A HIERARCHICAL CONTROL STRATEGY
FOR AIRCRAFT THERMAL SYSTEMS

BY
MATTHEW A. WILLIAMS

THESIS
Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Mechanical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2014

Urbana, Illinois

Adviser:
Professor Andrew Alleyne
Abstract

Aircraft systems present an interesting control problem given the fact that there exist multiple objectives, actuators, and physical dependencies throughout the system. For decades the onboard systems of aircraft have worked in isolation seeking to achieve their goals while giving little to no consideration to actions or constraints of other systems. As aircraft become more electric, aircraft systems can no longer work in isolation without regard to the constraints and limitations of other systems. Current generation aircraft are experiencing complications between electrical and thermal management systems that have emerged as a result of the increasing prevalence of electrical systems onboard aircraft. Due to these trends, new strategies for controlling aircraft systems are required so that decision making and communication can occur between the vehicle, system, subsystem, and component levels of an aircraft. This thesis looks at establishing the groundwork for the development of next-generation control for aircraft systems.

Since testing aircraft systems would be a costly endeavor, models of an aircraft’s electrical, thermal, hydraulic, and pneumatic systems are developed for computer based simulation. A five-level hierarchical control strategy is proposed that seeks to minimize objectives at the vehicle, system, subsystem, component, and physical level. This thesis looks specifically at the system and subsystem-levels of the hierarchy while focusing on the thermal system of the aircraft.

At the system level a model predictive controller is developed for minimizing total energy consumption while maintaining the temperature of thermal zones within some bounds. Control decisions made by the system-level controller are passed to a subsystem controller that is formulated as a mixed integer quadratic programming problem which seeks to minimize power consumption while meeting the command of the system-level controller. The subsystem-level controller is responsible for determining the optimal operational mode for each thermal subsystem.

An example system consisting of a passenger cabin, fuel tank, vapor compression system, air cycle machine, and ram air cooling loop is used to demonstrate the capabilities of the proposed system and subsystem-level controllers. Controller parameters are analyzed to determine the effect on total power consumption and temperature regulation. Preliminary results show that the two-level hierarchical controller is capable of maintaining temperatures within constraints while minimizing total power consumption.
To my parents,
who gave me the opportunity to pursue my dreams.
Acknowledgements

First, I would like to thank my advisor, Dr. Andrew Alleyne, for his continual commitment to my education, research, and helping me become a better researcher. Over my previous two years at the University of Illinois at Urbana-Champaign, he has been an excellent mentor and an outstanding example of what it means to be a professor, researcher, and leader. I look forward to working on my Ph.D. under his guidance and support.

Without my parents I would not be where I am today. As a child they helped facilitate my learning, provided me ample opportunities to grow in knowledge and character, and instilled in me good values and morals. They dedicated a large part of their life to raising me, and for that I am forever grateful.

I consider myself very lucky to be a part of the Alleyne Research Group and have the opportunity to work alongside such extraordinary colleagues. I would like to especially acknowledge Justin, Megan, Tim, Katie, Joey, Lindsey, Vikas, Neera, Herschel, Nanjun, Erick, Bin, Yangmin, Sarah, Ugo, Kasper, and Ehsan, who I have had the privilege of forming friendships and working with over the last two years. I hope that we can continue to keep in touch as we go our separate directions.

Finally, I would like to acknowledge the financial support provided to me by Rolls-Royce plc, the National Science Foundation, and the Graduate Research Fellowship Program. This support has given me the flexibility and assistance needed to achieve my goals.
# Table of Contents

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>Page #</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF TABLES</td>
<td>VII</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>VIII</td>
</tr>
<tr>
<td>NOMENCLATURE</td>
<td>XIII</td>
</tr>
</tbody>
</table>

## CHAPTER 1 INTRODUCTION

1.1. Motivation ........................................... 2
1.2. Research Objectives ................................ 4
  1.2.1. System Models .................................. 4
  1.2.2. Large-Scale Simulation Methodology ............... 5
  1.2.3. System and Subsystem-Level Control Design .......... 5
1.3. Organization of Thesis ................................... 5

## CHAPTER 2 AIRCRAFT ENERGY AND POWER SYSTEMS MODELING

2.1. Fuel and Oil Thermal Management System ................. 7
  2.1.1. Fuel and Oil Tank ................................ 8
  2.1.2. Fuel and Oil Pump ................................ 9
  2.1.3. Stochastic Heat Loads ............................. 12
2.2. Environmental Control System ........................... 13
  2.2.1. Thermal Zone/Cabin ................................ 13
  2.2.2. Heat Exchanger ................................... 16
  2.2.3. Mixing Junction ................................... 21
  2.2.4. Compressor and Turbine ........................... 22

