will reduce over-sizing but does not guarantee improved system-level performance. To guarantee improved system-level performance, an additional sensitivity or optimization study may be necessary to find an optimal power loading within the feasible region. For novel propulsion system failure modes, the optimal power loading may not be the largest one available in the feasible region. Therefore, in future aircraft designs involving advanced propulsion systems, the primary failure mode chosen will significantly impact system-level performance, particularly for aircraft leveraging advanced/distributed propulsion systems.

While the multi-propulsor failures presented in the experiment (particularly those involving four or more propulsor failures) are unlikely to occur, designing for such failure modes could be useful for overcoming additional flight-critical system failures, such as the loss of flight control electronics system. Rather than losing all pitch, roll, and yaw authority, some of the distributed propulsors could be shut down, providing differential thrust to maintain stability. The remaining distributed propulsors would then provide enough power to climb away from any obstacles and fly to a safe landing space. Such capabilities can only be provided by distributed propulsion systems, highlighting their advantages from both reliability and flight control perspectives.

Together, the generalized climb performance requirements and engine inoperative correction factor pave a pathway for sizing future aircraft concepts under conditions similar to those defined in FAA Part 25 regulations while providing engineers the flexibility to reap system-level benefits from advanced propulsion technologies. Furthermore, the generalized climb performance requirements can be used in Chapter 6 as constraints during the aircraft sizing process while designing future aircraft concepts with advanced propulsion systems.

Chapter 6

Bi-Level Optimization Framework

6.1 Introduction

The literature review revealed that there is a need for a unified system architecture optimization environment with problem-specific heuristics to efficiently explore the design space and identify the best system architecture. While combinatorial or exhaustive searches traverse the entire design space, significant computational resources are required, even if utilizing parallel or high-performance computing resources. Furthermore, designing a globally-optimal system architecture also requires component sizing, creating an additional optimization problem to be solved for each candidate system architecture. Additionally, no system safety analyses have been embedded in any prior optimization frameworks, neglecting system safety as either an objective or constraint. To overcome the challenges and limitations present in the literature, a safety-constrained system architecture optimization framework is constructed with problem-specific heuristics, enabling the best system architectures to be rapidly identified without the need for combinatorial or exhaustive design space searches.

6.1.1 Research Questions, Hypotheses, and Experiments

Advanced propulsion system architecture synthesis requires both the topology (component connections) and component sizes (via a power management strategy) to be optimized for system-level performance. Additionally, safety becomes an important objective in system architecture optimization, ensuring that there are no single points of failure in the propulsion system. Research Question 4.0 explores how propulsion system architectures can be optimized for both system-level performance and safety, supporting the design of future aircraft concepts and their subsystems:

RQ 4.0: How can propulsion system architecture topologies and corresponding power management strategies be synthesized to optimize the system-level performance and safety of an aircraft concept?

The literature review revealed three possible methods for designing advanced system architectures: 1) combinatorial searches, possibly involving graph enumeration; 2) multidisciplinary or numerical optimization; and 3) Constraint Programming or solving a Boolean Satisfiability Problem. As indicated previously, combinatorial or enumerative searches require significant computational resources, preventing the best design from being attained quickly [108]. Constraint Programs and Boolean Satisfiability Problems enable a *feasible* system architecture to be identified, but not necessarily an *optimal* one [125, 139]. MDO frameworks are capable of solving hierarchical optimization problems, enabling the system architecture topology and power management strategy to be designed together within a unified computational framework.

Traditionally, stochastic optimization methods are needed to strategically or randomly explore the design space without evaluating all candidate designs. Existing hierarchical MDO frameworks typically utilize evolutionary algorithms with generalized heuristics, such as multi-objective Genetic Algorithms [128], requiring multiple system architectures to be evaluated during each major optimization iteration. If the system architecture component sizes are also optimized, an additional gradient-based optimization is executed within each objective function evaluation. Surrogate optimization can be useful for rapidly finding a globally-optimal solution with minimal function evaluations, but does not perform well if there are hidden constraints in the design space [133], posing challenges for accommodating infeasible system architecture topologies.

Rather than using stochastic optimization techniques that perform global searches and/or possibly return infeasible system architectures, problem-specific system architecting heuristics can be embedded in an optimization framework, enabling the best system architecture to be identified with fewer objective function evaluations and computational resources. To this end, Hypothesis 4.0 proposes developing a bi-level optimization framework with embedded heuristics to answer Research Question 4.0, enabling the assembly and optimization of safety-feasible propulsion system architectures:

Hyp 4.0: A bi-level architecture/power management optimization framework that assembles safety-feasible propulsion system architectures from a predefined component library utilizing embedded, problem-specific heuristics will synthesize propulsion system architectures with improved system-level performance than those designed with fixed system

architectures or stochastic optimization methods that utilize generalized, problem-agnostic heuristics.

Three parts of Hypothesis 4.0 must be tested. First, the constructed bi-level optimization framework must be validated and benchmarked, demonstrating that safety-feasible propulsion systems are assembled and exhibit improved system-level performance and safety relative to *only* optimizing the power management strategy for a set of fixed system architectures, respectively. For varying system safety requirements, advanced propulsion system architectures can be synthesized, creating a Pareto Frontier to illustrate the tradeoffs between optimizing for system-level performance and safety. To compare the optimality of the system architectures synthesized, the Pareto Hypervolume can be computed, indicating whether or not the bi-level optimization framework assembles safety-feasible system architectures with improved system-level performance or safety relative to a fixed system architecture with an optimized power management strategy only. Experiment 4.0.1 describes the necessary steps to perform the validation and benchmarking study.

Experiment 4.0.1: Validate the bi-level optimization framework’s architecture synthesis capabilities against a known advanced aircraft concept with fixed propulsion system architectures satisfying different system-level safety constraints and an optimized power management strategy.

1. Define a representative benchmark aircraft and corresponding sets of system-level safety constraints and propulsion system failure modes for the analysis.
2. Identify a representative propulsion system architecture that satisfies each system-level safety constraint, possibly containing redundant component connections.
3. For each fixed propulsion system architecture, minimize the fuel burn by varying the power management strategy in a constrained optimization problem. Optimize the power management strategy and corresponding fuel burn. Plot a performance-safety Pareto Frontier for the optimized objective function value and system-level safety constraints. Compute the Pareto Hypervolume of each Pareto Frontier generated.
4. Create a library of all components utilized in the fixed propulsion system architecture optimization, listing each component’s key performance parameters, inputs, and outputs.
5. For each system-level safety constraint, utilize the bi-level optimization framework to synthesize a propulsion system architecture topology and corresponding power

management strategy that minimizes fuel burn. Plot a performance-safety Pareto Frontier for the optimized objective function value and system-level safety constraints.

6. Compute the difference in Pareto Hypervolumes between the Pareto Frontiers generated from the bi-level optimization framework and fixed architecture power management optimization.

After validating the bi-level optimization framework, the latter half of Hypothesis 4.0 can be tested, comparing the optimized system architectures synthesized relative to those from a stochastic optimizer that utilizes generalized, problem-agnostic heuristics. The bi-level optimization framework is compared against a Genetic Algorithm optimizer that requires multiple system architectures to be evaluated during each major optimization iteration. Experiment 4.0.2 describes the benchmarking process between the bi-level optimization framework and Genetic Algorithm optimizer.

Experiment 4.0.2: Benchmark the bi-level optimization framework against stochastic optimization methods that utilize generalized, problem-agnostic heuristics.

1. Generate a benchmark test with a representative component library and system-level safety constraint.
2. Run the bi-level optimization framework to optimize an objective function value, recording the time to attain the optimum.
3. Run the same optimization problem utilizing a Genetic Algorithm optimizer for the same number of major iterations as the bi-level optimization framework.
4. Compare the optimized system-level performance and wall time between the two approaches.

First, a system architecture synthesis problem is solved with the bi-level optimization framework, providing a baseline optimized configuration and corresponding power management strategy. The bi-level optimization framework runs for a fixed number of iterations, serving as a limit on the number of generations allowed in the Genetic Algorithm optimizer. Then, the system-level performance of the two optimized system architectures can be compared, as well as the computational time required to perform the optimization. In the bi-level optimization framework, computational resources are utilized to find safety-feasible system architectures, while the Genetic Algorithm optimizer solves a power management optimization problem for

each candidate system architecture. Thus, there is a tradeoff between the computational time required to identify a safety-feasible system architecture and that required evaluate more system architectures during a random search of the design space. The benchmarking study acts as a method to compare which optimization approach is superior.

While Experiments 4.0.1 and 4.0.2 test different parts of Hypothesis 4.0, an additional study is performed to evaluate the bi-level optimization framework’s robustness to different initial system architectures. Experiment 4.0.3 describes a sensitivity study performed to ensure that the embedded, problem-specific heuristics in the bi-level optimization framework are robust to different initial propulsion system architectures, confirming that the optimized system architecture selected is weakly sensitive to the initial propulsion system architecture guessed.

Experiment 4.0.3 Verify that the optimized propulsion system architecture synthesized is weakly sensitive to the initial propulsion system architecture guessed.

1. Generate 50 propulsion system architectures as initial guesses for the bi-level optimization.
2. For each initial guess generated, run the bi-level optimization framework for a fixed system safety constraint. Record the initial and optimized system-level performance.
3. For each initial and optimized system architecture, count the number of electric generators, electric motors, and batteries included. Additionally, count the number of redundant electric generator-motor connections.
4. Compare the distributions of system-level responses, component counts, and redundancy levels. Verify that the distribution bounds are acceptably tight for all initial guesses.

In Experiment 4.0.3, 50 propulsion system architectures will be constructed using different seeds in the in the bi-level optimization framework’s random number generators, serving as initial guesses. Then, the optimization problem is solved for the same technology levels, top-level aircraft requirements, and system-level safety constraints with the different initial guesses generated. After solving all optimization problems, the system-level performance, component counts, and redundancy in the initial and optimized system architectures can be compared. If the heuristics embedded in the bi-level optimization framework are robust, then *near-equivalent* optimized propulsion system architectures will be generated regardless of the

initial guess input, suggesting that the optimized propulsion system architecture synthesized is weakly sensitive to the initial guess provided.

In the sensitivity study, it is not guaranteed that the *exact* same optimized system architecture topologies are output from the optimization. The randomness inherent to a stochastic optimizer may result in the same number of connections between similarly typed components, but not necessarily the exact same topology, resulting in *near-equivalent* ones instead. Due to the *near-equivalent* topologies returned, the optimized objective function values may vary slightly. Thus, part of the sensitivity study focuses on ensuring that the distributions of the objective function values are acceptably tight, confirming the robustness of the heuristics embedded within the bi-level optimization framework.

6.1.2 Chapter Outline

The remainder of the chapter is as follows. First, the bi-level optimization framework is constructed in Section 6.2. An extended design structure matrix of the bi-level optimization framework introduces its overall construction, followed by descriptions of the propulsion system architecture and power management optimizations in Sections 6.2.1 and 6.2.2, respectively. Other core capabilities in the bi-level optimization framework including the Fault Tree Analysis and problem-specific heuristics to select safety-feasible system architectures are described in Sections 6.2.4 and 6.2.5, respectively.

Then, Experiment 4.0.1 is executed in Section 6.3 to validate the bi-level optimization framework constructed and benchmark it against a fixed system architecture optimization approach. After validating the optimization framework, Experiment 4.0.2 benchmarks the bi-level optimization framework against state-of-the-art optimization approaches with generalized heuristics in Section 6.4. The bi-level optimization framework’s robustness is then assessed in Section 6.5 by executing Experiment 4.0.3. Lastly, Section 6.6 concludes the chapter, summarizes the key contributions, and reviews the findings from all experiments.

6.2 Bi-Level Optimization Framework

The bi-level optimization framework is constructed according to the high-level extended design structure matrix illustrated in Fig. 6.1. Two optimization problems, the PROPULSION ARCHITECTURE OPTIMIZATION and POWER MANAGEMENT OPTIMIZATION (colored orange in Fig. 6.1), are solved hierarchically to minimize the aircraft’s fuel burn; alternative system-level responses such as maximum takeoff weight, total energy expenditure, environmental

impact, etc. may also serve as objective functions, but were not utilized in the present implementation.

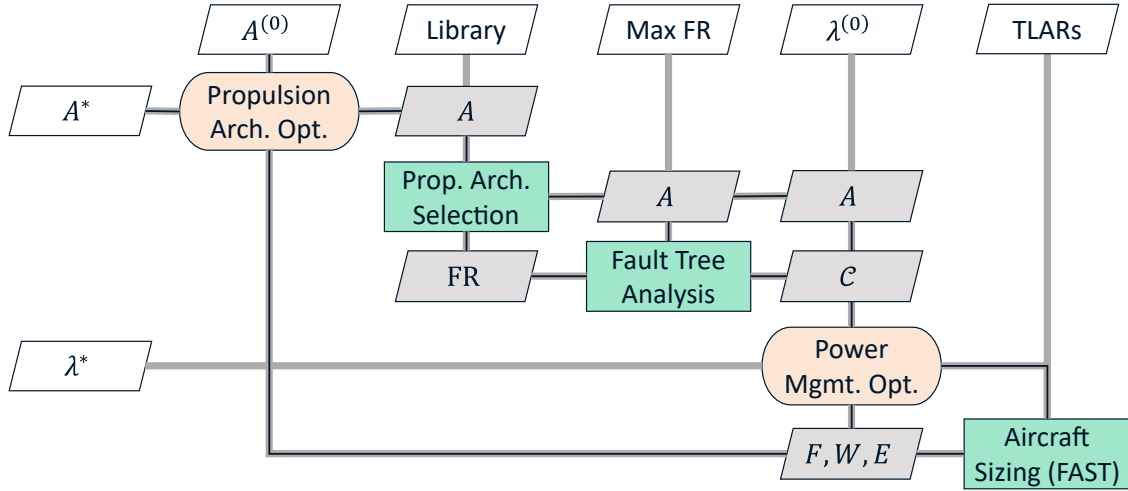


Figure 6.1: Extended design structure matrix illustrating the bi-level optimization framework's overarching workflow.

The bi-level framework's computational procedure is as follows. First, the PROPULSION ARCHITECTURE OPTIMIZATION problem commences with a randomly generated initial system architecture or one provided by the user, indicated by $A^{(0)}$. In subsequent iterations, the PROPULSION ARCHITECTURE SELECTION module utilizes heuristics to select the next candidate propulsion system architecture, incorporating information about the prior propulsion system architectures and their corresponding objective function values while satisfying both architectural and user-prescribed constraints. New system architectures and corresponding connections are selected based on the components available within a "Component Library" (indicated by "Library" in Fig. 6.1), containing lists of: 1) available components, 2) their inputs/outputs, 3) minimum/maximum allowable quantities, 4) nominal failure rates (per hour), and 5) overall efficiency.

After a propulsion system architecture is selected and meets all architectural and user-prescribed constraints, a FAULT TREE ANALYSIS identifies the system-level failure intensity and corresponding failure modes. Each propulsion system architecture's system-level failure rate must be less than or equal to the maximum allowable failure rate prescribed by the user (indicated by "Max. FR" in Fig. 6.1). Candidate propulsion system architectures that do not satisfy the aforementioned constraint are iteratively redesigned in the PROPULSION ARCHITECTURE SELECTION module by either adding more components or (additional)

redundant connections before the FAULT TREE ANALYSIS is executed again.

After a safety-feasible propulsion system architecture is identified, the configuration is sent to the POWER MANAGEMENT OPTIMIZATION, along with an initial guess for the optimum power management strategy, $\lambda^{(0)}$. During the POWER MANAGEMENT OPTIMIZATION, the candidate propulsion system architecture remains *fixed* and the power splits are varied by the optimizer. The objective function is computed by analyzing the aircraft in FAST, predicting the system-level responses for the given propulsion system architecture and power management strategy. After converging on an optimum power management strategy, the system-level response is passed back to the PROPULSION ARCHITECTURE OPTIMIZATION to inform the selection of the next propulsion system architecture. The process repeats iteratively until convergence.

In the remainder of this section, detailed descriptions of the optimization problem formulations and solvers utilized in the bi-level framework are introduced in Sections 6.2.1 and 6.2.2. Then, the FAULT TREE ANALYSIS and PROPULSION ARCHITECTURE SELECTION modules are introduced in Sections 6.2.4 and 6.2.5, respectively.

6.2.1 Propulsion Architecture Optimization Problem and Solver

6.2.1.1 Problem Formulation

The PROPULSION ARCHITECTURE OPTIMIZATION represents the high-level problem being solved in the bi-level optimization framework, identifying safety-feasible system architecture topologies to be analyzed further in the POWER MANAGEMENT OPTIMIZATION, and selecting the best one based on the estimated system-level responses. The optimization problem is defined in Eq. 6.1; the fuel burn (F) is minimized by varying the system architecture topology (A) and power management strategy (λ). Since the bi-level optimization framework solves a hierarchical problem, the PROPULSION ARCHITECTURE OPTIMIZATION problem contains the POWER MANAGEMENT OPTIMIZATION within it, optimizing the power management strategy for each safety-feasible system architecture assembled.

$$\begin{aligned}
& \text{MINIMIZE: } F(A) \\
& \text{WITH RESPECT TO: } A \\
& \text{SUBJECT TO: } \Lambda_{\text{fail,sys}}(A) \leq \Lambda_{\text{fail,max}} \quad (\text{System Safety}) \\
& \quad \begin{cases} 1 \leq \sum A_{i:} & \text{if } \sum A_{i:} > 0 \\ \sum A_{i:} = 0 & \text{if } \sum A_{i:} = 0 \end{cases} \quad (\text{Transmitter I/O}) \\
& \quad \sum L(A) = 0 \quad (\text{No Loops}) \\
& \quad \mathbf{g}(A) \leq 0 \quad (\text{User-Imposed Rules}) \\
& \text{WHILE MINIMIZING } F(\lambda; A) \\
& \text{FOR } \lambda
\end{aligned} \tag{6.1}$$

The constraints in the PROPULSION ARCHITECTURE OPTIMIZATION problem are focused on synthesizing safety-feasible system architectures. The first constraint, labeled “System-Safety”, requires that the system-level failure rate ($\Lambda_{\text{fail,sys}}$) must be less than or equal to the maximum allowable one ($\Lambda_{\text{fail,max}}$), and is evaluated in the Fault Tree Analysis, as depicted in Fig. 6.1.

The second and third constraints ensure that the propulsion system architecture is physically tractable and can be analyzed in the POWER MANAGEMENT OPTIMIZATION. The “Transmitter I/O” constraint ensures that all transmitters included in the propulsion system architecture have an appropriate number of inputs/outputs; see Chapter 3 for the definition of a transmitter. If a given component receives power, as indicated by $\sum A_{i:} > 0$, then it must output that power to at least one other component, as indicated by the inequality. Conversely, if the component receives no power from any components, as indicated by $\sum A_{i:} = 0$, then it cannot provide power to the other components in the system architecture, as indicated by $\sum A_{i:} = 0$. The “No Loops” constraint ensures that the propulsion system remains as a directed acyclic graph, a concept previously introduced in Chapter 3. The propulsion system architecture must be structured in such a form to ensure that the power flow algorithm introduced in Section 3.2.4 converges and that energy is conserved.

Lastly, the fourth constraint, labeled “User-Imposed Rules”, enables the user to prescribe any additional architectural constraints desired, as indicated by $\mathbf{g}(A)$. An example of a user-imposed constraint may require that similarly typed components have the same number of input/output connections.

6.2.1.2 Solver Employed

Since the PROPULSION ARCHITECTURE OPTIMIZATION problem is governed by binary variables, a gradient-free optimization technique is required to perform a global search of the design space. To solve the binary (or discrete) optimization problem, a custom-written Simulated Annealing algorithm was implemented in MATLAB. Rather than evaluating multiple candidate system architectures like a Genetic Algorithm, the Simulated Annealing algorithm utilizes a NEIGHBOR FUNCTION to select a successive candidate design, and has the ability to accommodate constraints by embedding logic that rejects infeasible designs [65]. Therefore, the NEIGHBOR FUNCTION provides capabilities to assemble safety-feasible system architectures from a component library, filtering out infeasible system architectures better than a Genetic Algorithm that naively evaluates any candidate system architecture and penalizes infeasible ones.

The Simulated Annealing Algorithm mimics the annealing process in metals; as the metal cools, the atoms are arranged in a minimum-energy state [65]. In the context of design optimization, the objective function acts as the “energy state,” and is sought to be minimized. The Simulated Annealing Algorithm progresses between iterations in one of two ways: 1) “cooling”; or 2) “quenching”.

During cooling, an improved objective function and corresponding set of design variables has been identified, and is accepted as the new global optimum. In a mathematical context, $f^* = f$ if $f \leq f^*$. Throughout the cooling process, the optimizer continues to progress towards the global optimum.

During quenching, a poorer objective function value ($f > f^*$) is accepted if the probability of jumping to a higher energy state is acceptably high, providing randomness to search different regions of the design space, thus accommodating multi-modal objective functions. The probability of jumping to a higher energy state is computed using Eq. 6.2, where T is the “temperature” of the metal [65].

$$P = \exp(-[f - f^*]/T) \quad (6.2)$$

The temperature of the metal decays throughout the simulation by a decay rate, α [65]. Both the initial temperature of the metal, T_0 , and the decay rate, α , are hyperparameters selected by the user, and depend on the problem being solved.

The NEIGHBOR FUNCTION utilized in the Simulated Annealing Algorithm is a user-developed heuristic that selects a new candidate set of design variables to analyze in search

of the best solution. The heuristics embedded in the NEIGHBOR FUNCTION are reviewed in Section 6.2.5, and are embedded in the PROPULSION ARCHITECTURE SELECTION block in Fig. 6.1.

6.2.2 Power Management Optimization Problem and Solver

6.2.2.1 Problem Formulation

The POWER MANAGEMENT OPTIMIZATION represents the low-level problem in the bi-level optimization framework, varying the power management strategy and corresponding component sizes for a *fixed* propulsion system architecture. As defined in Eq. 6.3, the optimizer minimizes the fuel burn, F , with respect to the power splits, λ .

$$\begin{aligned}
&\text{MINIMIZE: } F = f(\lambda; A) \\
&\text{WITH RESPECT TO: } \lambda \\
&\text{SUBJECT TO: } 0 \leq \lambda_{ij} \leq A_{ij} \quad (\text{Power Split Bounds}) \\
&\quad \sum_{j=1}^n \lambda_{ij} = 1 \quad (\text{Conservation of Energy}) \\
&\quad P_s(t) \geq 0 \quad (\text{Nonnegative Specific Excess Power}) \\
&\quad \gamma(t) \geq \hat{\gamma}(t) \quad (\text{Off-Nominal Climb Performance}) \\
&\quad m_c \geq 0 \quad (\text{Nonnegative Component Masses}) \\
&\quad \mathbf{g}(\lambda) \leq 0 \quad (\text{User-Imposed Constraints})
\end{aligned} \tag{6.3}$$

There are five constraints embedded in the POWER MANAGEMENT OPTIMIZATION to ensure that physically tractable power management strategies are identified and enable safe flight during off-nominal operational scenarios. The first constraint, labeled “Power Split Bounds,” requires that each power split is no larger than the corresponding architectural variable ascribed to the corresponding connection. If $A_{ij} = 1$, then the connection between components is active and may either partially or fully transmit the component’s available power. If $A_{ij} = 0$, then the connection is inactive and no power can be transmitted, resulting in an equality constraint requiring $\lambda_{ij} = 0$.

The second constraint, labeled “Conservation of Energy,” requires all power splits for an active source or transmitter to add to 1, ensuring that all its power is transmitted to other components in the propulsion system. Sinks are excluded from the constraint, as all their power is expended to perform a task such as propelling the aircraft or powering an onboard subsystem.

The “Nonnegative Specific Excess Power” constraint ensures that the aircraft has enough power to fly during nominal operations. In contrast, the “Off-Nominal Climb Performance” constraint ensures that the aircraft has enough power to climb during a propulsion system failure. All primary propulsion system failure modes identified during the FAULT TREE ANALYSIS are simulated to find the one with the most severe failure. Then, the generalized flight performance requirements developed in Chapter 5 are utilized to compute the required climb gradient for a given propulsion system failure severity. The maximum achievable climb gradients from the off-nominal flight simulation are compared to the required climb gradient, confirming whether or not the constraint has been satisfied. Lastly, the “Nonnegative Component Masses” constraint ensures that all sized components have nonnegative masses, guaranteeing that the optimizer returns a physically tractable solution.

Beyond the five constraints embedded in the optimization problem, the user may include additional constraints. Examples of such constraints include: 1) ensuring that all power flow paths between similarly typed components have equal power splits, guaranteeing that the components are sized identically; or 2) sizing the propulsors to ensure that a critical yaw moment is not exceeded during a propulsion system failure.

6.2.2.2 Solver Employed

In the POWER MANAGEMENT OPTIMIZATION, the power splits are governed by continuous variables, enabling gradient-based optimization techniques to be used. The optimization problem is solved using the embedded constrained optimizer in MATLAB, FMINCON, and solves a Sequential Quadratic Programming (SQP) problem. The SQP solver performs a local search of the design space using gradient-based optimization techniques.

6.2.3 Convergence to an Optimum

In the bi-level optimization framework, convergence to the global optimum is not guaranteed due to its hierarchical structure and solvers implemented. First, consider the hierarchical structure of the bi-level optimization framework. To select the best propulsion system architecture and corresponding power management strategy, two optimization problems are solved sequentially; the propulsion system architecture is selected and then the POWER MANAGEMENT OPTIMIZATION problem is solved for the *fixed* system architecture. The sequential/iterative optimization does not guarantee that the globally-optimal solution is found. If multiple optimization problems are solved sequentially, they must be unconstrained

to attain the globally-optimal solution [65]. In the bi-level optimization framework, there are constraints present in both the PROPULSION ARCHITECTURE OPTIMIZATION and POWER MANAGEMENT OPTIMIZATION problems, possibly preventing convergence to the true global optimum.

Second, consider the solvers utilized in the bi-level optimization framework. The Simulated Annealing optimizer used to solve the PROPULSION ARCHITECTURE OPTIMIZATION problem is a stochastic optimization method that approximates a full combinatorial search of the design space [65]. Although the design space is strategically searched with the embedded heuristics, there is no mathematical guarantee that the global optimum is reached. Additionally, the probability that the Simulated Annealing optimizer converges to the global optimum only approaches 1 if it is run for an infinitely long time and that the cooling schedule (impacted by the initial temperature and decay rate) decreases logarithmically [195]. Since the Simulated Annealing optimizer is run for a fixed number of iterations, there is no guarantee of convergence to the globally-optimal solution. Additionally, the SQP Method used to solve the POWER MANAGEMENT OPTIMIZATION problem is gradient-based, and can only perform a local search. Thus, there is no guarantee that the global optimum is reached unless the design space is proven to be convex.

Therefore, based on rationale presented, any optimized design is assumed to be a local optimum rather than a global one.

6.2.4 Fault Tree Analysis

To analyze system safety, a Fault Tree Analysis (FTA) was used to derive the system-level failure rate and minimal cut sets – combinations of the fewest component failures that can simultaneously occur to result in a system failure [196, 197] – within a system architecture. To execute the FTA, a system architecture must be provided along with component-level failure rates. This section describes the assumptions made for defining component-level failure rates and the FTA algorithm, along with a simple demonstration.

6.2.4.1 Component Failure Probability Estimation

A component's failure probability, P_{F_i} , describes the likelihood that the component fails during flight, and is derived from probability distributions governing component reliability and unreliability [197]. Component reliability, R_i (or probability of success, P_{S_i}), is assumed to be a constant-intensity exponential distribution, provided in Eq. 6.4, maintaining consistency with

practices established in SAE ARP4761A standards [196]. Conversely, component unreliability, Q_i (or failure probability, P_{F_i}), is the probability that the component is not reliable, as listed in Eq. 6.5. Component reliability and unreliability are complementary and mutually exclusive events ($R_i + Q_i = 1$), allowing Q_i to be defined as the complement of R_i .

$$R_i = P_{S_i} = e^{-\Lambda_i T} \quad (6.4)$$

$$Q_i = P_{F_i} = 1 - e^{-\Lambda_i T} \quad (6.5)$$

In Eqs. 6.4 and 6.5, Λ_i is defined as the component's nominal failure rate, and is traditionally published by its manufacturer in units of failures per hour. T is defined as the exposure time, or total time that the component is operated during a given flight. Multiplying the component's nominal failure rate by the exposure time returns the component's failure probability per flight. Although this step may seem trivial, it is necessary when considering components that may not operate during the entire flight. For example, in a parallel hybrid propulsion system architecture, the electric motor may only operate during takeoff and/or climb to boost the gas turbine engine power [18]; as a result of operating for a shorter time, its failure probability per flight is less than that of the gas turbine engine.

In this work, each component failure is assumed to be a rare event, implying that $\Lambda_i T \ll 1$, and enables the failure probability to be approximated by linearizing Eq. 6.5 about $\Lambda_i T = 0$. The resulting approximation yields Eq. 6.6, and estimates the component-level failure probabilities that are input into the FTA.

$$Q_i = P_{F_i} \approx \Lambda_i T \quad (6.6)$$

6.2.4.2 Fault Tree Analysis

In a FTA, minimal cut sets, $\mathcal{C}_j$, are obtained by traversing a graph of the propulsion system architecture (developed using the GPSA Framework introduced in Chapter 3) to identify all combinations of component failures that will result in a system-level failure occurring, known as failure modes. Three key assumptions are made in the FTA: 1) all component failures are independent; 2) all propulsion system failure modes are independent; and 3) system-level failures are rare events. The assumptions enable the system failure probability per flight to be approximated by Eq. 6.7.

$$Q_{\text{sys}} \approx \sum_j \prod_{i \in \mathcal{C}_j} Q_i \quad (6.7)$$

To align with guidance provided in the FAA Advisory Circular 25.1309-1B [198], the FTA returns the failure probability per flight hour (Λ_{sys}) by dividing the system failure probability by the exposure time, as listed in Eq. 6.8. In this case, the exposure time represents the block time, or total time that the aircraft is operating between its gate-to-gate departure and arrival.

$$\Lambda_{\text{sys}} = Q_{\text{sys}}/T \quad (6.8)$$

During systems architecting, probabilistic system-level safety constraints were imposed on the system-level failure rate, Λ_{sys} , measured in probability per flight hours. Conceptually, the constraint guarantees that the likelihood any possible propulsion system failure mode occurring is infrequent, demonstrating airworthiness for the requisite regulatory bodies. In the analysis, the constraint is enforced in the PROPULSION ARCHITECTURE OPTIMIZATION problem, requiring that the system-level failure rate be less than or equal to a maximum allowable failure rate. Any propulsion system architecture with a system-level failure rate exceeding the maximum allowable one is rejected, requiring a new propulsion system architecture to be constructed.

An additional systems architecting constraint requires that there be no single point of failure in the propulsion system architecture. From a conceptual perspective, the constraint requires that all thrust in the propulsion system cannot be lost from a single component failure, possibly resulting in a catastrophic failure and loss of life. To ensure that the constraint is satisfied conceptually, the system architecture must include either redundant components or an alternative means to power the propulsion system. The constraint is enforced in a problem-specific manner within the PROPULSION ARCHITECTURE OPTIMIZATION problem, as imposed by developing a custom (in)equality constraint that can be embedded in $\mathbf{g}(A)$ within Eq. 6.1.

Fault Tree Analysis Algorithm The FTA algorithm developed to compute the system failure probability is defined in Alg. 4, and utilizes both the propulsion system architecture matrix, A , and the component-level failure probabilities, Q . The system-level failure probability per flight, Q_{sys} , is returned, along with the minimal cut sets identified during the analysis, $\mathcal{C}$.

Algorithm 4: Fault tree analysis.

input : A and Q
output : $Q_{\text{sys}}, \mathcal{C}$

- 1 $\mathcal{C} \leftarrow \emptyset$
- 2 $j \leftarrow i : \sum_{k=1}^n A_{ik} = 0 \cap \sum_{k=1}^n A_{ki} > 0$ $\triangleright$ Find all components that are sinks
- 3 **for** j **do**
- 4 $\mathcal{C} \leftarrow \mathcal{C} \cup \text{Minimal Cut Sets}(A, Q, j)$ $\triangleright$ via Alg. 5
- 5 **end**
- 6 $\mathcal{C} \leftarrow \text{Idempotent Law}(\mathcal{C})$
- 7 $\mathcal{C} \leftarrow \text{Law of Absorption}(\mathcal{C})$
- 8 $Q_{\text{sys}} \leftarrow \sum_j \prod_{i \in \mathcal{C}_j} Q_i$

First, the minimal cut set is initialized to be empty, as no failure modes have been identified yet. Then, all sinks in the propulsion system architecture are identified mathematically (as previously introduced in Eq. 3.5 from Chapter 3), providing an initial component list to start searching for minimal cut sets. For each sink, the propulsion system architecture is traversed recursively using Alg. 5 until the sources are reached, returning Boolean expressions to represent each minimal cut set that could lead to a system-level failure.

Since the set of enumerated failures is large and may contain duplicate events, the Boolean expressions are simplified using the Idempotent Law and Law of Absorption [199], listed in Eqs. 6.9 and 6.10, respectively.

$$\text{Idempotent Law: } X \cap X = X \tag{6.9}$$

$$X \cup X = X$$

$$\text{Law of Absorption: } X \cap (X \cup Y) = X \tag{6.10}$$

$$X \cup (X \cap Y) = X$$

After simplifying the Boolean expressions, they are converted into mathematical expressions to compute the failure probability per flight. Any Boolean **AND**, represented by $\cap$, is replaced with scalar multiplication. Conversely, any Boolean **OR**, represented by $\cup$, is replaced with scalar addition. Then, the component failure probabilities are substituted into the Boolean expression, allowing the failure probability per flight, Q_{sys} , to be attained. The three assumptions previously described (independent component failures, independent failure modes, and that failures are rare events) allow the system-level failure probability per flight to be approximated by the sum of the failure probabilities across all minimal cut sets. Then,

the failure probability per flight is converted to a failure probability per hour using Eq. 6.8, with an exposure time corresponding to the design mission's block time.

Identifying Minimal Cut Sets The algorithm for identifying the minimal cut sets is introduced in Alg. 5, and utilizes the propulsion system architecture matrix, A , component-level failure probabilities, Q , and an initial component index to start searching at, i . Algorithm 5 searches for both “internal failures,” or failures inherent to the component currently being searched, and “downstream failures,” or failures that result from other component failures in the propulsion system. A Boolean expression for all minimal cut sets identified, $\mathcal{C}$, is returned.

Algorithm 5: Minimal cut set identification.

```

input :  $A, Q, i$ 
output :  $\mathcal{C}$ 
1 if  $Q_i > 0$  then  $\hat{\mathcal{C}} \leftarrow i$                                  $\triangleright$  The component has its own failure mode
2 else  $\hat{\mathcal{C}} \leftarrow \emptyset$                                      $\triangleright$  The component does not have its own failure mode
3  $\tilde{\mathcal{C}} \leftarrow \emptyset$                                         $\triangleright$  Assume there are no downstream failures
4  $j \leftarrow k : A_{ki} > 0$                                       $\triangleright$  Find all downstream component connections
5 if  $j \notin \emptyset$  then
6   for  $j$  do
7      $\tilde{\mathcal{C}}_j \leftarrow \text{Minimal Cut Sets}(A, Q, j)$             $\triangleright$  Recursively call Alg. 5
8   end
9    $K \leftarrow A_{ji}$                                             $\triangleright$  Entry indicates the number of components allowed to fail
10   $N \leftarrow \text{size}(j)$                                         $\triangleright$  Get the total number of elements considered
11   $\tilde{\mathcal{C}} \leftarrow K\text{-out-of-}N \text{ enumeration } (\tilde{\mathcal{C}})$             $\triangleright$  Enumerate all possible downstream faults
12   $\tilde{\mathcal{C}} \leftarrow \text{Idempotent Law } (\tilde{\mathcal{C}})$ 
13   $\tilde{\mathcal{C}} \leftarrow \text{Law of Absorption } (\tilde{\mathcal{C}})$ 
14 end
15  $\mathcal{C} \leftarrow \tilde{\mathcal{C}} \cup \hat{\mathcal{C}}$                                     $\triangleright$  Combine internal and downstream faults

```

First, Alg. 5 remembers the internal failure mode, $\hat{\mathcal{C}}$ if the component-level failure probability is greater than zero. Then, all downstream component connections are identified by querying the respective column in the architecture matrix. Assuming that at least one downstream connection exists, the minimal cut sets corresponding to the downstream failure modes, $\tilde{\mathcal{C}}$, are recursively identified for each component. In a Graph Theory context, the recursive search is analogous to a “Depth First Search,” visiting any adjacent vertex before

backtracking to the prior vertex analyzed.

After the downstream failure modes are identified, they are passed through a K -out-of- N gate, enumerating all combinations of K component failures, and are stored as minimal cut sets. The K -out-of- N gate indicates that the failure of K components causes Component i to fail even though N components are connected to it [197]. The number of components triggering a failure at Component i is defined by storing K in the i^{th} column of the architecture matrix. Therefore, the architecture matrix becomes a *weighted adjacency matrix* for the purposes of the FTA, deviating from its prior abstraction as an *adjacency matrix* previously introduced in Chapter 3.

After enumerating across the K -out-of- N gate, the minimal cut sets are again simplified utilizing the Idempotent Law and Law of Absorption, as previously described in Eqs. 6.9 and 6.10. Both the minimal cut sets corresponding to the internal and downstream failure modes are returned for further analysis.

All common-mode failures are also returned as part of the minimal cut sets. The FTA recursively searches the propulsion system starting at each sink and terminating once a source is reached. Therefore, if a given component is connected to multiple others, its failure mode will be recorded multiple times and simplified as a common-mode failure while applying the Law of Absorption in Alg. 5. For example, in a distributed propulsion system, a battery fire would be captured in the FTA as a common-mode failure, resulting in the failure of any propulsor that is connected to it if no backup battery exists.

Defining Variables for the K -out-of- N Gate To compute the minimal cut sets for the downstream failure modes, they must be passed through a K -out-of- N gate, requiring both K and N to be defined. In Alg. 5, N is defined as the total number of downstream connections to Component i , indicated by the number of elements in j .

However, K is defined differently depending on whether the component is a transmitter or a sink. For all transmitters, defining K is automated in the bi-level optimization framework by assuming that a transmitter will fail if *all* of its downstream components fail, requiring that $A_{:j} = N$. This assumption transforms the corresponding K -out-of- N gates to N -out-of- N gates, making them equivalent to an AND gate [197].

For sinks, K is defined by the user, indicating the number of downstream components that can fail before a system-level failure occurs. During the optimization problem setup, the user prescribes the number of components that may fail for each sink in the propulsion system architecture, enabling them to explore different system-level failure modes; the K -out-of- N

gate is not simplified to an AND gate automatically. The flexibility is essential for synthesizing configurations with distributed propulsion system architectures, as multiple failure modes may be considered during conceptual design. Such considerations were previously introduced in Chapter 5 while sizing the notional Elysian E9X for a variety of primary propulsion system failure modes.

Coupling the Safety and Performance Analyses The failure modes from the FTA are coupled to the POWER MANAGEMENT OPTIMIZATION, ensuring that the aircraft has sufficient power to satisfy a given off-nominal climb gradient requirement. To simulate a failure in the propulsion system, all upstream/downstream power splits corresponding to the failed components are set to zero, ensuring that power is not transferred in either direction. Mathematically, for a given minimal cut set, $\mathcal{C}_k$:

$$\begin{cases} \lambda_{i:}^u = 0 \text{ and } \lambda_{:i}^d = 0, \text{ if } i \in \mathcal{C}_k \\ \lambda_{i:}^u \text{ and } \lambda_{:i}^d \text{ remain unchanged, if } i \notin \mathcal{C}_k \end{cases} \quad (6.11)$$

After simulating the resultant propulsion system failure mode, the specific excess power loss fraction is estimated, enabling a required climb gradient to be computed from the generalized climb performance requirements introduced in Chapter 5. The required climb gradient is then compared to those attained while simulating the propulsion system failure mode, confirming if $\gamma(t) \geq \hat{\gamma}(t)$, as previously listed in Eq. 6.3 (the POWER MANAGEMENT OPTIMIZATION problem).

6.2.4.3 Fault Tree Analysis Demonstration

To demonstrate the algorithm, a simple four-component system is used, as illustrated in Fig. 6.2. The architecture matrix representing the configuration is listed in Eq. 6.12.

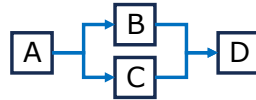


Figure 6.2: Example system architecture for FTA algorithm.

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (6.12)$$

The architecture matrix is now abstracted as a *weighted architecture matrix*, containing integer values between 1 and 2, indicating the number number of components needed to fail before triggering the K -out-of- N gate. The columns corresponding to Components B and C contain a 1, indicating that either component fails if its one downstream component (Component A) fails. However, the column corresponding to Component D contains a 2, indicating that both downstream components (Components B and C) must fail to trigger the K -out-of- N gate.

In the analysis, assume all components failure rates, Λ_i , are 10^{-2} per flight hour. Additionally, the exposure time, T , is assumed to be one hour, making it easier to trace each component's failure to the overall system-level failure probability.

The FTA begins at Component D, the most upstream component, and its failure mode is remembered. Then, the architecture matrix is checked to find the components that flow into Component D, namely, Components B and C.