## CHAPTER 3 VARIABLE FIDELITY MODELING

3.1. Relationship between Accuracy, Speed, and Fidelity .......... 26
  3.1.1. Effect of Model Fidelity on Simulation Accuracy ........ 28
  3.1.2. Effects of Model Fidelity on Computational Speed ........ 30
3.2. Modeling Framework ................................... 31
3.3. Supervisor Design ...................................... 32
  3.3.1. Filter ........................................... 32
  3.3.2. K\text{filter} and Discrete Logic ...................... 33
  3.3.3. Coordination of Fidelity Mode and t\text{dwell} ........... 35
3.4. Effect of K\text{filter} and t\text{dwell} on Computational Cost and Accuracy ....... 36
  3.4.1. Influence of K\text{filter} and t\text{dwell} on Accumulated Error ......... 37
  3.4.2. Influence of K\text{filter} and t\text{dwell} on Computational Cost ........ 39
  3.4.3. Optimal Trade-off between Computational Cost and Accuracy ....... 39
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5. Variable Fidelity of Closed Loop Dynamical Systems</td>
<td>42</td>
</tr>
<tr>
<td>3.5.1. System Description</td>
<td>42</td>
</tr>
<tr>
<td>3.5.2. Supervisor Setup for Variable Fidelity System Modeling</td>
<td>43</td>
</tr>
<tr>
<td>3.5.3. Simulation Case Study for Variable Fidelity System Modeling</td>
<td>46</td>
</tr>
<tr>
<td>CHAPTER 4 MODEL PREDICTIVE CONTROL</td>
<td>52</td>
</tr>
<tr>
<td>4.1. Overview</td>
<td>52</td>
</tr>
<tr>
<td>4.2. Basic Formulation</td>
<td>53</td>
</tr>
<tr>
<td>4.3. MPC Algorithm Summary</td>
<td>56</td>
</tr>
<tr>
<td>CHAPTER 5 HIERARCHICAL ENERGY AND POWER MANAGEMENT</td>
<td>57</td>
</tr>
<tr>
<td>5.1. Proposed Two-Level Hierarchical Control Strategy</td>
<td>59</td>
</tr>
<tr>
<td>5.1.1. System-level Control Using Model Predictive Control</td>
<td>60</td>
</tr>
<tr>
<td>5.1.2. Subsystem Operating Mode Selection Using Mixed Integer Quadratic Programming</td>
<td>62</td>
</tr>
<tr>
<td>CHAPTER 6 CASE STUDY USING AN AIRCRAFT THERMAL SYSTEM</td>
<td>65</td>
</tr>
<tr>
<td>6.1. Example System Configuration</td>
<td>65</td>
</tr>
<tr>
<td>6.2. Mission Profile and System Parameters</td>
<td>66</td>
</tr>
<tr>
<td>6.2.1. Dynamic Model of Thermal System</td>
<td>70</td>
</tr>
<tr>
<td>6.3. System-level MPC Formulation</td>
<td>71</td>
</tr>
<tr>
<td>6.3.1. Tunable Parameters</td>
<td>74</td>
</tr>
<tr>
<td>6.4. Subsystem-Level MIQP Formulation</td>
<td>74</td>
</tr>
<tr>
<td>6.4.1. Tunable Parameters</td>
<td>76</td>
</tr>
<tr>
<td>6.5. Simulation Results</td>
<td>76</td>
</tr>
<tr>
<td>6.5.1. Baseline Simulation</td>
<td>78</td>
</tr>
<tr>
<td>6.5.2. Effect of Load Preview</td>
<td>81</td>
</tr>
<tr>
<td>6.5.3. Effect of Regulation</td>
<td>84</td>
</tr>
<tr>
<td>6.5.4. Effect of Switching Penalty</td>
<td>87</td>
</tr>
<tr>
<td>6.5.5. Total Power Consumed Comparison</td>
<td>90</td>
</tr>
<tr>
<td>6.6. Case Study Remarks</td>
<td>91</td>
</tr>
<tr>
<td>CHAPTER 7 CONCLUSION</td>
<td>92</td>
</tr>
<tr>
<td>7.1. Summary of Research Contributions</td>
<td>92</td>
</tr>
<tr>
<td>7.2. Future Work</td>
<td>93</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>95</td>
</tr>
<tr>
<td>APPENDIX A</td>
<td>98</td>
</tr>
<tr>
<td>APPENDIX B</td>
<td>118</td>
</tr>
<tr>
<td>APPENDIX C</td>
<td>123</td>
</tr>
</tbody>
</table>
List of Tables