Component D will also fail if both Components B and C fail. Thus, the system-level failure probability is:

$$P(\text{fail}) = P(\text{D fail}) \cup [P(\text{B fail}) \cap P(\text{C fail})]$$

Next, Components B and C are searched recursively. For both Components B and C, their only input is Component A. Component B and C will fail either individually or if Component A fails. Thus, the probability that Components B and C fail are:

$$P(\text{B fail}) = P(\text{B fail}) \cup P(\text{A fail})$$

$$P(\text{C fail}) = P(\text{C fail}) \cup P(\text{A fail})$$

These failure modes cannot be simplified using the Idempotent Law or Law of Absorption, so they are returned as written. Then, they are input into the system-level failure probability expression:

$$\begin{aligned}
P(\text{fail}) = & P(\text{D fail}) \cup \\
& \{ [P(\text{B fail}) \cup P(\text{A fail})] \cap \\
& [P(\text{C fail}) \cup P(\text{A fail})] \}
\end{aligned}$$

Next, the failures are enumerated by expanding the AND operators ($\cap$) in the Boolean expression:

$$\begin{aligned}
P(\text{fail}) = & P(\text{D fail}) \cup \\
& [P(\text{B fail}) \cap P(\text{C fail})] \cup \\
& [P(\text{B fail}) \cap P(\text{A fail})] \cup \\
& [P(\text{C fail}) \cap P(\text{A fail})] \cup \\
& [P(\text{A fail}) \cap P(\text{A fail})]
\end{aligned}$$

Using the Idempotent Law, any duplicated event is reduced from the Boolean expression. In this example, the failure of Component A is duplicated in the last term. The updated Boolean expression becomes:

$$\begin{aligned}
P(\text{fail}) = & P(\text{D fail}) \cup \\
& [P(\text{B fail}) \cap P(\text{C fail})] \cup \\
& [P(\text{B fail}) \cap P(\text{A fail})] \cup \\
& [P(\text{C fail}) \cap P(\text{A fail})] \cup \\
& P(\text{A fail})
\end{aligned}$$

Then, the Law of Absorption is used to eliminate redundant events. In this example, the failure of Component A also exists in the failure of Components B/A and Components C/A. After reducing, the Boolean expression is:

$$\begin{aligned}
P(\text{fail}) = & P(\text{D fail}) \cup \\
& [P(\text{B fail}) \cap P(\text{C fail})] \cup \\
& P(\text{A fail})
\end{aligned}$$

Lastly, the component-level failure probabilities are substituted into the fully reduced Boolean expression:

$$\begin{aligned}
P(\text{fail}) &= 10^{-2} + (10^{-2} \times 10^{-2}) + 10^{-2} \\
&= 2.01 \times 10^{-2}
\end{aligned}$$

Evaluating the expression yields a system-level failure probability of 2.01×10^{-2} . The result is dominated by single-point failures of Components A or D (each with a failure probability of 10^{-2}). The redundant connections provided by Components B and C contributes 10^{-4} to the system-level failure probability.

Since the exposure time is one hour, the system-level failure intensity is equal to the probability of failure. This is merely a coincidence and a result of selecting a convenient exposure time for the example.

6.2.5 Propulsion Architecture Selection

The PROPULSION ARCHITECTURE SELECTION module selects a new propulsion system architecture to analyze within the PROPULSION ARCHITECTURE OPTIMIZER and serves as the NEIGHBOR FUNCTION within the Simulated Annealing Algorithm introduced in Section 6.2.1. The NEIGHBOR FUNCTION is a heuristic-based function, utilizing commonsense arguments to select the next set of design variables [65].

In the bi-level optimization framework, the NEIGHBOR FUNCTION considers both the objective function value (fuel burn, maximum takeoff weight, energy expenditure, etc.), f , and the number “active components” (components present in the current system architecture), N_{act} , to select the next propulsion system architecture. The following heuristics are utilized to determine whether components/connections should be added/removed from the propulsion system between successive iterations:

1. If $f^{(i)} \leq f^{(i+1)}$ and $N_{\text{act}}^{(i)} \leq N_{\text{act}}^{(i+1)}$ or vice versa, then the objective function value decreases by utilizing fewer components in the propulsion system architecture, possibly favoring an aircraft concept with gas turbine engines that is boosted by electrical power (either in series or parallel). Remove components/connections in the propulsion system architecture as long as the system safety constraints are satisfied.
2. If $f^{(i)} \leq f^{(i+1)}$ and $N_{\text{act}}^{(i)} \geq N_{\text{act}}^{(i+1)}$ or vice versa, then the objective function value decreases by utilizing more components in the propulsion system architecture, possibly favoring an aircraft concept with distributed propulsion. Add components/connections in the propulsion system architecture as long as they are available (i.e., not all are active).
3. Otherwise, randomly decide whether to add/remove components from the system architecture.

Regardless of the propulsion system architecture that is favored by the aforementioned heuristics, redundancy is assumed to limit system performance (requiring more fuel, increasing MTOW, increasing the total energy expenditure, etc.) and should only be included to satisfy the system safety constraints. However, redundant configurations must be explored to better inform the heuristics and identify system architectures that both perform the best and satisfy the system-level safety constraints. Therefore, the level of redundancy in the propulsion system architecture is selected by randomly choosing a number of redundant connections between 1 and the minimum number of redundant connections in the last two system architectures analyzed.

Algorithm 6 describes how the NEIGHBOR FUNCTION selects a candidate propulsion system architecture to be analyzed in the next major iteration of the Propulsion Architecture Optimization. The heuristics within the NEIGHBOR FUNCTION rely on the propulsion system architectures and objective function values from the previous two iterations. As a consequence, random changes are made to the system architecture after the first iteration, and the NEIGHBOR FUNCTION is utilized to select the next candidate propulsion system architecture for the remaining iterations.

Algorithm 6: Neighbor function for selecting a new propulsion system architecture.

input : $i, f^{(i)}, f^{(i-1)}, A^{(i)}, A^{(i-1)}, \Lambda_{\max}, Q$
output : $A^{(i+1)}, \Lambda^{(i+1)}, \mathcal{C}^{(i+1)}$

- 1 $\text{Iter} \leftarrow 0$
- 2 $\Lambda_{\text{sys}} \leftarrow 1$
- 3 **while** $\Lambda_{\text{sys}} > \Lambda_{\max} \cap \text{Iter} < \text{Max. Iter}$ **do**
- 4 $N_{\text{act}}^{(i-1)}, N_{\text{act}}^{(i)} \leftarrow$ Active Components in $A^{(i-1)}$ and $A^{(i)}$, respectively
- 5 $R_{\text{EG}}^{(i-1)}, R_{\text{EG}}^{(i)} \leftarrow$
 Number of EMs each EG connects to in $A^{(i-1)}$ and $A^{(i)}$, respectively
- 6 $R_{\text{EM}}^{(i-1)}, R_{\text{EM}}^{(i)} \leftarrow$
 Number of EGs each EM connects to in $A^{(i-1)}$ and $A^{(i)}$, respectively
- 7 $R_{\text{EG}}^* \leftarrow \min \left\{ R_{\text{EG}}^{(i-1)}, R_{\text{EG}}^{(i)} \right\}$
- 8 $R_{\text{EM}}^* \leftarrow \min \left\{ R_{\text{EM}}^{(i-1)}, R_{\text{EM}}^{(i)} \right\}$
- 9 $\overline{\text{EG}} \leftarrow 1 + \text{round} \left(1 + (R_{\text{EG}}^* - 1) \times \mathcal{U}(0, 1) \right)$
- 10 $\overline{\text{EM}} \leftarrow 1 + \text{round} \left(1 + (R_{\text{EM}}^* - 1) \times \mathcal{U}(0, 1) \right)$
- 11 $\Delta \leftarrow 0$
- 12 **if** $\left(f^{(i)} \leq f^{(i-1)} \cap N_{\text{act}}^{(i)} \geq N_{\text{act}}^{(i-1)} \right) \cup \left(f^{(i-1)} \leq f^{(i)} \cap N_{\text{act}}^{(i-1)} \geq N_{\text{act}}^{(i)} \right)$ **then**
- 13 | $\Delta \leftarrow +1$
- 14 **end**
- 15 **if** $\left(f^{(i)} \leq f^{(i-1)} \cap N_{\text{act}}^{(i)} \leq N_{\text{act}}^{(i-1)} \right) \cup \left(f^{(i-1)} \leq f^{(i)} \cap N_{\text{act}}^{(i-1)} \leq N_{\text{act}}^{(i)} \right)$ **then**
- 16 | $\Delta \leftarrow -1$
- 17 **end**
- 18 **if** $\Delta = 0$ **then**
- 19 | $r \leftarrow \mathcal{U}(0, 1)$
- 20 | **if** $r > 0.5$ **then** $\Delta = +1$
- 21 | **else** $\Delta = -1$
- 22 | $\overline{\text{EG}}, \overline{\text{EM}} \leftarrow 0$
- 23 **end**
- 24 $A^{(i+1)} \leftarrow \text{CHANGETOPOLOGY} \left(A^{(i)}, \Delta, \overline{\text{EG}}, \overline{\text{EM}}, g \right)$
- 25 $A^{(i+1)} \leftarrow \text{REROUTECABLES} \left(A^{(i+1)} \right)$
- 26 $\Lambda_{\text{sys}}, \mathcal{C} \leftarrow \text{FAULTTREEANALYSIS} \left(A^{(i+1)}, Q \right)$
- 27 $\text{Iter} \leftarrow \text{Iter} + 1$
- 28 **end**

The NEIGHBOR FUNCTION modifies the prior propulsion system architecture iteratively until an iteration limit is reached or the system-level failure rate is less than or equal to the maximum allowable failure rate. Four key steps occur in each iteration:

1. **Select how the system architecture changes (Lines 4 through 23):** the previously described heuristic is utilized to determine if connections should be added or removed from the system architecture, as indicated by Δ . If the heuristics are inconclusive, then a uniformly-distributed random decimal number is selected between 0 and 1 (exclusive); connections are added/removed depending on the decimal value output from the random selection. Additionally, the number of redundant connections is selected based on the heuristic previously described (Lines 9 and 10).
2. **Modify the system architecture (Line 24):** the CHANGETOPOLOGY function, described in Section 6.2.5.1, randomly selects connections to be added/removed based on the remaining available ones. After adding/removing the connections, the system architecture is validated to ensure that it satisfies both the baseline constraints embedded in the PROPULSION ARCHITECTURE OPTIMIZATION, as well as any user-prescribed constraints. Section 6.2.5.2 describes the system architecture validation process in greater detail.
3. **Reroute any cables present (Line 25):** an additional stochastic optimization is performed to minimize the cable length in the propulsion system architecture by rerouting any cable connections between the electric generators and motors. The optimizer, described in Section 6.2.5.3 changes the electric generator-motor connection arrangement while ensuring that each electric generator/motor has the same number of inputs/outputs as the initial propulsion system architecture provided.
4. **Perform a system-safety analysis (Line 26):** a FAULT TREE ANALYSIS, previously described in Section 6.2.4, is executed to estimate the system-level failure rate and identify the minimal cut sets that could cause a system-level failure.

After performing the four key steps, the system-level failure rate is then compared to the maximum allowable system-level failure rate to check convergence. If the test fails and the maximum iteration limit has not been met, the system architecture is changed again until converging on a valid one that satisfies the prescribed system-level safety constraint.

6.2.5.1 System Architecture Modification

The system architecture is modified utilizing the `CHANGETOPOLOGY` function, previously introduced in Section 6.2.5, and adds/removes compatible component connections in the system architecture. Each time the function is called, only one connection is added/removed, but more might be added/removed during system architecture validation, ensuring that the architectural and user-prescribed constraints in the `PROPULSION ARCHITECTURE OPTIMIZATION` are satisfied. Algorithm 7 describes the `CHANGETOPOLOGY` function in greater detail, illustrating how random connections are added/removed from the system architecture.

Algorithm 7: `CHANGETOPOLOGY` function.

```

input  :  $A, \Delta, \overline{EG}, \overline{EM}, g$ 
output :  $A^*$ 
1  $n \leftarrow \sqrt{\text{size}(A)}$        $\triangleright$  Number of possible components in the system architecture
2  $P \leftarrow \emptyset$ 
3 while  $P = \emptyset \cap \text{Iter} \leq \text{Max. Iter}$  do
4    $i \leftarrow \text{round}((n-1) \times \mathcal{U}(0,1)) + 1$        $\triangleright$  Randomly select a component
5   if  $\Delta = +1$  then
6      $P \leftarrow j : C_{ij} = 1 \cap A_{ij} = 0$        $\triangleright$  Find possible connections to add
7   end
8   if  $\Delta = -1$  then
9      $P \leftarrow j : C_{ij} = 1 \cap A_{ij} = 1$        $\triangleright$  Find possible connections to remove
10  end
11   $\text{Iter} \leftarrow \text{Iter} + 1$ 
12 end
13 if  $P = \emptyset$  then
14    $A^* \leftarrow A$        $\triangleright$  No changes are made
15   exit
16 end
17  $m \leftarrow \text{size}(P)$ 
18  $k \leftarrow \text{round}((m-1) \times \mathcal{U}(0,1)) + 1$        $\triangleright$  Randomly select an available connection;
    $j \leftarrow P_k$        $\triangleright$  Get the index of the selected connection
19  $A_{ij} \leftarrow \sim A_{ij}$        $\triangleright$  Flip the bit to add/remove the connection
20  $A^* \leftarrow \text{VALIDATEARCHITECTURE}(A, \Delta, \overline{EG}, \overline{EM}, g)$ 

```

First, an iteration is performed to identify a component with possible connections, P , that can be added/removed from the system architecture (Lines 3-12). Components are selected randomly, enabling different regions of the design space to be searched without introducing bias into the system architecting process (Line 4). A ‘‘Compatibility Matrix,’’ C , is constructed

to aid in the identification process, and is based on each component's inputs/outputs, as stored in the Component Library (previously introduced in Section 6.2.1). The Compatibility Matrix is a square matrix of size n , where n is the number of possible components in the system architecture, indicating the permitted connections between components in the system architecture. Components with matching output-input types (such as mechanical power, electrical power, thrust power, etc.) are flagged with a 1, indicating their compatibility and ability to be connected together. For example, a gas turbine engine outputs mechanical power, enabling connections to both electric generators and propellers which require mechanical power to operate. Conversely, components with different output-input types are flagged with a 0, indicating that they should not be connected together. For example, a gas turbine engine cannot connect to an electric motor because the electric motor requires electrical power to be input.

If components are being added to the system architecture, compatible connections are found that *do not already exist* in the system architecture, requiring $C_{ij} = 1$ and $A_{ij} = 0$ (Lines 5-7). Conversely, if component are being removed from the system architecture, compatible connections are found that already exist in the system architecture, still requiring $C_{ij} = 1$ but $A_{ij} = 1$ as well (Lines 8-10). If P is empty, indicating that no compatible connections can be added/removed from the system architecture, another component is chosen randomly until P is non-empty. If the iteration limit is met and P remains empty, then no changes to the system architecture are made and the initial system architecture input to the algorithm is returned (Lines 13-16).

If P is non-empty, then one of the previously identified component connections is chosen at random and its bit is flipped in the architecture matrix (Lines 17-19). Bit flipping is a convenient method to change a 1 to a 0 or vice versa in the architecture matrix while remaining general enough to add or remove connections, thus reducing the computational complexity of the code. After changing the system architecture, it is validated using the `VALIDATEARCHITECTURE` function, introduced in Section 6.2.5.2, ensuring that the architectural and user-prescribed constraints in the `PROPULSION ARCHITECTURE OPTIMIZATION` are satisfied. The validated system architecture is then returned to the `PROPULSION ARCHITECTURE SELECTION` module for further analysis.

6.2.5.2 System Architecture Validation

After changing the system architecture, a validation process is performed to ensure that all architectural and user-prescribed constraints are satisfied. Within the `PROPULSION`

ARCHITECTURE OPTIMIZER, there are two constraints that all system architectures must satisfy. First, each component must have a valid number of input/output connections, ensuring that power can flow through the physical system. Since sources only output power and sinks only input power, the check is only necessary for all transmitters in the system, as they may be active (requiring at least one input and one output) or inactive (requiring no inputs or outputs). Second, there must not be any loops in the propulsion system architecture. In Chapter 3, the propulsion system architectures were required to be a directed acyclic graph to ensure that the energy conservation equations are obeyed and the power flow algorithm converges.

Algorithm 8 describes the steps taken to validate the system architecture and modify it to satisfy the two aforementioned constraints. For any user-defined system architecture constraints, $\mathbf{g}$, the user must prescribe the routine to modify the system architecture such that the additional constraints are satisfied.

In Alg. 8, sequential changes are made to the system architecture until both of the aforementioned constraints are satisfied. To ensure that the transmitters have the correct number of inputs and outputs, the row and column sums in the architecture matrix are checked. If the transmitter is active (has at least one input and one output) or inactive (has no inputs and outputs), the constraint is satisfied. However, if the transmitter has at least one input and no outputs or no inputs and at least one output, the system architecture must be modified by adding/removing connections until the component is either active or inactive. If connections were added to the system architecture before validating, more connections are added to ensure that the component violating the constraint becomes active. Conversely, if connections were removed from the system architecture before validating, connections are removed to ensure that the component violating the constraint becomes inactive.

To check that the propulsion system architecture contains no loops, MATLAB's depth-first search function, `DFSEARCH`, was utilized. One of the `DFSEARCH`'s capabilities includes the ability to identify edges that connect to previously discovered nodes from the depth-first search. Such edges cause the propulsion system architecture to be a *cyclic* graph, thus violating the constraint that no loops exist. To ensure that the propulsion system architecture remains as a *directed acyclic graph*, any edges identified from the `DFSEARCH` are removed.

Again, any user-prescribed constraints must be validated using separate routines provided by the user. The routines must return the architecture matrix and a flag indicating whether or not any changes to the system architecture were made.

Algorithm 8: VALIDATEARCHITECTURE function

Input : $A, \Delta, \overline{EG}, \overline{EM}, g$
Output : A^*

```

1   $n \leftarrow \sqrt{\text{size}(A)}$  ▷ Number of possible components in the system architecture
2   $\text{Changed} \leftarrow 1$  ▷ Assume the system architecture changed
3   $\text{Iter} \leftarrow 0$ 
4   $A^* \leftarrow A$ 
5  while  $\text{Changed} = 1 \cap \text{Iter} \leq \text{MaxIter}$  do
6     $\text{Changed} \leftarrow 0$ 
7     $\text{Iter} \leftarrow \text{Iter} + 1$ 
8    for  $i = 1 : n$  do
9       $N_{\text{in}} \leftarrow \sum_{k=1}^n A_{ki}$ 
10      $N_{\text{out}} \leftarrow \sum_{k=1}^n A_{ik}$ 
11     if  $N_{\text{in}} = 0 \cap N_{\text{out}} = 0$  then continue
12     if Component  $i$  is not a transmitter then continue
13     if  $(N_{\text{in}} = 0 \cap N_{\text{out}} > 0) \cup (N_{\text{in}} > 0 \cap N_{\text{out}} = 0)$  then
14        $\text{Changed} \leftarrow 1$ 
15        $P \leftarrow \emptyset$ 
16       if  $\Delta = +1$  then
17          $P \leftarrow \text{Repeat Lines 5-7 from Alg. 7}$ 
18       end
19       if  $\Delta = -1$  then
20          $P \leftarrow \text{Repeat Lines 8-10 from Alg. 7}$ 
21       end
22     end
23     if  $P = \emptyset$  then
24        $A^* \leftarrow \text{Repeat Lines 17-19 from Alg. 7}$ 
25     end
26     else
27        $A^* \leftarrow \text{CHANGETOPOLOGY}(A^*, \Delta, \overline{EG}, \overline{EM}, g)$ 
28     end
29   end
30    $L \leftarrow \{(i, j) : A^* \text{ contains a loop at } (i, j)\}$ 
31   if  $L \neq \emptyset$  then
32      $\text{Changed} \leftarrow 1$ 
33     for  $(i, j) \in L$  do
34        $A^*_{ij} \leftarrow 0$  ▷ Remove connections that caused any loops to exist
35     end
36   end
37    $(A^*, \text{Changed}) \leftarrow \text{Evaluate}(g(A^*))$ 
38 end
39 if  $\text{Iter} = \text{MaxIter}$  then
40    $r \leftarrow \mathcal{U}(0, 1)$ 
41   if  $r > 0.5$  then  $A^* \leftarrow \text{CHANGETOPOLOGY}(A^*, +1, \overline{EG}, \overline{EM}, g)$ 
42   else  $A^* \leftarrow \text{CHANGETOPOLOGY}(A^*, -1, \overline{EG}, \overline{EM}, g)$ 
43 end

```

If any changes are made to the propulsion system architecture during the validation process, the iteration continues as a result of the activated flag. Once no more changes are flagged, the iteration ends and the system architecture is fully validated.

6.2.5.3 Cable Routing Optimization

In the event that there are redundant connections between any electric generators and motors, an additional optimization is performed to minimize the total cable length, L_{cables} , by modifying the cable connection routes within the propulsion system architecture. Equation 6.13 defines the unconstrained optimization problem and is a function of the propulsion system architecture, A , and the individual cable lengths, l_{cables} .

$$\begin{aligned} \text{MINIMIZE: } & L_{\text{cables}} = f(A; l_{\text{cables}}) \\ \text{WITH RESPECT TO: } & A \end{aligned} \tag{6.13}$$

Since each cable connection is governed by a binary variable – the 1 or 0 in the architecture matrix – the optimization problems remains discrete and must be solved using a stochastic optimization algorithm [65]. The cable routing optimization problem is solved using the Simulated Annealing algorithm. The objective function computes the total cable length, and is dependent upon the propulsion system architecture and individual cable lengths between the electric generators and motors. The user must prescribe a function to return the individual cable lengths to compute the objective function value, as they are specific to the systems architecting problem being solved based on the Component Library provided.

The NEIGHBOR FUNCTION to reroute the cables is described in Alg. 9. The routine changes the electric generator-motor connection arrangement while ensuring that each electric generator and motor has the same number of inputs/outputs as the initial propulsion system architecture provided.

If two or more electric generators are in the propulsion system architecture, Alg. 9 randomly selects two of them. After selecting the electric generators, their corresponding rows in the architecture and compatibility matrices are checked to find all active component connections and additional possible components that the electric generator could connect to. Based on the number of active connections available, a random number of them are selected to be swapped, followed by attempts to make each swap. Two electric generator-motor connections are swapped if both electric generators are not currently connected to the proposed electric motors that they will connect to if the swap were made. Once a swap is made, either the next swap is attempted or the algorithm is terminated by returning the new system architecture.

Algorithm 9: Neighbor function for the cable routing optimization problem.

```

input :  $A, C$ 
output :  $A^*$ 
1  $n_{EG} \leftarrow$  Find the number of active electric generators
2 if  $n_{EG} > 1$  then
3    $r_1, r_2 \leftarrow$  Select two electric generators in the system architecture randomly
4    $i_1 \leftarrow i_1 : C_{r_1 i_1} = 1 \cap A_{r_1 i_1} = 1$ 
5    $i_2 \leftarrow i_2 : C_{r_2 i_2} = 1 \cap A_{r_2 i_2} = 1$ 
6    $n_{swap} \leftarrow [\min\{1, \text{size}(i_1)\} - 1] \times \mathcal{U}(0, 1) + 1 \triangleright$  Random number of connections to
      swap
7   for  $k = 1 : n_{swap}$  do
8     for  $Iter = 1 : \text{Max. Iter}$  do
9        $s_1, s_2 \leftarrow$  Select a random cable from  $i_1$  and  $i_2$ , respectively
10      if  $A_{i_1 s_2} = 0 \cap A_{i_2 s_1} = 0$  then
11         $A_{i_1 s_1} = 0$ 
12         $A_{i_2 s_2} = 0$ 
13         $A_{i_1 s_2} = 1$ 
14         $A_{i_2 s_1} = 1$ 
15        BREAK
16      end
17    end
18  end
19 end
20  $A^* \leftarrow A$ 

```

6.3 Optimization Framework Validation (Experiment 4.0.1)

The bi-level optimization framework constructed must be validated against a known aircraft concept to ensure that safety-feasible propulsion system architectures can be assembled from a predefined component library utilizing embedded, problem-specific heuristics. Additionally, the bi-level optimization framework must be able to capture tradeoffs between system-level performance (fuel burn) and safety (failure rate) for propulsion system architectures with varying levels of redundancy. A validation study is performed on NASA's SUSAN concept [7, 42], previously introduced in Chapter 1 while describing nascent advanced aircraft designs and in Chapter 3 while validating the GPSA Framework. Prior work on SUSAN investigated the system-level performance and safety tradeoffs on a distributed partially turboelectric propulsion system with redundant electric generator-motor cross-connections [174], and is expanded upon in Experiment 4.0.1.

The remainder of this section is structured as follows. First, Section 6.3.1 describes how SUSAN was modeled, with a focus on the cable lengths between the electric generators and motors. Next, five fixed propulsion system architectures with varying levels of redundancy are proposed in Section 6.3.2, and their power management strategies are optimized to minimize fuel burn. After optimizing the power management strategy, a Pareto Frontier is generated for various propulsion system failure modes and involves selecting the propulsion system architecture that minimizes fuel burn for a given system-level safety constraint. The resulting Pareto Frontier serves as a baseline for comparing system-level performance and safety for multiple fixed propulsion system architectures against those output from the bi-level optimization framework.

6.3.1 SUSAN Modeling Approach

SUSAN was modeled using the Future Aircraft Sizing Tool (FAST) [177], introduced in Chapter 4. The parameters used to model SUSAN are listed in Appendix D.1.

To simplify the propulsion system analysis, all electrical components (buses, circuit breakers, contactors, relays, etc.) *except cables* were neglected. Instead, these components were accounted for by allocating a constant 9,178 kg to an electrified aircraft propulsion weight group (hereinafter referred to as “EAP Weight”), consistent with prior system-level analyses performed [187].

In prior SUSAN design studies [187], only the turbofan engine, electric generators, and electric motors were modeled. However, system-level performance is greatly impacted by the cables connecting the electric generators and motors – redundant connections require more cables to be installed on the aircraft, thus increasing the structural weight required to support the airframe and propulsion system. Therefore, this study includes the cable weights and efficiencies in the propulsion system model.

Table 6.1 lists the electrical component performance parameters used in this study. Each of the values in the table are the “nominal” values predicted in Ref. [47] and were also used in Ref. [7] to develop SUSAN’s propulsion system model.

To determine each cable’s weight and overall efficiency, its length must be estimated. This was accomplished by digitizing the “SUSAN Electrofan power system diagram” in Fig. 6.3 (shown as Fig. 5 in Ref. [7]) and computing the total cable length from the electric generators to each electric motor. Since there is no reference length in the schematic, the digitizer was calibrated using the wingspan provided in Ref. [7].

Table 6.1: Electrical component performance parameters.

Component	Power-Weight Ratio	Efficiency
Electric Generator	25 kW/kg	98.5%
Electric Motor	20 kW/kg	99.0%
Cables	2 kg/m/MW	0.040% loss/m

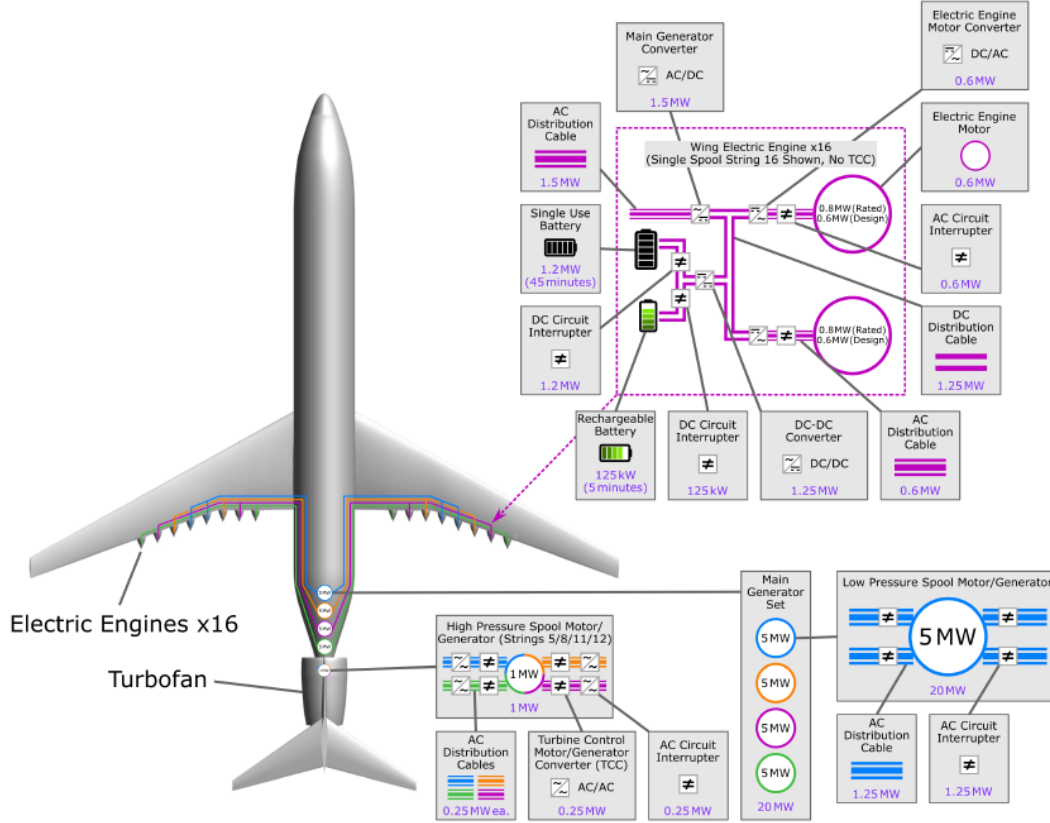


Figure 6.3: Schematic digitized to estimate the cable lengths, adapted from Ref. [7].

Existing SUSAN literature does not contain any information about the exact generator placement. However, prior studies [7, 47, 200] all refer to the same schematic as Fig. 6.3, which places the generators consecutively along the fuselage centerline in the tailcone. This is likely done to minimize the weight of the gearboxes and mechanical couplings required to spin the generators from the turbofan engine. Thus, each generator's position was assumed to be the center of each circle on the schematic.

Generators 1 and 4 are the closest and farthest from the turbofan engine, respectively;

Motors 1 and 16 are the most port and starboard positions along the wingspan, respectively. Using these component positions while digitizing Fig. 6.3, the cable lengths were computed, and are included in Appendix D.2. The cable lengths were multiplied by the nominal loss per unit length to compute the overall efficiency of each cable, as listed in Tab. 6.2.

Table 6.2: Estimated cable efficiency after digitizing the propulsion system schematic (all in %).

	Generator 1	Generator 2	Generator 3	Generator 4
Motors 1/16	99.25	99.28	99.30	99.31
Motors 2/15	99.30	99.32	99.35	99.36
Motors 3/14	99.34	99.37	99.39	99.41
Motors 4/13	99.39	99.41	99.44	99.45
Motors 5/12	99.43	99.45	99.48	99.49
Motors 6/11	99.47	99.49	99.52	99.54
Motors 7/10	99.51	99.53	99.56	99.57
Motors 8/9	99.55	99.57	99.60	99.61

After collecting data to model the propulsion system, SUSAN was calibrated in FAST while flying the design mission profile established in a prior study [187]. This was accomplished by imposing calibration factors on the airframe weight and fuel flow rate to match the MTOW and fuel burn for a 4,630 km mission with reserves. Extra attention was focused on ensuring that the fuel burn closely matched the value from literature because it will be used in the upcoming trade studies in Section 6.3.2. Table 6.3 lists the calibration factors and Tab. 6.4 compares the MTOW and fuel burn output by FAST relative to the same values provided in literature.

Table 6.3: Calibration factors applied to model SUSAN in FAST.

Calibration Factor	Value
Airframe Weight	1.095
Fuel Flow	0.950

The tuned calibration factors are within 10% of unity, indicating that SUSAN’s behavior is not significantly altered by the model calibration. After calibrating, the resultant SUSAN model in FAST returns MTOW and fuel burn values that closely match those provided in

literature [7]. The validated model of SUSAN provides a reference system-level performance threshold for further trade studies to be performed.

Table 6.4: Weight comparison between the SUSAN model in FAST and literature (all weights in kg).

Weight	FAST Model	Literature	Percent Error
MTOW	86,456	86,586	-0.15%
Fuel Burn	13,706	13,791	-0.61%
OEW	54,008	54,862	-1.56%

6.3.2 Performance-Safety Tradeoffs for Fixed Propulsion System Architectures

Using the validated SUSAN model, a trade study is executed to identify propulsion system architectures and corresponding power management strategies that maximize both system-level performance, measured via fuel burn, and safety, measured by failure rate. Four propulsion system architectures with a *fixed topology* were considered, each with different redundant electric generator-motor cross-connection schemes:

Baseline: each electric motor is connected to one electric generator.

Double Redundancy: each electric motor is connected to two electric generators.

Triple Redundancy: each electric motor is connected to three electric generators.

Quadruple Redundancy: each electric motor is connected to all four electric generators.

The baseline propulsion system architecture is illustrated in Fig. 6.4 and the redundant ones are illustrated in Fig. D.1 within Appendix D.3.

The redundancy included in the system architectures improves system safety while requiring more fuel to fly the same mission. As redundant connections are included, additional cables are required to transmit power between the generators and motors, thus increasing the component-level weights.

Two key changes were made to SUSAN’s propulsion system for the trade study. First, the turbofan engine is replaced with a turboshaft engine, eliminating the need to optimize the power split between its fan and electric generators. Second, a battery may be included to

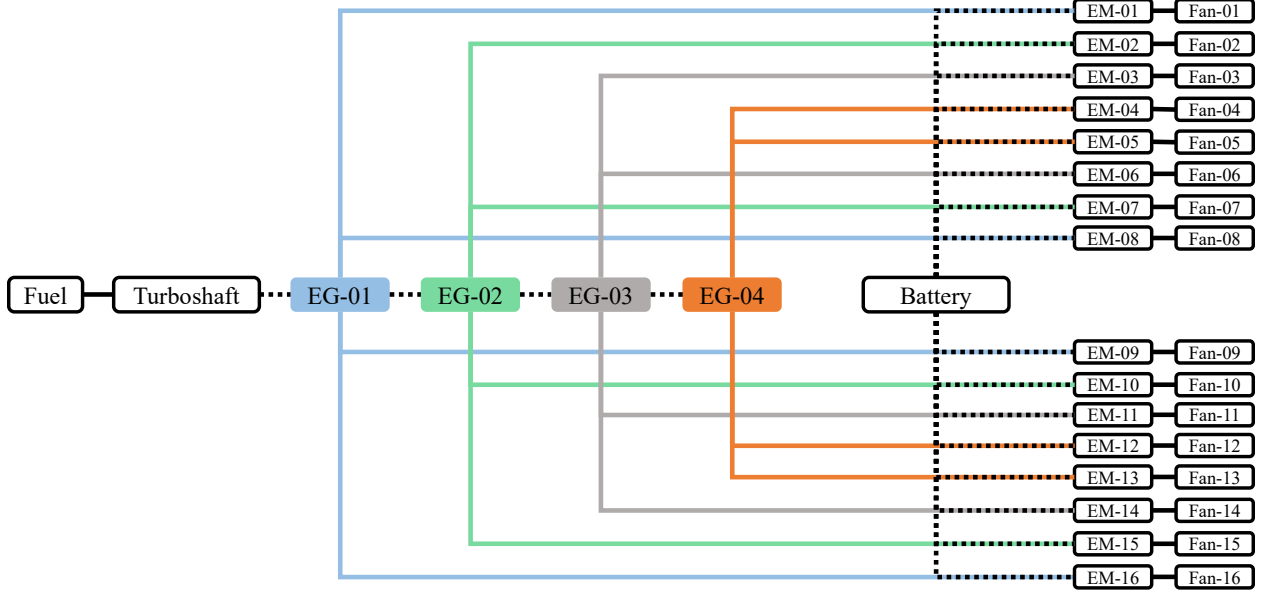


Figure 6.4: Baseline propulsion architecture.

boost the electric motor power during takeoff and climb at a *constant power split*, possibly requiring optimization to identify the power split that minimizes the fuel burn.

An additional assumption was made for sizing the electric generators in the presence of redundant electric generator-motor cross-connections. In a prior trade study [174], the electric generators in each of the redundant propulsion system architectures were sized to fully power eight electric motors, permitting nominal operations if one electric generator fails. If multiple electric generators fail, then the overall propulsion system performance is degraded because the electric motors can no longer operate at their rated power – this is standard for any abnormal in-flight scenario involving a propulsion system failure. The decision to increase each electric generator’s size contrasts prior SUSAN design studies; the generators were previously sized to only power four motors [7].

The same electric generator sizing logic is embedded in the bi-level optimization framework constructed in Section 6.2, but was generalized for any number of electric generator-motor connections – for propulsion system architectures with redundant electric generator-motor cross-connections, the electric generators are sized for double their required rated power, ensuring that the electric motors can still operate at their rated power if one electric generator fails.

6.3.2.1 System Performance: Fuel Burn and Weight Breakdown

The power management strategies for each of the four propulsion system architectures introduced in Section 6.3.2 were optimized using MATLAB's FMINCON function to minimize fuel burn, as previously defined in Eq. 6.3 while constructing the bi-level optimization framework.

A weight breakdown in Tab. 6.5 reveals the optimum power split and system-level responses for each propulsion system architecture.

In the worst case scenario (Quadruple Redundancy), there is a 5.83% fuel burn penalty. Aside from the fuel burn penalty, the cable and generator weights change the most; additional cables provide redundancy and the generators were resized to power eight motors (as opposed to four in the baseline configuration). The other aircraft components are resized accordingly to accommodate the additional cable and generator weight. However, their impact on MTOW is negligible relative to that from the cables and generators.

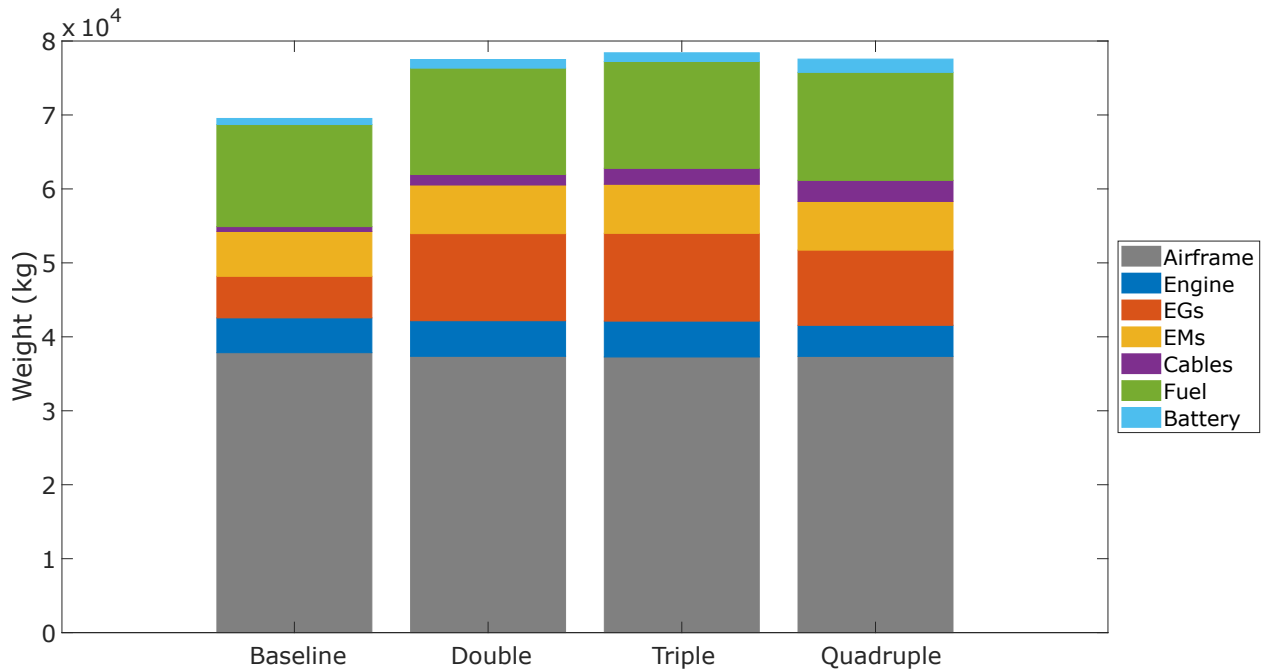
Table 6.5: System-level weight breakdowns (all in kg). The percent difference relative to the baseline configuration is included in parenthesis for each redundant configuration.