Table 2.1. Energy Domain Color Coding ................................................................. 7
Table 3.1 Comparison of Root-Mean-Square Error (RMSE) of Low and Switched Fidelity
Models Relative to High-fidelity .............................................................................. 48
Table 5.1. Proposed 5-level hierarchical control strategy for aircraft ....................... 59
Table 6.1. Constraints on the thermal system during sample mission ....................... 68
Table 6.2. Performance parameters for the ACM and VCS during sample mission .......... 70
Table 6.3. Subsystem operational modes and color designations ................................ 77
Table 6.4. Baseline case study optimization parameters ............................................. 78
Table 6.5. Case study optimization parameters with no preview of disturbances ........... 81
Table 6.6. Case study optimization parameters with no regulation of temperatures .......... 84
Table 6.7. Case study optimization parameters with high penalty on switching ............ 87
Table 6.8. Total work in the system over the course of the mission using each controller .... 91
List of Figures

Figure 1.1. Evolution of electrical power requirements in aircraft [1][2][3][4] ........................................3
Figure 1.2. Power and thermal requirements for military aircraft [6] ........................................................4
Figure 2.1. Fluid Tank Input and Output Energy Domains .................................................................9
Figure 2.2. Pump Performance Curve .........................................................................................10
Figure 2.3. Pump Efficiency Curve ..........................................................................................11
Figure 2.4. Fuel Pump Input and Output Energy Domains ..........................................................12
Figure 2.5. Thermal Zone Input and Output Energy Domains ....................................................16
Figure 2.6 One Layer of an Offset Strip-Fin Heat Exchanger [11] ............................................16
Figure 2.7 Geometric Parameters of the Offset Strip-Fin Heat Exchanger [11] ....................17
Figure 2.8. Fin Thermal Resistance [11] .................................................................................19
Figure 2.9. Core Heat Exchange Model ..................................................................................20
Figure 2.10. Heat Exchanger Input and Output Energy Domains .............................................21
Figure 2.11. Turbine Input and Output Energy Domains ..........................................................24
Figure 2.12. Compressor Input and Output Energy Domains ...................................................24
Figure 3.1. Trade-off between computational speed and accuracy for high (■), low (●), and
variable (★) fidelity models ........................................................................................................26
Figure 3.2. Dynamic space of high and low-fidelity models and a simulation (white line)
that switches at the boundaries ..............................................................................................27
Figure 3.3. Comparison of the steady-state error between high and low-fidelity model
outputs ..................................................................................................................................29
Figure 3.4. Comparison transient response error between high and low-fidelity model
outputs ..................................................................................................................................30
Figure 3.5. Comparison of simulation speed and level of fidelity ..............................................30
Figure 3.6. Signal flow of reference and disturbance signals through the supervisor and time
delay to the simulated system ...............................................................................................31
Figure 3.7. Supervisor signal flow ........................................................................................32
Figure 3.8. Filter behavior with an input of a step (a), the corresponding output (b), input of a ramp (c), and the corresponding output (d) ................................................................. 33
Figure 3.9. Scaling by $K_{filter}$ of the filter output for the step (a) and ramp (c) and the corresponding discrete logic for the step (b) and the ramp (d) ........................................ 34
Figure 3.10. Application of dwell time to level of fidelity switching logic ...................... 35
Figure 3.11. Supervisor output after the application of dwell time .................................. 36
Figure 3.12. Effect of $K_{filter}$ and $t_{dwell}$ on the error accumulation in the primary flow exit pressure ....................................................................................................................... 38
Figure 3.13. Effect of $K_{filter}$ and $t_{dwell}$ on the error accumulation in the secondary flow exit temperature ....................................................................................................................... 38
Figure 3.14. Effect of $K_{filter}$ and $t_{dwell}$ on computational cost ..................................... 39
Figure 3.15. Cost associated with different values of $K_{filter}$ and $t_{dwell}$ ..................... 40
Figure 3.16. Computational speed for low, high, and switched-fidelity models .............. 41
Figure 3.17. Accumulated pressure and temperature errors between the low and switched-fidelity models .......................................................................................................................... 41
Figure 3.18. Four component vapor compression system configuration ...................... 43
Figure 3.19. Configuration of Reference and Disturbance Signal Propagation to Supervisor and System ......................................................................................................................... 44
Figure 3.20. Switched fidelity logic tree ........................................................................... 45
Figure 3.21. Four component vapor compression system configuration ...................... 46
Figure 3.22. Reference temperature (top) and disturbance heat load (bottom) profiles for simulation case study ............................................................................................................. 47
Figure 3.23. Normalized error between high and low-fidelity system inputs $u_1$ and $u_2$ .... 49
Figure 3.24. Normalized system inputs $u_1$ and $u_2$ time history for simulation case study .... 50
Figure 3.25. Tracking of system outputs $y_1$ and $y_2$ for simulation case study ............. 50
Figure 3.26. Comparison of time rations for low, high, and switched-fidelity models .......... 51
Figure 4.1. A discrete model predictive control scheme .................................................. 53
Figure 5.1. Physical hierarchy of aircraft systems ............................................................ 57
Figure 5.2. Tradeoff between cooling capacity and power consumption for thermal management unit

Figure 6.1. Example system layout containing three loads, two thermal zones, two thermal management units, and two thermal sinks

Figure 6.2. Mission profile for simulation case study

Figure 6.3. Profile of heat loads into the cabin and fuel tank for the simulation case study

Figure 6.4. Maximum $\dot{Q}$ capable of being dumped via ambient air, fuel tank, and ram air

Figure 6.5. Nominal temperature and upper/lower constraints of the cabin for mission phase

Figure 6.6. Tradeoff between cooling capacity and power consumption for thermal management unit

Figure 6.7. Performance curves for the VCS and ACM

Figure 6.8. Performance of the baseline controller

Figure 6.9. Cabin and fuel temperature for the baseline controller

Figure 6.10. Operational mode throughout the mission for the baseline controller

Figure 6.11. Performance of the controllers with no system-level preview of disturbances

Figure 6.12. Cabin and fuel temperature with no preview of disturbances

Figure 6.13. Operational mode throughout the mission with no preview of disturbances

Figure 6.14. Performance of the controllers with no regulation on fuel tank or cabin temperature

Figure 6.15. Cabin and fuel temperature with no regulation

Figure 6.16. Operational mode throughout the mission when not regulating temperatures

Figure 6.17. Performance of the controllers with higher penalty on switching frequency

Figure 6.18. Cabin and fuel temperature with high penalty on switching frequency

Figure 6.19. Operational mode throughout the mission with high penalty on switching frequency

Figure 6.20. Total power consumption over the course of the mission for each controller