Configuration	Baseline	Double	Triple	Quadruple
MTOW	97,446	105,418	106,324	105,473
	—	(+8.18%)	(+9.11%)	(+8.24%)
Fuel	13,801	14,384	14,447	14,607
	—	(+4.22%)	(+4.67%)	(+5.83%)
Battery	819	1,161	1,177	1,791
	—	(+41.76%)	(+43.71%)	(+118.68%)
Airframe	37,857	37,365	37,278	37,360
	—	(-1.30%)	(-1.53%)	(-1.31%)
Engine	4,707	4,815	4,851	4,199
	—	(+2.29%)	(+3.06%)	(-10.79%)
Electric Generators	5,621	11,774	11,864	10,167
	—	(+109.46%)	(+111.07%)	(+80.88%)
Electric Motors	6,062	6,575	6,634	6,579
	—	(+8.46%)	(+9.44%)	(+8.53%)
Cables	658	1,423	2,155	2,850
	—	(+116.26%)	(+227.51%)	(+333.13%)
Optimized Power Split	15.13%	19.77%	19.87%	30.43%

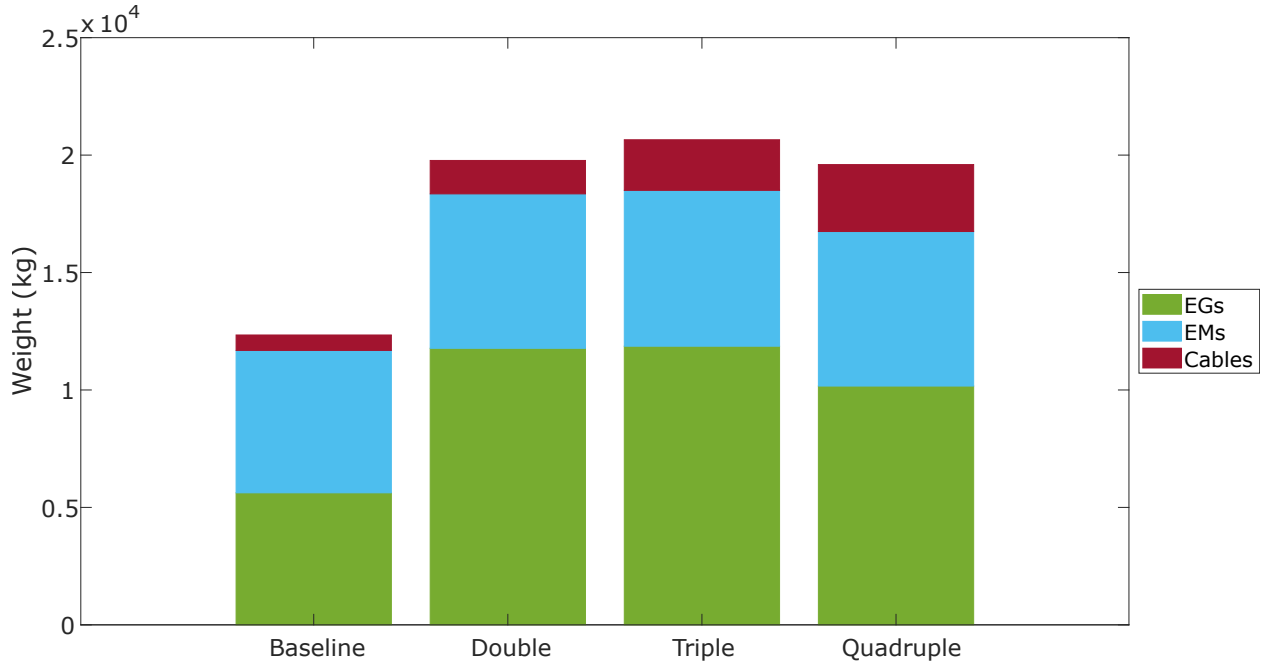
Weight breakdowns are plotted in Fig. 6.5 to graphically represent the impacts of adding redundancy to the propulsion system architecture, revealing a more qualitative summary of

the results. Fig. 6.5a includes all weight groups, while Fig. 6.5b only includes the electric generator, electric motor, and cable weights, better revealing the impacts of adding redundant electric generator-motor connections and doubling each electric generator’s size. Recall that a constant 9,178 kg was allocated to “EAP Weight,” and represents all of the systems on SUSAN’s electrified propulsion system (batteries, buses, circuit breakers, etc.) *excluding cables*. Future work could involve sizing these other components, but is too granular for this system-level study.

The combined electric generator, electric motor, and cable weights in the redundant propulsion systems are approximately 7,200 to 8,300 kg heavier than the baseline. Besides the additional cables included, system performance is most impacted by the decision to size the generators to power eight electric motors. To accommodate additional electric generator failures, it is possible to size them to power 12 or 16 electric motors. However, any electric generator failure on SUSAN can be likened to an engine failure on a conventional aircraft, thus constituting an abnormal flight scenario and requiring a diversion to an alternate airport. The decision to size the electric generators to power eight electric motors can be further justified by considering conventional aircraft configurations with two, three, or four turbofan engines – a loss of one engine results in significantly degraded flight performance characteristics.



(a) System-level weights.



(b) Electrical component weights.

Figure 6.5: Weight breakdowns for all configurations examined.

Therefore, it is unrealistic to expect SUSAN to perform nominally during an abnormal flight scenario. Although the engineer may be tempted to over-size the electric generators to ensure maximum performance during abnormal scenarios, SUSAN's nominal performance during standard operations would be further degraded, requiring more fuel to fly as a result of the heavier electric generators.

Additionally, the presence of the distributed electric propulsion system allows SUSAN's performance to degrade more gracefully than a conventional turbofan aircraft. If one or two of SUSAN's electric propulsors fail, approximately 4-8% of its power available is lost. In contrast, an engine failure on a conventional turbofan aircraft causes 25-50% of its power available to be lost. Based on this argument, SUSAN can perform better in abnormal scenarios than conventional turbofan aircraft, further justifying why the electric generators do not need to be significantly over-sized.

6.3.2.2 System Safety: Failure Intensity and Minimal Cut Sets

For the system-level studies performed in this work, only electric generator and motor failures were considered in the safety analysis. Other electrical components (batteries, buses, circuit breakers, etc.) *including cables* were assumed to be fully functional and excluded from the safety analysis. The same assumption was applied to the aft turbofan engine and fans in the distributed electric propulsion system.

Each component’s “Applied Failure Rate”, listed in Tab. 6.6, was estimated by multiplying its “Baseline Failure Rate” by an “Applied Environmental Factor”, as previously done in Ref. [201]. Since the “Baseline Failure Rate” was identified from failure rates of similar components, the “Applied Environmental Factor” serves to conservatively estimate a reasonable failure rate for state-of-the-art technologies, primarily due to a lack of technological maturity. Each component’s probability of failure per flight was computed by multiplying the “Applied Failure Rate” by the block time for SUSAN’s design mission (5.78 hours).

Table 6.6: Component failure rates and probabilities.

	Electric Generator	Electric Motor
Baseline Failure Rate (failures per 10^6 flight hours)	13.00	9.24
Applied Environmental Factor	10	10
Applied Failure Rate (failures per 10^6 flight hours)	130.0	92.4
Probability of Failure (per flight)	7.52×10^{-4}	5.35×10^{-4}

In this study, only failure modes pertaining to SUSAN’s distributed electric propulsion system are assessed. Since certification regulations have yet to be developed for advanced aircraft concepts, five failure modes were considered, consisting of a minimum of either 1-, 2-, 4-, 8-, or 16-electric propulsor failures (referred to as “ k -EP failure modes/scenarios” hereafter). Each failure mode is characterized by a *minimum* number of component failures because one component failure can result in the loss of many electric propulsors, allowing multiple proposed failure modes to be realized. For example, a generator failure in SUSAN’s baseline propulsion system architecture causes four electric propulsors to fail, satisfying the minimum criteria for three of the five proposed failure modes.

The system-level failure rates in Tab. 6.7 illustrate the system safety improvements for increased redundancy in the propulsion system. Across each row, the failure rate monotonically decreases as a result of adding more redundant connections in the propulsion system

architecture. Down each column, the failure rate monotonically decreases as a result of permitting more electric propulsors to fail before a system-level failure is triggered. Depending on the failure mode selected, the redundant system architectures may provide system safety improvements that span multiple orders of magnitude, as shown for the 2-, 4-, and 8-EP failure scenarios. However, other failure scenarios (1- and 16-EP) do not benefit from increased redundancy.

Table 6.7: Failure intensities (per flight hour) for each configuration under varying failure modes.

Failure Mode	Baseline	Double	Triple	Quadruple
1-EP	2.00×10^{-03}	1.50×10^{-03}	1.50×10^{-03}	1.50×10^{-03}
2-EP	5.26×10^{-04}	6.12×10^{-06}	5.93×10^{-06}	5.93×10^{-06}
4-EP	5.20×10^{-04}	1.96×10^{-07}	3.20×10^{-10}	2.58×10^{-11}
8-EP	5.87×10^{-07}	1.96×10^{-07}	5.53×10^{-14}	5.53×10^{-14}
16-EP	5.53×10^{-14}	5.53×10^{-14}	5.53×10^{-14}	5.53×10^{-14}

The results from each failure mode are discussed separately to indicate why redundancy does (or does not) improve system safety. Since the generator and motor failure probabilities are approximately the same order of magnitude, a nominal component failure rate, x , is defined for the remainder of the discussions in this section. The order of magnitude analysis performed in the following discussions is predicated on the assumption that each failure is a rare event ($x \ll 1$).

The analyses in the following discussions consider failures involving only generators or only motors. For some failure modes, it is possible that combinations of generators and motors could fail. However, these are less impactful on the system-level failure intensities and obfuscate the general findings of this work.

Some of the following discussions refer to the FAA’s “quantitative probability terms” [198], defining how frequently a system-level failure may occur. The terms and corresponding probability bounds are listed in Tab. 6.8.

Table 6.8: Quantitative probability terms defined in FAA Advisory Circular 25.1309-1B.

Term	Lower Bound (/flight hour)	Upper Bound (/flight hour)
Probable	1×10^{-5}	1×10^{-3}
Remote	1×10^{-7}	1×10^{-5}
Extremely Remote	1×10^{-9}	1×10^{-7}
Extremely Improbable	—	1×10^{-9}

1-EP Failure Scenario On the baseline architecture, only one generator or motor must fail. Thus, the system-level failure probability is obtained by adding all component failure probabilities together:

$$16x + 4x = 20x$$

Once redundancy is added, multiple generators or a single motor must fail. Since the generator and motor component failure probabilities are nominally the same order of magnitude, a motor failure is much more likely to occur than multiple generator failures.

For example, in the Double Redundancy architecture, the failure mode is triggered by one of 16 possible motors failing, or one of two pairs of generators failing. Thus, the system-level failure probability is:

$$16x + 2x^2 \approx 16x$$

In summary, a single electric propulsor failure in the redundant propulsion system architectures is dominated by a motor failure. The likelihood of multiple generators failing is negligible relative to that of a single motor, explaining why there is minimal improvement as more redundancy is added for this failure mode.

2-EP Failure Scenario For two electric propulsors to fail on the baseline architecture, either one (of four) generators or two motors (of the 120 possible combinations) must fail. The system-level failure probability is:

$$120x^2 + 4x \approx 4x$$

Thus, for the baseline architecture, it is easier to fail one generator than it is to fail multiple motors. However, this is not true for the redundant architectures. In the Double

Redundancy architecture, either one of the two redundant generator pairs or two of the motors must fail. The system-level failure probability is:

$$120x^2 + 2x^2 \approx 120x^2$$

Once redundancy is added, it becomes just as difficult to fail two generators as it does to fail two motors. As a result, the 120 possible motor failure combinations are more likely to occur. In the remaining redundant system architectures, the exponent on the second addend increases (corresponding to additional redundant connections between each generator and motor), making the contribution of the generator failures less significant, thus explaining the small system safety improvements observed.

4-EP Failure Scenario On the baseline architecture, a failure occurs if one of the generators fails or four of the motors fail (1,820 possible combinations). The system-level failure probability is approximately:

$$1,820x^4 + 4x \approx 4x$$

Since the generator failure modes are dominant, there is minimal safety improvement by adopting the 4-EP failure scenario relative to a 2-EP failure scenario. The minimal improvement comes from the requirement that four motors must fail instead of two.

In the redundant architectures, the generator failures remain more likely to occur than the motor failures. However, this trend changes once the Quadruple Redundancy configuration is considered. At this point, the system-level failure probability is approximately:

$$1,820x^4 + x^4 \approx 1,820x^4$$

The motor failures now dominate the expression because both failure modes require four components to fail. The thousands of possible motor failure combinations outweigh the likelihood of all four generators failing.

8-EP Failure Scenario For eight electric propulsors to fail on the baseline architecture, either two of the four generators (6 possible combinations) or eight motors (12,870 possible combinations) must fail. The nominal system-level failure probability is:

$$12,870x^8 + 6x^2 \approx 6x^2$$

which is dominated by the generator failures.

As redundancy is added to the system architecture, a similar behavior is observed as demonstrated in the 4-EP failure scenario; it is easier to fail multiple generators than it is to fail eight motors. For example, in the Quadruple Redundancy configuration, the nominal system-level failure probability is:

$$12,870x^8 + x^4 \approx x^4$$

highlighting how it is significantly more likely that all four generators fail than eight motors.

16-EP Failure Scenario For all propulsion system architectures considered, either all four generators or all 16 motors must fail. Thus, the system-level failure probability is:

$$x^{16} + x^4 \approx x^4$$

and remains constant, regardless of the redundancy included in the system architecture. The likelihood of all four generators failing is always greater than that of all 16 motors failing. Although it is highly unlikely that this failure mode would be selected for certification, it suggests that there are no system safety benefits from adding redundant generator-motor connections.

6.3.2.3 Baseline Pareto Frontier Generation

A Pareto Frontier was developed to assess the tradeoff between system-level performance and safety. Figure 6.6 plots the common logarithm of the system-level failure rate against the fuel burn increase.

The dashed lines on the plot represent the *upper bounds* of the FAA’s “quantitative probability terms” [198], and describes the frequency of a failure occurring in a (sub)system onboard an aircraft. Each term and corresponding probability bounds were previously defined in Tab. 6.8.

Overall, the redundant system architectures improve system safety while requiring more fuel to fly the same mission. As redundant connections are included, additional cables are required to transmit power between the generators and motors, thus increasing the component-level weights.

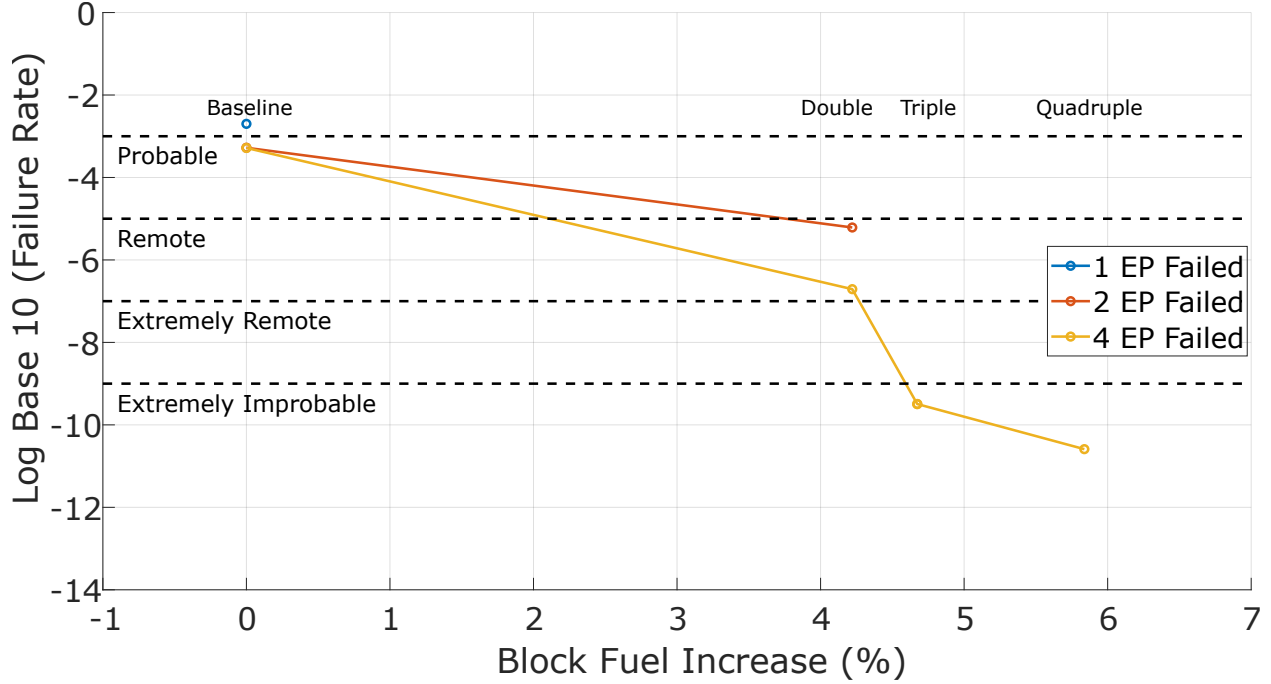


Figure 6.6: Pareto Frontiers for different propulsion system failure modes synthesized under a fixed architecture with power management optimization only. System architectures corresponding to each point are labeled at the top).

Furthermore, the failure mode selected greatly affects whether the redundant configurations improve system safety or not. For a 1-EP failure scenario, the likelihood of a failure occurring is always greater than the “Probable” limits imposed by the FAA, regardless of the redundancy included in the propulsion system architecture, resulting in none of the propulsion architectures being safety-feasible. In contrast, for the 2- and 4-EP failure modes, the baseline configuration falls within the “Probable” limits imposed by the FAA, rendering it as a possible system architecture that could be installed on an aircraft, depending on the severity of each failure mode classified [202, 203].

The Pareto Frontiers generated for the 2- and 4-EP failure modes provide safety-feasible propulsion system architectures that are either optimized for system-level performance or system-level safety. For both failure modes, the baseline configuration optimizes system-level performance, meeting the minimum required safety level. All other propulsion system architectures maximize system safety for various fuel burn penalties.

In the 2-EP failure mode, system safety improves by two orders of magnitude under the Double Redundancy configuration, but only improves marginally for the Triple and Quadruple

Redundancy configurations. Thus, the primary performance-safety tradeoff is between the Baseline and Double Redundancy configurations, and were included on the Pareto Frontier. In the 4-EP failure mode, system safety improves by three orders of magnitude between the Baseline, Double Redundancy, and Triple Redundancy configurations, and improves by an additional order of magnitude between the Triple and Quadruple Redundancy configurations. As a result of the continuously improving system safety, all four system architectures were included on the Pareto Frontier.

Impact of Certification Regulations on Configuration Selection From the Pareto Frontier, select designs could be integrated into a future aircraft concept depending on the severity of a 2- or 4-EP failure in SUSAN’s distributed propulsion system. FAA certification regulations of aircraft engines, as well as the aircraft’s systems and subsystems, require all hazardous failures to be *extremely remote* and all major failures to be *remote* [203, 204]. Therefore, the Double Redundancy or Triple Redundancy propulsion system architectures would minimize fuel burn while meeting the required safety margin, depending on the severity classification of 2- and 4-EP failures.

The same design logic can be extended for normal category aircraft certified under FAA Part 23 regulations [205], as the system and subsystem safety requirements do not change between the two aircraft categories. However, the design logic cannot be extended to experimental aircraft in the normal and transport categories, as they are certified under *Special Airworthiness Certificates* as opposed to *Standard Airworthiness Certificates* [206]; the airframe manufacturer and certification authorities collaborate to outline sufficient safety requirements to demonstrate airworthiness.

6.3.2.4 Benchmarking: Pareto Hypervolume

Based on the Pareto Frontier generated in Fig. 6.6, estimated failure rates, and the FAA’s quantitative probability terms, eight benchmarking points can be selected for benchmarking the bi-level optimization framework. The benchmarking points are listed in Tab. 6.9, along with the corresponding optimized *fixed* system architecture.

The Pareto Hypervolume is computed to serve as a benchmark against the Pareto Frontiers generated using the bi-level optimization framework. Since there are two objective function values, the Pareto Hypervolume is computed as an area underneath the Pareto Frontier for each corresponding failure mode. In the optimization problem, fuel burn and system-level failure rate are minimized, suggesting that the Pareto-optimal solution is towards the bottom-

Table 6.9: System safety constraints for benchmarking and corresponding optimized propulsion system architectures.

Test	Failure Rate (/hr)	Electric Propulsors Failed	Optimized Architecture
A	10^{-02}	1	Baseline
B	10^{-03}	2	Baseline
C	10^{-05}	2	Double Redundancy
D	10^{-03}	4	Baseline
E	10^{-05}	4	Double Redundancy
F	10^{-07}	4	Triple Redundancy
G	10^{-09}	4	Triple Redundancy
H	10^{-10}	4	Quadruple Redundancy

left corner of Fig. 6.6. Therefore, improved objective function values will result in a smaller Pareto Hypervolume; either less fuel is burned or the system-level failure rate is smaller.

To compute the Pareto Hypervolume, a trapezoidal integration technique is used. The bases of each trapezoid are computed as the difference between the configuration’s system-level failure rate and a baseline system-level failure rate of 10^{-12} /hr. The height of each trapezoid is computed as the difference in block fuel increase between consecutive points on the Pareto Frontier. The results of the trapezoidal integration for the baseline Pareto Frontier in Fig. 6.6 are listed in Tab. 6.10 for the 2- and 4-EP failure modes.

Table 6.10: Pareto Hypervolume from *fixed* architecture optimization.

Electric Propulsors Failed	Pareto Hypervolume
2	32.73
4	33.61

6.3.3 Performance-Safety Tradeoffs using the Bi-Level Optimization Framework

After establishing the baseline performance-safety tradeoffs under *fixed* architecture optimization, the bi-level optimization framework can be utilized to repeat the prior optimizations outlined in Tab. 6.9 by changing *both* the system architecture topology and corresponding power management strategy.

6.3.3.1 Problem Setup

To execute the optimization, a component library and any additional user-prescribed architectural or power management strategy constraints must be defined, as described next. All technology levels and top-level aircraft requirements used in Section 6.3.2 for the *fixed* architecture optimization are utilized in the bi-level optimization framework, as well as the design mission profile. Holding such parameters constant ensures that the possible improvements in system-level performance and safety are from optimizing both the propulsion system architecture topology and corresponding power management strategy.

Component Catalog Using the components in SUSAN’s propulsion system as a baseline, a component catalog was created, as listed in Tab. 6.11.

Table 6.11: Component catalog used in the bi-level optimization framework.

Component	Type	Min. Num.	Max. Num.	Input Power	Output Power	Efficiency (%)	Failure Rate (/hr)
Fuel	SRC	1	1	—	Chemical	—	—
Battery	SRC	0	1	—	Electrical	—	—
Cable	TRN	0	1	Electrical	Electrical	0.996 /m	—
Turboshaft	TRN	1	1	Chemical	Mechanical	—	—
Elec. Gen.	TRN	1	4	Mechanical	Electrical	98.5	1.30×10^{-4}
Elec. Motor	TRN	2	16	Electrical	Mechanical	99.0	9.24×10^{-5}
Ducted Fan	TRN	2	16	Mechanical	Thrust	99.0	—
Thrust Power	SNK	1	1	Thrust	—	—	—

Each component is classified as a source (SRC), transmitter (TRN), or sink (SNK), as established by the GPSA Framework in Chapter 3. The minimum number of allowable components is checked in the bi-level optimization framework as a user-imposed architectural constraint (described next). The maximum number of allowable components sets the architecture, operational, and efficiency matrices’ sizes. The input/output power describes the type of power each component receives and transmits to others, driving the compatibility matrix (C) that the PROPULSION ARCHITECTURE OPTIMIZATION utilizes to add/remove connections in the system architecture. The fixed component efficiencies and failure rates are utilized during the POWER MANAGEMENT OPTIMIZATION and FAULT TREE ANALYSIS, respectively.

Compatibility Matrix Modifications In the bi-level optimization framework, the `CHANGETOPOLOGY` function described in Alg. 7 may add/remove connections between any two components with the matching power output/input types. As a result, some connections in the compatibility matrix must be suppressed to avoid producing physically intractable system architectures. Three modifications are made to the compatibility matrix:

1. Turboshift engines are not allowed to connect to ducted fans directly. Their power must be transmitted by electrical components.
2. Each electric motor connects to its own, unique ducted fan, eliminating any possible redundant connections between the two components.
3. Electric motors cannot output power to electric generators, and prevents the system architecture from becoming unnecessarily complex to maximize system safety.

Additional Architectural Constraints Four additional architectural constraints were included in the `PROPULSION ARCHITECTURE OPTIMIZATION`:

1. The minimum number of components listed in Tab. 6.11 must be included in the propulsion system.
2. Each electric generator must output power to the same number of electric motors.
3. Each electric motor must receive the same number of electric generator inputs.
4. All electric motors either do or do not connect to the cables from the battery.

The constraint on the minimum number of components ensures that a propulsion system architecture with a gas turbine engine is generated rather than a battery-electric propulsion system, which remains intractable for single-aisle aircraft with existing battery technology [34–37]. The constraints on the electric generator/motor output/input connections are present to ensure that the same level of redundancy exists throughout the propulsion system, helping to reproduce variants of the propulsion system architectures introduced in Section 6.3.2. The Cable Routing Optimization introduced in Section 6.2.5.3 may elect to arrange the cable connections between the electric generators and motors differently, generating *near-equivalent* topologies and possibly resulting in improved system-level performance or safety.

Additional Power Management Optimization Constraints One additional constraint was added to the POWER MANAGEMENT OPTIMIZATION, requiring that the downstream power splits between similarly typed components are equal. For example, the downstream power split between the thrust power sink and ducted fans must be equal, ensuring that each one outputs the same amount of power to eliminate any adverse yaw moments from occurring during flight. Although the electric motors output the same power, the cables used to route power from the electric generators vary in length, resulting in different power loads for each electric generator and may modestly impact system-level performance.

Optimization Hyperparameters In the Simulated Annealing Optimizer, there are three hyperparameters: 1) the initial temperature, T_0 , impacting the probability that a sub-optimal solution is accepted to explore new regions of the design space; 2) the decay rate, α , impacting how the probability that a sub-optimal solution is accepted decreases between iterations; and 3) the number of major iterations to execute. The hyperparameters for the PROPULSION ARCHITECTURE OPTIMIZATION and Cable Routing Optimization problems are listed in Tab. 6.12.

Table 6.12: Hyperparameters utilized for the Simulated Annealing Optimizer.

Optimization Problem	T_0	α	Iterations
Propulsion Architecture Optimization	1,200	0.85	50
Cable Routing Optimization	1	0.99	3,000

6.3.3.2 Pareto Frontier Generation

Again, a Pareto Frontier was developed to assess the tradeoff between system-level performance and safety. Figure 6.7 plots the common logarithm of the system-level failure rate against the fuel burn increase.

There is one noticeable difference between the Pareto Frontier generated under the bi-level optimization framework relative to that generated from the fixed architecture power management optimization illustrated previously in Fig. 6.6. For a Double Redundancy propulsion system architecture in Fig. 6.7, the likelihood of a propulsion system failure occurring is “extremely remote,” as opposed to “remote” in Fig. 6.6. Thus, the bi-level optimization framework improved system safety without additional redundancy, enhancing the overall system safety relative to the Double Redundancy propulsion system architecture

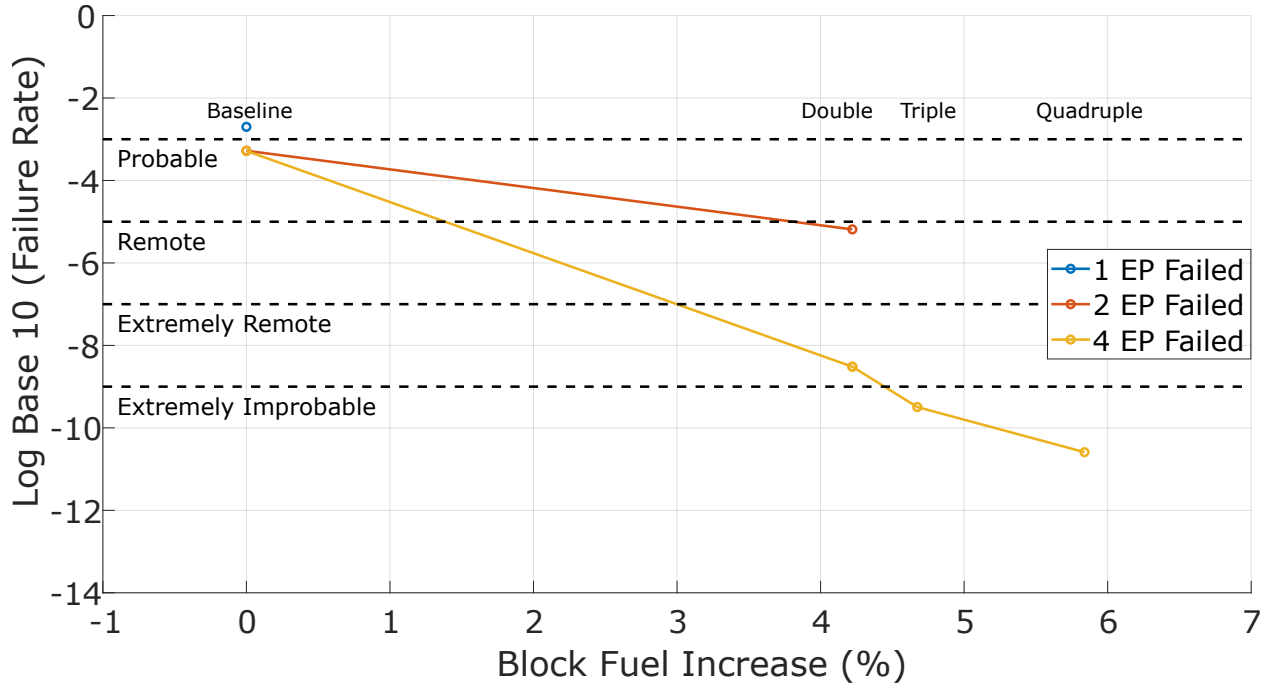


Figure 6.7: Pareto Frontiers for different propulsion system failure modes synthesized under the bi-level optimization framework. System architectures corresponding to each point are labeled at the top.

utilized in Section 6.3.2, resulting in a new non-dominated point on the Pareto Frontier. The improved propulsion system architecture is illustrated in Fig. 6.8 and arranges the redundant connections asymmetrically, a contrast to the symmetric arrangements illustrated in Fig. D.1 in Appendix D.3.

In the symmetric arrangement, Generators 1 and 2 powered all eight outboard motors (1 through 4 and 13 through 16), while Generators 3 and 4 powered all eight inboard motors (5 through 12). As a result, each generator connected to four electric motors on each side of the distributed electric propulsion system. From the FAULT TREE ANALYSIS, four electric motors would fail if either: 1) both Generators 1 and 2 failed; 2) both Generators 3 and 4 failed; or 3) any combination of four electric motors failed.

In contrast, the asymmetric connection arrangement in Fig. 6.8 significantly improves system safety. Generators 1 and 3 connect to five electric motors on the left half of the wing and three electric motors on the right half. Conversely, Generators 2 and 4 connect to three electric motors on the left half of the wing and five electric motors on the right half. As a result of the asymmetric arrangement, system safety improves because a maximum of three

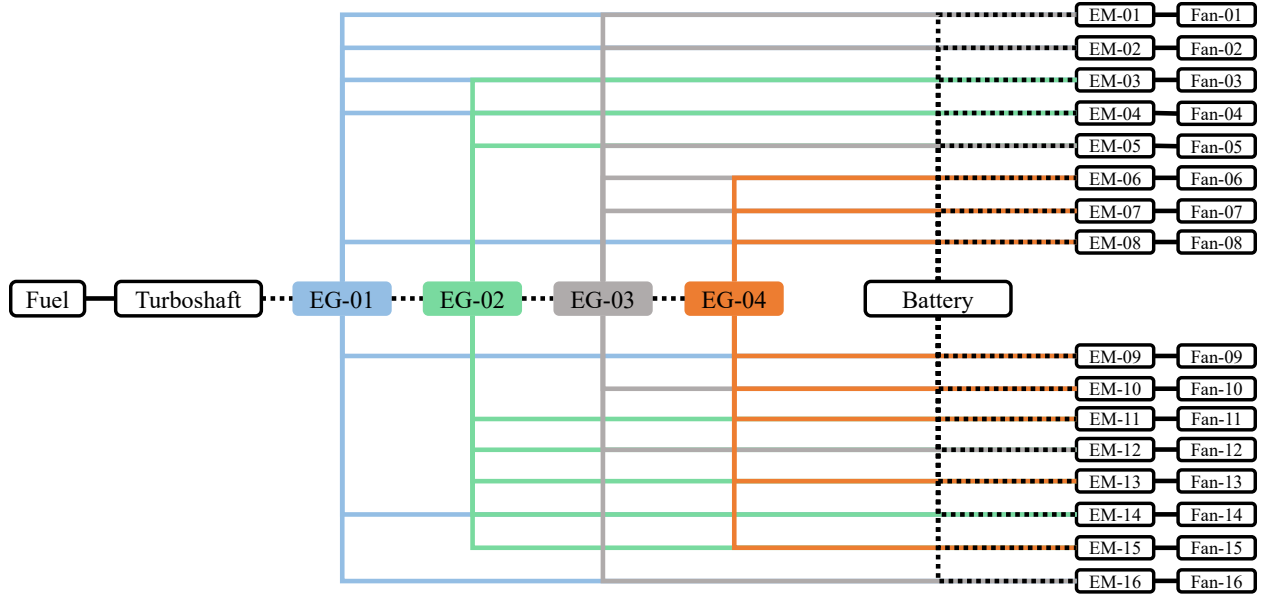


Figure 6.8: Improved Double Redundancy propulsion system architecture from the bi-level optimization framework.

electric motors fail if two electric generators fail. Table 6.13 lists all of the electric motors that fail if two electric generators fail. The asymmetric connections in the propulsion system architecture improve system safety because four electric motors fail if *three* electric generators fail.

Table 6.13: Electric motors that fail from two electric generators failing.

Generator Failure 1	Generator Failure 2	Electric Motors Failed
1	2	3, 4, 14
1	3	1, 2, 16
1	4	8, 9
2	3	5, 12
2	4	11, 13, 15
3	4	6, 7, 10

6.3.3.3 Benchmarking: Pareto Hypervolume

The eight benchmarking points previously listed in Tab. 6.9 are now listed in Tab. 6.14 with the optimized propulsion system architectures output from the bi-level optimization framework.

Previously, the Triple Redundancy propulsion system architecture was necessary to meet a 4-EP failure mode with a maximum allowable failure rate of 10^{-7} /hr. With the bi-level optimization framework, a novel arrangement of electric generator-motor cross-connections was synthesized to only require a Double Redundancy propulsion system architecture to satisfy the maximum allowable failure rate under the 4-EP failure mode.

Table 6.14: System safety constraints for benchmarking and corresponding optimized propulsion system architectures. Optimized architectures in *italics* represent configurations with improved system safety relative to the fixed architecture benchmarks.

Test	Failure Rate (/hr)	Electric Propulsors Failed	Optimized System Architecture	Optimized Power Split (%)
A	10^{-02}	1	Baseline	15.13
B	10^{-03}	2	Baseline	15.13
C	10^{-05}	2	Double Redundancy	19.77
D	10^{-03}	4	Baseline	15.13
E	10^{-05}	4	Double Redundancy	19.77
F	10^{-07}	4	<i>Double Redundancy</i>	19.77
G	10^{-09}	4	Triple Redundancy	19.87
H	10^{-10}	4	Quadruple Redundancy	30.43

Again, the Pareto Hypervolume, or area under the curve for each corresponding failure mode, is computed to serve as a benchmark against the Pareto Frontiers generated from the *fixed* architecture power management optimization. As done in the previous trapezoidal integration, a failure rate of 10^{-12} was set as the reference line to perform the integration. The resulting Pareto Hypervolumes and percent difference relative to those computed for the *fixed* architecture power management optimization are listed in Tab. 6.15.

Table 6.15: Pareto Hypervolume from *fixed* architecture optimization.

Electric Propulsors Failed	Pareto Hypervolume	Percent Difference
2	32.79	+0.18%
4	29.38	-12.59%

For the 2-EP failure mode, the Pareto Hypervolume is negligibly larger, resulting in no significant difference between the two Pareto Frontiers. For the 4-EP failure mode, however, the Pareto Hypervolume is about 12.5% smaller, resulting in an improved Pareto

Frontier. Furthermore, the reduced Pareto Hypervolume indicates that the propulsion system architecture synthesized from the bi-level optimization framework exhibit improved system-level performance and safety relative to the fixed system architectures introduced in Section 6.3.2 that only optimized the power management strategy.

6.4 Heuristics Benchmarking (Experiment 4.0.2)

Experiment 4.0.1 validated the bi-level optimization framework against a fixed architecture trade study and demonstrated its ability to generate advanced propulsion system architectures with improved system-level performance and safety relative to fixed architecture. In Experiment 4.0.2, the bi-level optimization framework was benchmarked against MATLAB’s implementation of the Genetic Algorithm optimizer, an existing state-of-the-art stochastic optimization method that utilizes generalized, problem-agnostic heuristics to identify the best propulsion system architecture. The Genetic Algorithm is a nature-inspired search method to explore the design space globally, utilizing multiple design points, known as “individuals”, during each major iteration, or “generation” [65]. Based on the objective function value, or “fitness”, of each individual in the generation, some are kept for evaluation in further generations, while others are utilized for “crossover” or “mutation,” generating new individuals in the population. Additionally, new individuals are added to each generation by randomly exploring the design space.

In Experiment 4.0.1, the design space in the SYSTEM ARCHITECTURE OPTIMIZATION problem consisted of 117 binary variables, resulting in 1.662×10^{35} possible system architectures. Assuming that the optimization process took 1 millisecond and that high performance computing resources could evaluate 100 candidate system architectures simultaneously, it would still take 5.27×10^{16} years to evaluate all system architectures and identify the globally-optimal one. As a result of the intractable computational load required to identify the globally-optimal solution, benchmarking against a state-of-the-art method like the Genetic Algorithm optimizer was selected instead.

In addition to benchmarking the Simulated Annealing optimizer against the Genetic Algorithm optimizer, the tradeoffs between the two were assessed. Three primary tradeoffs between the optimizers include the: 1) ability to explore the design space fully; 2) ability to filter out infeasible designs; and 3) computational efficiency to attain a solution.

6.4.1 Experimental Setup

The PROPULSION ARCHITECTURE OPTIMIZATION problem is solved using the bi-level optimization framework constructed in Section 6.2 with two different optimizers to select the best propulsion system architecture: 1) the Simulated Annealing optimizer initially included in the bi-level optimization framework; and 2) the Genetic Algorithm optimizer. The Simulated Annealing optimizer included in the framework utilizes the NEIGHBOR FUNCTION introduced in Alg. 6 to select the next system architecture via problem-specific heuristics. The Genetic Algorithm optimizer utilizes a similar problem formulation in MATLAB; each constraint in the PROPULSION ARCHITECTURE OPTIMIZATION problem (Eq. 6.1) is still included as a constraint. Rather than utilizing the constraints to filter out infeasible designs, as done in the Simulated Annealing optimizer, the Genetic Algorithm optimizer constructs an unconstrained optimization problem with a penalty on any system architectures that violate one or more of the constraints [207].

To benchmark the Simulated Annealing optimizer against the Genetic Algorithm optimizer, Test A from Section 6.3.2 is utilized, corresponding to a 1-EP failure mode with a maximum allowable failure rate of 10^{-02} /hr. The Simulated Annealing optimizer runs utilizing the same hyperparameters used to validate the framework and benchmark it against a fixed architecture optimization (refer to Tab. 6.12 from Section 6.3). The Genetic Algorithm optimizer also utilizes three hyperparameters: 1) the crossover fraction, the fraction of the next generation's population that is created by crossing the genes of existing individuals; 2) the population size; and 3) the maximum allowable number of iterations. The crossover fraction is varied between 10% and 60% in increments of 10%. The population size is set to either 50 or 100. Since the Simulated Annealing optimizer always converged in less than 24 hours, the Genetic Algorithm optimizer was allowed to run for up to 24 hours, providing a comparison of the best objective function value attained relative to the wall time available. The resulting system-level performance (fuel burn), safety (propulsion system failure rate), and wall time to execute the optimization are recorded for comparison between the two methods.

6.4.2 Results

The baseline performance of the bi-level optimization framework utilizing the Simulated Annealing optimizer is listed in Tab. 6.16. From Section 6.3, the optimized propulsion system architecture was the baseline configuration, containing no redundant electric generator-motor

cross-connections, and took nearly 12 hours to solve.

Table 6.16: System-level performance and safety baselines from the bi-level optimization framework.

Optimized Configuration	Fuel Burn (kg)	Failure Rate (/hr)	Wall Time (hh:mm:ss)
Baseline	13,801	2.00×10^{-03}	11:43:34

The optimized propulsion system architectures, system-level responses, and wall times required to run the Genetic Algorithm optimizer for 50 generations are included in Tabs. 6.17 and 6.18 for population sizes of 50 and 100, respectively. In all tests run, the Genetic Algorithm optimizer found that the Quadruple Redundancy configuration is optimized, resulting in a fuel burn of 14,510 kg and a system-level failure rate of 1.50×10^{-3} /hr. Regardless of the population size analyzed, the optimization took approximately 18 to 20 hours to run.

Table 6.17: Optimized system-level performance and computational resources required, 50 individuals per generation.

Crossover Fraction (%)	Optimized Configuration	Fuel Burn (kg)	Failure Rate (/hr)	Wall Time (hh:mm:ss)
10	Quadruple	14,510	1.50×10^{-03}	18:17:33
20	Quadruple	14,510	1.50×10^{-03}	18:24:58
30	Quadruple	14,510	1.50×10^{-03}	18:45:09
40	Quadruple	14,510	1.50×10^{-03}	18:46:47
50	Quadruple	14,510	1.50×10^{-03}	18:29:04
60	Quadruple	14,510	1.50×10^{-03}	18:28:24

In the Genetic Algorithm Optimizer, a feasible solution is identified, maximizing system safety rather than minimizing fuel burn and returning a sub-optimal system architecture. The solution returned by the Genetic Algorithm optimizer, however, is Pareto-optimal. The benchmarking test focused on identifying the best propulsion system architecture by running the Genetic Algorithm optimizer for a longer time than it took for the Simulated Annealing optimizer to converge. As a result of the problem-agnostic heuristics and evolutionary behavior inherent to the Genetic Algorithm optimizer, additional generations or a larger population size may have been necessary to identify the best propulsion system architecture. However, the Genetic optimizer already required 50 to 66% more time than the Simulated

Table 6.18: Optimized system-level performance and computational resources required, 100 individuals per generation.