Figure A.1. Structural diagram of an exciter system

Figure A.2. GUI for synchronous generation unit with exciter

Figure A.3. Battery constants for multiple time scales

Figure A.4. Battery constants for multiple time scales
Figure A.5. Battery GUI ................................................................. 104
Figure A.6. Inverter inputs and outputs ........................................... 105
Figure A.7. Inverter GUI ............................................................... 106
Figure A.8. AC loads GUI ............................................................. 107
Figure A.9. Simplified circuit diagram of the modeled transformer ........... 108
Figure A.10. Transformer inputs and outputs ..................................... 108
Figure A.11. Transformer GUI ....................................................... 109
Figure A.12. Engine driven pump inputs and outputs ............................ 111
Figure A.13. EDP general GUI ....................................................... 111
Figure A.14. EDP Hagen-Poiseuille GUI ......................................... 111
Figure A.15. Fluid reservoir general GUI ......................................... 113
Figure A.16. Fluid reservoir initial conditions GUI ............................... 113
Figure A.17. Fuel tank geometry GUI .............................................. 114
Figure A.18. Fuel tank initial conditions GUI ..................................... 114
Figure A.19. Fuel tank thermal GUI ............................................... 114
Figure A.20. Fuel pump general GUI .............................................. 114
Figure A.21. Fuel pump power GUI ............................................... 115
Figure A.22. Heat loads GUI .......................................................... 115
Figure A.23. Cabin zone parameters GUI .......................................... 115
Figure A.24. Cabin zone wall parameters GUI .................................... 115
Figure A.25. Cabin zone passengers GUI .......................................... 116
Figure A.26. Cabin zone heat transfer GUI ....................................... 116
Figure A.27. Air-air heat exchanger parameters GUI ............................ 116
Figure A.28. Air-air heat exchanger fin parameters GUI ......................... 116
Figure A.29. Mixing junction GUI ................................................... 117
Figure A.30. Turbine GUI .............................................................. 117
Figure B.1. Layout of the variable fidelity VCS model ......................... 118
Figure B.2. Supervisory logic GUI ................................................. 118
Figure B.3. Supervisory logic .......................................................... 119
Figure B.4. Hi/Low switch block diagram ......................................... 119
Figure B.5. EEV setup .................................................................................................................. 120
Figure B.6. Compressor setup....................................................................................................... 120
Figure B.7. Switched evaporator setup ....................................................................................... 121
Figure B.8. Switched condenser setup ....................................................................................... 121
Figure B.9. Controller framework for temperature regulation .................................................. 122
Figure B.10. Controller framework for superheat regulation ..................................................... 122
## Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>Area</td>
<td>m²</td>
</tr>
<tr>
<td>$C$</td>
<td>Thermal capacity</td>
<td>J/K</td>
</tr>
<tr>
<td>$C_p$</td>
<td>Specific heat</td>
<td>J/g°C</td>
</tr>
<tr>
<td>$d$</td>
<td>Disturbance</td>
<td>~</td>
</tr>
<tr>
<td>$D_h$</td>
<td>Hydraulic diameter</td>
<td>m</td>
</tr>
<tr>
<td>$E_{ac}$</td>
<td>Energy accumulation</td>
<td>J</td>
</tr>
<tr>
<td>$f$</td>
<td>Friction factor</td>
<td>~</td>
</tr>
<tr>
<td>$G$</td>
<td>Irradiation</td>
<td>W/m²</td>
</tr>
<tr>
<td>$h$</td>
<td>Heat transfer coefficient</td>
<td>W/(m²·K)</td>
</tr>
<tr>
<td>$h$</td>
<td>Height</td>
<td>m</td>
</tr>
<tr>
<td>$j$</td>
<td>Colburn factor</td>
<td>~</td>
</tr>
<tr>
<td>$J$</td>
<td>Inertia</td>
<td>kg·m²</td>
</tr>
<tr>
<td>$K_n$</td>
<td>Fin thermal conductivity</td>
<td>W/(m·K)</td>
</tr>
<tr>
<td>$K_p$</td>
<td>Plate thermal conductivity</td>
<td>W/(m·K)</td>
</tr>
<tr>
<td>$l$</td>
<td>Length</td>
<td>m</td>
</tr>
<tr>
<td>$L$</td>
<td>Total length</td>
<td>m</td>
</tr>
<tr>
<td>$m$</td>
<td>Mass</td>
<td>kg</td>
</tr>
<tr>
<td>$\dot{m}$</td>
<td>Mass flow rate</td>
<td>kg/s</td>
</tr>
<tr>
<td>$M$</td>
<td>Mach number</td>
<td>~</td>
</tr>
<tr>
<td>$n$</td>
<td>Number/Count</td>
<td>~</td>
</tr>
<tr>
<td>$n_s$</td>
<td>Number of thermal management units</td>
<td>~</td>
</tr>
<tr>
<td>$n_z$</td>
<td>Number of thermal zones</td>
<td>~</td>
</tr>
<tr>
<td>$N$</td>
<td>Rotational speed</td>
<td>RPM</td>
</tr>
<tr>
<td>$N_f$</td>
<td>Number of fins</td>
<td>~</td>
</tr>
<tr>
<td>$N_p$</td>
<td>Prediction horizon</td>
<td>~</td>
</tr>
<tr>
<td>$N_u$</td>
<td>Control horizon</td>
<td>~</td>
</tr>
<tr>
<td>$pf$</td>
<td>Power factor</td>
<td>~</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td>Unit</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>$P$</td>
<td>Pressure</td>
<td>kPa</td>
</tr>
<tr>
<td>$Q$</td>
<td>Reactive power</td>
<td>var</td>
</tr>
<tr>
<td>$\dot{Q}$</td>
<td>Heat transfer per unit time/Cooling capacity</td>
<td>kW</td>
</tr>
<tr>
<td>$r$</td>
<td>Reference</td>
<td>~</td>
</tr>
<tr>
<td>$R$</td>
<td>Gas Constant</td>
<td>J/(kg·K)</td>
</tr>
<tr>
<td>$Re$</td>
<td>Reynolds number</td>
<td>~</td>
</tr>
<tr>
<td>$s$</td>
<td>Spacing</td>
<td>m</td>
</tr>
<tr>
<td>$t$</td>
<td>Thickness</td>
<td>m</td>
</tr>
<tr>
<td>$T$</td>
<td>Temperature</td>
<td>°C</td>
</tr>
<tr>
<td>$\dot{T}$</td>
<td>Temperature rate of change</td>
<td>°C/sec.</td>
</tr>
<tr>
<td>$u_b$</td>
<td>Boolean decision variable</td>
<td>~</td>
</tr>
<tr>
<td>$u_c$</td>
<td>Continuous decision variable</td>
<td>~</td>
</tr>
<tr>
<td>$U$</td>
<td>Overall heat transfer coefficient</td>
<td>W/(m²·K)</td>
</tr>
<tr>
<td>$v$</td>
<td>Noise</td>
<td>~</td>
</tr>
<tr>
<td>$\dot{v}$</td>
<td>Volumetric flow rate</td>
<td>m³/sec.</td>
</tr>
<tr>
<td>$V$</td>
<td>Volume</td>
<td>m³</td>
</tr>
<tr>
<td>$\dot{W}$</td>
<td>Power</td>
<td>W</td>
</tr>
<tr>
<td>$z$</td>
<td>Time delay</td>
<td>sec.</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Absorptivity</td>
<td>~</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>General coefficient</td>
<td>~</td>
</tr>
<tr>
<td>$\beta$</td>
<td>General coefficient</td>
<td>~</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Specific heat ratio</td>
<td>~</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Weighting factor</td>
<td>~</td>
</tr>
<tr>
<td>$\delta$</td>
<td>General coefficient</td>
<td>~</td>
</tr>
<tr>
<td>$\Delta P$</td>
<td>Pressure differential</td>
<td>kPa</td>
</tr>
<tr>
<td>$\varepsilon_f$</td>
<td>Fin efficiency</td>
<td>~</td>
</tr>
<tr>
<td>$\varepsilon$</td>
<td>General coefficient</td>
<td>~</td>
</tr>
<tr>
<td>$\eta$</td>
<td>Efficiency</td>
<td>~</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Weighting factor</td>
<td>~</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Fluid dynamic viscosity</td>
<td>Pa·sec.</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td>Unit</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------</td>
<td>----------</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Density</td>
<td>kg/m$^3$</td>
</tr>
<tr>
<td>$\tau$</td>
<td>Time constant/Sample rate</td>
<td>sec.</td>
</tr>
<tr>
<td>$\tau$</td>
<td>Torque</td>
<td>J</td>
</tr>
<tr>
<td>$\omega$</td>
<td>Angular velocity</td>
<td>rad./sec.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subscripts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abs</td>
<td>Absorbed</td>
</tr>
<tr>
<td>acc</td>
<td>Accumulated</td>
</tr>
<tr>
<td>air</td>
<td>Air</td>
</tr>
<tr>
<td>amb</td>
<td>Ambient</td>
</tr>
<tr>
<td>bot</td>
<td>Bottom of aircraft</td>
</tr>
<tr>
<td>cab</td>
<td>Cabin</td>
</tr>
<tr>
<td>ci</td>
<td>Cold fluid into control volume</td>
</tr>
<tr>
<td>c</td>
<td>Cold, Compressor</td>
</tr>
<tr>
<td>cm</td>
<td>Core mass</td>
</tr>
<tr>
<td>d</td>
<td>Dry</td>
</tr>
<tr>
<td>des</td>
<td>Desired</td>
</tr>
<tr>
<td>elec</td>
<td>Electric</td>
</tr>
<tr>
<td>e</td>
<td>Evaporator</td>
</tr>
<tr>
<td>EA</td>
<td>Exhaust Air</td>
</tr>
<tr>
<td>f</td>
<td>Fluid/Fuel</td>
</tr>
<tr>
<td>fg</td>
<td>Fluid-Ullage boundary</td>
</tr>
<tr>
<td>fus</td>
<td>Fuselage</td>
</tr>
<tr>
<td>g</td>
<td>Ullage</td>
</tr>
<tr>
<td>gen</td>
<td>Generated</td>
</tr>
<tr>
<td>grd</td>
<td>Ground</td>
</tr>
<tr>
<td>h</td>
<td>Hot</td>
</tr>
<tr>
<td>hi</td>
<td>Hot fluid into control volume</td>
</tr>
<tr>
<td>ho</td>
<td>Hot fluid out of control volume</td>
</tr>
<tr>
<td>in</td>
<td>Into the control volume</td>
</tr>
</tbody>
</table>
$junc$ & Junction \\
$kh$ & Kinetic Heating \\
$out$ & Out of the control volume \\
$pax$ & Passengers \\
$p$ & Plate \\
$RA$ & Recirculation Air \\
$rec$ & Recovery \\
$sat$ & Saturation \\
$super$ & Superheat \\
$SA$ & Supply Air \\
$top$ & Top of aircraft \\
$tot$ & Total \\
$t$ & Turbine \\
$w$ & Wet \\

<table>
<thead>
<tr>
<th>Abbreviations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A330</td>
<td>Airbus 330</td>
</tr>
<tr>
<td>A380</td>
<td>Airbus 380</td>
</tr>
<tr>
<td>ACM</td>
<td>Air Cycle Machine</td>
</tr>
<tr>
<td>AFRL</td>
<td>Air Force Research Lab</td>
</tr>
<tr>
<td>ATTMO</td>
<td>AFRL Transient Thermal Modeling and Optimization</td>
</tr>
<tr>
<td>B737</td>
<td>Boeing 737</td>
</tr>
<tr>
<td>B767</td>
<td>Boeing 767</td>
</tr>
<tr>
<td>B777</td>
<td>Boeing 777</td>
</tr>
<tr>
<td>B787</td>
<td>Boeing 787</td>
</tr>
<tr>
<td>CO2</td>
<td>Carbon Dioxide</td>
</tr>
<tr>
<td>CTOL</td>
<td>Conventional Take-Off and Landing</td>
</tr>
<tr>
<td>DC-8</td>
<td>McDonnell Douglas DC-8</td>
</tr>
<tr>
<td>ECS</td>
<td>Environmental Control System</td>
</tr>
<tr>
<td>EEV</td>
<td>Electronic Expansion Valve</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ELLA</td>
<td>Electric Laser on Large Aircraft</td>
</tr>
<tr>
<td>EPS</td>
<td>Electrical Power System</td>
</tr>
<tr>
<td>FCS</td>
<td>Flight Control System</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HX</td>
<td>Heat Exchanger</td>
</tr>
<tr>
<td>LRS</td>
<td>Long Range Surveillance</td>
</tr>
<tr>
<td>MEA</td>
<td>More Electric Aircraft</td>
</tr>
<tr>
<td>MIMO</td>
<td>Multi-Input Multi-Output</td>
</tr>
<tr>
<td>MIQP</td>
<td>Mixed Integer Quadratic Programming</td>
</tr>
<tr>
<td>MPC</td>
<td>Model Predictive Control</td>
</tr>
<tr>
<td>PACK</td>
<td>Pneumatic Air Conditional Kit</td>
</tr>
<tr>
<td>RMSE</td>
<td>Root-Mean-Square Error</td>
</tr>
<tr>
<td>RPM</td>
<td>Revolutions per Minute</td>
</tr>
<tr>
<td>STOVL</td>
<td>Short Take-Off Vertical Landing</td>
</tr>
<tr>
<td>TMS</td>
<td>Thermal Management System</td>
</tr>
<tr>
<td>VCS</td>
<td>Vapor Compression System</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