Crossover Fraction (%)	Optimized Configuration	Fuel Burn (kg)	Failure Rate (/hr)	Wall Time (hh:mm:ss)
10	Quadruple	14,510	1.50×10^{-03}	18:15:51
20	Quadruple	14,510	1.50×10^{-03}	18:14:10
30	Quadruple	14,510	1.50×10^{-03}	18:17:28
40	Quadruple	14,510	1.50×10^{-03}	18:14:04
50	Quadruple	14,510	1.50×10^{-03}	19:19:48
60	Quadruple	14,510	1.50×10^{-03}	20:07:07

Annealing optimizer and failed to reach the best propulsion system architecture, suggesting that its problem-agnostic heuristics are not the most appropriate to solve the PROPULSION ARCHITECTURE OPTIMIZATION problem.

Therefore, the Simulated Annealing optimizer utilizing embedded problem-specific heuristics synthesizes propulsion system architectures with improved system-level performance than those designed with stochastic optimization methods that utilize generalized, problem-agnostic heuristics. The problem-specific heuristics embedded in the NEIGHBOR FUNCTION enable the Simulated Annealing optimizer to strategically search the design space while accommodating randomness and exploring different regions of the design space. Since only one POWER MANAGEMENT OPTIMIZATION is required during each major iteration, computational resources are saved from evaluating the POWER MANAGEMENT OPTIMIZATION on all individuals in the populations, as done in the Genetic Algorithm optimizer.

The Genetic Algorithm optimizer returned the same solution for each set of hyperparameters used, suggesting that the optimizer may be sensitive to the problem setup. In the Genetic Algorithm optimizer, the problem is formulated as a penalty function [207], meaning that improvements in the penalty function could either be improvements in the objective function or feasibility of the candidate solution. Therefore, future benchmarking studies may need to further refine the problem formulation such that infeasible system architectures are penalized more, allowing the Genetic Algorithm optimizer to improve how feasible and infeasible system architectures are distinguished from each other. The results presented are not indicative of a definitive Genetic Algorithm optimizer failure, but rather an area to improve the problem formulation and hyperparameter selection for future benchmarking studies.

Previously, three primary tradeoffs were introduced, serving as criteria to compare the

optimizers' performance against each other, and included the: 1) ability to explore the design space fully; 2) ability to filter out infeasible designs; and 3) computational efficiency to attain a solution. In the Simulated Annealing optimizer, only one configuration is evaluated during each major iteration, resulting in a sparse exploration of the design space, while the Genetic Algorithm optimizer evaluates multiple configurations in each generation, resulting in a more thorough exploration of the design space. The problem-specific heuristics embedded in the NEIGHBOR FUNCTION enable the Simulated Annealing optimizer to strategically search the design space while accommodating randomness and exploring different regions of the design space. However, if formulated improperly, the heuristics may fail to identify the best solution. In contrast, the Genetic Algorithm optimizer utilizes problem-agnostic heuristics, selecting the configurations with the best fitness in each generation, resulting in a less strategic search of the design space that does not leverage any prior knowledge about the problem.

The Simulated Annealing optimizer filters out infeasible designs while searching for the next configuration to analyze in the NEIGHBOR FUNCTION. In contrast, the Genetic Algorithm optimizer utilizes a penalty function to evaluate all individuals [207], penalizing infeasible ones without eliminating them from the population. Thus, there is a tradeoff in the computational efforts between optimization algorithms. The Simulating Annealing optimizer spends more effort filtering out infeasible designs, running the POWER MANAGEMENT OPTIMIZATION on feasible ones only; the Genetic Algorithm optimizer runs the POWER MANAGEMENT OPTIMIZATION on each individual in the population every generation, using less effort to find and filter out infeasible configurations.

6.4.3 Conclusions

In Experiment 4.0.2, the Simulated Annealing optimizer was benchmarked against a Genetic Algorithm optimizer to solve the PROPULSION ARCHITECTURE OPTIMIZATION problem in the bi-level optimization framework constructed in Section 6.2. Both optimizers were run for 50 iterations, enabling a comparison of the best propulsion system architectures selected. The benchmarking study also enabled a comparison of the tradeoffs between utilizing Simulated Annealing and Genetic Algorithm optimizers.

While both optimizers returned safety-feasible, Pareto-optimal propulsion system architectures, the Simulated Annealing optimizer returned a configuration that further minimized the objective function than that of the Genetic Algorithm optimizer. The Simulated Annealing optimizer spent more time filtering out infeasible designs and strategically searching the design space while the Genetic Algorithm optimizer spent more time optimizing the power

management strategy on all propulsion system architectures in each generation, regardless of whether they were physically tractable or not. Thus, computational resources in the Simulated Annealing optimizer were focused on assembling safety-feasible propulsion system architectures, overcoming the challenge of evaluating few propulsion system architectures relative to the size of the combinatorial design space. Additionally, the unconstrained optimization problem solved by the Genetic Algorithm optimizer may have impacted its ability to optimize the propulsion system architecture for minimum fuel burn, as all violated constraints penalized the objective function value. An improvement in the objective function value in the Genetic Algorithm optimizer may not have corresponded to a decrease in fuel burn, but rather an improvement towards reaching a safety-feasible propulsion system architecture.

Overall, Experiment 4.0.2 revealed that utilizing embedded problem-specific heuristics synthesizes propulsion system architectures with improved system-level performance than those designed with stochastic optimization methods that utilize generalized, problem-agnostic heuristics. The results of Experiment 4.0.2 justified the choice to construct the bi-level optimization framework with the Simulated Annealing optimizer and developing problem-specific heuristics to strategically traverse the design space.

6.5 Optimization Framework and Heuristic Robustness (Experiment 4.0.3)

Experiments 4.0.1 and 4.0.2 validated the bi-level optimization framework’s ability to assemble safety-feasible propulsion system architectures with improved system-level performance than those design with a fixed system architecture or stochastic optimization methods that utilize generalized, problem-agnostic heuristics. An additional study is necessary to test the robustness of the heuristics embedded in the bi-level optimization framework, ensuring that the optimized propulsion system architecture is weakly sensitive to the initial guess input to the PROPULSION ARCHITECTURE OPTIMIZER. Experiment 4.0.3 tests the robustness of the optimizer and heuristics by solving the same optimization problem for 50 different initial propulsion system architecture guesses and comparing the objective function values output from the bi-level optimization framework.

6.5.1 Experimental Setup

Using the bi-level optimization framework constructed in Section 6.2, MATLAB's random number generators utilized to add/remove components were seeded with an integer value between 1 and 50. Each seed causes different random numbers to be selected, effectively generating different initial propulsion system architectures to be evaluated in the bi-level optimization framework.

The Simulated Annealing Optimizer runs using the system safety constraints from Tests A and E in Section 6.3, corresponding to a 1-EP failure mode with a maximum allowable failure rate of 10^{-02} /hr and a 4-EP failure mode with a maximum allowable failure rate of 10^{-05} /hr, respectively. The Simulated Annealing Optimizer runs utilizing the same hyperparameters in Experiments 4.0.1 and 4.0.2 (refer to Tab. 6.12 from Section 6.3).

6.5.2 Results

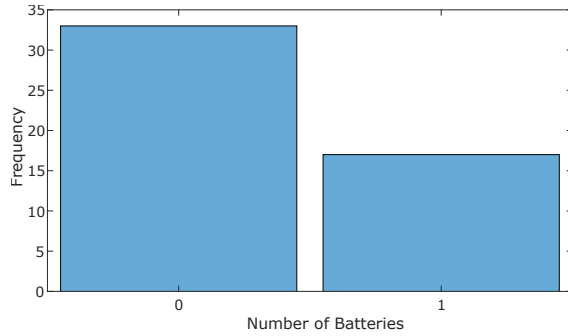
To examine the robustness of the heuristics and bi-level optimization framework, the component counts, levels of redundancy, and objective function values were examined.

6.5.2.1 Test A

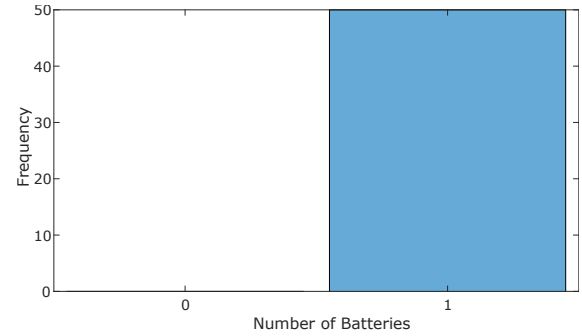
From Test A in Section 6.3, the best propulsion system architecture was the baseline configuration (each electric motor was powered by one electric generator), resulting in a fuel burn of 13,801 kg and system-level failure rate of 2.00×10^{-03} /hr.

Figures 6.9 through 6.11 list the number of batteries, electric generators, and electric motors in each propulsion system architecture. Across the initial propulsion system architecture guesses, the majority do not include a battery. The number of electric generators utilized in the initial propulsion system architectures vary, with two being the most common. A majority of propulsion systems also utilize 2 electric motors, also resulting in two ducted fans installed onboard at the beginning of the optimization.

After optimizing, all configurations utilize a battery, four electric generators, and 16 electric motors, resulting in a distributed propulsion system. While the maximum allowable number of components are present, the result does not indicate the level of redundancy in the propulsion system, possibly resulting in different configurations with the same number of components, but different levels of redundant connections. To verify that the optimizer converged on the baseline propulsion system architecture, the level of redundancy was also plotted in Fig. 6.12 across the initial and optimized propulsion system architectures. Initially,

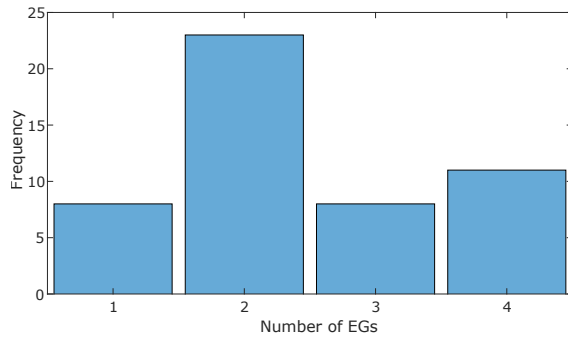


(a) Initial guesses.

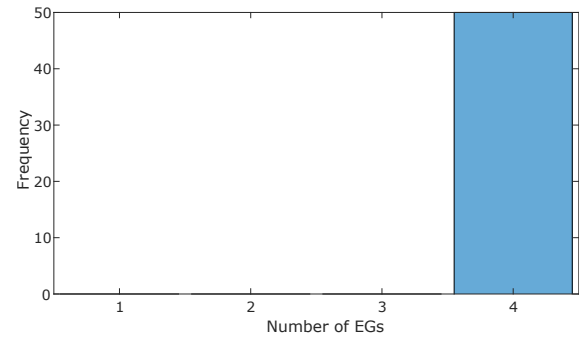


(b) Optimized propulsion system architectures.

Figure 6.9: Batteries included in the propulsion system architectures.



(a) Initial guesses.

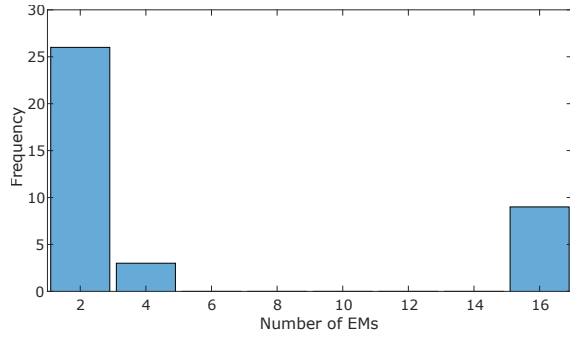


(b) Optimized propulsion system architectures.

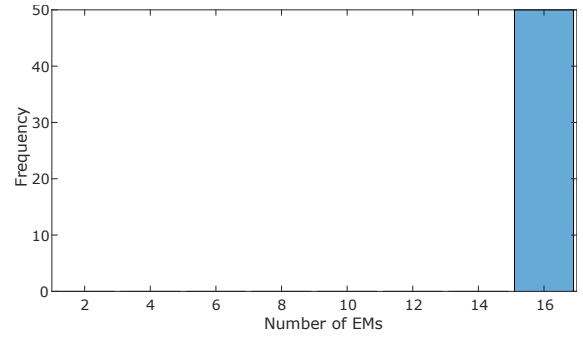
Figure 6.10: Electric generators included in the propulsion system architectures.

about half of the propulsion system architectures contain redundant connections. After optimizing, all configurations have one level of redundancy, corresponding to the baseline propulsion system architecture expected from Test A in Section 6.3.

To further confirm that the bi-level optimization framework converged on the baseline propulsion system architecture, the initial and optimized fuel burn distributions are plotted in Fig. 6.13. Initially, the fuel burn varied between 13,600 to 15,000 kg. After optimization, the fuel burn distribution is centered around 13,800 kg, remaining tightly bound. The final fuel burn distribution must be examined closer to assess the distribution of optimized values, as illustrated in Fig. 6.14.

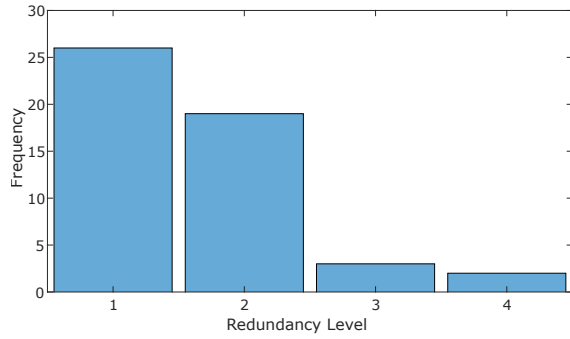


(a) Initial guesses.

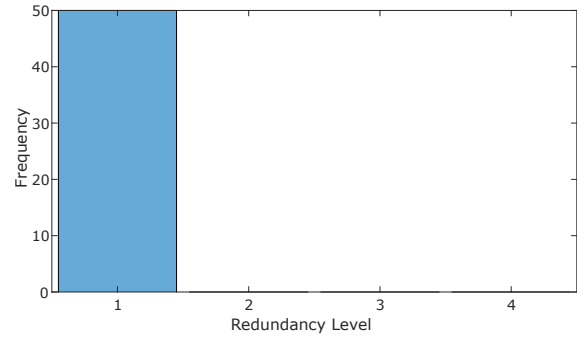


(b) Optimized propulsion system architectures.

Figure 6.11: Electric motors included in the propulsion system architectures.



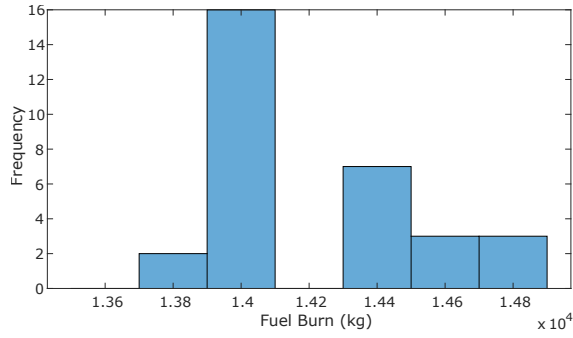
(a) Initial guesses.



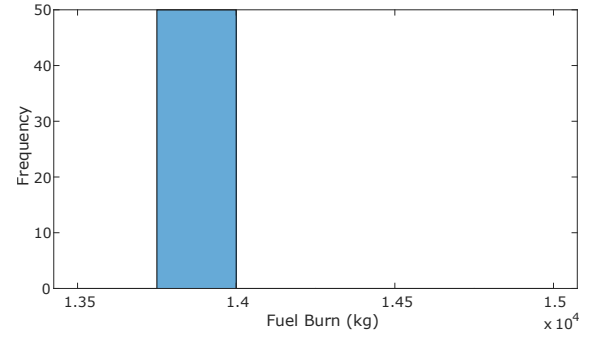
(b) Optimized propulsion system architectures.

Figure 6.12: Redundancy levels included in the propulsion system architectures.

After magnifying the final fuel burn distribution in Fig. 6.14, the distribution is centered around 13,801 kg, matching the optimized fuel burn for the baseline propulsion system architecture. The absolute difference between the minimum and maximum fuel burn within the distribution is 0.06 kg, reflecting a $4 \times 10^{-4}\%$ relative difference, suggesting that the optimizer converges to approximately the same objective function value in each run. The small relative difference indicates that the best propulsion system architecture is weakly sensitive to the initial propulsion system architecture guessed.



(a) Initial guesses.



(b) Optimized propulsion system architectures.

Figure 6.13: Fuel burn distribution.

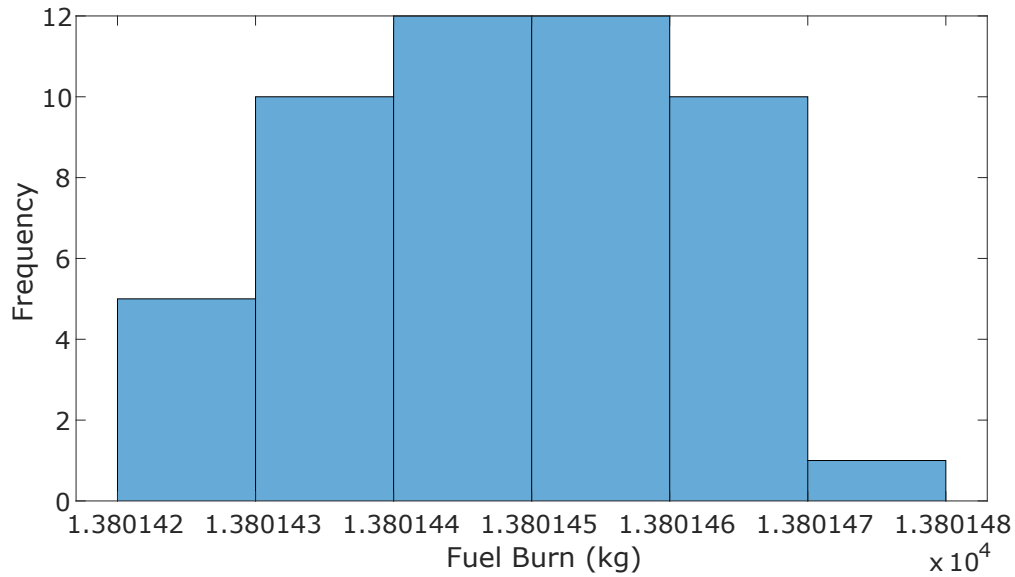
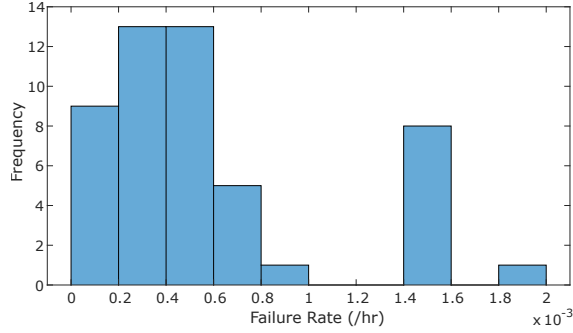
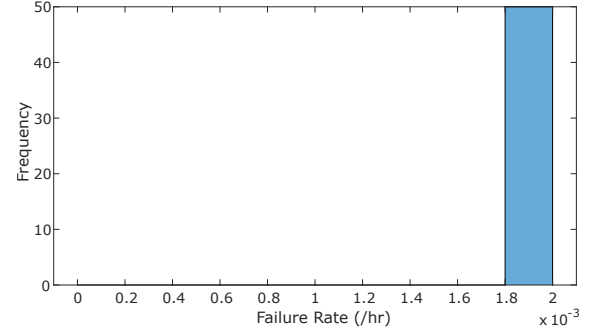


Figure 6.14: Magnified final fuel burn distribution.

The marginal differences in the optimized fuel burn can be attributed to the Cable Routing Optimizer converging on “near-equivalent topologies,” the number of components remains the same but the connections between similarly typed components change. In a near-equivalent topology for the baseline propulsion system architecture, each electric generator connects to four electric motors, but the connections between the generators and motors vary. As a result, the cables connecting the electric generators and motors are marginally different, causing slight variations in system-level performance.



(a) Initial guesses.



(b) Optimized propulsion system architectures.

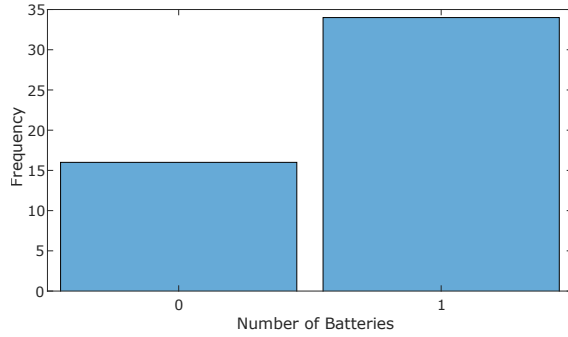
Figure 6.15: Failure rate distribution.

Despite the slight variations in system-level performance, system-level safety does not vary between the near-equivalent topologies, with each optimized propulsion system architecture exhibiting a failure rate of 2.00×10^{-3} /hr. Again, the identical failure rates indicate that the optimized propulsion system architecture is weakly sensitive to the initial propulsion system architecture guessed.

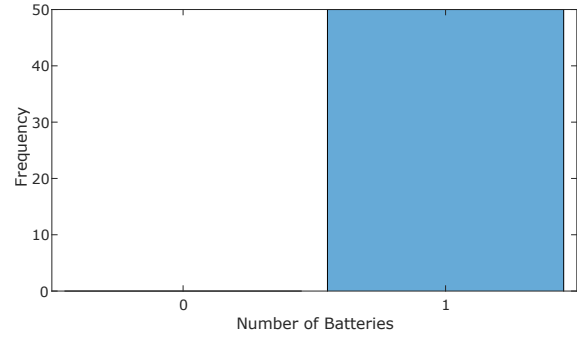
Initially, the system-level failure rate varies, and is lower if redundant electric generator-motor cross-connections are included. However, after optimizing, there is no variation in the system-level failure rate. Any near-equivalent topology of the baseline propulsion system architecture has two primary failure modes causing a 1-EP failure: 1) a single electric generator failure; or 2) a single electric motor failure. The same result would occur if the test were run for a 2-EP failure mode. As a result of both failure modes relying on one (or two) electric motor(s) to fail, the cable routing has no impact on system safety. However, if the test were run on a 4-EP failure mode, the cable routing would impact system safety because the redundant connections require different numbers of electric generators to fail before four electric motors fail. Thus, system safety could vary by multiple orders of magnitude for the same system-level performance (fuel burn). In the cable routing shown in Fig. 6.8 from Section 6.3, the asymmetric cable connections require three electric generators to fail before four electric motors fail, as opposed to two in the fixed Double Redundancy system architecture introduced in Section 6.3.2.

6.5.2.2 Test E

From Test E in Section 6.3, the best propulsion system architecture was the double redundancy configuration (each electric motor was powered by two electric generators), resulting in a fuel burn of 14,384 kg and system-level failure rate of 3.04×10^{-09} /hr. Figures 6.16 through 6.18 list the number of batteries, electric generators, and electric motors in each propulsion system architecture.

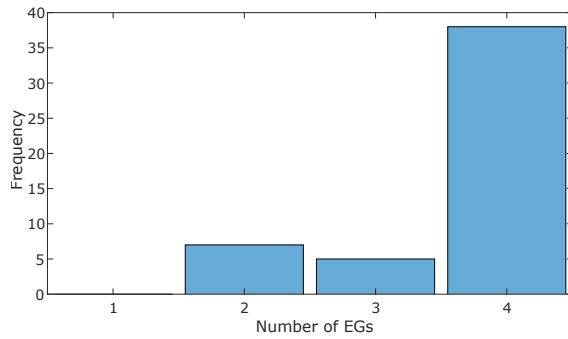


(a) Initial guesses.

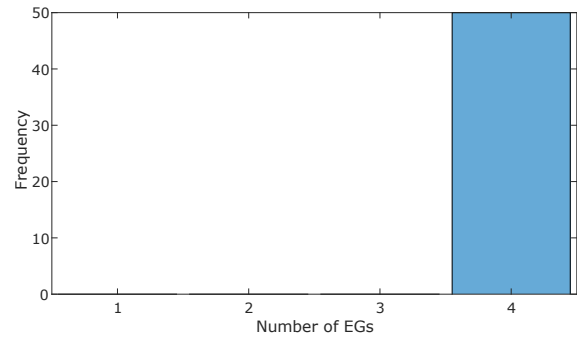


(b) Optimized propulsion system architectures.

Figure 6.16: Batteries included in the propulsion system architectures.

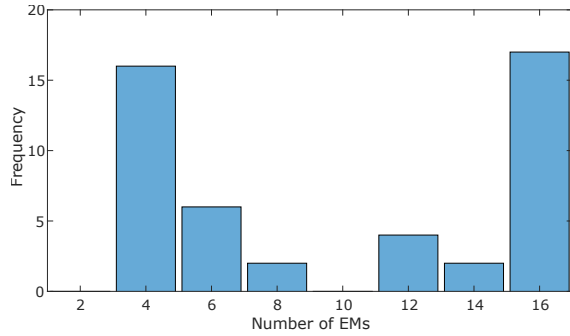


(a) Initial guesses.

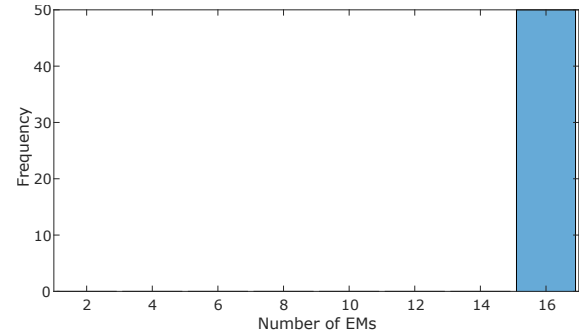


(b) Optimized propulsion system architectures.

Figure 6.17: Electric generators included in the propulsion system architectures.



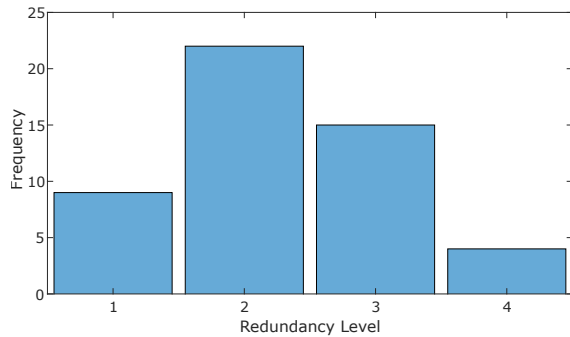
(a) Initial guesses.



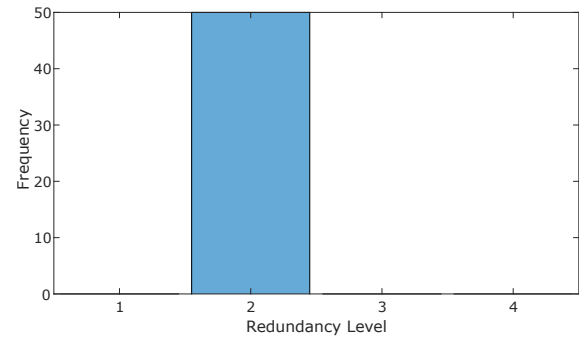
(b) Optimized propulsion system architectures.

Figure 6.18: Electric motors included in the propulsion system architectures.

After optimizing, all configurations utilize a battery, four electric generators, and 16 electric motors, resulting in a distributed propulsion system. Again, the result does not indicate the level of redundancy in the propulsion system, and is plotted in Fig. 6.19 to confirm that a Double Redundancy propulsion system architecture is consistently selected.



(a) Initial guesses.



(b) Optimized propulsion system architectures.

Figure 6.19: Redundancy levels included in the propulsion system architectures.

To further confirm that the bi-level optimization framework converged on the Double Redundancy propulsion system architecture, the initial and optimized fuel burn distributions are plotted in Fig. 6.20. Initially, the fuel burn varied between 13,600 to 14,800 kg. After optimization, the fuel burn distribution is centered around 13,384 kg, remaining tightly bound. The final fuel burn distribution must be examined closer to assess the distribution of

optimized values, as illustrated in Fig. 6.21.

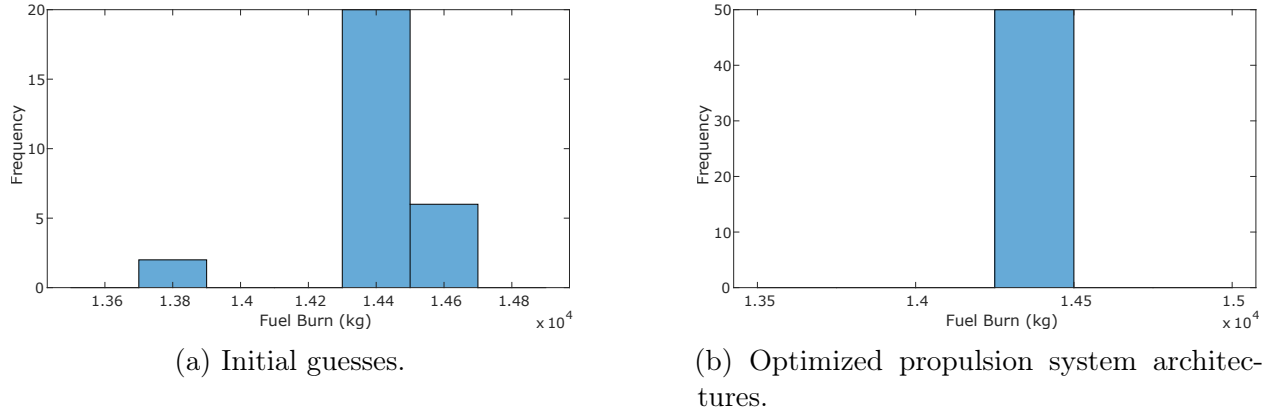


Figure 6.20: Fuel burn distribution.

After magnifying the final fuel burn distribution in Fig. 6.21, the distribution is centered around 14,384 kg, matching the optimized fuel burn for the Double Redundancy propulsion system architecture. The absolute difference between the minimum and maximum fuel burn within the distribution is 0.10 kg, reflecting a $7 \times 10^{-4}\%$ relative difference, suggesting that the optimizer converges to approximately the same objective function value in each run. The small relative difference again indicates that the best propulsion system architecture is weakly sensitive to the initial propulsion system architecture guessed. Again, the marginal differences in the optimized fuel burn can be attributed to the Cable Routing Optimizer converging on “near-equivalent topologies,” causing slight variations in system-level performance.

However, under the 4-EP failure mode, system-level safety varies between the near-equivalent topologies, with failure rates of 3.04×10^{-09} , 1.96×10^{-07} , and 3.91×10^{-07} /hr after the sensitivity study. Although the near-equivalent topologies return different system-level failure rates, the total cable length to connect the electric generators and motors are all within a relative difference of 0.0557%. Figure 6.23 illustrates that the CABLE ROUTING OPTIMIZATION converges to similar cable lengths for each of the 50 optimizations run, while Fig. 6.24 highlights the distribution of cable lengths that the optimizer converged to.

Before minimizing the total cable length, the PROPULSION ARCHITECTURE OPTIMIZER selects the necessary electric generator-motor cross-connection scheme that satisfies the system-level safety constraints, providing an upper bound for the propulsion system architecture’s system-level failure rate. In the CABLE ROUTING OPTIMIZATION, the embedded heuristics randomly change the connections between the electric generators and motors, focusing

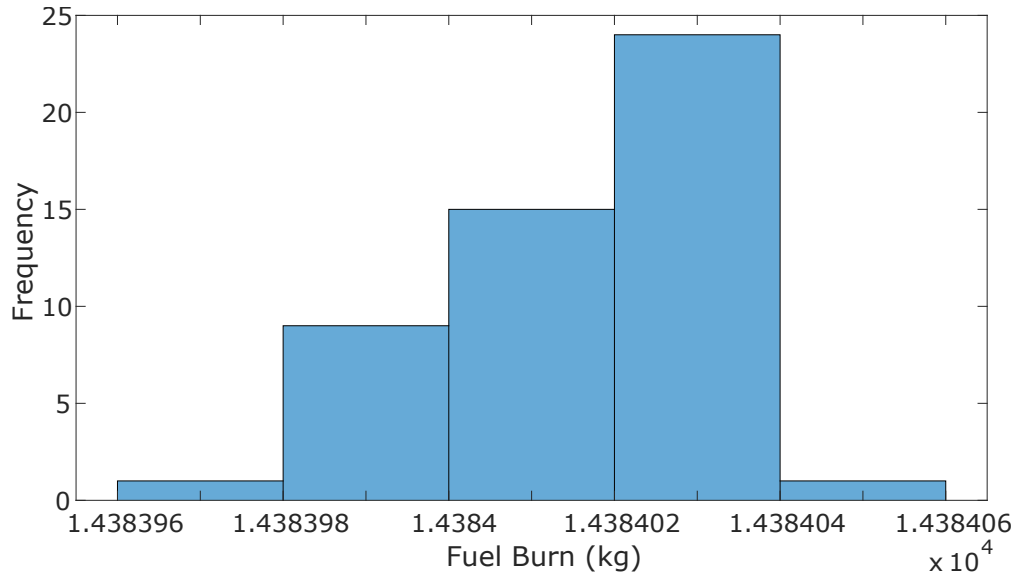
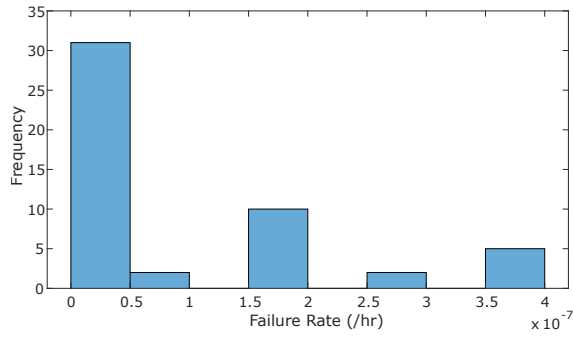
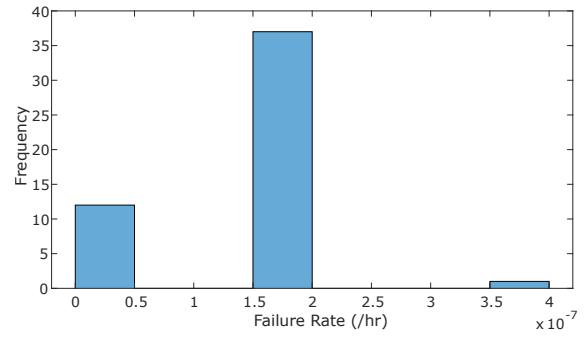


Figure 6.21: Magnified final fuel burn distribution.



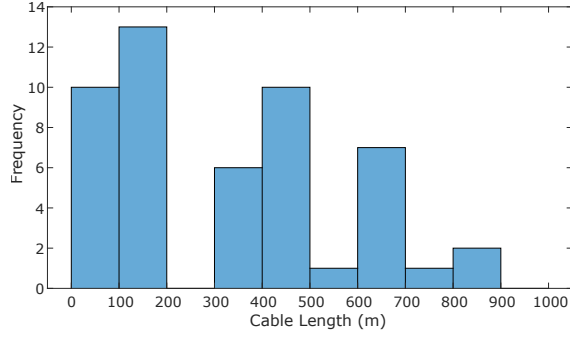
(a) Initial guesses.



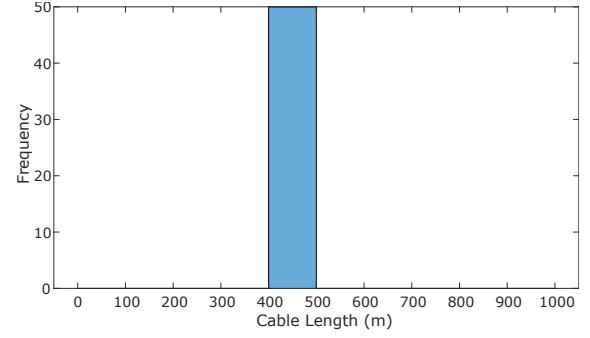
(b) Optimized propulsion system architectures.

Figure 6.22: Failure rate distribution.

on minimizing the total cable length without considering system safety. As a result, the optimized electric generator-motor cross-connection scheme may exhibit system-level safety improvements but will always satisfy the system-level safety constraints. Based on this investigation, system-level performance and the cable length are weakly sensitive to the CABLE ROUTING OPTIMIZATION, while system-level safety remains sensitive to the optimized electric generator-motor cross-connection schemes.



(a) Initial guesses.



(b) Optimized propulsion system architectures.

Figure 6.23: Cable length distribution.

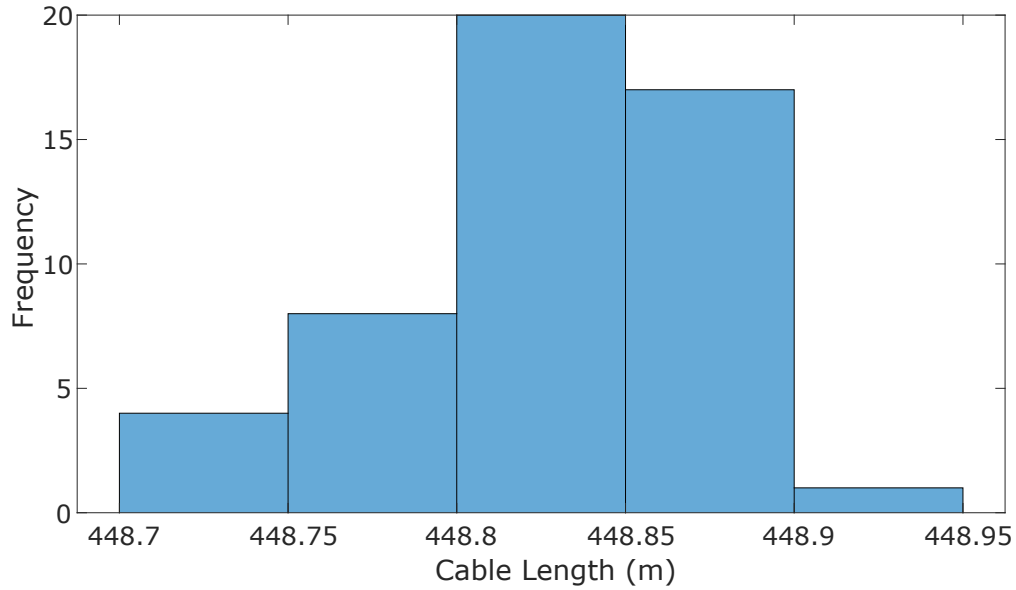


Figure 6.24: Magnified final cable length distribution.

6.5.3 Conclusions

In Experiment 4.0.3, the robustness of the bi-level optimization framework and its embedded, problem-specific heuristics were assessed on two different test cases. By seeding the random number generators in MATLAB, 50 different initial propulsion system architectures were constructed and optimized in the bi-level optimization framework. The initial propulsion system architectures had 1 to 4 electric generators, 2 to 16 electric motors, varying levels of

redundant connections, and, in some cases, a battery to boost the electric motor power. In both test cases, the initial fuel burn distributions spanned 13,600 to 15,000 kg. For the 1- and 4-EP failure modes, the system-level failure rates spanned values up to 2.00×10^{-3} and 4.00×10^{-7} , respectively.

For the 1-EP failure mode, the optimum propulsion system architecture returned from each run was the baseline propulsion system architecture, resulting in a fuel burn of approximately 13,801 kg and a failure rate of 2.00×10^{-3} /hr. Due to the randomness in the heuristics and the Cable Routing Optimizer, each optimized propulsion system architecture contains the same number of components, but not necessarily the same connections between similarly typed components. As a result of slightly different cable connections between the electric generators and motors, the fuel burn varies by 0.06 kg, a $4 \times 10^{-4}\%$ difference, between the largest and smallest values. The cable connections, however, do not impact system safety for the 1-EP failure mode.

In contrast, the 4-EP failure mode always returned a Double Redundancy propulsion system architecture with similar system-level performance – a fuel burn of approximately 14,384 kg and a total cable length of about 448.85 m – but varied system-level safety. The system-level failure rates returned spanned values between 3.04×10^{-09} to 3.91×10^{-07} /hr, resulting in significantly different failure probabilities. Although the CABLE ROUTING OPTIMIZER only focused on minimizing the total cable length, system safety was sometimes improved based on the initial propulsion system architecture provided. The propulsion system architectures designed under a 4-EP failure mode were significantly impacted because the electric generator-motor cross-connection schemes impact whether or not some of the electric propulsors remain operational after an electric generator failure. Thus, the fuel-minimizing propulsion system architecture synthesized is weakly sensitive to the initial propulsion system architecture guessed, indicating that the problem-specific heuristics in the bi-level optimization framework are robust, but may exhibit varied system safety characteristics.

6.6 Chapter Summary

State-of-the-art system architecture synthesis methods have enabled broad exploration of combinatorial design spaces, identifying the best system architectures. Such methods include graph-based enumeration and exhaustive search techniques, multidisciplinary design optimization, and Constraint Programming (or solving a Boolean Satisfiability Problem). While graph-based enumeration techniques are efficient, a computational bottleneck emerges if all

system architectures generated must be analyzed in a physics-based model, even after filtering out infeasible propulsion system architectures. Multidisciplinary design optimization is useful for searching the multi-modal design space strategically, requiring problem-specific heuristics to be computationally efficient. Constraint Programming enables *feasible* system architectures to be identified, but are not guaranteed to be *optimal*, resulting in additional computational overhead to optimize the component sizes. Beyond the aforementioned shortcomings, system safety analyses have not been incorporated into conceptual design workflows, opening a larger gap in the literature for identifying and optimizing safety-feasible propulsion system architectures.

To address the gaps in the literature, Research Question 4.0 explored how propulsion system architecture topologies and corresponding power management strategies could be synthesized to optimize the system-level performance and safety of an aircraft concept. Hypothesis 4.0 addressed Research Question 4.0 by posing that a bi-level architecture/power management optimization framework that assembles safety-feasible propulsion system architectures from a predefined component library utilizing embedded, problem-specific heuristics would synthesize propulsion system architectures with improved system-level performance than those design with fixed system architecture or stochastic optimization methods that utilize generalized, problem-agnostic heuristics.