The work presented herein investigates several topics that are currently relevant to large-scale system design and control. In this context, the term “large-scale” refers to systems that are truly “systems of systems” and encompass multiple time-scales, energy domains, and sizes. Examples of such systems include aircraft, spacecraft, mining vehicles, large farm equipment, and naval vessels. In general these systems are so complex that their components and systems are designed in isolation from other systems and the control strategies implemented are rather conservative due to the large number of system actuators and states, as well as complicated control objectives.

While the past few decades have seen substantial improvement in engineering design due to the increasing levels of computer aided design, because of still existent computational limitations, often total system efficiency is sacrificed for ease of design. Furthermore, since money is the often the ultimate reason for design decisions, solutions that enable a system designer to improve design decisions at little to no additional cost are desirable. Later in this thesis, this topic is addressed through the development of a simulation toolkit designed to help model the large-scale integration of aircraft energy domains down to a component level. Additionally, a simulation method is proposed for switching the fidelity of a model during simulation in order to maintain simulation accuracy while increasing computational speed.

Control of large systems can be a difficult and daunting task. The sheer number of system states, outputs, actuators, and potential control objectives make centralized system-level control quite difficult, and sometimes nearly impossible. While control is a relatively new engineering discipline, it has seen a large increase in research efforts over the last half century resulting in multiple methods for controlling systems of different natures. The field of predictive control is one such area that has seen substantial growth in research and industry acceptance, as the advantage of being able to predict what the system will do provides significant benefits for improving efficiency and performance. The remaining sections of this thesis will look at several
forms of predictive control and how it can be developed and integrated into a large-scale system to improve efficiency and performance.

1.1. Motivation

Less than two decades after the Wright brothers first manned powered flight, aircraft began to see the integration of electrically powered systems. During World War I, aircraft were outfitted with 1.75 horsepower auxiliary engines for the purpose of powering radio transmitters and lights. Since these systems were so small, and often operated in favorable conditions, the thermal aspects were negligible. But as technology advanced, and aircraft began flying higher, faster, and more often, the integration of electrical systems became vital to the operation of the aircraft. World War II brought rapid developments to the field of aircraft propulsion and systems such as radar, navigation, flight control, and communication. Each of these systems required electrical power in order to operate, and produced heat as a by-product of inefficient operation.

The introduction of the jet age after WWII brought along an increase in passenger air travel with airlines desiring advanced technology for passenger comfort and safety. Improved avionics, flight controls, navigation/communication systems were introduced and thermal management systems were developed to pressurize the cabins and remove the heat generated by these new electrical systems.