To test Hypothesis 4.0, a bi-level optimization framework was constructed to synthesize advanced propulsion system architectures and their corresponding power management strategy. The PROPULSION ARCHITECTURE OPTIMIZATION problem acts as the high-level problem, identifying a safety-feasible propulsion system architecture that minimizes the fuel burn, and is solved using a Simulated Annealing optimization algorithm. Embedded within the PROPULSION ARCHITECTURE OPTIMIZATION problem is a POWER MANAGEMENT OPTIMIZATION problem, solving for the optimum power splits after *fixing* the propulsion system architecture using the SQP Method within MATLAB's FMINCON function. The hierarchical MDO problem is structured as an abstraction of a Multidisciplinary feasible MDO architecture; the propulsion system architecture connections and power splits represent the design and state variables, respectively. There are two novelties within the bi-level optimization framework: 1) the development of problem-specific heuristics that select candidate propulsion system architectures based on the objective function values and number of active components included in prior ones, as well as filtering out infeasible designs based on the systems architecting constraints; and 2) embedding a FAULT TREE ANALYSIS in the optimization framework, enabling safety-feasible system architectures to be identified.

After constructing the bi-level optimization framework, Experiments 4.0.1 through 4.0.3 tested its ability to synthesize safety-feasible propulsion system architectures against pre-selected fixed architectures and stochastic optimization methods utilizing problem-agnostic heuristics, and verified its robustness to identify the optimized propulsion system architecture for a variety of different initial guesses, respectively. In Experiment 4.0.1, propulsion system architectures were synthesized under multiple system-level safety constraints, resulting in a Pareto Frontier created for each propulsion system failure mode assessed. The redundant connections between the electric generators and motors served as a mechanism for trading system-level performance and safety objectives. The Pareto Frontier highlights that fuel burn is minimized by limiting redundant connections in the propulsion system, thus reducing the cable weight and size of the generators. In contrast, safety is maximized by adding redundant connections between the generators and motors in the propulsion system.

Then, the Pareto Hypervolume was computed for the Pareto Frontiers generated under 2- and 4-EP failure modes, enabling a comparison of the system-level performance and safety of the optimized propulsion system architectures against a fixed architecture design with power management optimization only. For the 2-EP failure mode, there was no significant difference between the Pareto Hypervolume corresponding to fixed architecture optimization and the bi-level optimization framework. However, for the 4-EP failure mode, the Pareto Hypervolume corresponding to the bi-level optimization framework was approximately 12.5% smaller than that corresponding to the fixed architecture design with power management optimization only. The reduction in Pareto Hypervolume was attributed to the Cable Routing Optimization embedded within the PROPULSION ARCHITECTURE OPTIMIZATION; unique, asymmetric electric generator-motor cross-connections improved system safety by multiple orders of magnitude without penalizing system-level performance.

In Experiment 4.0.2, two different optimizers were implemented to solve the PROPULSION ARCHITECTURE OPTIMIZATION problem: 1) the Simulated Annealing optimizer previously embedded in the bi-level optimization framework; and 2) a Genetic Algorithm optimizer. Both optimizers were run for the same number of iterations, enabling a comparison of the system-level performance and safety returned by each one. The Simulated Annealing optimizer more strategically searched the design space and synthesized a propulsion system architecture that minimized the fuel burn while satisfying the system safety constraints. In contrast, the Genetic Algorithm optimizer identified a safety-feasible propulsion system architecture but failed to minimize fuel burn. Additionally, the Genetic Algorithm optimizer required 40 to 60% more time, requiring additional computational resources to identify a

sub-optimal propulsion system architecture. The heuristics embedded in the Simulated Annealing optimizer enabled the design space to be searched strategically, spending more computational resources to identify promising designs and filter out infeasible ones. Since the Genetic Algorithm optimizer uses a penalty function to minimize the objective, more computational resources were spent on solving POWER MANAGEMENT OPTIMIZATION problems for each propulsion system architecture in the population, even if it was infeasible. Thus, the Genetic Algorithm optimizer may be sensitive to the penalty function formulated to solve the optimization problem, and should not be interpreted as a definitive failure of the Genetic Algorithm.

Lastly, Experiment 4.0.3 assessed the robustness of the bi-level optimization framework and its embedded heuristics, verifying that the optimized propulsion system architecture was weakly sensitive to the initial propulsion system architecture guessed. After running the bi-level optimization framework with 50 different initial guesses, the optimized system architecture contained the same number of components and level of redundancy between the electric generator-motor cross-connections. Although all 50 optimized propulsion system architectures contained the same number of components and connections, their topologies were not identical. Rather, they were *near-equivalent*, meaning that the same number of connections existed between similarly typed components, but the connections themselves were not identical. The different connections between the electric generators and motors were due to the randomness in the heuristics embedded in the Cable Routing Optimization, resulting in marginally different fuel burn values, with a relative difference of $4.00 \times 10^{-4}\%$ between the largest and smallest ones. Under a 1-EP failure mode, system safety was not impacted by the Cable Routing Optimization, resulting in the same system-level failure rate being returned for each optimized propulsion system architecture. However, under a 4-EP failure mode, system safety was impacted by the Cable Routing Optimization, returning similar objective function values (fuel burn) despite different system-level failure rates.

Overall, the work to address Research Question 4.0 enhances the state-of-the-art by developing advanced MDO methods with problem-specific heuristics and an embedded system safety analysis to assemble propulsion system architectures that account for both system-level performance and safety objectives during conceptual design.

Chapter 7

Conclusions

7.1 Introduction

This thesis addressed the need for developing a unified, configuration-agnostic design framework to synthesize advanced aircraft propulsion architectures while accounting for both nominal and off-nominal performance considerations. In the literature review, five gaps were identified, indicating the lack of configuration-agnostic design environments for future aircraft concepts and their advanced propulsion system architectures, as well as poor applicability of existing off-nominal flight performance requirements. Additionally, system safety was not traditionally considered during conceptual design, but has become necessary due to the many possible failure modes that advanced propulsion system architectures may exhibit. The research gaps identified in Chapter 2 were encapsulated in Research Question 1.0, serving as the overarching research question:

RQ 1.0: How can a unified conceptual aircraft design framework represent, evaluate, and synthesize advanced aircraft propulsion architectures while accounting for nominal aircraft performance, off-nominal propulsion system failure scenarios, and system-level safety constraints in a configuration-agnostic manner?

Hypothesis 1.0 then proposed the necessary elements in a computational framework to synthesize advanced propulsion system architectures in a configuration-agnostic manner and compare them against previously optimized baseline designs:

Hyp 1.0: A framework combining graph-based propulsion architecture analyses, off-nominal climb performance constraints based on specific excess power losses under critical failure modes, and joint optimization of the architecture topology and its corresponding power management strategy will identify architectures with a smaller Pareto Hypervolume

than baselines designed with a fixed architecture or stochastic optimization method with problem-agnostic heuristics.

Research Question 1.0 was then divided into three different research questions, with each one focusing on a critical technological advancement necessary to develop a configuration-agnostic conceptual aircraft design framework:

RQ 2.0: How can propulsion system architectures be formally represented and analyzed to quantify system-level tradeoffs in a configuration-agnostic design environment?

RQ 3.0: How can the currently published FAA climb performance requirements for aircraft with discrete, homogeneously sized propulsors be generalized to accommodate propulsion systems with heterogeneously sized or distributed propulsors and their corresponding critical failure modes?

RQ 4.0: How can propulsion system architecture topologies and corresponding power management strategies be synthesized to optimize the system-level performance and safety of an aircraft concept?

7.1.1 Chapter Outline

Research Questions 2.0, 3.0, and 4.0 are answered first in Section 7.2, followed by an answer to Research Question 1.0. After that, key lessons learned for designing future aircraft concepts are listed in Section 7.3. Then, all technical contributions made in this thesis are described in Section 7.4. Lastly, areas for future work are discussed in Section 7.5, highlighting areas for further advancements to the state-of-the-art and applications beyond the work presented in this thesis.

7.2 Answers to Research Questions

7.2.1 Research Question 2.0: Graph-based Propulsion System Analysis Framework and Parametric Aircraft Sizing Tool

To answer Research Question 2.0, Hypothesis 2.0 outlined a structure for a graph-based propulsion system analysis framework and proposed comparisons against state-of-the-art benchmarks:

Hyp 2.0: A graph-based representation with component constitutive models at each vertex and power transfer relationships along each edge will reproduce benchmark component power loads and overall energy conservation equations across representative conventional and electrified propulsion architectures within predefined accuracy thresholds while reducing topology-specific model setup effort and execution time relative to existing graph-based methods and propulsion architecture-specific analysis implementations.

The GPSA Framework was developed to flexibly and robustly model a wide range of propulsion system architectures. After constructing the GPSA Framework, Experiment 2.0.1 tested Hypothesis 2.0 by validating the GPSA Framework construction on multiple advanced propulsion system architectures, and benchmarking the framework’s modeling accuracy and model setup efforts against existing state-of-the-art propulsion system analysis frameworks. The results from Experiment 2.0.1 partially supported Hypothesis 2.0. Although advanced propulsion system architectures were modeled with improved accuracy relative to prior state-of-the-art methods, topology-specific model setup increased and the execution time were not significantly reduced.

Improved modeling accuracy was attained with the GPSA framework by mathematically abstracting the propulsion system architecture as a graph, enabling any propulsion system architecture topology, power management strategy, and power flow losses to be simulated in a configuration-agnostic manner. The fixed-point iteration included in Alg. 1 enabled power to be transmitted between an *unlimited* number of components across different power flow paths, overcoming limitations imposed by prior numerical algorithms and propulsion systems architecture generalizations. However, the topology-specific model setup increased relative to prior state-of-the-art propulsion system analysis methods. Despite the increased topology-specific model setup, there was no significant difference between the wall time required to simulate a power flow using either the GPSA Framework relative to prior state-of-the-art methods.

Thus, the GPSA Framework’s graph-based representation with component constitutive models at each vertex and power transfer relationships along each edge reproduced benchmark component power loads and overall energy conservation equations across representative conventional and electrified propulsion architectures within predefined accuracy thresholds. Thus, propulsion system architectures can be formally represented and analyzed using a typed graph representation with component constitutive models at each vertex and power transfer relationships along each edge, accommodating any topological arrangement of components and corresponding power management strategies. Additionally, power flows in the propulsion

system architectures can be simulated using a fixed point iteration, enabling bi-directional power flows and novel power management strategies to be analyzed.

The GPSA Framework was then integrated into the Future Aircraft Sizing Tool (FAST) to quantify system-level tradeoffs between aircraft designs with different propulsion system architectures and power management strategies. Using the Embraer ERJ175LR as a baseline, the aircraft was resized with four different electrified propulsion system architectures while maintaining the same top-level aircraft requirements, technology levels, and design mission profile. Under the assumptions made in the study, two key tradeoffs were identified. First, the aircraft sized with the parallel hybrid propulsion system saved fuel by downsizing the gas turbine engine and operating it at a higher throttle setting during cruise (reducing its specific fuel consumption) while boosting its power with an electric motor during takeoff. Second, any aircraft with a gas turbine engine, electric generator, and electric motor in series incurred a significant weight penalty because the electric motor needed to generate all power necessary to spin the propeller. Overall, the study demonstrated that system-level tradeoffs can be quantified by integrating the typed graph representation and power flow simulation algorithm in a parametric aircraft sizing tool that utilizes an energy-based mission analysis and first-order physics-based models to approximate the aircraft as a point mass.

More broadly, tradeoffs between architectures were systematically quantified because the GPSA Framework decouples the propulsion system architecture, operational strategy, and component efficiency definitions from each other. As a result, large-scale design space exploration and sensitivity studies can be seamlessly executed, providing information about the most promising aircraft configurations as well as the design parameters that have the largest impact on system-level performance. Together, the information from the sensitivity studies can inform the aircraft design process, highlighting areas for present and future technology investment.

7.2.2 Research Question 3.0: Generalized Engine Inoperative Performance Requirements

To answer Research Question 3.0, Hypothesis 3.0 proposed a method to generalize the currently published FAA climb performance requirements for aircraft with discrete, homogeneously sized propulsors, accommodating propulsion systems with heterogeneously sized or distributed propulsors:

Hyp 3.0: A monotonic, bounded mapping between the specific excess power loss and the required climb gradient will reproduce the climb gradient requirements implied by the published regulations for conventional aircraft utilizing discrete, homogeneously sized propulsors while extending the same flight performance logic to accommodate aircraft concepts utilizing propulsion systems with heterogeneously sized or distributed propulsors.

Future flight performance requirements for future aircraft concepts must be parameterized in a configuration-agnostic manner, accommodating propulsion systems powered by both discrete, homogeneously-sized propulsors, as well as heterogeneously-sized or distributed propulsors. The specific excess power was found to be a suitable parameter, providing a physically consistent mapping between the propulsion system failure severity and off-nominal climb performance requirements, supporting the claims made in Hypothesis 3.0. The specific excess power captures the aero-propulsive impacts of a propulsion system failure, accounting for the installed power lost, windmilling drag, and trim drag. Other parameters, such as the installed power available loss, are less suitable because they only account for the propulsive impacts of a propulsion system failure and neglect all aerodynamic considerations.

Using the aforementioned parameterization, the climb performance requirements can be constructed by mapping the nominal specific excess power losses of conventional aircraft concepts to their FAA-published climb performance requirements. The Sigmoid Curves utilized in the construction provided a monotonic, bounded mapping, ensuring that the same flight performance logic applied to conventional aircraft concepts was extended to aircraft concepts with advanced propulsion systems: as the propulsion system failure severity increases, the required climb gradient decreases. Using the aforementioned requirement construction, the presently-published climb performance constraints were reproduced with 0.15% and 0.015% accuracy for 2- and 4-engine conventional aircraft, respectively, supporting the claim made in Hypothesis 3.0. Since 3-engine aircraft are less prevalent in today's global airline fleet, the Sigmoid Curve constructions were deemed reasonable, even though the presently-published climb gradients were reproduced with less accuracy than those of the 2- and 4-engine aircraft.

Overall, parameterizing the flight performance requirements and engine inoperative correction factor as a function of the specific excess power loss provides a foundation for sizing the propulsion system as a function of various primary failure modes. While the design heuristics developed do not replace the certification requirements, they provide a first-order estimate of the necessary flight performance requirements in a configuration-agnostic manner,

supporting the design of future aircraft concepts and their advanced propulsion systems. More specifically, trade studies can be executed to identify possible primary propulsion system failure modes that maximize an aircraft’s system-level performance, informing propulsion system architecting decisions throughout conceptual design.

7.2.2.1 Research Question 3.1: Advanced Aircraft Sizing with Generalized Flight Performance Requirements

While Research Question 3.0 addressed the need for configuration-agnostic flight performance requirements and an engine inoperative correction factor, an additional research question was raised, focusing on the system-level impacts of sizing an advanced aircraft concept with the requirement constructions under Experiments 3.0.1 through 3.0.3:

Research Question 3.1: What are the system-level impacts of sizing an advanced aircraft concept with the generalized climb performance requirements and engine inoperative correction factor relative to one sized under FAA-published requirements and traditional propulsion system scaling factors for the same design mission?

To answer Research Question 3.1, Hypothesis 3.1 described the expected system-level impacts of sizing an aircraft with the generalized climb performance requirements and engine inoperative correction factor:

Hypothesis 3.1: For an advanced aircraft concept with a propulsion system failure mode that produces a smaller specific excess power loss than a conventional aircraft under a legacy one-engine inoperative failure mode, sizing under the generalized climb performance requirements and engine inoperative correction factor will reduce over-sizing and improve system-level performance while maintaining the required off-nominal climb gradient for the corresponding failure mode.

Hypothesis 3.1 was tested by sizing an Elysian E9X, a novel battery-electric aircraft concept leveraging distributed electric propulsion, under both the existing FAA-published requirements and the generalized flight performance requirements constructed to answer Research Question 3.0. For propulsion system failures that produced a smaller specific excess power loss than legacy one-engine inoperative failure modes, the design point (power loading and wing loading) remained the same as if the aircraft were designed with the FAA-published climb requirements for a 4-engine aircraft and traditional $n/(n - 1)$ engine

inoperative correction factor. Constraints corresponding to the takeoff field length and all-engine operative climb gradient remained active, preventing the propulsion system from being downsized. However, if the operational constraints were neglected and the climb constraints were active, then the propulsion system can be downsized.

The results from the sizing study indicate that Hypothesis 3.1 is partially supported. For an advanced aircraft concept with a propulsion system failure mode that produces a smaller specific excess power loss than a legacy one-engine inoperative failure mode, sizing under the generalized climb performance requirements and engine inoperative correction factor reduces over-sizing, but system-level performance improvements are not guaranteed. If the power loading is significantly reduced, system-level performance is hampered due to the additional time required to climb to cruise altitude.

There are two key implications for sizing future aircraft concepts under novel propulsion system failure modes. First, if the generalized climb performance constraints are active and the design space widens, additional optimization or trade studies may be necessary to find a power loading that improves system-level performance. Traditionally, the maximum power loading is selected, minimizing the propulsion system size. However, for novel failure modes that are significantly less severe than legacy one-engine inoperative failure modes, maximizing the power loading may not maximize system-level performance. Therefore, the propulsion system size must be carefully selected to minimize the propulsor size without significantly degrading nominal flight performance during all-engine operative conditions.

Second, operational constraints may limit downsizing of future aircraft concepts with advanced propulsion systems. Although the novel failure modes may produce a smaller specific excess power loss than legacy one-engine inoperative failure modes, future aircraft concepts must remain compatible with existing airport infrastructure to be seamlessly integrated into the global airline fleet, particularly for meeting the required takeoff/landing field lengths. Additional operational constraints, such as flying Standard Instrument Departure (SID) and Standard Arrival (STAR) routes present in the National Airspace System, may also impact the ability to downsize a future aircraft concept's propulsion system.

7.2.3 Research Question 4.0: Bi-Level Optimization Framework

To answer Research Question 4.0, Hypothesis 4.0 proposed that a bi-level optimization framework can synthesize performance- and safety-optimal propulsion system architecture topologies and corresponding power management strategies for advanced aircraft concepts:

Hyp 4.0: A bi-level architecture/power management optimization framework that assembles safety-feasible propulsion system architectures from a predefined component library utilizing embedded, problem-specific heuristics will synthesize propulsion system architectures with improved system-level performance than those designed with fixed system architectures or stochastic optimization methods that utilize generalized, problem-agnostic heuristics.

A bi-level optimization framework was constructed to assemble safety-feasible propulsion system architectures and identify an optimal power management strategy for each architecture synthesized. To test Hypothesis 4.0, a model of NASA’s SUSAN aircraft was utilized to synthesize distributed series hybrid electric propulsion systems under 2- and 4-EP inoperative failure modes. Those synthesized under a 2-EP failure mode demonstrated similar system-level performance relative to the fixed architecture baselines, resulting in a 0.7% larger Pareto Hypervolume. In contrast, system-level performance improvements were realized for the propulsion system architectures synthesized under a 4-EP failure mode relative to the fixed architecture baselines, resulting in a 12.5% smaller Pareto Hypervolume. Thus, Hypothesis 4.0 is partially supported, as system-level performance improvements are possible, but not guaranteed, from synthesizing architectures using the bi-level optimization framework.

The system-level performance improvements realized under the 4-EP failure mode were due to the CABLE ROUTING OPTIMIZER selecting asymmetric electric generator-motor cross-connection schemes. Although the CABLE ROUTING OPTIMIZER minimized cable length and, hence, maximized system-level performance, the less intuitive cross-connection schemes also improved system-level safety without additional redundant electric generator-motor connections. System-level safety was improved because the novel cross-connection schemes required three electric generators to fail before the 4-EP failure mode was realized. In contrast, the symmetric, fixed architecture cross-connection schemes only required two electric generators to fail before the failure mode was realized. Thus, the PROPULSION ARCHITECTURE OPTIMIZATION and embedded CABLE ROUTING OPTIMIZER served as a mechanism for assembling safety-feasible propulsion system architectures using problem-specific heuristics with possibly improved system-level performance than those designed with fixed system architectures. Additionally, the novel cross-connection schemes identified from the CABLE ROUTING OPTIMIZER can serve as a baseline for improving system safety with minimal fuel burn penalty in future aircraft concepts leveraging distributed propulsion systems.

More broadly, the systems architecting study highlights the system-level safety advantages

of integrated distributed electric propulsion systems into future aircraft concepts. The repeated propulsor units within the system act as a built-in redundancy, bolstering system safety while facilitating a more graceful degradation of the propulsion system’s performance. However, depending on the failure mode selected, the redundancy offered by the distributed propulsor arrays may not be satisfactory, requiring additional redundancy mechanisms such as the electric generator-motor cross-connections.

The Simulated Annealing optimizer utilized in the PROPULSION ARCHITECTURE OPTIMIZATION problem searched the design space more strategically than the Genetic Algorithm optimizer, suggesting that the use of problem-specific heuristics may be more advantageous than problem-agnostic ones. While both optimizers assembled safety-feasible propulsion system architectures, the one assembled under the Simulated Annealing optimizer exhibited improved system-level performance (reduced fuel burn) than that assembled under the Genetic Algorithm optimizer. Although the design space was explored more sparsely and fewer configurations were analyzed at the aircraft-level, the Simulated Annealing optimizer leveraged the embedded, problem-specific heuristics to filter out infeasible designs and exploit the attributes of promising designs identified in prior iterations. In contrast, the Genetic Algorithm optimizer explored a larger portion of the design space by evaluating multiple individuals in each generation, but spent more computational resources on solving the POWER MANAGEMENT OPTIMIZATION problem for each individual. Additionally, the objective function in the Genetic Algorithm optimizer was penalized for infeasible propulsion system architectures. As a result, improvements in the objective function may not have corresponded to a decreased fuel burn, but rather an improvement towards reaching a safety-feasible propulsion architecture, obfuscating optimal designs from feasible ones. Thus, the benchmarking study supported the claims in Hypothesis 4.0 that the problem-specific heuristics embedded in the bi-level optimization framework will synthesize propulsion system architectures with improved system-level performance than those designed with stochastic optimization methods that utilize generalized, problem-agnostic heuristics.

Therefore, propulsion system architecture topologies and corresponding power management strategies can be synthesized to optimize system-level performance and safety using a bi-level optimization framework that assembles safety-feasible propulsion system architectures from a predefined component library utilizing embedded, problem-specific heuristics. Employing problem-specific heuristics enables the design space to be explored strategically while filtering out infeasible designs in each major optimization iteration, as well as exploiting the attributes of promising system architectures in later designs, ensuring optimal system-level performance.

Including a system safety analysis in the bi-level optimization framework allows the system-level performance and safety considerations to be coupled during aircraft conceptual design, and is critical if future aircraft concepts leverage advanced propulsion system architectures. The safety-performance analysis coupling in the POWER MANAGEMENT OPTIMIZATION problem accounts for both nominal and off-nominal performance considerations, resulting in a propulsion architecture and power management strategy that ensures safe flight and provides a pathway for demonstrating compliance during the aircraft certification process.

Beyond identifying promising propulsion system architectures, the bi-level optimization framework can be utilized to explore how variations certification requirements impact the system architecture selected by the optimizer. In this case, the optimizer is used as a decision-making tool, revealing the sensitivity of the optimal objective function values with respect to the certification requirements. For example, the Pareto Frontiers generated in Experiment 4.0.1 identify the most promising system architectures for a variety of system safety constraints. Since the certification regulations have yet to be defined for advanced aircraft concepts, the Pareto Frontier illustrates the possible propulsion system architectures that could be installed on the aircraft as well as its corresponding failure modes and probabilities. Such information can be useful for engaging in dialogues with the certification authorities, particularly for establishing requirements that are commensurate to those already used for certifying conventional aircraft. Additionally, similar decision-making approaches can be utilized to identify the technologies that have the largest impact on overall system-level performance, thus guiding future technology investments. As the technology levels improve, either in component efficiency or reliability, the most promising propulsion system architecture topology and power management strategies for future aircraft concepts may change.

One way to compute the sensitivities of the objective function values with respect to the certification requirements and technology levels is by performing a Design of Experiments (DOE) that varies the aforementioned parameters. For each experiment, the bi-level optimization framework is run, returning a selected propulsion system architecture its corresponding system-level performance and safety. After all experiments are run, a surrogate model can be developed, defining the design space with a continuous function and providing a mechanism to predict the sensitivities of the objective function values. The factors with the largest sensitivities highlight areas that have the potential to significantly impact system-level performance, possibly warranting further financial investments to reduce risk and uncertainty in the overall aircraft design.

7.2.4 Research Question 1.0: Overarching Thesis

To answer Research Question 1.0, Hypothesis 1.0 proposed the construction of a conceptual aircraft design framework to represent, evaluate, and synthesize advanced aircraft propulsion system architectures while accounting for nominal aircraft performance, off-nominal propulsion system failure scenarios, and system-level safety constraints in a configuration-agnostic manner:

Hyp 1.0: A framework combining graph-based propulsion architecture analyses, off-nominal climb performance constraints based on specific excess power losses under critical failure modes, and joint optimization of the architecture topology and its corresponding power management strategy will identify architectures with a smaller Pareto Hypervolume than baselines designed with a fixed architecture or stochastic optimization method with problem-agnostic heuristics.

The answers to Research Questions 2.0 through 4.0 and the corresponding experiments serve as evidence to answer Research Question 1.0 in greater detail. The conclusions are conditionally based on the results presented in Chapters 3 through 6, and may not be universal.

A unified conceptual aircraft design framework can synthesize safety-feasible advanced propulsion system architectures using a predefined component library and a bi-level optimization framework with embedded, problem-specific heuristics. The heuristics in the optimization framework support a strategic exploration of the design space, utilizing information from previously evaluated propulsion system architectures to generate designs with improved system-level performance. For each safety-feasible design assembled, its power management strategy can be optimized, ensuring that system-level performance is maximized while satisfying all operational constraints. To ensure that the off-nominal operational constraints are satisfied, system-level performance and safety must be coupled in the conceptual design framework, ensuring that the aircraft can safely divert and land if one of its critical propulsion system failure modes were to occur.

Depending on the failure mode considered, the bi-level optimization framework can synthesize propulsion system architectures with system-level performance that is no worse than fixed architecture baselines, partially supporting Hypothesis 1.0. Furthermore, the bi-level optimization framework identified preferred propulsion system architectures faster than stochastic optimization methods that utilize generalized, problem-agnostic heuristics, and with improved system-level performance, supporting the latter final part of Hypothesis 1.0.

Representing and analyzing advanced propulsion systems in a configuration-agnostic manner requires a typed graph representation with component constitutive models at each vertex and power transfer relationships along each edge. Such a representation enables any propulsion system architecture topology to be analyzed in a configuration-agnostic manner, with possibly redundant connections, and is also easily represented in a machine-readable manner, lending itself to parametric trade studies or numerical optimization. Iterative power flow simulations also accurately estimate the required power load for each component, further supporting a configuration-agnostic analysis, regardless of the number of branches and path lengths in the propulsion system.

To ensure that the propulsion system performance is accounted for at the system-level (or aircraft-level), the typed graph representation and power flow simulation capabilities can be integrated into a parametric aircraft sizing tool. Since a unified conceptual design framework is being developed, the parametric aircraft sizing tool should utilize first-order physics-based models to perform an energy-based mission analysis. However, depending on the level of fidelity desired or if the framework were to be used for preliminary or final design, more detailed physics-based models could be interchanged with the ones proposed.

The off-nominal flight conditions also impact the size of the propulsion system components because a set of generalized flight performance requirements act as constraints on the propulsion system's power management strategy. The generalized flight performance requirements included in the conceptual aircraft design framework are better parameterized by the propulsion system failure severity rather than the number of engines installed on the aircraft, accommodating any propulsion system architecture and critical failure mode in a configuration-agnostic manner. The formulation enables a flexible mapping, allowing others to utilize a different set of climb performance requirements than the FAA-published ones used in this work.

The constructed conceptual aircraft design framework can then be used to synthesize future aircraft concepts with advanced propulsion system architectures. Aside from selecting the best propulsion system architecture, the primary propulsion system failure mode chosen by the airframe manufacturer can impact system-level performance and nominal flight operations. For propulsion system failure modes that produce smaller specific excess power losses than conventional aircraft under a legacy one-engine inoperative failure mode, then the aircraft's power loading will either increase or remain the same, and depends on whether the climb constraints are active or not, respectively. If the climb constraints are active, then additional optimization may be needed to select a power loading that maximizes system-level

performance without degrading nominal flight operations. For propulsion system failure modes that produce larger specific excess power losses than conventional aircraft under a legacy one-engine inoperative failure mode, then the aircraft's power loading will either remain the same or decrease, and depends on whether the climb constraints are active or not, respectively. Therefore, system-level performance can only worsen and cannot be improved due to the decreased power loading.

7.3 Key Lessons Learned for Advanced Aircraft Design

From this thesis, there are multiple key lessons learned that can be leveraged for designing future aircraft concepts.

Tradeoffs can be systematically quantified between disparate aircraft concepts if its subsystems are represented and analyzed in a graph-based framework assuming the same technology levels and design mission profile. A graph-based framework enables connections between components to be included/excluded easily and selected either manually or via a numerical optimization algorithm. Beyond component connections, different configurational options could be selected, such as engine placement (wing-mounted, tail-mounted, or both) or tail type (conventional tail, T-tail, etc.), broadening the analysis to identify the most promising future aircraft concepts at a configuration-level.

Configuration-agnostic aircraft design requires performance and safety analyses to be coupled while simultaneously accounting for nominal and off-nominal flight performance. The possible failure modes that can occur depend on the system architecture synthesized, requiring a system safety analysis during conceptual design. The resultant failure modes can be used to perform off-nominal flight performance simulations, ensuring that the aircraft can meet the required climb gradients across all possible failure modes. Understanding the system's behavior and possible failure modes are more critical in advanced concepts as the architectural complexity increases, resulting in novel failure modes inherent to each propulsion system architecture synthesized. Furthermore, by considering the possible failure modes during conceptual design, system-level performance can be maximized, ensuring that the benefits of installing advanced technologies on future aircraft concepts are fully reaped.

Off-nominal performance must be parameterized by the failure mode severity rather than the number of engines installed on the aircraft. Future aircraft concepts may leverage advanced propulsion systems with heterogeneously-sized or distributed

propulsion systems, requiring certification regulations beyond those for aircraft with 2, 3, or 4 homogeneously-sized propulsors. Additionally, common-mode failures challenge existing design paradigms focused on one-engine inoperative conditions, particularly for distributed propulsion systems, resulting in multiple propulsors failing under a given failure mode.

The failure mode severity selected by the airframe manufacturer can significantly impact system-level performance, requiring additional effort to identify a suitable propulsion system size. For failure modes that exhibit a smaller severity than legacy one-engine inoperative conditions, the design space may widen far enough that nominal flight performance is degraded if too large of a power loading is selected. Thus, new tradeoffs exist between the propulsion system failure severity and nominal flight performance, requiring either numerical optimization or parametric trade studies to quantify the system-level benefits (if any) of downsizing the propulsion system.

Operational constraints inherent to existing airport infrastructures and in-flight operations may limit system-level performance improvements for advanced aircraft concepts. Although future aircraft concepts may be designed for failure modes that are less severe than legacy one-engine inoperative conditions, operational constraints such as maximum takeoff/landing field lengths or minimum cruise/diversion speeds may remain as active constraints, preventing the design space from widening. As a result, the propulsion system may not be downsized, requiring additional analyses to assess if system-level performance improvements are possible with the present technologies available for aircraft electrification.

Less intuitive subsystem designs have the potential to improve system-level safety while minimizing system-level performance penalties. Traditionally, redundant component connections may be arranged symmetrically or in a more systematic fashion, but may not maximize system-level performance for a given set of safety constraints. Numerical optimization can support the aircraft subsystems architecting by identifying novel connection schemes between components, improving system-level safety with minimal system-level performance penalties or vice versa.

7.4 Contributions

Three technical contributions have been developed in this thesis and enhance the state-of-the-art in configuration-agnostic design for future aircraft concepts: 1) a graph-based propulsion system analysis framework that analyzes any propulsion system architecture; 2)

generalized flight performance requirements parameterized by the specific excess power loss during a critical propulsion system failure mode; and 3) a bi-level optimization framework that couples system-level performance and safety, utilizing problem-specific heuristics to strategically search the design space for safety-feasible propulsion system designs. Each technical contribution is discussed further and explains the relevant enhancements made to the state-of-the-art.

The graph-based propulsion system analysis framework enables any propulsion system architecture to be analyzed in a flexible and scalable manner, independent of the component arrangement and architectural complexity. All propulsion system component connections are embedded in one, machine-readable matrix, lending itself to parametric trade studies or numerical optimization. There are four reasons that the framework developed is an enhancement over prior state-of-the-art methods. First, the energy conservation equations are inherently defined in the operational matrices by establishing power flow distributions between components, eliminating the need to derive them manually [81]. Second, the power management strategy (established in the operational matrices) is defined separately from the component efficiencies, decoupling the two and allowing parametric propulsion system analyses to be executed seamlessly. The interactions between the power management strategy and component efficiencies are accounted for by element-wise multiplying/dividing the two matrices prior to simulating the power flow, unlike other approaches that include both terms in the same matrix [81]. Third, power splits are defined at junctions between successive components, allowing the designer to easily understand the ramifications of the power management strategy selected [81]. This is an enhancement over prior techniques that relied on hybridization factors that cannot always be intuitively identified for complex propulsion systems. Fourth, the framework enables any propulsion system to be analyzed, including ones with power off-takes, and places no limits on the number of components that can be arranged in series [8], improving its robustness over prior state-of-the-art propulsion system analysis methods.

There are three contributions made from constructing the generalized flight performance requirements. First, an aircraft operational analysis demonstrated the validity of parameterizing off-nominal climb performance by the specific excess power loss. Second, configuration-agnostic flight performance requirements were constructed by mapping the specific excess power loss during a propulsion system failure to the presently published FAA climb performance requirements. Prior flight performance requirements were parameterized by the number of engines installed on the aircraft [89], a discrete variable, and only

accommodated homogeneously sized propulsion systems. The configuration-agnostic flight performance requirements are parameterized by the specific excess power loss, a proxy for the propulsion system failure severity, and accommodates heterogeneously sized or distributed propulsion systems. Third, a configuration-agnostic engine inoperative correction factor was also constructed as a function of the specific excess power loss. As a result, the propulsion system size is scaled by the propulsion system failure severity rather than the number of engines installed on the aircraft [103], again serving to accommodate heterogeneously sized propulsion systems.

More broadly, the generalized engine inoperative requirements enabled the validation of prior assumptions made during the previous design of advanced aircraft concepts in the literature. Prior advanced aircraft sizing studies utilized the FAA’s 4-engine climb performance requirements [9, 97, 100], particularly for sizing aircraft with distributed propulsion systems. Assuming that a distributed propulsion system failure involves one or two propulsors, the specific excess power loss fraction is likely to be less than 60%, aligning with the upper asymptote on the Sigmoid Curves that corresponds to the 4-engine climb requirements prescribed by the FAA. Thus, it is reasonable to design future aircraft concepts with advanced propulsion systems utilizing the FAA-published flight performance requirements corresponding to those for a 4-engine aircraft until improved regulations are released.

There are multiple technical contributions made within the bi-level optimization framework. First, to the author’s knowledge, the bi-level optimization framework is the first architecture synthesis framework that facilitates the design of redundant (propulsion) system architectures, enhancing the state-of-the-art in advanced propulsion system design. Second, off-nominal system performance and safety are coupled in the aircraft propulsion system architecting process. The coupling enables the power management strategy to be optimized for both nominal and off-nominal performance considerations, accommodating novel, configuration-specific failure modes in a configuration-agnostic manner. The performance-safety coupling is critical for ensuring that a future aircraft concept can fly safely under all possible propulsion system failures, particularly those that deviate from the traditional one-engine inoperative failure scenario. Third, problem-specific heuristics are utilized to strategically search the design space for safety-feasible propulsion system designs. The heuristics are able to drive the optimizer towards either an electrified propulsion system that boosts a gas turbine engine’s power during takeoff and climb, or a distributed propulsion system. Additionally, the heuristics minimize the redundancy in the propulsion system to improve system-level performance while satisfying the system safety constraints. Previous state-of-the-art methods

utilize stochastic optimization methods that search broader regions of the design space, but fall short in exploiting the distinguishing characteristics of optimal propulsion system architectures. Thus, the bi-level optimization framework and its embedded heuristics advance the state-of-the-art. Fourth, the Fault Tree Analysis is formulated directly from the propulsion system architecture matrices to evaluate multiple novel failure scenarios. Other system safety analysis software packages require the Fault Tree to be constructed manually, limiting its ability to be included in parametric trade studies or numerical optimization. As a result, the propulsion system architecture matrices can directly interact with a system safety analysis, enabling widespread system architecture optimization studies that assemble safety-feasible configurations.

7.5 Future Work

7.5.1 GPSA Framework

Future work involving the GPSA Framework spans both computational advances and applications to other networked systems. First, for propulsion systems with ten or more components, such as the distributed electric propulsion architecture in Fig. 3.10 from Chapter 3, the matrices become cumbersome to construct, read, and store, suggesting that a sparse matrix representation (with corresponding updates to the power flow algorithm) may enhance clarity and computational efficiency. The sparse matrix representation would also reduce the topology-specific model setup required, overcoming a weakness of the GPSA Framework’s present methodology. Second, while being recharged, the battery operates as a sink rather than a source, complicating the traditional component classifications provided. Refinement of these definitions to accommodate components that can function in dual roles will further improve the framework’s robustness, particularly for the propulsion system architecture exhibited in Appendix A.3 that violates the current component classification definitions. Lastly, when simulating a bi-directional power flow, upstream and downstream operations may be conflated, particularly for bi-directional power flows that include charging. It may be simpler to eliminate the notion of “upstream” and “downstream” flows, allowing the operational and efficiency matrices to be generalized for various operational scenarios, such as nominal operations, off-nominal operations, and charging.

The GPSA Framework can also be easily extended and deployed in other aerospace design and analysis tools. For example, two other aircraft design codes besides FAST [177], namely SUAVE [55] and Aviary [58], already analyze electrified propulsion architecture systems and

could adapt this framework to enhance their analysis capabilities.

While the GPSA Framework was demonstrated for aircraft propulsion systems, it is extensible to any networked system that contains energy or matter flows. Examples include electrical grids, transportation networks, or electro-mechanical system design. In the context of an electrical grid analysis, alternative physical quantities such as voltage or current may be propagated instead of power. In an electrical grid, examples of uni-directional flows include; 1) using the upstream operational matrices to model power transfer from a generation station to a consumer; or 2) using the downstream operational matrices to size a power generation station for a given consumer demand. Furthermore, bi-directional flows could be used to model a consumer with solar panels – the consumer produces excess electricity during the daytime and returns it to the grid, but extracts power from the grid at nighttime. In this case, the consumer acts as both a source and a sink depending on the time of day (similar to battery discharging/charging).

7.5.2 Parametric Aircraft Sizing Tool

There are two suggested improvements to enhance FAST and its aircraft analysis capabilities. First, the governing equations for the climb, cruise, and descent segments could be unified into one, overarching mission analysis. Despite each segment utilizing its own code, the underlying energy-based analysis remains the same. Unifying the code would improve its analytical robustness and usability while streamlining both code maintenance and aircraft performance evaluation.

Additionally, analytical differentiation should be implemented in FAST to enable gradient-based optimization without the need for computing derivatives via finite differences. During the POWER MANAGEMENT OPTIMIZATION in the bi-level optimization framework, an aircraft must be sized $m + 1$ times, where m is the number of power splits being optimized, and requires significant computational resources. Analytically differentiating the entire code, or, at a minimum, the propulsion system analysis, would support future design studies that optimize both the aircraft's propulsion system architecture and corresponding power management strategy with less computational overhead. Knowledge of the gradients can also be useful for sensitivity analyses during early-phase conceptual design, particularly for identifying the design parameters that will have the largest impact on system-level performance.

7.5.3 Generalized Engine Inoperative Performance Requirements

The data collected to approximate the specific excess power loss fractions (SEPLFs) used to construct the generalized engine inoperative performance requirements only consisted of turbofan-powered aircraft certified under FAA Part 25 regulations. However, there are many aircraft certified under FAA Part 25 regulations that are powered by turboprops, such as the ATR 42/72 families. Including data on turboprop-powered aircraft that carry 19-80 passengers would augment the existing dataset, and refine the estimated SEPLFs. Furthermore, the SEPLFs between the turboprop- and turbofan-powered aircraft could be compared to ensure that the same generalized flight performance requirement constructions can be broadly used across both aircraft types.

Second, the methodology developed in this work could be extended to normal category aircraft certified under FAA Part 23 requirements. Currently, climb performance requirements under FAA Part 25 are prescribed based on categorical variables such as the number of engines installed, number of passengers carried, and their cruise speed (faster or slower than Mach 0.6). Applying the methodology from Chapter 5 could unify the existing flight performance requirements under a similar mapping via the specific excess power loss.

Third, the Sigmoid Curve constructions in Experiment 3.0.2 revealed that future climb performance requirements could be governed by an upper/lower limit depending on the propulsion system failure severity. Future work is necessary to identify the corresponding propulsion system severity threshold that would cause a switch between the upper/lower limits, better contextualizing the FAA-prescribed climb gradients for advanced aircraft and propulsion system design.

More broadly, the generalized flight performance requirements and engine inoperative correction factor could be utilized in future design studies to explore the tradeoffs reducing the propulsion system size and its impacts on nominal climb performance. The results from Experiment 3.1.1 indicated that nominal climb performance is reduced by downsizing the propulsion system and, in some cases, can degrade system-level performance. Additional trade studies exploring the relationship between the propulsion system size and nominal/off-nominal climb performance for advanced propulsion system architectures, such as distributed electric propulsion architectures or ones with heterogeneously sized propulsors, may provide insights as to how much the propulsion system can be downsized before system-level performance is hampered. Like Experiment 3.1.1, it may be found that selecting the maximum admissible power loading from a constraint diagram is no longer appropriate for advanced propulsion systems, requiring further design optimization or trade studies during conceptual design.

7.5.4 System Safety Analysis

Future system safety analyses can be incorporated into the framework, extending beyond the FTA currently utilized. Although combinations of component-level failures are identified in the FTA, the only common-mode failures identified are those within the propulsion system caused by components that are included in the architecture matrix. As a result, common-mode failures outside of the propulsion system are not accounted for, such as bus faults, battery thermal runaway propagation, software failures, flight control system failures, cable routing common exposures, and external events. Multiple analyses could be incorporated into the framework for identifying other common failure modes, including: 1) Common-Cause/Beta-Factor Modeling, identifying how commonalities in hardware or software development could lead to system-level failures in the future; 2) Particular Risk Analysis, identifying how a common-mode failure could impact the aircraft's ability to survive the failure; 3) or a Zonal Safety Analysis, identifying how co-located equipment could cause potential safety hazards [196].

7.5.5 Bi-Level Optimization Framework

Future work involving the bi-level optimization framework includes both methodological improvements as well as alternative trade studies on SUSAN's propulsion system architecture design. First, the FAULT TREE ANALYSIS (FTA) assigned a component-level failure probability to each component, neglecting the more granular failure modes that each component could exhibit. For further subsystem- and component-level studies, more detail can be added to the FTA by considering the different possible failure modes within each component, adding more realism to the study. Also, the FTA currently assumes that all components operate for the entire flight, reflecting a constant exposure time. However, nominal exposure times for different propulsion system components could be integrated into the study for additional realism, particularly for propulsion system architectures that use an electric motor to boost the gas turbine engine power during takeoff and/or climb.

Second, the Pareto-optimal system architectures, in combination with the FAA's failure condition classifications, could be utilized to select a primary propulsion system failure mode for SUSAN's propulsion system, paving a pathway for certification in the future. Aside from identifying a reasonable primary propulsion system failure mode, tradeoffs between redundancy mechanisms and their impacts on the propulsion system weight/safety could be explored in future work. While this work proposed installing redundant connections

between the generators and motors, additional redundancy could also be achieved by adding more propulsors to the distributed electric propulsion system. However, including additional propulsors in the distributed propulsion system would increase the number of required cable connections from the electric generators, possibly increasing the aircraft's structural weight. Aside from weight considerations, aerodynamic effects must also be accounted for, as decreasing the fan/propeller diameter may reduce its efficiency.

A third area of future work involves integrating flight stability and dynamics considerations in the bi-level optimization framework. The work in this thesis focused on conceptual aircraft design and neglected flight stability and dynamics in the analysis. However, distributed propulsion systems are able to generate roll and yaw moments by changing the throttle settings on specific propulsor sub-groups. Therefore, the distributed propulsion system operation (and corresponding failure modes) can impact the aircraft's lateral and directional flight stability in a more nuanced manner than conventional propulsion systems. The bi-level optimization framework could be expanded to include an aircraft flight dynamics analysis with integrated flight stability constraints (from FAA Part 25.143 through 25.181 regulations) in the POWER MANAGEMENT OPTIMIZATION problem, exploring how specific propulsion system failure modes impact the aircraft's overall flight performance and stability.

While it is possible to solve an additional optimization problem for control co-design, such levels of detail may be unnecessary for conceptual aircraft design. Since conceptual aircraft design studies are intended to down-select on promising configurations for preliminary design and identify areas for technological development, fully designing a control system and corresponding controller gains may be too detailed and require additional computational overhead that could slow the design process. Rather, controller design should occur during preliminary design as the flight control surfaces are arranged and sized. If, however, a control co-design problem were to be integrated into the bi-level optimization framework, it would be divided into two parts. The PROPULSION ARCHITECTURE OPTIMIZATION problem could be broadened to a SUBSYSTEM ARCHITECTURE OPTIMIZATION problem and involve multiple Fault Tree Analyses for both the propulsion and flight control systems, ensuring that reliability requirements/objectives are satisfied for each subsystem. After selecting both the propulsion and flight control system architectures, the POWER MANAGEMENT OPTIMIZATION problem can be solved in conjunction with a CONTROL DESIGN OPTIMIZATION problem, identifying the optimal power management strategy and controller gains, respectively, during both nominal and off-nominal flight conditions.

Further work could also consider utilizing the distributed propulsion system as an ad-

ditional level of redundancy for the flight control system. Thus, the bi-level optimization framework could be modified to accommodate the propulsion and flight control systems architecting problems previously described.

A fourth area of future work involves investigating how certification requirements beyond FAA Part 25 can be integrated into the bi-level optimization framework. Unmanned aerial vehicles (UAVs) are becoming increasingly popular in the defense industry for reconnaissance, surveillance, and aerial attack missions because they can be remotely piloted, enabling aerial combat without the risks of carrying a service-member. However, certification of military aircraft in the United States is governed by the Department of Defense, remaining separate from the FAA. Therefore, further research is needed to investigate how military aircraft certification regulations, particularly those from MIL-HDBK-516C, can be integrated into the bi-level optimization framework for advanced aerospace systems design.

One final area of work involves benchmarking the proposed heuristic framework against an existing commercial optimization software that is capable of solving MINLP problems such as Gurobi ¹ or CPLEX ². Currently, the binary variables in the bi-level optimization framework change the propulsion system architecture, representing a categorical topology, preventing Branch-and-Bound Methods from being used and necessitating the use of a stochastic optimization algorithm with problem-specific heuristics. To maintain compatibility with commercial solvers that utilize Branch-and-Bound Methods, the optimization problem must be reformulated such that the variables governing the system architecture topology represent numerical design choices – such as the number of electric generators in the system architecture or the number of redundant connections in the electric generator-motor cross-connection scheme – enabling the “Systems Architecting” problem to be solved as a MINLP problem or a relaxation of a Mixed-Integer Linear Programming (MILP) problem.

7.5.6 Broader Systems Architecting Problems

The bi-level optimization framework in Chapter 6 was created to optimize for both performance and safety objectives, namely minimizing fuel burn and the propulsion system failure rate. While fuel burn is an important metric for reducing environmental impact, and acts as a proxy for minimizing carbon emissions, there are additional quantities of interest for other aircraft system and subsystem architecting studies.

One emergent subfield within aerospace engineering is the design of UAVs, and have

¹<https://www.gurobi.com/>

²<https://www.ibm.com/products/ilog-cplex-optimization-studio>

applications in humanitarian aid, logistics, and urban air mobility (beyond the defense applications previously described). Some UAVs designed with battery-electric propulsion systems to carry small packages or transport passengers between transportation hubs and urban centers. Instead of fuel burn, weight becomes a more critical quantity of interest, and must be minimized. Therefore, the computational framework could be applied to the design of UAVs by optimizing for both weight and reliability objectives. For example, the UAV's weight could be minimized subject to a reliability constraint. Alternatively, the UAV's reliability could be maximized for a given weight budget.

Aside from weight, another quantity of interest is the overall cost and economic impacts of flying an aircraft with an advanced propulsion system. Rather than minimizing fuel burn, as done in the optimization framework defined in Chapter 6, the objective function could be to minimize the direct operating cost of the aircraft, accounting for both fuel and maintenance considerations. For electrified aircraft with a battery, it may be necessary to minimize the total operating cost of the aircraft, capturing the additional operational expenses at an airport such as battery charging or additional gate time (if batteries cannot be simultaneously charged during fueling or passenger loading/unloading).

Beyond propulsion system architecture optimization, the bi-level optimization framework could be utilized for designing redundant flight control electronics or sensing systems as well. In the context of flight control electronics systems, there are weight-reliability tradeoffs for connecting additional actuators to each control surface or adding redundant cable connections from battery packs to the actuators. While the added redundancy ensures that the aircraft maintains directional and longitudinal stability during off-nominal scenarios, there is a weight penalty associated with carrying the additional components. Thus, the bi-level optimization framework could be utilized to identify whether a duplex, triplex, or quadruplex flight control system is necessary while optimizing for both weight and reliability objectives. Similar optimization studies could be performed for onboard sensing systems; including redundant sensors onboard the aircraft requires other power electronics such as switches to be installed, penalizing the UAVs takeoff weight but improving its overall reliability. Such studies would also require detailed flight dynamics analysis codes to be integrated into the bi-level optimization framework. Rather than optimizing the power management strategy, the gains in the flight controllers could be optimized, ensuring that the necessary flight stability and control requirements are met in both nominal and off-nominal operational scenarios.

Appendix A

Additional Electrified Propulsion Architectures Defined for Analysis the GPSA Framework

A.1 Turboelectric Propulsion Architecture

The turboelectric propulsion architecture is illustrated in Fig. A.1.

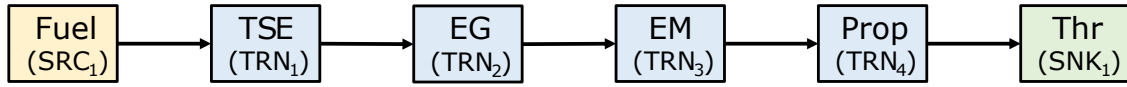


Figure A.1: Turboelectric propulsion architecture schematic.

Its architecture matrix is:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Its upstream and downstream operational matrices are:

$$\lambda^u = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \lambda^d = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

A.2 Simplified NASA SUSAN Concept

One of NASA's novel aircraft concepts includes the **S**U**B**sonic **S**ingle **A**ft **e**Ngine (**SUSAN**) and is powered by a series/parallel-partial hybrid electric propulsion system [208]. Using advanced electrification technologies, an aft turbofan engine generates thrust and also powers a set of 8 electric propulsors along each wing [7], which output equal amounts of power at any given time. A simplified version of the propulsion architecture with one propulsor on each wing is provided in Fig. A.2.

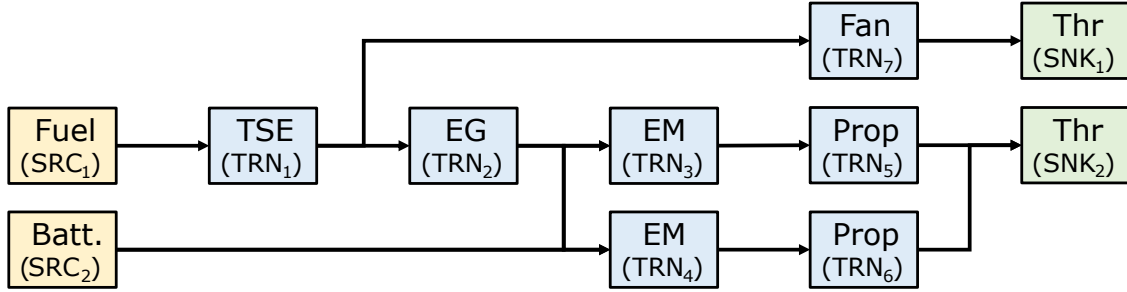


Figure A.2: NASA SUSAN propulsion architecture schematic (simplified).

Its architectural matrix is:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Its upstream and downstream operational matrices are:

$$\lambda^u = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda_{\text{Batt,EM}} & \lambda_{\text{Batt,EM}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_{\text{TSE,EG}} & 0 & 0 & 0 & 0 & \lambda_{\text{TSE,Fan}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda_{\text{EG,EM}} & \lambda_{\text{EG,EM}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\lambda^d = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda_{\text{Thr,Prop}} & \lambda_{\text{Thr,Prop}} & 0 & 0 & 0 & 0 \end{bmatrix}$$

A.3 Series Hybrid Electric with Recharging

In the series hybrid electric propulsion architecture provided in Fig. A.3, the battery may be recharged by either operating the electric motor in “reverse mode”, or by directing power from the electric generator to the battery. Typically, the battery is considered a source, which can only output power and receive inputs from outside of the system boundary during nominal operations. However, in this example, the battery can receive power from the electric generator and output it to the electric motor at any time. Therefore, the battery violates the definition of a source – the column sum for sources should be 0, but is now 1. Based on the component definitions, the battery is now a transmitter, although it can store energy like a source. Therefore, this propulsion architecture violates both the existing component classifications and the requirement that all power splits in each row must add to one (the battery’s power split can be zero when charging, or less than one to show that it is both charging and providing power to the electric motor). As discussed in Chapter 7, future work involves improving the component classifications to accommodate additional advanced propulsion system architectures.

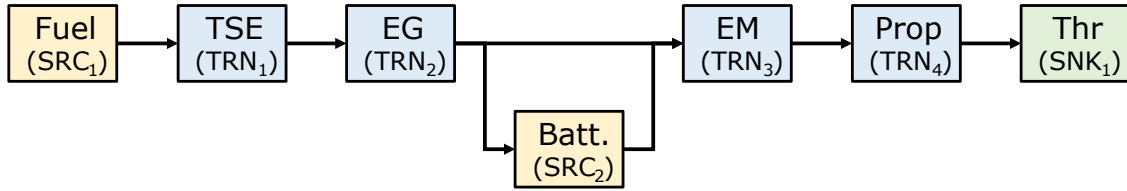


Figure A.3: Series hybrid propulsion architecture schematic with multiple battery recharging options.

Its architectural matrix is:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Its upstream and downstream operational matrices are:

$$\lambda^u = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda_{\text{Batt,EM}} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \lambda_{\text{EG,Batt}} & 0 & 0 & \lambda_{\text{EG,EM}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\lambda^d = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \lambda_{\text{EM,Batt}} & 0 & \lambda_{\text{EM,EG}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Appendix B

Design Mission Profiles Simulated in Experiments 2.0.2 and 2.0.3

Table B.1 lists the design mission profile for the Embraer ERJ175LR analyzed in Experiments 2.0.2 and 2.0.3. The initial design mission spans 2,150 nmi, followed by two reserve missions: 1) a 100 nmi diversion mission; and 2) a 45-minute loiter.

Table B.1: Notional ERJ175LR design mission profile.

Mission	Segment	Initial Altitude (ft)	Final Altitude (ft)	Initial Airspeed (kts or Mach)	Final Airspeed (kts or Mach)
Design Mission – 2,150 nmi	Takeoff	0	0	0 KTAS	135 KTAS
	Initial Climb	0	3,000	135 KTAS	200 KEAS
	Main Climb	3,000	35,000	200 KEAS	200 KEAS
	Accelerate (Climb)	35,000	35,000	200 KEAS	Mach 0.78
	Cruise	35,000	35,000	Mach 0.78	Mach 0.78
	Decelerate	35,000	35,000	Mach 0.78	210 KEAS
	Main Descent	35,000	3,000	210 KEAS	210 KEAS
	Final Descent	3,000	1,500	210 KEAS	162 KTAS
Diversion – 100 nmi	Initial Climb	1,500	3,000	162 KTAS	200 KEAS
	Main Climb	3,000	9,000	200 KEAS	200 KEAS
	Final Climb	9,000	10,000	200 KEAS	250 KTAS
	Cruise	10,000	10,000	250 KTAS	250 KTAS
Loiter – 45 min	Cruise	10,000	10,000	250 KTAS	250 KTAS
	Initial Descent	10,000	9,000	250 KTAS	200 KEAS
	Main Descent	9,000	3,000	200 KEAS	200 KEAS
	Final Descent	3,000	0	200 KEAS	162 KTAS
	Landing	0	0	162 KTAS	0 KTAS

Table B.2 lists the design mission profile for NASA's SUSAN aircraft concept analyzed in Experiment 2.0.2. Its design mission spans 2,500 nmi, and utilizes the same diversion and loiter reserve missions as the Embraer ERJ175LR.

Table B.2: Notional SUSAN design mission profile.

Mission	Segment	Initial Altitude (ft)	Final Altitude (ft)	Initial Airspeed (kts or Mach)	Final Airspeed (kts or Mach)
Design Mission – 2,500 nmi	Takeoff	0	0	0 KEAS	160 KEAS
	Initial Climb	0	1,500	160 KEAS	250 KEAS
	Accelerated Climb	1,500	10,000	250 KEAS	250 KEAS
	Final Climb	10,000	35,000	250 KEAS	Mach 0.775
	Cruise	35,000	35,000	Mach 0.775	Mach 0.775
	Descent	35,000	100	Mach 0.775	165 KEAS
Diversion – 100 nmi	Climb	100	15,000	165 KEAS	Mach 0.500
	Cruise	15,000	15,000	Mach 0.500	Mach 0.500
	Descent	15,000	1,500	Mach 0.500	200 KEAS
Loiter – 45 min	Cruise	1,500	1,500	200 KEAS	200 KEAS
	Approach	1,500	0	200 KEAS	165 KEAS
	Landing	0	0	165 KEAS	0 KEAS

Appendix C

Generalized Engine Inoperative Performance Requirements

C.1 Diversion Mission Profile Boundary Conditions

In Section 5.2, a diversion mission was flown to estimate the specific excess power loss after an engine failure. Table C.1 lists the boundary conditions for each segment flown. The engine was assumed to fail immediately after takeoff as the aircraft begins the first climb segment. Both the takeoff and climb speeds, V_{Tko} and V_{Clb} , respectively, were varied during the operational analysis.

Table C.1: Generic diversion mission profile. All airspeeds are reported as true airspeeds.

Segment	Initial Altitude (m)	Final Altitude (m)	Initial Airspeed (m/s)	Final Airspeed (m/s)
Takeoff	0	0	0	V_{Tko}
Climb	0	914	V_{Tko}	V_{Clb}
Climb	914	3,048	V_{Clb}	128.6
Cruise	3,048	3,048	128.6	128.6
Descent	3,048	0	128.6	$1.1V_{Tko}$
Landing	0	0	$1.1V_{Tko}$	0

C.2 Lift-to-Drag Correction Factors

In Section 5.2.1.3, a correction factor was imposed on the cruise and climb/descent lift-to-drag ratios to account for windmilling and trim drag caused by an engine failure. Tables C.2 and C.3 list the correction factors for 2- and 4-engine aircraft, respectively, under the defined operational strategies. The correction factors for 3-engine aircraft were computed as the mean of each corresponding correction factors for 2- and 4-engine aircraft.

Table C.2: Lift-to-drag ratio correction factors for 2-engine aircraft under one engine in-operative conditions. ΔV represents the difference between the takeoff and climb speeds, previously outlined in Tab. 5.2.

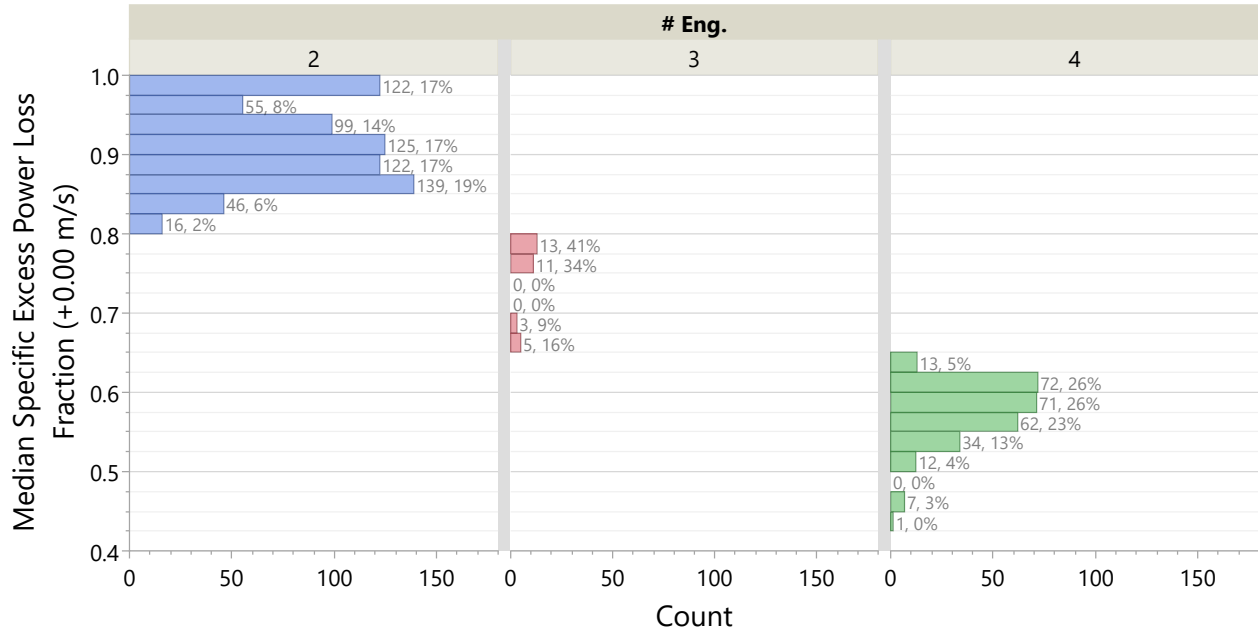
V_{Tko} (m/s)	ΔV (m/s)					
	0.00	5.14	10.29	15.43	20.58	25.72
66.88	0.8091	0.8084	0.8082	0.8039	0.8043	0.8021
72.02	0.8082	0.8082	0.8047	0.8027	0.8004	0.8030
77.17	0.8054	0.8036	0.8016	0.7993	0.8050	0.8000
82.31	0.8029	0.8009	0.7986	0.7954	0.7999	0.7979
87.46	0.8118	0.8045	0.8011	0.7993	0.7978	0.7962
92.60	0.7988	0.8012	0.7994	0.7980	0.7964	0.7873
97.74	0.8016	0.7998	0.7984	0.7967	0.7876	0.7791
102.89	0.8005	0.7990	0.7973	0.7880	0.7802	0.7731

Table C.3: Lift-to-drag ratio correction factors for 4-engine aircraft under one engine in-operative conditions. ΔV represents the difference between the takeoff and climb speeds, previously outlined in Tab. 5.2.

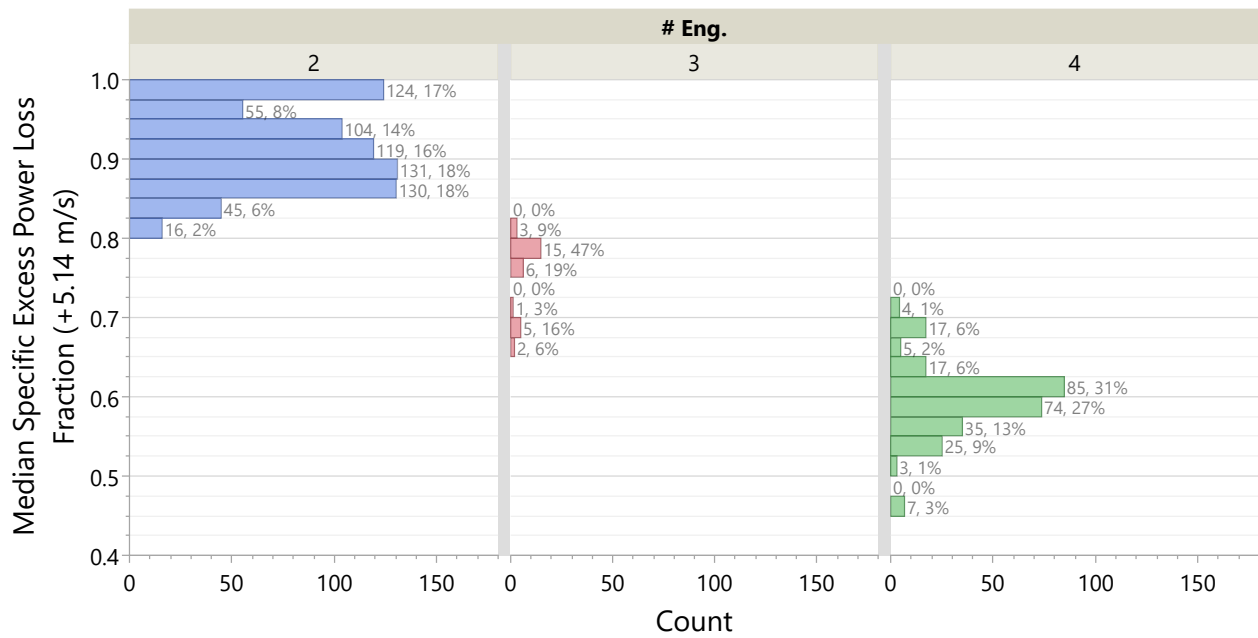
V_{Tko} (m/s)	ΔV (m/s)					
	0.00	5.14	10.29	15.43	20.58	25.72
66.88	0.8244	0.8386	0.8607	0.7676	0.7628	0.8554
72.02	0.8571	0.7590	0.7678	0.7629	0.7558	0.8846
77.17	0.8733	0.8455	0.8349	0.8310	0.8284	0.8263
82.31	0.8498	0.8329	0.8295	0.8271	0.8251	0.8233
87.46	0.8324	0.8290	0.8268	0.8248	0.8231	0.8214
92.60	0.8290	0.8267	0.8249	0.8231	0.8214	0.8110
97.74	0.8268	0.8249	0.8232	0.8215	0.8110	0.7954
102.89	0.8253	0.8236	0.8219	0.8113	0.7958	0.7826

C.3 Specific Excess Power Loss Fractions

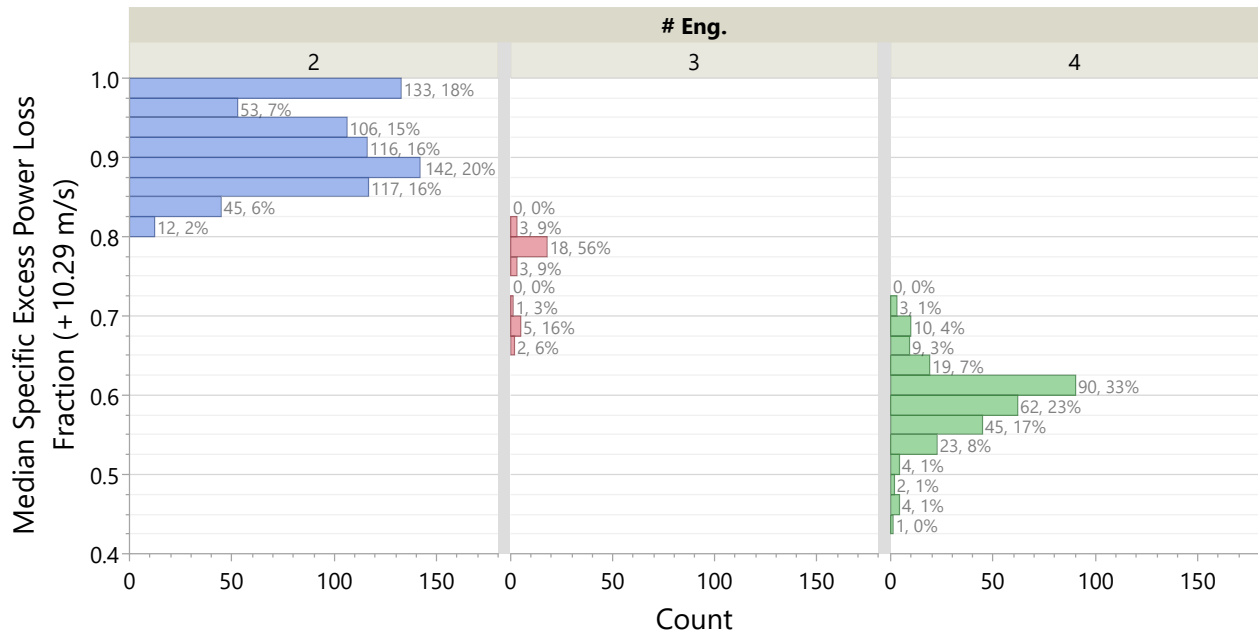
In Section 5.4.1, a histogram of the median SEPLF was plotted for a ΔV of 25.72 m/s. Additional histograms for ΔV between 0.00 and 20.58 m/s are included in Fig. C.1.



(a) $\Delta V = 0.00$ m/s.



(b) $\Delta V = 5.14$ m/s.



(c) $\Delta V = 10.29$ m/s.

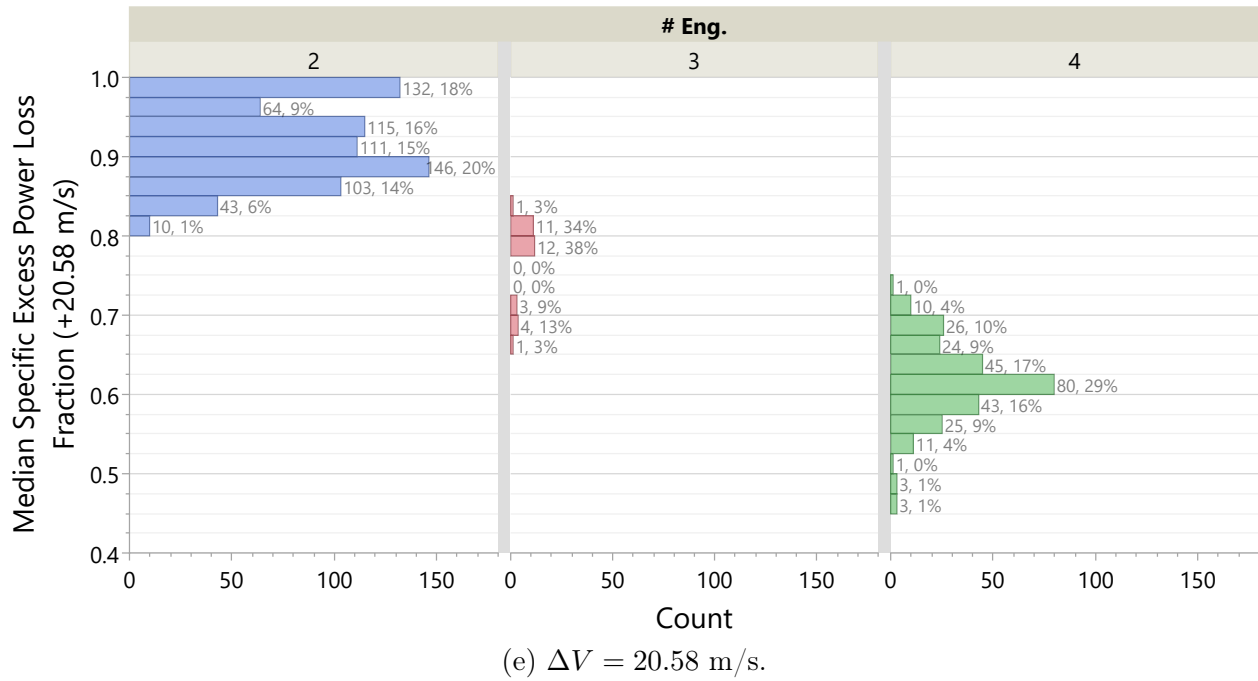
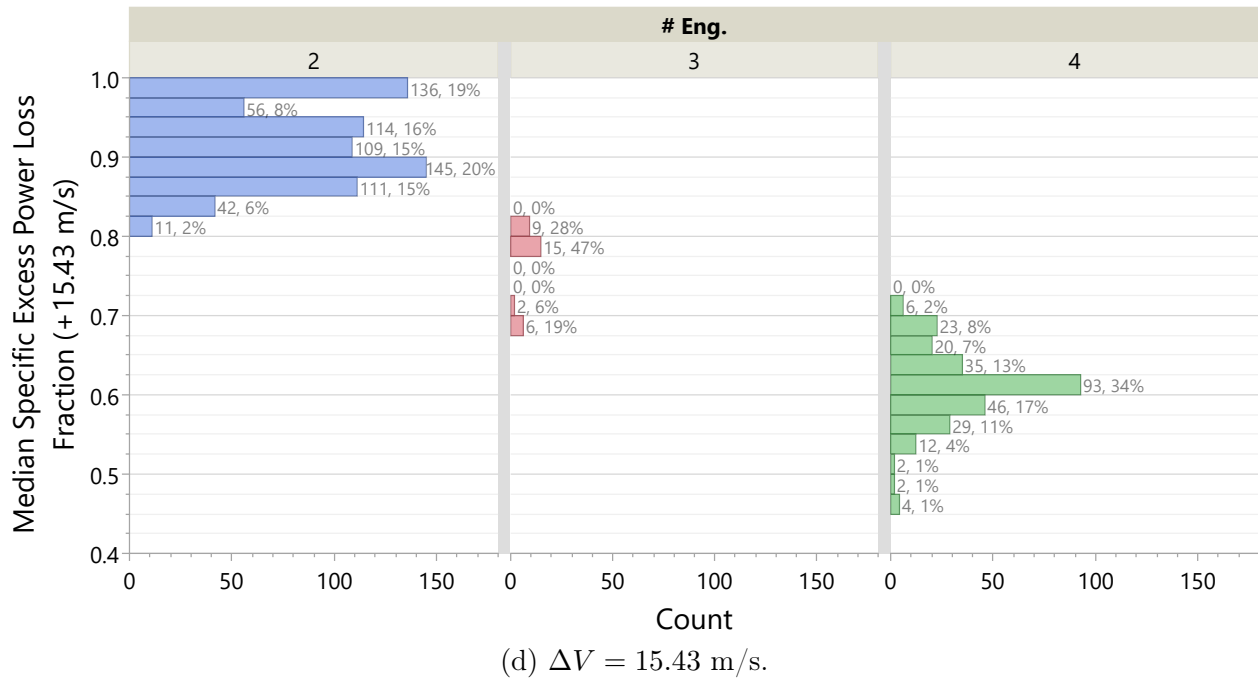


Figure C.1: SEP loss fraction distributions.

As done in Tab. 5.6 within Section 5.4.1, the distributional anchors for the remaining ΔV are included in Tabs. C.4 through C.8.

Table C.4: Selected quantities across the SEPLF distributions for $\Delta V = 0.00$ m/s.

Percentile	Engines Installed		
	2	3	4
10	0.8532	0.6683	0.5339
25	0.8720	0.6984	0.5567
50	0.9099	0.7692	0.5812
75	0.9489	0.7842	0.6034
90	0.9816	0.7859	0.6165

Table C.5: Selected quantities across the SEPLF distributions for $\Delta V = 5.14$ m/s.

Percentile	Engines Installed		
	2	3	4
10	0.8544	0.6754	0.5435
25	0.8737	0.7222	0.5735
50	0.9113	0.7796	0.5976
75	0.9494	0.7861	0.6132
90	0.9822	0.8091	0.6392

Table C.6: Selected quantities across the SEPLF distributions for $\Delta V = 10.29$ m/s.

Percentile	Engines Installed		
	2	3	4
10	0.8556	0.6780	0.5448
25	0.8753	0.7170	0.5653
50	0.9128	0.7827	0.5989
75	0.9512	0.7876	0.6155
90	0.9824	0.8072	0.6410

Table C.7: Selected quantities across the SEPLF distributions for $\Delta V = 15.43$ m/s.

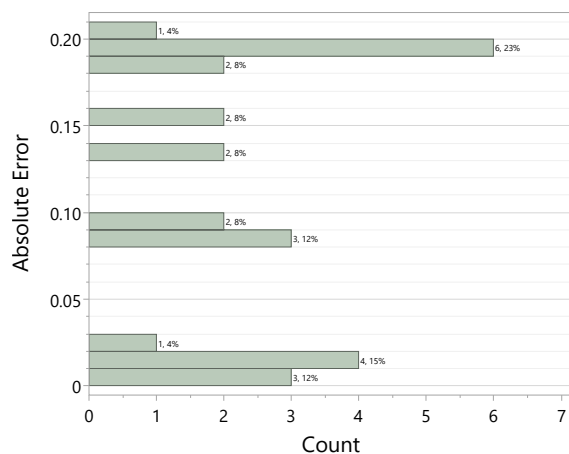
Percentile	Engines Installed		
	2	3	4
10	0.8585	0.6806	0.5558
25	0.8769	0.7252	0.5855
50	0.9144	0.7864	0.6099
75	0.9536	0.8010	0.6337
90	0.9826	0.8175	0.6771

Table C.8: Selected quantities across the SEPLF distributions for $\Delta V = 20.58$ m/s.

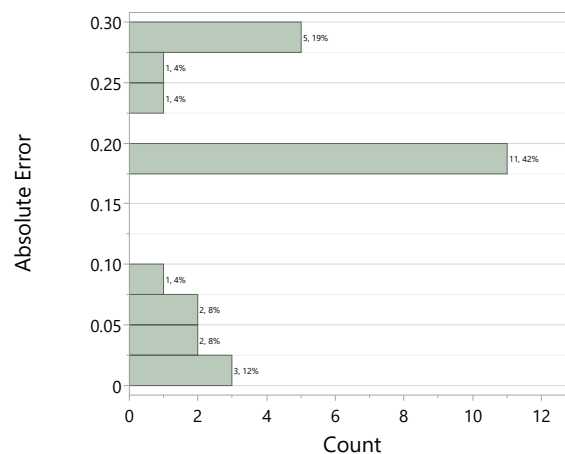
Percentile	Engines Installed		
	2	3	4
10	0.8571	0.6816	0.5601
25	0.8766	0.7281	0.5924
50	0.9141	0.7812	0.6146
75	0.9557	0.8149	0.6442
90	0.9829	0.8222	0.6847

C.4 Climb Gradient Comparisons

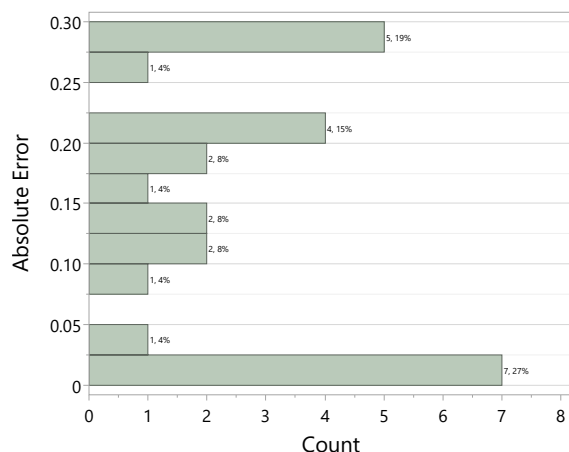
In Section 5.4.3, a test was performed to compare the required climb gradients output by the generalized and FAA Part 25 flight performance requirements. Figure 5.8 in Section 5.4.3 reported histograms for 2-engine aircraft only. Additional histograms for 3- and 4-engine aircraft are illustrated in Figs. C.2 and C.3, respectively.



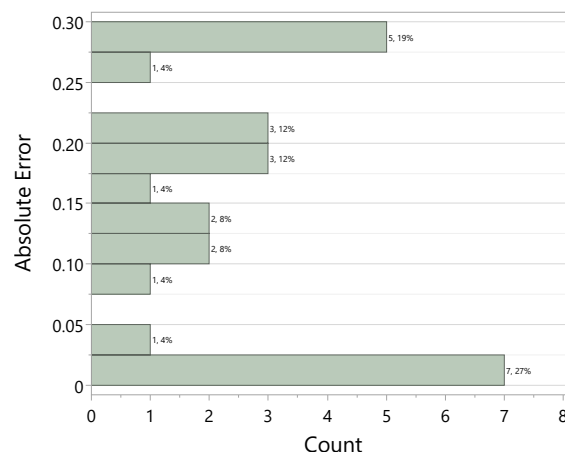
(a) 25.111 and 25.121c.



(b) 25.121a.

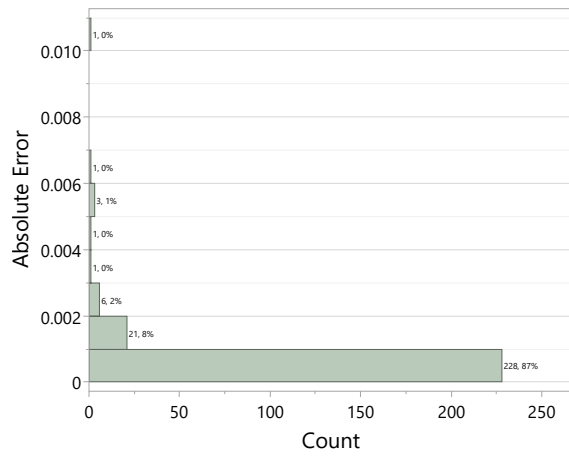


(c) 25.121b.

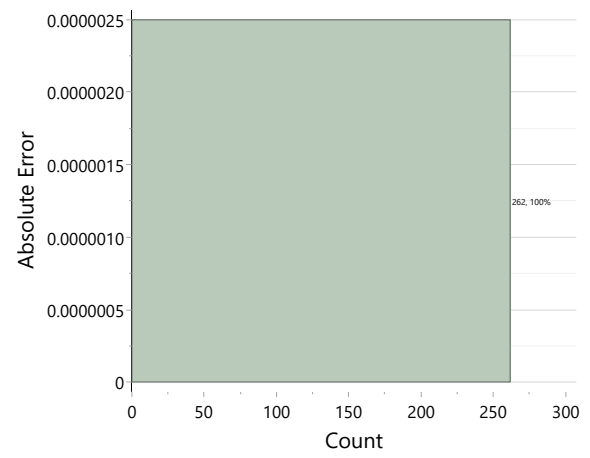


(d) 25.121d.

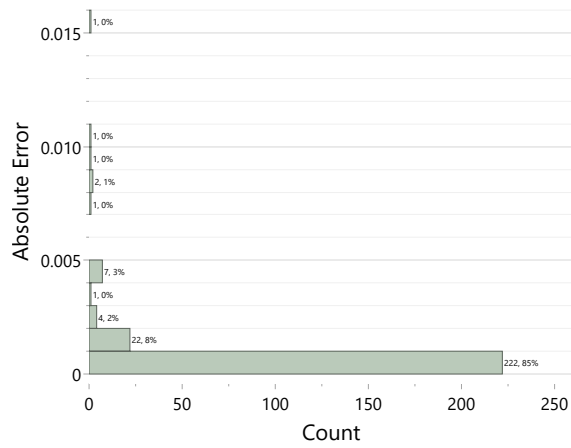
Figure C.2: Absolute error between FAA and proposed flight performance requirements for 3-engine aircraft.



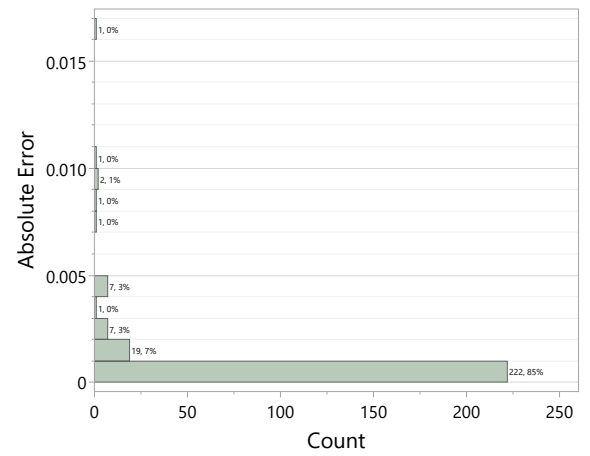
(a) 25.111 and 25.121c.



(b) 25.121a.



(c) 25.121b.

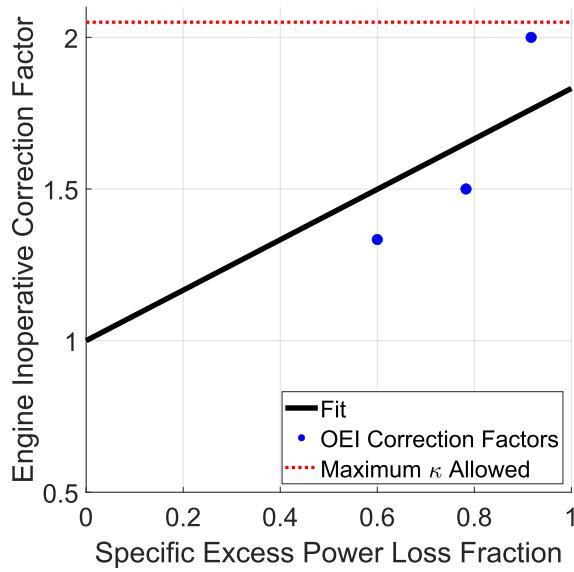


(d) 25.121d.

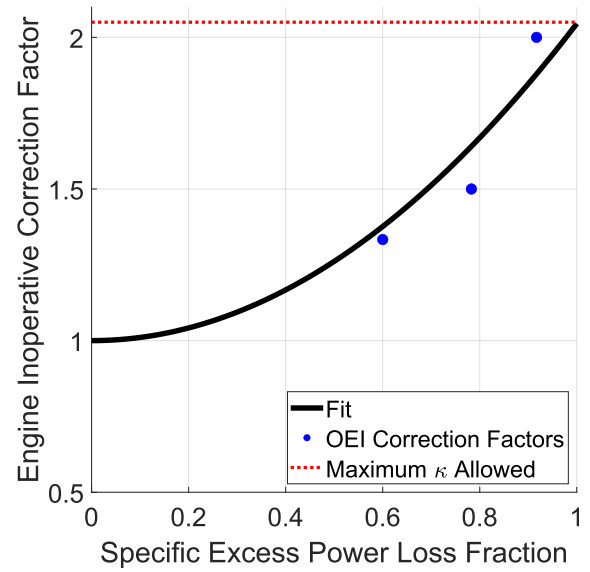
Figure C.3: Absolute error between FAA and proposed flight performance requirements for 4-engine aircraft.

C.5 Alternative Engine Inoperative Correction Factor Function Constructions

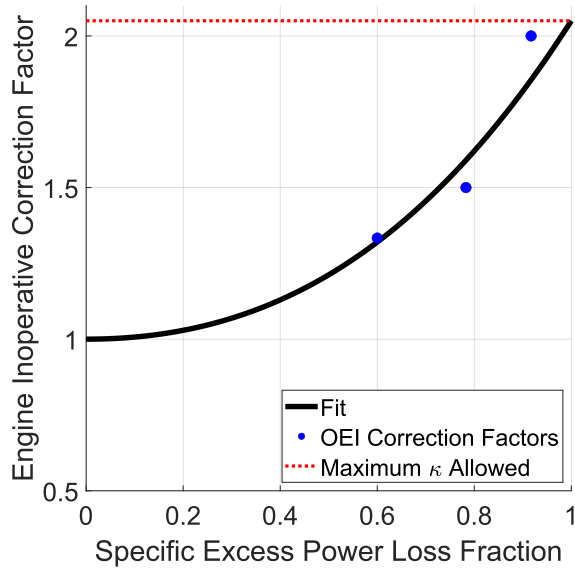
In Section 5.5.5, curve fits for defining the engine inoperative correction factor were generated by replacing the inequality constraint of $\hat{\kappa}(1) \leq 2.25$ with $\hat{\kappa}(1) \leq 2.05$. The resulting curve fits are illustrated in Fig. C.4.



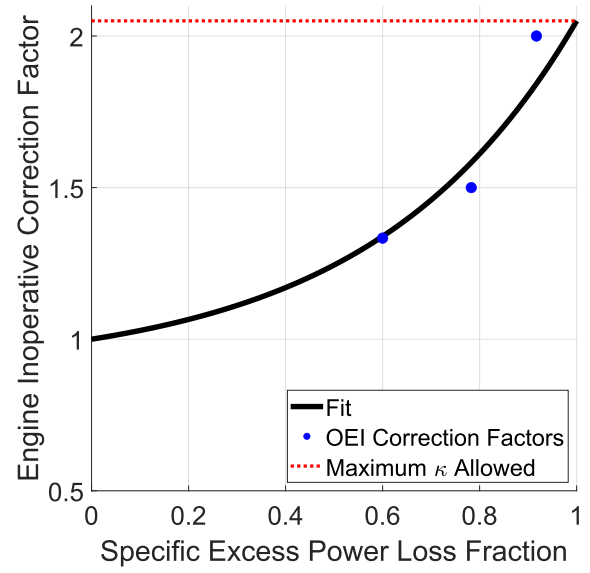
(a) Linear function.



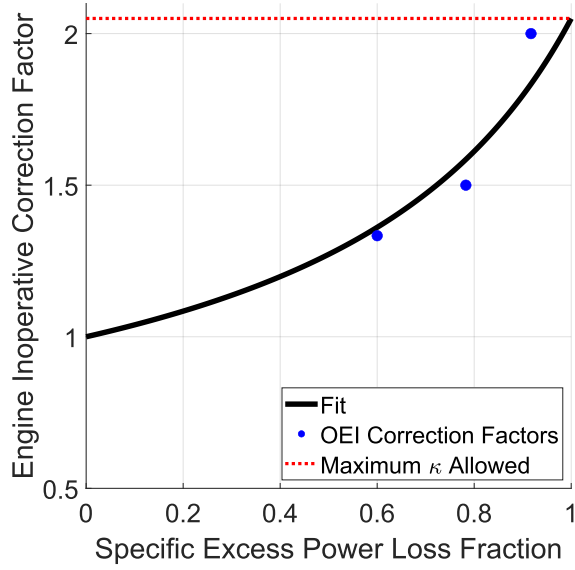
(b) Quadratic function.



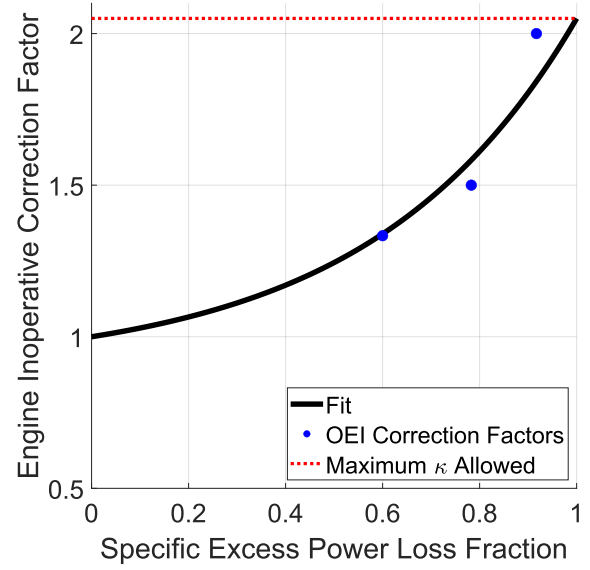
(c) Cubic function.



(d) Exponential function.



(e) Tangent function.



(f) Hyperbolic sine function.

Figure C.4: Candidate engine inoperative correction factor curve fits.

C.6 Notional Elysian E9X Modeling

A notional Elysian E9X was modeled in Section 5.6 to demonstrate how an aircraft is sized using the proposed performance requirements and engine inoperative correction factor. Appendix C.6 lists the modeling parameters and assumptions made during the various modeling stages.

C.6.1 Sizing in FAST

First, Tab. C.9 lists the parameters utilized to size the notional E9X in FAST [177]. In addition to the modeling information provided in Section 5.6, three assumptions were made:

1. The takeoff speed was assumed to be 10% faster than the stall speed.
2. The maximum rate of climb was set as a heuristic to ensure passenger comfort.
3. The climb/descent lift-to-drag ratio was approximated as 80% of the cruise lift-to-drag ratio.

Table C.9: Notional E9X sizing parameters. Values with asterisks were either assumed or modeled using information outside of Ref. [9].

Parameter	Value	Units
Takeoff Altitude*	0	m
Cruise Altitude	7,600	m
Takeoff Speed*	62.857	m/s
Approach Speed	74.594	m/s
Cruise Speed	0.600	Mach
Design Range	1,000	km
Maximum Rate of Climb*	11.43	m/s
Maximum Takeoff Weight	76,000	kg
Airframe Weight Calibration Factor	0.829	—
Cruise Lift-to-Drag Ratio	23.00	—
Climb/Descent Lift-to-Drag Ratio*	18.40	—
Wing Loading	509.68	kg/m ²
Power Loading	0.0666	N/W
Propeller Efficiency	87	%
Electric Motor Efficiency	95	%
Battery Efficiency	95	%
Battery Gravimetric Specific Energy	360	Wh/kg
Electric Motor Power-to-Weight Ratio	5	kW/kg

The notional E9X was sized using the design mission profile specified by the boundary conditions in Tab. C.10.

Table C.10: Assumed notional E9X design mission profile. Airspeeds listed in m/s are denoted as true or equivalent airspeeds.

Segment	Initial Altitude [m]	Final Altitude [m]	Initial Airspeed [m/s or Mach]	Final Airspeed [m/s or Mach]
Takeoff	0	0	0.00 (TAS)	62.86 (TAS)
Climb	0	914	62.86 (TAS)	102.89 (EAS)
Climb	914	7,600	102.89 (EAS)	102.89 (EAS)
Acceleration	7,600	7,600	102.89 (EAS)	M0.60
Cruise	7,600	7,600	M0.60	M0.60
Deceleration	7,600	7,600	M0.60	102.89 (EAS)
Descent	7,600	914	102.89 (EAS)	102.89 (EAS)
Descent	914	0	102.89 (EAS)	74.59 (TAS)
Landing	0	0	74.59 (TAS)	0.00 (TAS)

C.6.2 Constraint Diagram Modeling

Constraint diagrams were generated to assess the impacts of the generalized flight performance requirements and engine inoperative correction factor. The parameters used to model the notional E9X are included in Tab. C.11. With exception to the constraint modeling differences discussed in Section 5.6, all other constraints were modeled using the procedure provided in Ref. [9]. Given the limited public information available about the E9X, multiple values were either modeled using first-order estimations or assumed to be a fixed value, as described in the following paragraphs.

Table C.11: Notional E9X constraint diagram modeling parameters. Values with asterisks were either assumed or modeled using information outside of Ref. [9].

Parameter	Value	Units
Takeoff Altitude*	0	m
Cruise Altitude	7,600	m
Service Ceiling	8,000	m
Stall Speed	57.143	m/s
Takeoff Speed*	62.857	m/s
Approach Speed	74.594	m/s
Cruise Speed	0.600	Mach
Diversion Speed	0.400	Mach
Takeoff Field Length	2,000	m
Landing Field Length	2,000	m
Maximum Takeoff Weight	76,000	kg
Landing Weight Fraction	1.00	—
Aspect Ratio	12.00	—
Takeoff Lift Coefficient	2.50	—
Cruise Lift Coefficient*	1.00	—
Landing Lift Coefficient	2.50	—
Takeoff Parasite Drag Coefficient*	0.0618	—
Cruise Parasite Drag Coefficient*	0.0168	—
Landing Parasite Drag Coefficient*	0.1068	—
Takeoff Oswald Efficiency*	0.7468	—
Cruise Oswald Efficiency*	0.7861	—
Landing Oswald Efficiency*	0.7075	—
Cruise Power Calibration Factor	0.10	—
Diversion Power Calibration Factor	0.20	—

Cruise Lift Coefficient Estimation The cruise lift coefficient was varied until the constraint contour for FAA Part 25.121c reasonably matched that of the constraint diagram in the literature [9], and was estimated to be 1.0. As a result of the limited geometric information known about the aircraft, the cruise lift coefficient is likely overestimated to ensure that the contours on the constraint diagram reasonably matched those in literature.

Parasite Drag Coefficient Estimation The parasite drag coefficients were estimated using Raymer’s Equivalent Skin Friction Method for transport aircraft [183]. This approach assumes that the wetted area is a function of the maximum takeoff weight, $S_{\text{wet}} = 10^c (\text{MTOW})^d$, where $c = 0.0199$, and $d = 0.7531$. The cruise parasite drag coefficient is then estimated as $C_{D_0} = C_f S_{\text{wet}} / S_{\text{ref}}$, with $C_f = 0.003$. The takeoff and landing parasite drag coefficients were estimated by adding 0.05 and 0.10, respectively, to the cruise parasite drag coefficient to account for flap deflections and landing gear deployment.

Oswald Efficiency Factor Estimation To estimate the Oswald efficiency factor, the authors of Ref. [9] utilize Obert’s *Aerodynamic Design of Transport Aircraft* (Fig. 40.36) [209]. In this figure, there are three equations relating the aspect ratio to the Oswald efficiency factor. Since the E9X is a transformative flight concept, the most aggressive (drag-minimizing) relation was selected: $e = (1.05 + 0.0070\pi\mathcal{AR})^{-1}$. Since the E9X’s fuselage is shorter relative to its wingspan, the authors of Ref. [9] also add 0.025 to the Oswald efficiency factor. The climb and landing Oswald efficiency factors were obtained by multiplying the cruise Oswald efficiency factor by 95% and 90%, respectively, to account for flap deflections and landing gear deployment.

Cruise and Diversion Constraints As discussed in Section 5.6, the cruise and diversion constraints were scaled by a factor of $1/\sigma^a$, where σ is the density ratio at the respective altitude, and a is a user-selected parameter to tune the constraint. For the cruise and diversion constraints, a was tuned to be 0.1 and 0.2, respectively. These values are reasonably close to 0, highlighting the minimal tuning required for these constraints, especially since the E9X is powered by electric motors that do not lapse at altitudes above sea level.

Such scale factors, however, are typically used to account for gas turbine engine lapsing. Although the E9X is a battery-electric aircraft, utilizing the tuning parameter enables the constraints to be seamlessly adapted to propulsion systems utilizing gas turbine engines, especially if the E9X’s Reserve Energy System is modeled in future work.

C.6.3 Constraint Diagram Validation

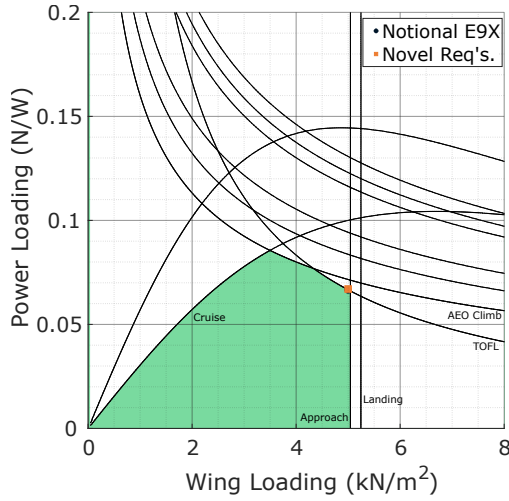
In Section 5.6, the results from Tab. 5.15 validated the constraint diagram at a wing loading of 4 kN/m². Table C.12 validates the constraint diagram at a wing loading of 6 kN/m².

Table C.12: Constraint diagram comparison at a 6 kN/m² wing loading.

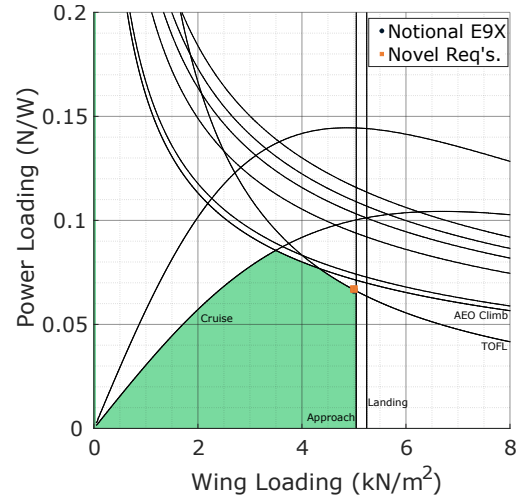
Constraint	Model	Literature	Absolute Error	Relative Error
FAA 25.121a	0.1175	0.1363	-0.0188	-13.793%
FAA 25.121b	0.1045	0.1063	-0.0018	-1.693%
FAA 25.121c	0.1105	0.1201	-0.0096	-7.993%
FAA 25.121d	0.0750	0.0797	-0.0047	-5.897%
FAA 25.119	0.0870	0.0856	+0.0014	+1.636%
AEO Climb	0.0660	0.0662	-0.0002	-0.302%
Cruise	0.1050	0.1040	+0.0010	+0.961%
Diversion	0.1433	0.1445	-0.0012	-0.830%
Takeoff Field Length	0.0560	0.0505	+0.0055	+10.891%

C.6.4 Constraint Diagrams under Engine Inoperative Conditions

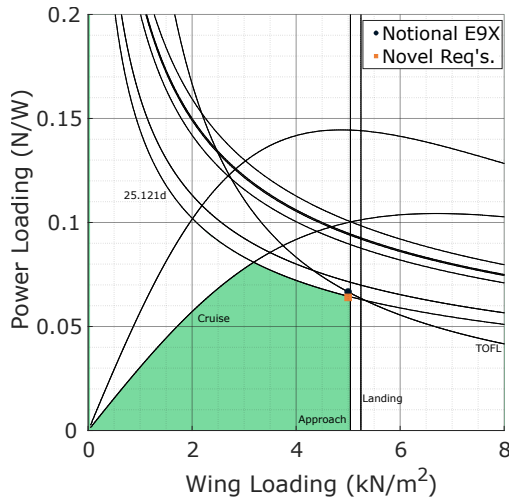
In Section 5.6, Fig. 5.12 illustrated the constraint diagrams for 1- and 6-engine inoperative conditions. Figure C.5 illustrates the constraint diagrams for 2-, 3-, 4-, and 5-engine inoperative conditions.



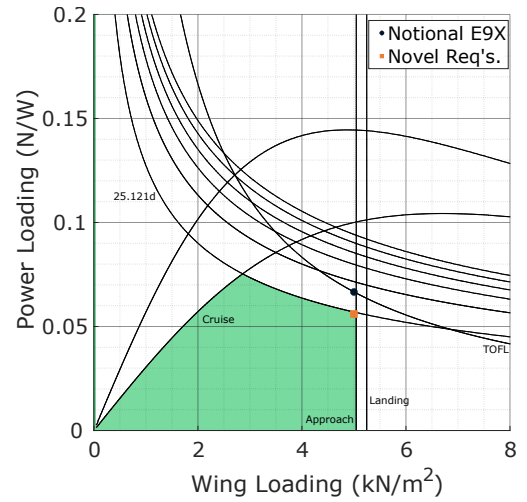
(a) Two-Engines Inoperative.



(b) Three-Engines Inoperative.



(c) Four-Engines Inoperative.



(d) Five-Engines Inoperative.

Figure C.5: Constraint diagrams using the proposed climb requirements and engine inoperative correction factor. The feasible region is shaded in green.

Appendix D

Modeling NASA’s SUSAN Aircraft

D.1 SUSAN Modeling Parameters

The parameters used to model SUSAN in FAST are listed in Tab. D.1. Recall that, to simplify the propulsion system analysis, all parts of the electrified propulsion system (batteries, buses, circuit breakers, etc.) *excluding* the cables were removed from the propulsion system; their mass was accounted for in the “EAP Weight” group. To simplify the safety analysis, only generator and motor failures were considered; all other electrical components were assumed to be fully operational, as well as the turbofan engine and fans in the distributed propulsion system.

Table D.1: Parameters to model SUSAN in FAST.

Parameter	Value	Units
Fuel Flow Calibration Factor	0.950	—
Airframe Calibration Factor	1.094	—
Entry into Service Year	2040	—
Passengers	190	—
Takeoff Speed	82.3	m/s
Cruise Speed	0.775	Mach
Cruise Altitude	11,278	m
Design Range	4,630	km
MTOW	86,586	kg
Wing Loading	634	kg/m ²
Fuel Weight	13,762	kg
EAP Weight	9,178	kg
Thrust-to-Weight Ratio	0.298	—
Battery Gravimetric Specific Energy	1,500	Wh/kg

D.2 Cable Length Modeling

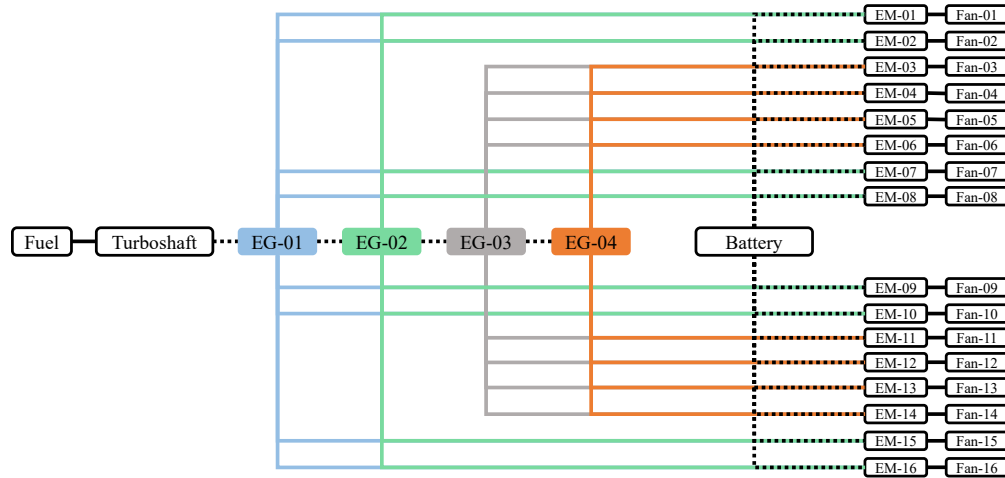
Table D.2 lists the cable lengths estimated from the digitized propulsion system schematic in Fig. 6.3.

Table D.2: Estimated cable lengths (all in m).

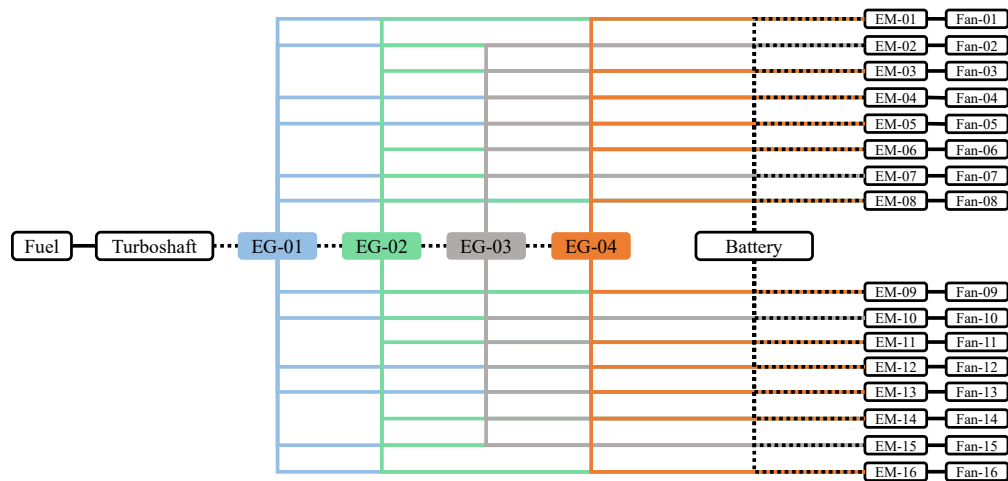
	Generator 1	Generator 2	Generator 3	Generator 4
Motors 01/16	18.66	18.07	17.50	17.15
Motors 02/15	17.50	16.92	16.34	15.98
Motors 03/14	16.40	15.80	15.22	14.85
Motors 04/13	15.35	14.72	14.11	13.73
Motors 05/12	14.37	13.68	13.05	12.66
Motors 06/11	13.36	12.70	12.06	11.62
Motors 07/10	12.36	11.71	11.08	10.64
Motors 08/09	11.34	10.70	10.08	9.66

D.3 Redundant Propulsion System Architectures

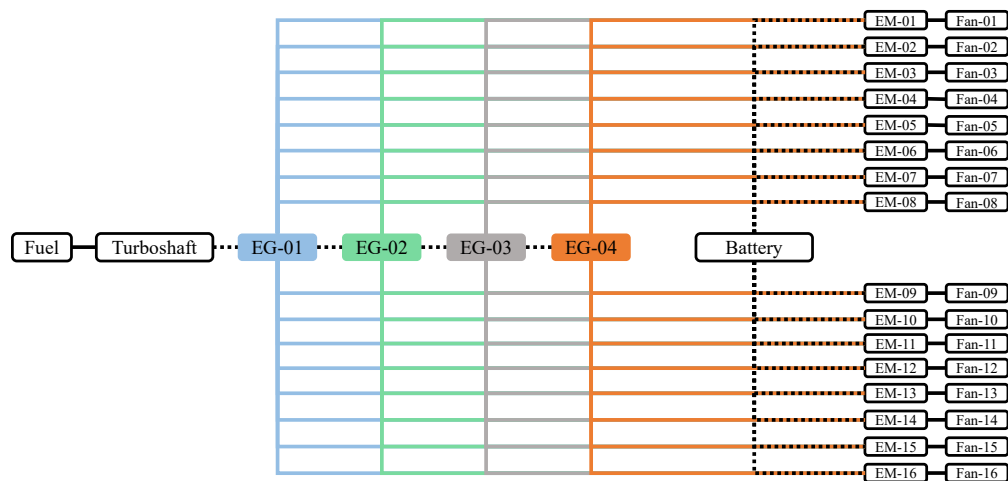
Figure D.1 illustrates the redundant propulsion system architectures installed on SUSAN and analyzed in FAST. The baseline propulsion architecture was previously illustrated in Fig. 6.4 in Section 6.3.2.



(a) Double Redundancy propulsion architecture.



(b) Triple Redundancy propulsion architecture.



(c) Quadruple Redundancy propulsion architecture.

Figure D.1: Propulsion system architectures analyzed in prior performance-safety trade studies.

Bibliography

- [1] National Aeronautics and Space Administration, “Electrified Aircraft Propulsion,” , 2024. URL <https://www1.grc.nasa.gov/aeronautics/eap/>.
- [2] Federal Aviation Administration, “Flight Test Guide for Certification of Transport Category Airplanes,” Tech. Rep. AC No. 25-7D, United States Department of Transportation, 2018.
- [3] Gray, J. S., Hwang, J. T., Martins, J. R. R. A., Moore, K. T., and Naylor, B. A., “OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization,” *Structural and Multidisciplinary Optimization*, Vol. 59, No. 4, 2019, pp. 1075–1104. doi:10.1007/s00158-019-02211-z.
- [4] Parsa, L., “Flux-Switching Machine Based All-Electric Power Train for Future Aircraft,” Tech. rep., United States Department of Energy, 2025. doi:10.2172/2570817.
- [5] Walsh, P. P., and Fletcher, P., *Gas turbine performance*, John Wiley & Sons, 2004.
- [6] Nelson, R. C., *Flight Stability and Automatic Control*, 2nd ed., WCB/McGraw Hill New York, 1998.
- [7] Chau, T., and Duensing, J., “Conceptual Design of the Hybrid-Electric Subsonic Single Aft Engine (SUSAN) Electrofan Transport Aircraft,” *AIAA SCITECH 2024 Forum*, 2024, p. 1326. doi:10.2514/6.2024-1326.
- [8] Cinar, G., Garcia, E., and Mavris, D. N., “A framework for electrified propulsion architecture and operation analysis,” *Aircraft Engineering and Aerospace Technology*, Vol. 92, No. 5, 2020, pp. 675–684. doi:10.1108/AEAT-06-2019-0118.
- [9] de Vries, R., Wolleswinkel, R. E., Hoogreef, M., and Vos, R., “A new perspective on battery-electric aviation, part II: Conceptual design of a 90-seater,” *AIAA Scitech 2024 Forum*, 2024, p. 1490. doi:10.2514/6.2024-1490.
- [10] Spierling, T. A., Lynch, A., and Bradley, T. H., “Battery Optimization for Sustainable Aviation: A Comprehensive Analysis for Battery Electric Aircraft,” *Journal of Aircraft*, 2025, pp. 1–6. doi:10.2514/1.C038197.

- [11] Air Transportation Action Group, “Waypoint 2050,” , 2020. URL <https://aviationbenefits.org/other-environmental-challenges/climate-action/waypoint-2050/>.
- [12] International Civil Aviation Organization, “2022 Environmental Report,” , 2022. URL <https://www.icao.int/sites/default/files/sp-files/environmental-protection/Documents/EnvironmentReport-2010/ICAO-ENV-Report-2022-F4.pdf>.
- [13] McDonald, R. A., German, B. J., Takahashi, T., Bil, C., Anemaat, W., Chaput, A., Vos, R., and Harrison, N., “Future aircraft concepts and design methods,” *The Aeronautical Journal*, Vol. 126, No. 1295, 2022, pp. 92–124. doi:10.1017/aer.2021.110.
- [14] Bravo-Mosquera, P. D., Catalano, F. M., and Zingg, D. W., “Unconventional aircraft for civil aviation: A review of concepts and design methodologies,” *Progress in Aerospace Sciences*, Vol. 131, 2022, p. 100813. doi:10.1016/j.paerosci.2022.100813.
- [15] Brelje, B. J., and Martins, J. R., “Electric, hybrid, and turboelectric fixed-wing aircraft: A review of concepts, models, and design approaches,” *Progress in Aerospace Sciences*, Vol. 104, 2019, pp. 1–19. doi:10.1016/j.paerosci.2018.06.004.
- [16] Jansen, R., Brown, G. V., Felder, J. L., and Duffy, K. P., “Turboelectric aircraft drive key performance parameters and functional requirements,” *51st AIAA/SAE/ASEE joint propulsion conference*, 2015, p. 3890. doi:10.2514/6.2015-3890.
- [17] Abolmoali, P., Donovan, A., Patnaik, S. S., McCarthy, P., Dierker, D., Jones, N., and Buettner, R., “Integrated propulsive and thermal management system design for optimal hybrid electric aircraft performance,” *2020 AIAA/IEEE Electric Aircraft Technologies Symposium (EATS)*, IEEE, 2020, pp. 1–17. doi:10.2514/6.2020-3557.
- [18] Lents, C. E., Hardin, L. W., Rheume, J., and Kohlman, L., “Parallel hybrid gas-electric geared turbofan engine conceptual design and benefits analysis,” *52nd AIAA/SAE/ASEE joint propulsion conference*, 2016, p. 4610. doi:10.2514/6.2016-4610.
- [19] Decerio, D. P., and Hall, D. K., “Benefits of parallel hybrid electric propulsion for transport aircraft,” *IEEE Transactions on Transportation Electrification*, Vol. 8, No. 4, 2022, pp. 4054–4066. doi:10.1109/TTE.2022.3193622.
- [20] Clarke, S., Redifer, M., Papathakis, K., Samuel, A., and Foster, T., “X-57 power and command system design,” *2017 IEEE Transportation Electrification Conference and Expo (ITEC)*, IEEE, 2017, pp. 393–400. doi:10.1109/ITEC.2017.7993303.
- [21] Orefice, F., Della Vecchia, P., Ciliberti, D., and Nicolosi, F., “Aircraft conceptual design including powertrain system architecture and distributed propulsion,” *2019 AIAA/IEEE Electric Aircraft Technologies Symposium (EATS)*, IEEE, 2019, pp. 1–20. doi:10.2514/6.2019-4465.

- [22] Erhard, R. M., Clarke, M. A., and Alonso, J. J., “A low-cost aero-propulsive analysis of distributed electric propulsion aircraft,” *AIAA Scitech 2021 Forum*, 2021, p. 1200. doi:10.2514/6.2021-1200.
- [23] Tom, L., Khowja, M., Vakil, G., and Gerada, C., “Commercial aircraft electrification—Current state and future scope,” *Energies*, Vol. 14, No. 24, 2021, p. 8381. doi:10.3390/en14248381.
- [24] Adler, E. J., and Martins, J. R., “Hydrogen-powered aircraft: Fundamental concepts, key technologies, and environmental impacts,” *Progress in Aerospace Sciences*, Vol. 141, 2023, p. 100922. doi:10.1016/j.paerosci.2023.100922.
- [25] Adler, E., “Hydrogen-Powered Aircraft Design Optimization,” Ph.D. thesis, University of Michigan, 2024. doi:10.7302/25775.
- [26] World Economic Forum, “Clean Skies for Tomorrow - Sustainable Aviation Fuels as a Pathway to Net-Zero Aviation,” https://www3.weforum.org/docs/WEF_Clean_Skies_Tomorrow_SAF_Analytics_2020.pdf, 2020.
- [27] Marciello, V., Di Stasio, M., Ruocco, M., Trifari, V., Nicolosi, F., Meindl, M., Lemoine, B., and Caliandro, P., “Design exploration for sustainable regional hybrid-electric aircraft: A study based on technology forecasts,” *Aerospace*, Vol. 10, No. 2, 2023, p. 165. doi:10.3390/aerospace10020165.
- [28] Voskuijl, M., Van Bogaert, J., and Rao, A. G., “Analysis and design of hybrid electric regional turboprop aircraft,” *CEAS Aeronautical Journal*, Vol. 9, No. 1, 2018, pp. 15–25. doi:10.1007/s13272-017-0272-1.
- [29] Cinar, G., Cai, Y., Bendarkar, M. V., Burrell, A. I., Denney, R. K., and Mavris, D. N., “System analysis and design space exploration of regional aircraft with electrified powertrains,” *Journal of Aircraft*, Vol. 60, No. 2, 2023, pp. 382–409. doi:10.2514/1.C036919.
- [30] Cai, Y., Xie, J., Cinar, G., and Mavris, D. N., “Advanced 2030 turboprop aircraft modeling for the electrified powertrain flight demonstration program,” *2022 IEEE Transportation Electrification Conference & Expo (ITEC)*, IEEE, 2022, pp. 664–669. doi:10.1109/ITEC53557.2022.9813858.
- [31] Cinar, G., Cai, Y., Denney, R. K., and Mavris, D. N., “Modeling and Simulation of a Parallel Hybrid Electric Regional Aircraft for the Electrified Powertrain Flight Demonstration (EPFD) Program,” *2022 IEEE Transportation Electrification Conference & Expo (ITEC)*, IEEE, 2022, pp. 670–675. doi:10.1109/ITEC53557.2022.9813832.
- [32] Gnadt, A. R., Speth, R. L., Sabnis, J. S., and Barrett, S. R., “Technical and environmental assessment of all-electric 180-passenger commercial aircraft,” *Progress in Aerospace Sciences*, Vol. 105, 2019, pp. 1–30. doi:10.1016/j.paerosci.2018.11.002.

- [33] Milios, K., Hall, C., Burrell, A., Brooks, J., Kenny, J., Gladin, J., and Mavris, D., “Modeling and Simulation of a Parallel Hybrid-Electric Propulsion System-Electrified Powertrain Flight Demonstration (EPFD) Program,” *2022 IEEE Transportation Electrification Conference & Expo (ITEC)*, IEEE, 2022, pp. 682–687. doi:10.1109/ITEC53557.2022.9813981.
- [34] Fichtner, M., “Recent research and progress in batteries for electric vehicles,” *Batteries & Supercaps*, Vol. 5, No. 2, 2022, p. e202100224. doi:10.1002/batt.202100224.
- [35] Houache, M. S., Yim, C.-H., Karkar, Z., and Abu-Lebdeh, Y., “On the current and future outlook of battery chemistries for electric vehicles—mini review,” *Batteries*, Vol. 8, No. 7, 2022, p. 70. doi:10.3390/batteries8070070.
- [36] Tiede, B., O’Meara, C., and Jansen, R., “Battery key performance projections based on historical trends and chemistries,” *2022 IEEE Transportation Electrification Conference & Expo (ITEC)*, IEEE, 2022, pp. 754–759. doi:10.1109/ITEC53557.2022.9814008.
- [37] Epstein, A. H., and O’Flarity, S. M., “Considerations for reducing aviation’s co 2 with aircraft electric propulsion,” *Journal of propulsion and power*, Vol. 35, No. 3, 2019, pp. 572–582. doi:10.2514/1.B37015.
- [38] Beard, J. E., and Takahashi, T. T., “(Un) controlled Flight Into Terrain: A History of Obstacle Clearance Regulations,” *2018 AIAA Aerospace Sciences Meeting*, 2018, p. 1614. doi:10.2514/6.2018-1614.
- [39] Harish, A., Shi, M., Gladin, J., and Mavris, D., “Advanced 2030 single aisle aircraft modeling for the Electrified Powertrain Flight Demonstration program,” *2022 IEEE Transportation Electrification Conference & Expo (ITEC)*, IEEE, 2022, pp. 676–681. doi:10.1109/ITEC53557.2022.9813859.
- [40] de Vries, R., Wolleswinkel, R. E., Exalto, J., van den Berg, P., Vos, R., and Hoogreef, M., “Conceptual Redesign of a 90-Seater Battery-Electric Aircraft,” *AIAA AVIATION FORUM AND ASCEND 2025*, 2025, p. 3153. doi:10.2514/6.2025-3153.
- [41] Electra.aero, “Harnessing Electric Propulsion,” , 2026. URL <https://www.electra.aero/technology>.
- [42] Chau, T., Kenway, G., and Kiris, C. C., “Conceptual Exploration of Aircraft Configurations for the SUSAN Electrofan,” *AIAA SciTech 2022 Forum*, 2022, p. 2181. doi:10.2514/6.2022-2181.
- [43] Jansen, R., Kiris, C. C., Chau, T., Machado, L. M., Duensing, J. C., Mirhashemi, A., Chapman, J., French, B. D., Miller, L., Litt, J. S., et al., “Subsonic Single Aft Engine (SUSAN) Transport Aircraft Concept and Trade Space Exploration,” *AIAA SciTech 2022 Forum*, 2022, p. 2179. doi:10.2514/6.2022-2179.

- [44] Mirhashemi, A., Chapman, J., Miller, C., Julia, S., and Jansen, R., “Tail-Mounted Engine Architecture and Design for the Subsonic Single Aft Engine Electrofan Aircraft,” *AIAA SciTech 2022 Forum*, 2022, p. 2182. doi:10.2514/6.2022-2182.
- [45] Machado, L. M., Chau, T., Kenway, G. K., Duensing, J. C., and Kiris, C. C., “High-Fidelity Aerodynamic Analysis and Optimization of the SUSAN Electrofan Concept,” *AIAA SciTech 2022 Forum*, 2022, p. 2304. doi:10.2514/6.2022-2304.
- [46] Heersema, N., and Jansen, R., “Thermal Management System Trade Study for SUSAN Electrofan Aircraft,” *AIAA SciTech 2022 Forum*, 2022, p. 2302. doi:10.2514/6.2022-2302.
- [47] Haglage, J., Dever, T., Jansen, R., and Lewis, M., “Electrical System Trade Study for SUSAN Electrofan Concept Vehicle,” *AIAA SciTech 2022 Forum*, 2022, p. 2183. doi:10.2514/6.2022-2183.
- [48] Lynde, M. N., Campbell, R. L., and Hiller, B. R., “A Design Exploration of Natural Laminar Flow Applications for the SUSAN Electrofan Concept,” *AIAA Scitech 2022 Forum*, 2022, p. 2303. doi:10.2514/6.2022-2303.
- [49] Machado, L. M., Chau, T., Kenway, G. K., Duensing, J. C., and Kiris, C. C., “Preliminary Assessment of a Distributed Electric Propulsion System for the SUSAN Electrofan,” *AIAA SciTech 2023 Forum*, 2023, p. 1748. doi:10.2514/6.2023-1748.
- [50] Denham, C. L., and Jansen, R., “Initial regulatory and certification approach for the SUSAN electrofan concept,” *AIAA SciTech 2022 Forum*, 2022, p. 2180. doi:10.2514/6.2022-2180.
- [51] Denham, C. L., Chau, T., Ryan, W., and Jansen, R., “Mission Profiles for the SUSAN Electrofan Concept,” *AIAA SciTech 2023 Forum*, 2023, p. 1938. doi:10.2514/6.2023-1938.
- [52] Wolleswinkel, R. E., de Vries, R., Hoogreef, M., and Vos, R., “A new perspective on battery-electric aviation, part I: Reassessment of achievable range,” *AIAA Scitech 2024 Forum*, 2024, p. 1489. doi:10.2514/6.2024-1489.
- [53] Wang, Z., De Breuker, R., Sodja, J., Wolleswinkel, R. E., de Vries, R., and Giuffré, A., “A study on wing structural design and sizing for a large battery-electric aircraft,” *AIAA Aviation Forum and Ascend 2025*, 2025, p. 3470. doi:10.2514/6.2025-3470.
- [54] Margalida, G., Sinnige, T., de Vries, R., Exalto, J., and Wolleswinkel, R. E., “Performance Analysis of Distributed Propellers: Efficiency and Noise Trade-offs in Full-Electric Regional Aircraft,” *AIAA AVIATION FORUM AND ASCEND 2025*, 2025, p. 3086. doi:10.2514/6.2025-3086.

- [55] Lukaczyk, T. W., Wendorff, A. D., Colonna, M., Economon, T. D., Alonso, J. J., Orra, T. H., and Ilario, C., “SUAVE: an open-source environment for multi-fidelity conceptual vehicle design,” *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2015, p. 3087. doi:10.2514/6.2015-3087.
- [56] Kruger, M., “The Challenges and Potential Benefits of Electrified Propulsion for Aircraft,” Ph.D. thesis, University of Southern California, 2022.
- [57] Wu, E., Kenway, G., Mader, C. A., Jasa, J., and Martins, J. R., “pyOptSparse: A Python framework for large-scale constrained nonlinear optimization of sparse systems,” *Journal of Open Source Software*, Vol. 5, No. 54, 2020, p. 2564. doi:10.21105/joss.02564.
- [58] Gratz, J., Kirk, J., Recine, C., Jasa, J., Aretskin-Hariton, E., Moore, K., and Marfatia, K., “Aviary: An Open-Source Multidisciplinary Design, Analysis, and Optimization Tool for Modeling Aircraft with Analytic Gradients,” *AIAA AVIATION FORUM AND ASCEND 2024*, 2024, p. 4219. doi:10.2514/6.2024-4219.
- [59] David, C., Delbecq, S., Defoort, S., Schmollgruber, P., Benard, E., and Pommier-Budinger, V., “From FAST to FAST-OAD: An open source framework for rapid Overall Aircraft Design,” *IOP Conference Series: Materials Science and Engineering*, Vol. 1024, IOP Publishing, 2021, p. 012062. doi:10.1088/1757-899X/1024/1/012062.
- [60] De Vries, R., Hoogreef, M. F., and Vos, R., “Range equation for hybrid-electric aircraft with constant power split,” *Journal of Aircraft*, Vol. 57, No. 3, 2020, pp. 552–557. doi:10.2514/1.C035734.
- [61] Harish, A., Gladin, J., and Mavris, D., “Universal Range Equation for Unconventional Aircraft Concepts,” *2022 IEEE Transportation Electrification Conference & Expo (ITEC)*, IEEE, 2022, pp. 1057–1063. doi:10.1109/ITEC53557.2022.9814012.
- [62] Trifari, V., Ruocco, M., Cusati, V., Nicolosi, F., and De Marco, A., “Multi-disciplinary analysis and optimization Java tool for aircraft design,” *Proceedings of the 31st ICAS Conference (International Council of the Aeronautical Sciences)*, Belo Horizonte, Brazil, 2018, pp. 9–14.
- [63] Nicolosi, F., De Marco, A., Attanasio, L., and Vecchia, P. D., “Development of a Java-based framework for aircraft preliminary design and optimization,” *Journal of Aerospace Information Systems*, Vol. 13, No. 6, 2016, pp. 234–242. doi:10.2514/1.I010404.
- [64] Trifari, V., De Marco, A., Di Stasio, M., Ruocco, M., Nicolosi, F., Grazioso, G., Ahuja, V., and Hartfield, R., “An aircraft design workflow using the automatic knowledge-based modelling tool JPAD Modeller,” *AIAA AVIATION 2022 Forum*, 2022, p. 3737. doi:10.2514/6.2022-3737.
- [65] Martins, J. R. R. A., and Ning, A., *Engineering Design Optimization*, Cambridge University Press, 2022.

- [66] Cinar, G., “A methodology for dynamic sizing of electric power generation and distribution architectures,” Ph.D. thesis, Georgia Institute of Technology, 2018.
- [67] Cinar, G., Cai, Y., Chakraborty, I., and Mavris, D. N., “Sizing and optimization of novel general aviation vehicles and propulsion system architectures,” *2018 Aviation Technology, Integration, and Operations Conference*, 2018, p. 3974. doi:10.2514/6.2018-3974.
- [68] Anderson, J. D., *Introduction to flight*, 7th ed., McGraw Hill, New York, 2012.
- [69] Doff-Sotta, M., Cannon, M., and Bacic, M., “Optimal energy management for hybrid electric aircraft,” *IFAC-PapersOnLine*, Vol. 53, No. 2, 2020, pp. 6043–6049. doi:10.1016/j.ifacol.2020.12.1672, 21st IFAC World Congress.
- [70] Leite, J. P. S. P., and Voskuijl, M., “Optimal energy management for hybrid-electric aircraft,” *Aircraft Engineering and Aerospace Technology*, Vol. 92, No. 6, 2020, pp. 851–861. doi:10.1108/AEAT-03-2019-0046.
- [71] Misley, A., D’Arpino, M., Ramesh, P., and Canova, M., “A real-time energy management strategy for hybrid electric aircraft propulsion systems,” *2021 AIAA/IEEE Electric Aircraft Technologies Symposium (EATS)*, IEEE, 2021, pp. 1–11. doi:10.23919/EATS52162.2021.9704831.
- [72] Trawick, D., Milios, K., Gladin, J. C., and Mavris, D. N., “A method for determining optimal power management schedules for hybrid electric airplanes,” *2019 AIAA/IEEE Electric Aircraft Technologies Symposium (EATS)*, IEEE, 2019, pp. 1–21. doi:10.2514/6.2019-4500.
- [73] Barufaldi, G., Morales, M., and Silva, R. G., “Energy optimal climb performance of electric aircraft,” *AIAA Scitech 2019 Forum*, 2019, p. 0830. doi:10.2514/6.2019-0830.
- [74] Oliviero, F., and Swannet, K., “Optimal Energy Management Strategies and Mission Profiles for a generic Hybrid Aircraft,” *AIAA AVIATION 2023 Forum*, 2023, p. 4224. doi:10.2514/6.2023-4224.
- [75] Lorenz, L., Seitz, A., Kuhn, H., and Sizmann, A., *Hybrid power trains for future mobility*, Deutsche Gesellschaft für Luft-und Raumfahrt-Lilienthal-Oberth eV Bonn, Germany, 2014.
- [76] Isikveren, A. T., Kaiser, S., Pornet, C., and Vratny, P. C., “Pre-design strategies and sizing techniques for dual-energy aircraft,” *Aircraft Engineering and Aerospace Technology: An International Journal*, Vol. 86, No. 6, 2014, pp. 525–542. doi:10.1108/AEAT-08-2014-0122.
- [77] Cinar, G., Mavris, D. N., Emeneth, M., Schneegans, A., Riediger, C., Fefermann, Y., and Isikveren, A., “Sizing, integration and performance evaluation of hybrid electric

- propulsion subsystem architectures,” *55th AIAA aerospace sciences meeting*, 2017, p. 1183. doi:10.2514/6.2017-1183.
- [78] Pornet, C., Gologan, C., Vratny, P. C., Seitz, A., Schmitz, O., Isikveren, A. T., and Hornung, M., “Methodology for sizing and performance assessment of hybrid energy aircraft,” *Journal of Aircraft*, Vol. 52, No. 1, 2015, pp. 341–352. doi:10.2514/1.C032716.
 - [79] Papadopoulos, K. I., Nasoulis, C. P., Ntouvelos, E. G., Gkoutzamanis, V. G., and Kalfas, A. I., “Power flow optimization for a hybrid-electric propulsion system,” *Journal of Engineering for Gas Turbines and Power*, Vol. 144, No. 11, 2022, p. 111013. doi:10.1115/1.4055478.
 - [80] Kruger, M., Byahut, S., Uranga, A., Gonzalez, J., Hall, D. K., and Dowdle, A., “Electrified aircraft trade-space exploration,” *2018 Aviation Technology, Integration, and Operations Conference*, 2018, p. 4227. doi:10.2514/6.2018-4227.
 - [81] De Vries, R., Brown, M., and Vos, R., “Preliminary sizing method for hybrid-electric distributed-propulsion aircraft,” *Journal of Aircraft*, Vol. 56, No. 6, 2019, pp. 2172–2188. doi:10.2514/1.C035388.
 - [82] Ascher, U. M., and Greif, C., *A first course on numerical methods*, SIAM, 2011.
 - [83] Lawhorn, D., Rallabandi, V., and Ionel, D. M., “Scalable graph theory approach for electric aircraft power system optimization,” *2019 AIAA/IEEE Electric Aircraft Technologies Symposium (EATS)*, IEEE, 2019, pp. 1–5. doi:10.2514/6.2019-4488.
 - [84] Lawhorn, D., Rallabandi, V., and Ionel, D. M., “A network graph technique for the design of electric aircraft power systems,” *2020 IEEE Transportation Electrification Conference & Expo (ITEC)*, IEEE, 2020, pp. 809–813. doi:10.1109/ITEC48692.2020.9161492.
 - [85] Federal Aviation Administration, “Part 23 — Airworthiness Standards: Normal Category Airplanes,” , 2025. URL <https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-23>.
 - [86] Federal Aviation Administration, “Part 25 — Airworthiness Standards: Transport Category Airplanes,” , 2025. URL <https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-25>.
 - [87] European Union Aviation Safety Agency, “CS-23 Initial Issue,” , 2003. URL <https://www.easa.europa.eu/en/document-library/certification-specifications/cs-23-initial-issue>.
 - [88] European Union Aviation Safety Agency, “CS-25 Amendment 28,” , 2023. URL <https://www.easa.europa.eu/en/document-library/certification-specifications/cs-25-amendment-28>.

- [89] Federal Aviation Administration, “Subpart B – Flight,” , 2025. URL <https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-25/subpart-B>.
- [90] Borer, N. K., Patterson, M. D., Viken, J. K., Moore, M. D., Bevirt, J., Stoll, A. M., and Gibson, A. R., “Design and performance of the NASA SCEPTOR distributed electric propulsion flight demonstrator,” *16th AIAA aviation technology, integration, and operations conference*, 2016, p. 3920. doi:10.2514/6.2016-3920.
- [91] Schlickenmaier, H. W., Anderson, M., Harrison, E., Hooper, E. H., Knickerbocker, J., Larson, M., Lawson, J., Voss, M., Wilkinson, R., and Turnberg, J., “X-57 Maxwell Airworthiness Validation Plan,” Tech. Rep. NASA/CR-20220015049, National Aeronautics and Space Administration, 2023.
- [92] Litt, J. S., Sachs-Wetstone, J. J., Simon, D. L., Sowers, T. S., Owen, A. K., Bell, M. E., Guthrie, B. E., Lehan, J., and Castro, A., “Flight Simulator Demonstration of Powertrain Failure Mitigation in a Partial Turboelectric Aircraft,” *AIAA AVIATION 2023 Forum*, 2023, p. 4156. doi:10.2514/6.2023-4156.
- [93] Xie, J., Cai, Y., Sarojini, D., Harrison, E. D., and Mavris, D. N., “Certification-constrained vertical tail sizing and power split optimization for distributed electric propulsion aircraft,” *Journal of Aircraft*, Vol. 60, No. 4, 2023, pp. 1272–1289. doi:10.2514/1.C037239.
- [94] Borer, N. K., Wallace, R. D., Reynolds, J. R., Cox, D. E., Sales, C., Williams, T. L., and Ringelberg, W. M., “Distributed thrust takeoff for the NASA X-57 Mod IV flight demonstrator,” *34th Congress of the International Council of the Aeronautical Sciences*, 2024.
- [95] Vechtel, D., and Buch, J.-P., “Aspects of yaw control design of an aircraft with distributed electric propulsion,” *CEAS aeronautical journal*, Vol. 13, No. 4, 2022, pp. 847–860. doi:10.1007/s13272-022-00595-1.
- [96] Orefice, F., Nicolosi, F., Della Vecchia, P., and Ciliberti, D., “Conceptual Design of Commuter Aircraft Including Distributed Electric Propulsion,” *AIAA Aviation 2020 Forum*, 2020, p. 2627. doi:10.2514/6.2020-2627.
- [97] Biser, S., Filipenko, M., Boll, M., Kastner, N., Atanasov, G., Hepperle, M., Keller, D., Vechtel, D., and Noe, M., “Design space exploration study and optimization of a distributed turbo-electric propulsion system for a regional passenger aircraft,” *2020 AIAA/IEEE Electric Aircraft Technologies Symposium (EATS)*, IEEE, 2020, pp. 1–27. doi:10.2514/6.2020-3592.
- [98] Nguyen Van, E., Troillard, P., Jézégou, J., Alazard, D., Pastor, P., and Döll, C., “Reduction of vertical tail using differential thrust: Influence on flight control and certification,” *AEGATS 2018*, 2018, pp. 1–8.

- [99] Hoogreef, M. F., and Soikkeli, J. S., “Flight dynamics and control assessment for differential thrust aircraft in engine inoperative conditions including aero-propulsive effects,” *CEAS Aeronautical Journal*, Vol. 13, No. 3, 2022, pp. 739–762. doi:10.1007/s13272-022-00591-5.
- [100] Xie, J., Bendarkar, M. V., Cai, Y., and Mavris, D. N., “Evaluating “Critical-Engine-Inoperative” Conditions for Distributed Hybrid-Electric Aircraft,” *Journal of Aircraft*, 2025, pp. 1–18. doi:10.2514/1.C038127.
- [101] Steiner, H.-J., Vratny, P. C., Gologan, C., Wieczorek, K., Isikveren, A. T., and Hornung, M., “Optimum number of engines for transport aircraft employing electrically powered distributed propulsion,” *CEAS Aeronautical Journal*, Vol. 5, No. 2, 2014, pp. 157–170. doi:10.1007/s13272-013-0096-6.
- [102] Bendarkar, M. V., Sarojini, D., Harrison, E., and Mavris, D. N., “Evaluation of off-nominal performance and reliability of a distributed electric propulsion aircraft during early design,” *AIAA scitech 2021 forum*, 2021, p. 1723. doi:10.2514/6.2021-1723.
- [103] Roskam, J., *Airplane Design*, DAR Corporation, 1997.
- [104] Blaesser, N. J., “Propeller-Wing Integration on the Parallel Electric-Gas Architecture with Synergistic Utilization Scheme (PEGASUS) Aircraft,” *AIAA Scitech 2019 Forum*, 2019, p. 1809. doi:10.2514/6.2019-1809.
- [105] Nüesch, T., Ott, T., Ebbesen, S., and Guzzella, L., “Cost and fuel-optimal selection of HEV topologies using particle swarm optimization and dynamic programming,” *2012 American Control Conference (ACC)*, IEEE, 2012, pp. 1302–1307. doi:10.1109/ACC.2012.6314868.
- [106] Buettner, R., Herber, D. R., Abolmoali, P. C., and Patnaik, S. S., “An automated design tool for generation and selection of optimal aircraft thermal management system architectures,” *AIAA Propulsion and Energy 2021 Forum*, 2021, p. 3718. doi:10.2514/6.2021-3718.
- [107] Herber, D. R., Guo, T., and Allison, J. T., “Enumeration of architectures with perfect matchings,” *Journal of Mechanical Design*, Vol. 139, No. 5, 2017, p. 051403. doi:10.1115/1.4036132.
- [108] Herber, D. R., Allison, J. T., Buettner, R., Abolmoali, P., and Patnaik, S. S., “Architecture generation and performance evaluation of aircraft thermal management systems through graph-based techniques,” *AIAA Scitech 2020 Forum*, 2020, p. 0159. doi:10.2514/6.2020-0159.
- [109] Bragalli, C., D’Ambrosio, C., Lee, J., Lodi, A., and Toth, P., “On the optimal design of water distribution networks: a practical MINLP approach,” *Optimization and Engineering*, Vol. 13, No. 2, 2012, pp. 219–246. doi:10.1007/s11081-011-9141-7.

- [110] Castillo, L., and González, A., “Distribution network optimization: finding the most economic solution by using genetic algorithms,” *European Journal of Operational Research*, Vol. 108, No. 3, 1998, pp. 527–537. doi:10.1016/S0377-2217(97)00050-7.
- [111] D’Ambrosio, C., Lodi, A., Wiese, S., and Bragalli, C., “Mathematical programming techniques in water network optimization,” *European Journal of Operational Research*, Vol. 243, No. 3, 2015, pp. 774–788. doi:10.1016/j.ejor.2014.12.039.
- [112] Pei, M., Lin, P., Du, J., Li, X., and Chen, Z., “Vehicle dispatching in modular transit networks: A mixed-integer nonlinear programming model,” *Transportation Research Part E: Logistics and Transportation Review*, Vol. 147, 2021, p. 102240. doi:10.1016/j.tre.2021.102240.
- [113] Ploskas, N., Laughman, C., Raghunathan, A. U., and Sahinidis, N. V., “Optimization of circuitry arrangements for heat exchangers using derivative-free optimization,” *Chemical Engineering Research and Design*, Vol. 131, 2018, pp. 16–28. doi:10.1016/j.cherd.2017.05.015.
- [114] Wyatt, D. F., Wynn, D. C., Jarrett, J. P., and Clarkson, P. J., “Supporting product architecture design using computational design synthesis with network structure constraints,” *Research in Engineering Design*, Vol. 23, No. 1, 2012, pp. 17–52. doi:10.1007/s00163-011-0112-y.
- [115] Herber, D. R., “Enhancements to the perfect matching approach for graph enumeration-based engineering challenges,” *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 84003, American Society of Mechanical Engineers, 2020, p. V11AT11A047. doi:10.1115/DETC2020-22774.
- [116] Buettner, R., Herber, D. R., and Patnaik, S. S., “Innovating Air Cycle Machine Design: A MAGMA-Driven Approach for Aircraft Thermal Management,” *AIAA SCITECH 2025 Forum*, 2025, p. 1473. doi:10.2514/6.2025-1473.
- [117] Peddada, S. R., Herber, D. R., Pangborn, H. C., Alleyne, A. G., and Allison, J. T., “Optimal flow control and single split architecture exploration for fluid-based thermal management,” *Journal of Mechanical Design*, Vol. 141, No. 8, 2019, p. 083401. doi:10.1115/1.4043203.
- [118] Bayrak, A. E., Ren, Y., and Papalambros, P. Y., “Design of hybrid-electric vehicle architectures using auto-generation of feasible driving modes,” *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 55843, American Society of Mechanical Engineers, 2013, p. V001T01A005. doi:10.1115/DETC2013-13043.

- [119] Bayrak, A. E., Ren, Y., and Papalambros, P. Y., “Topology generation for hybrid electric vehicle architecture design,” *Journal of Mechanical Design*, Vol. 138, No. 8, 2016, p. 081401. doi:10.1115/1.4033656.
- [120] Arney, D. C., and Wilhite, A. W., “Modeling space system architectures with graph theory,” *Journal of Spacecraft and Rockets*, Vol. 51, No. 5, 2014, pp. 1413–1429. doi:10.2514/1.A32578.
- [121] Williams, M. A., Koeln, J. P., Pangborn, H. C., and Alleyne, A. G., “Dynamical graph models of aircraft electrical, thermal, and turbomachinery components,” *Journal of Dynamic Systems, Measurement, and Control*, Vol. 140, No. 4, 2018, p. 041013. doi:10.1115/1.4038341.
- [122] Laird, C., Docimo, D., Aksland, C. T., and Alleyne, A. G., “Graph-based design and control optimization of a hybrid electrical energy storage system,” *Dynamic Systems and Control Conference*, Vol. 84270, American Society of Mechanical Engineers, 2020, p. V001T11A004. doi:10.1115/DSCC2020-3233.
- [123] Thompson, A. F., Iezzi, A. J., Pangborn, H. C., Madabhushi, R. K., and Atassi, O. V., “Combined Design and Open-Loop Control Optimization for Propulsion, Power, and Thermal Management of Hybrid-electric Aircraft,” *2023 IEEE Conference on Control Technology and Applications (CCTA)*, IEEE, 2023, pp. 955–962. doi:10.1109/CCTA54093.2023.10252309.
- [124] Docimo, D. J., Kang, Z., James, K. A., and Alleyne, A. G., “A novel framework for simultaneous topology and sizing optimization of complex, multi-domain systems-of-systems,” *Journal of Mechanical Design*, Vol. 142, No. 9, 2020, p. 091701. doi:10.1115/1.4046066.
- [125] Hartmann, C., Chenouard, R., Mermoz, E., and Bernard, A., “A framework for automatic architectural synthesis in conceptual design phase,” *Journal of Engineering Design*, Vol. 29, No. 11, 2018, pp. 665–689. doi:10.1080/09544828.2018.1532494.
- [126] Münzer, C., Shea, K., and Helms, B., “Automated parametric design synthesis using graph grammars and constraint solving,” *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 45066, American Society of Mechanical Engineers, 2012, pp. 517–528. doi:10.1115/DETC2012-70313.
- [127] Shi, M., Gladin, J., and Mavris, D. N., “A systematic methodology for populating the aircraft thermal management system architecture space,” *AIAA Scitech 2021 Forum*, 2021, p. 1295. doi:10.2514/6.2021-1295.
- [128] Bussemaker, J., Sánchez, R. G., Fouda, M., Boggero, L., and Nagel, B., “Function-Based Architecture Optimization: An Application to Hybrid-Electric Propulsion Systems,” *INCOS international symposium*, Vol. 33, Wiley Online Library, 2023, pp. 251–272. doi:10.1002/iis2.13020.

- [129] Swantner, A., and Campbell, M. I., “Topological and parametric optimization of gear trains,” *Engineering Optimization*, Vol. 44, No. 11, 2012, pp. 1351–1368. doi:10.1080/0305215X.2011.646264.
- [130] Bayrak, A. E., Kang, N., and Papalambros, P. Y., “Decomposition-based design optimization of hybrid electric powertrain architectures: Simultaneous configuration and sizing design,” *Journal of Mechanical Design*, Vol. 138, No. 7, 2016, p. 071405. doi:10.1115/1.4033655.
- [131] Bussemaker, J. H., Ciampa, P. D., and Nagel, B., “System architecture design space exploration: An approach to modeling and optimization,” *AIAA Aviation 2020 Forum*, 2020, p. 3172. doi:10.2514/6.2020-3172.
- [132] Bussemaker, J. H., Boggero, L., and Nagel, B., “System architecture design space exploration: Integration with computational environments and efficient optimization,” *AIAA AVIATION FORUM AND ASCEND 2024*, 2024, p. 4647. doi:10.2514/6.2024-4647.
- [133] Bussemaker, J. H., Bartoli, N., Lefebvre, T., Ciampa, P. D., and Nagel, B., “Effectiveness of surrogate-based optimization algorithms for system architecture optimization,” *AIAA Aviation 2021 forum*, 2021, p. 3095. doi:10.2514/6.2021-3095.
- [134] Bussemaker, J. H., Saves, P., Bartoli, N., Lefebvre, T., and Nagel, B., “Surrogate-based optimization of system architectures subject to hidden constraints,” *AIAA AVIATION FORUM AND ASCEND 2024*, 2024, p. 4401. doi:10.2514/6.2024-4401.
- [135] Bussemaker, J. H., De Smedt, T., La Rocca, G., Ciampa, P. D., and Nagel, B., “System architecture optimization: An open source multidisciplinary aircraft jet engine architecting problem,” *AIAA Aviation 2021 Forum*, 2021, p. 3078. doi:10.2514/6.2021-3078.
- [136] Bussemaker, J. H., Saves, P., Bartoli, N., Lefebvre, T., and Lafage, R., “System architecture optimization strategies: dealing with expensive hierarchical problems,” *Journal of Global Optimization*, Vol. 91, No. 4, 2025, pp. 851–895. doi:10.1007/s10898-024-01443-8.
- [137] Garg, S., Garcia Sanchez, R., Bussemaker, J. H., Boggero, L., and Nagel, B., “Dynamic formulation and execution of mdao workflows for architecture optimization,” *AIAA AVIATION FORUM AND ASCEND 2024*, 2024, p. 4402. doi:10.2514/6.2024-4402.
- [138] Silvas, E., Hofman, T., Serebrenik, A., and Steinbuch, M., “Functional and cost-based automatic generator for hybrid vehicles topologies,” *IEEE/ASME Transactions on Mechatronics*, Vol. 20, No. 4, 2015, pp. 1561–1572. doi:10.1109/TMECH.2015.2405473.
- [139] Münzer, C., Helms, B., and Shea, K., “Automatically transforming object-oriented graph-based representations into Boolean satisfiability problems for computational

- design synthesis,” *Journal of Mechanical Design*, Vol. 135, No. 10, 2013, p. 101001. doi:10.1115/1.4024850.
- [140] Arora, J. S., *Introduction to Optimum Design*, 4th ed., Academic Press, 2016.
 - [141] Mane, M., Crossley, W. A., and Nusawardhana, “System-of-systems inspired aircraft sizing and airline resource allocation via decomposition,” *Journal of aircraft*, Vol. 44, No. 4, 2007, pp. 1222–1235. doi:10.2514/1.26333.
 - [142] Roy, S., Moore, K., Hwang, J. T., Gray, J. S., Crossley, W. A., and Martins, J. R., “A mixed integer efficient global optimization algorithm for the simultaneous aircraft allocation-mission-design problem,” *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2017, p. 1305. doi:10.2514/6.2017-1305.
 - [143] Davendralingam, N., and DeLaurentis, D. A., “A robust portfolio optimization approach to system of system architectures,” *Systems Engineering*, Vol. 18, No. 3, 2015, pp. 269–283. doi:10.1002/sys.21302.
 - [144] Ozkan, O., Ermis, M., and Bekmezci, I., “Reliable communication network design: The hybridisation of metaheuristics with the branch and bound method,” *Journal of the Operational Research Society*, Vol. 71, No. 5, 2020, pp. 784–799. doi:10.1080/01605682.2019.1582587.
 - [145] Beale, E. M., “Branch and bound methods for mathematical programming systems,” *Annals of Discrete Mathematics*, Vol. 5, Elsevier, 1979, pp. 201–219. doi:10.1016/S0167-5060(08)70351-0.
 - [146] Lawler, E. L., and Wood, D. E., “Branch-and-bound methods: A survey,” *Operations research*, Vol. 14, No. 4, 1966, pp. 699–719. doi:10.1287/opre.14.4.699.
 - [147] Leon, F., and Caşcaval, P., “Optimization Method for Reliability–Redundancy Allocation Problem in Large Hybrid Binary Systems,” *Mathematics*, Vol. 13, No. 15, 2025, p. 2450. doi:10.3390/math13152450.
 - [148] Bauer, C., Lagadec, K., Bès, C., and Mongeau, M., “Flight control system architecture optimization for fly-by-wire airliners,” *Journal of guidance, control, and dynamics*, Vol. 30, No. 4, 2007, pp. 1023–1029. doi:10.2514/1.26311.
 - [149] Bussemaker, J., Boggero, L., and Nagel, B., “The AGILE 4.0 Project: MBSE to Support Cyber-Physical Collaborative Aircraft Development,” *INCOSE International Symposium*, Vol. 33, Wiley Online Library, 2023, pp. 163–182. doi:10.1002/iis2.13015.
 - [150] Delligatti, L., *SysML distilled: A brief guide to the systems modeling language*, Addison-Wesley, 2013.

- [151] Habermehl, C., Höpfner, G., Berroth, J., Neumann, S., and Jacobs, G., “Optimization workflows for linking model-based systems engineering (MBSE) and multidisciplinary analysis and optimization (MDAO),” *Applied Sciences*, Vol. 12, No. 11, 2022, p. 5316. doi:10.3390/app12115316.
- [152] Jeyaraj, A. K., Tabesh, N., and Liscouet-Hanke, S., “Connecting Model-based Systems Engineering and Multidisciplinary Design Analysis and Optimization for Aircraft Systems Architecting,” *AIAA AVIATION 2021 FORUM*, 2021, p. 3077. doi:10.2514/6.2021-3077.
- [153] van Gent, I., and La Rocca, G., “Formulation and integration of MDAO systems for collaborative design: A graph-based methodological approach,” *Aerospace Science and Technology*, Vol. 90, 2019, pp. 410–433. doi:10.1016/j.ast.2019.04.039.
- [154] Ciampa, P. D., Prakasha, P. S., Torrigiani, F., Walther, J.-N., Lefebvre, T., Bartoli, N., Timmermans, H., Della Vecchia, P., Stingo, L., Rajpal, D., et al., “Streamlining cross-organizational aircraft development: results from the AGILE project,” *AIAA Aviation 2019 Forum*, 2019, p. 3454. doi:10.2514/6.2019-3454.
- [155] Ciampa, P. D., and Nagel, B., “Accelerating the Development of Complex Systems in Aeronautics via MBSE and MDAO: a Roadmap to Agility,” *AIAA Aviation 2021 Forum*, 2021, p. 3056. doi:10.2514/6.2021-3056.
- [156] Lupp, C. A., Kao, J. Y., Novotny, N., Wontor, T., Xu, A., Singleton, J., and Sandler, D. A., “A Theoretical Foundation and Practical Demonstration of Integrating MDO, MBSE and the Digital Thread,” *AIAA SciTech Forum*, 2026, p. 1608. doi:10.2514/6.2026-1608.
- [157] Kolonay, R. M., “MDAO - An Air Force Perspective,” , 2021. URL https://www.phoenix-int.com/wp-content/uploads/2021/01/Kolonay_PIMDAOEvent2021Rev3.pdf.
- [158] Lupp, C. A., and Xu, A., “Creating a Universal Communication Standard to Enable Heterogeneous Multidisciplinary Design Optimization,” *AIAA SCITECH Forum*, 2024, p. 1799. doi:10.2514/6.2024-1799.
- [159] Alder, M., Moerland, E., Jepsen, J., and Nagel, B., “Recent advances in establishing a common language for aircraft design with CPACS,” , 2020. URL https://www.researchgate.net/profile/Marko-Alder-2/publication/344346475_Recent_Advances_in_Establishing_a_Common_Language_for_Aircraft_Design_with_CPACS/links/678184023e33dd0be9f7264c/Recent-Advances-in-Establishing-a-Common-Language-for-Aircraft-Design-with-CPACS.pdf.
- [160] van Gent, I., La Rocca, G., and Hoogreef, M. F., “CMDOWS: a proposed new standard to store and exchange MDO systems,” *CEAS Aeronautical Journal*, Vol. 9, No. 4, 2018, pp. 607–627. doi:10.1007/s13272-018-0307-2.

- [161] Ciampa, P. D., La Rocca, G., and Nagel, B., “A MBSE approach to MDAO systems for the development of complex products,” *AIAA Aviation 2020 Forum*, 2020, p. 3150. doi:10.2514/6.2020-3150.
- [162] Aiello, O., Kandel, D. S. D. R., Chaudemar, J.-C., Poitou, O., and de Saqui-Sannes, P., “Populating MBSE models from MDAO analysis,” *2021 IEEE International Symposium on Systems Engineering (ISSE)*, IEEE, 2021, pp. 1–8. doi:10.1109/ISSE51541.2021.9582519.
- [163] Hossain, N. U. I., Lutfi, M., Ahmed, I., Akundi, A., and Cobb, D., “Modeling and analysis of unmanned aerial vehicle system leveraging systems modeling language (SysML),” *Systems*, Vol. 10, No. 6, 2022, p. 264. doi:10.3390/systems10060264.
- [164] Aiello, O., Poitou, O., Chaudemar, J.-C., and De Saqui-Sannes, P., “Sizing a Drone Battery by coupling MBSE and MDAO,” *11th European Congress Embedded Real Time Systems (ERTS) 2022*, 2022, p. 03740634.
- [165] Bussemaker, J., Boggero, L., and Ciampa, P. D., “From system architecting to system design and optimization: A link between MBSE and MDAO,” *INCOSE International Symposium*, Vol. 32, Wiley Online Library, 2022, pp. 343–359. doi:10.1002/iis2.12935.
- [166] Bruggeman, A.-L. M., and La Rocca, G., “From requirements to product: An MBSE approach for the digitalization of the aircraft design process,” *INCOSE International Symposium*, Vol. 33, Wiley Online Library, 2023, pp. 1688–1706. doi:10.1002/iis2.13107.
- [167] Chaudemar, J.-C., and de Saqui-Sannes, P., “MBSE and MDAO for early validation of design decisions: a bibliography survey,” *2021 IEEE International Systems Conference (SysCon)*, IEEE, 2021, pp. 1–8. doi:10.1109/SysCon48628.2021.9447140.
- [168] Swaminathan, R., Sarojini, D., and Hwang, J. T., “Integrating mbse and mdo through an extended requirements-functional-logical-physical (rflp) framework,” *AIAA AVIATION 2023 Forum*, 2023, p. 3908. doi:10.2514/6.2023-3908.
- [169] Institut Mines-Télécom, “TTool,” , 2016. URL <https://gitlab.telecom-paris.fr/mbe-tools/TTool/>.
- [170] Bruggeman, A.-L., van Manen, B., van der Laan, T., van den Berg, T., and La Rocca, G., “An MBSE-based requirement verification framework to support the MDAO process,” *AIAA Aviation 2022 Forum*, 2022, p. 3722. doi:10.2514/6.2022-3722.
- [171] Campbell, M., Cagan, J., and Kotovsky, K., “Agent-based synthesis of electro-mechanical design configurations,” *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 80333, American Society of Mechanical Engineers, 1998, p. V003T03A003. doi:10.1115/DETC98/DTM-5673.

- [172] Renkert, P. M., and Alleyne, A. G., “Component-based design optimization of multirotor aircraft,” *2022 American Control Conference (ACC)*, IEEE, 2022, pp. 3985–3990. doi:10.23919/ACC53348.2022.9867163.
- [173] Sirico, A., and Herber, D. R., “Geometric deep learning towards the iterative classification of graph-based aircraft thermal management systems,” *AIAA SCITECH 2024 Forum*, 2024, p. 0684. doi:10.2514/6.2024-0684.
- [174] Mokotoff, P. R., and Cinar, G., “Reliability and Performance Trade Studies for NASA’s Subsonic Single Aft Engine Aircraft,” , 2025. doi:10.7302/wg2f-nv21, Data Set.
- [175] Anderson, J. D., *Aircraft performance and design*, McGraw-Hill New York, 1999.
- [176] Wallis, W. D., *A beginner’s guide to graph theory*, Springer, 2007.
- [177] Mokotoff, P. R., Arnson, M., Wang, Y.-C., and Cinar, G., “FAST: A Future Aircraft Sizing Tool for Conventional and Electrified Aircraft Design,” *Journal of Aircraft*, 2025, pp. 1–18. doi:10.2514/1.C038452.
- [178] IDEAS Laboratory, “FAST Aerobase,” , 2024. URL <https://github.com/ideas-um/FAST/tree/main/%2BDatabasePkg>.
- [179] Arnson, M., Aljaber, R., and Cinar, G., “Predicting Aircraft Design Parameters using Gaussian Process Regressions on Historical Data,” *AIAA SciTech 2025 Forum*, 2025, p. 1287. doi:10.2514/6.2025-1287.
- [180] Khailany, N., Mokotoff, P. R., and Cinar, G., “Aircraft Geometry and Propulsion Architecture Visualization for the Future Aircraft Sizing Tool (FAST),” *AIAA SciTech Forum*, 2025, p. 1288. doi:10.2514/6.2025-1288.
- [181] De Vries, R., Hoogreef, M. F., and Vos, R., “Range equation for hybrid-electric aircraft with constant power split,” *Journal of Aircraft*, Vol. 57, No. 3, 2020, pp. 552–557. doi:10.2514/1.C035734.
- [182] Duffy, K. P., and Jansen, R. H., “Turboelectric and hybrid electric aircraft drive key performance parameters,” *2018 AIAA/IEEE Electric Aircraft Technologies Symposium (EATS)*, IEEE, 2018, pp. 1–19. doi:10.2514/6.2018-5023.
- [183] Raymer, D. P., *Aircraft Design: A Conceptual Approach*, 1st ed., American Institute of Aeronautics and Astronautics, 1989.
- [184] Tremblay, O., and Dessaint, L.-A., “Experimental Validation of a Battery Dynamic Model for EV Applications,” *World Electric Vehicle Journal*, Vol. 3, No. 2, 2009, pp. 289–298. doi:10.3390/wevj3020289.
- [185] Federal Aviation Administration, “FAA Type Certificate Data Sheet No. A56NM,” , 2022. URL <https://drs.faa.gov/browse/excelExternalWindow/22037ED8FF40F56A862587D4004E3443.0001>.

- [186] EMBRAER S.A., “Airport planning manual,” https://www.embraercommercialaviation.com/wp-content/uploads/2017/02/APM_E175.pdf, 2005.
- [187] Wang, Y.-C., Stockhausen, M., Mokotoff, P. R., and Cinar, G., “SUBsonic Single Aft eNginE (SUSAN) System Integration Analysis with the Future Aircraft Sizing Tool (FAST),” *AIAA SciTech 2025 Forum*, 2025, p. 2376. doi:10.2514/6.2025-2376.
- [188] McDonald, R. A., “Establishing mission requirements based on consideration of aircraft operations,” *Journal of aircraft*, Vol. 50, No. 3, 2013, pp. 741–751. doi:10.2514/1.C031878.
- [189] National Archives and Records Administration, “What is a special condition? (14 CFR 11.19),” , 2026. URL <https://www.ecfr.gov/current/title-14/chapter-I/subchapter-B/part-11/subpart-A/subject-group-ECFR8ebf6bddc82be9b/section-11.19>.
- [190] Acar, H., Arnson, M., Tsai, M., and Cinar, G., “Historical Trends and Future Projections of Key Performance Parameters in Aircraft Design,” *Journal of Aircraft*, 2025, pp. 1–23. doi:10.2514/1.C038340.
- [191] Mokotoff, P. R., and Cinar, G., “A Graph-based Framework for Advanced Aircraft Propulsion System Analysis,” *Aerospace Science and Technology*, 2025, p. 110798. doi:10.1016/j.ast.2025.110798.
- [192] Airbus, “AutoCAD 3-View Aircraft Drawings,” , 2026. URL <https://www.aircraft.airbus.com/en/customer-care/fleet-wide-care/airport-operations-and-aircraft-characteristics/autocad-3-view-aircraft-drawings>.
- [193] OpenVSP Airshow, “Boeing 777-200ER,” , 2025. URL <https://airshow.openvsp.org/vsp/AVcBIGvTSkWi8b3KI5X4>.
- [194] Torenbeek, E., *Advanced aircraft design: conceptual design, analysis and optimization of subsonic civil airplanes*, John Wiley & Sons, 2013.
- [195] Guilmeau, T., Chouzenoux, E., and Elvira, V., “Simulated annealing: A review and a new scheme,” *2021 IEEE statistical signal processing workshop (SSP)*, IEEE, 2021, pp. 101–105. doi:10.1109/SSP49050.2021.9513782.
- [196] Society of Automotive Engineers, “Guidelines for Conducting the Safety Assessment Process Civil Aircraft, Systems, and Equipment,” Tech. rep., Society of Automotive Engineers, 2023. ARP4761A.
- [197] Limnios, N., *Fault Trees*, John Wiley & Sons, 2013.
- [198] Federal Aviation Administration, “System Design and Analysis (AC 25.1309-1B),” , 2024. URL https://www.faa.gov/regulations_policies/advisory_circulars/index.cfm/go/document.information/documentID/1043037.

- [199] Vesely, W. E., Goldberg, F. F., Roberts, N. H., and Haasl, D. F., “Fault Tree Handbook,” Tech. rep., U.S. Nuclear Regulatory Commission, 1981. NUREG-0492.
- [200] Chapman, J. W., Kratz, J. L., Dever, T., Mirhashemi, A., Heersema, N., and Jansen, R., “SUSAN Concept Vehicle Power and Propulsion System Study,” *AIAA SciTech 2023 Forum*, 2023, p. 1749. doi:10.2514/6.2023-1749.
- [201] Darmstadt, P. R., Catanese, R., Beiderman, A., Dones, F., Chen, E., Mistry, M. P., Babie, B., Beckman, M., and Preator, R., “Hazards Analysis and Failure Modes and Effects Criticality Analysis (FMECA) of Four Concept Vehicle Propulsion Systems,” Tech. rep., National Aeronautics and Space Administration, 2019. URL <https://ntrs.nasa.gov/citations/20190026443>.
- [202] National Archives and Records Administration, “Definitions (14 CFR 25.4),” , 2025. URL <https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-25/subpart-A/section-25.4>.
- [203] National Archives and Records Administration, “Equipment, Systems, and Installations (14 CFR 25.1309),” , 2025. URL <https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-25/subpart-F/subject-group-ECFR9f24bf451b0d2b1/section-25.1309>.
- [204] National Archives and Records Administration, “Safety Analysis (14 CFR 33.75),” , 2026. URL <https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-33/subpart-E/section-33.75>.
- [205] National Archives and Records Administration, “Equipment, Systems, and Installations (14 CFR 23.2510),” , 2026. URL <https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-23/subpart-F/section-23.2510>.
- [206] National Archives and Records Administration, “Airworthiness Certificates (14 CFR 21, Subpart H),” , 2026. URL <https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-21/subpart-H>.
- [207] The MathWorks, Inc., “Nonlinear Constraint Solver Algorithms for Genetic Algorithm,” , 2026. URL <https://www.mathworks.com/help/gads/description-of-the-nonlinear-constraint-solver.html>.
- [208] NASA Glenn Researcher Center, “Subsonic Single Aft Engine (SUSAN) Aircraft,” , 2023. URL <https://www1.grc.nasa.gov/aeronautics/eap/airplane-concepts/susan/>.
- [209] Obert, E., *Aerodynamic Design of Transport Aircraft*, IOS press, 2009.