As the industry moves forward today, there is an emphasis on lighter and more efficient aircraft, which has led towards a rationalization of more electric aircraft (MEA) and the replacement of traditional aircraft systems. This trend is emphasized in Figure 1.1 which displays the total generated power across several aircraft platforms from the past 50 years. While conventional aircraft integrate systems spanning pneumatic, hydraulic, electrical and mechanical power domains, recent efforts attempt to remove pneumatic and hydraulic systems in exchange for lighter weight electrical systems. The Boeing 787 is an excellent example since it completely removed bleed air systems in exchange for electrically powered anti-icing and environmental control systems. It is therefore reasonable to assume that next-generation commercial aircraft will continue this trend, perhaps at a temporary exponential rate as hydraulic systems are also phased out in exchange for electrically actuated control surfaces.
With the development of MEA comes the increased demand for electrical power and the reduction of bleed air being siphoned from the engine. While this reduces aircraft weight, fuel consumption, and CO$_2$ emissions, it also introduces advanced dynamical interactions between the thermal and electrical systems, which Boeing quickly discovered with their lithium-ion batteries on the 787 [5]. This highlights the need for novel types of cooling techniques and systems that will introduce additional degrees of freedom in controlling the interactions between thermal and electrical systems. Therefore, advanced thermal management control techniques that operate the system in an efficient manner are necessary for the advancement of commercial aviation.

Similar to the commercial aviation industry, the military expects to see higher thermal and electrical loading in next generation aircraft as advanced weapon systems and avionics are integrated into new platforms. This trend is apparent in Figure 1.2 which shows past, present, and future military flight platforms and the power/thermal requirements (adapted from [6]). Additionally, with unmanned aircraft seeing increased use, the demand for thermal and electrical systems becomes increasingly difficult to address due to smaller platforms with difficult operating environments.

Unlike commercial aviation, military aircraft have the added concern of thermal signature to address during missions. During certain mission phases, it may not be possible to remove heat to
the ambient because of a desire to reduce thermal signature of the aircraft. In this event, it is necessary for the aircraft to have to ability to store heat or dissipate the heat in a manner that is not traceable via IR signature.

With these increasing power and thermal requirements comes the need for improved heat sink technology and control. In the past and present, the most common heat sink was fuel and ram air, but with increasing loads comes the need for additional heat sinks and “heat movers.” Control algorithms are also needed to determine how to operate the thermal and electrical systems efficiently and within constraints of the vehicle.

![Figure 1.2. Power and thermal requirements for military aircraft](image)

**1.2. Research Objectives**

This thesis is broken into three goals that aim at addressing the problems and concerns discussed in the previous section. These goals are introduced in the following subsections, and will be developed in-depth throughout the thesis.

**1.2.1. System Models**

In order to benefit the development and testing of next-generation aircraft architectures, an analysis toolset that is capable of analyzing an aircraft’s integrated propulsion, power and thermal management systems needs to be developed. This variable-fidelity toolset will be developed in the MATLAB/Simulink [7] modeling and simulation environment and will be...
capable of simulating dynamic operating conditions of electrical, pneumatic, hydraulic, and thermal power systems. Additionally, the toolset will be capable of integrating with control algorithms.

1.2.2. Large-Scale Simulation Methodology

With the development of large scale models, the need for simulation tools that increase computational speed are desired. Methods will be developed that help increase the computational speed of system models while maintaining suitable levels of accuracy in the simulation outputs.

1.2.3. System and Subsystem-Level Control Design

The final objective of this thesis is the preliminary development of a hierarchical control methodology for aircraft thermal systems. This will involve designing control algorithms with the purpose of managing and optimizing system level objectives and constraints, and designing subsystem-level controllers that work in collaboration with the system-level controllers in order to improve the operating efficiency of the entire system while meeting operational requirements.

1.3. Organization of Thesis

The rest of the thesis is organized into six chapters. Chapter two presents the mathematical models used to simulate an aircraft’s thermal system and is supplemented by Appendix A which contains the mathematical models for other system components that are modeled in MATLAB/Simulink. In chapter three a variable fidelity method is presented that allows a simulation to autonomously switch the fidelity of a model during transient events so that computational speed can be increased while accuracy is maintained at suitable levels in the simulation variables. Chapter four contains a high level explanation of model predictive control (MPC) which is used in the following two chapters. The hierarchical control approach for system and subsystem operation is presented in chapter five, while chapter six provides simulation results for a sample system. Conclusions and future work are present in chapter seven.

The appendices of this thesis are meant to help a reader fully understand the work that has been done. In Appendix A, extensive detail is given to the model development of a toolset used
for modeling aircraft thermal, electrical, hydraulic, and pneumatic power systems. Appendix B presents the modeling setup used to determine optimal switching criteria for the variable fidelity models in Chapter 3. Appendix C contains the code used in Chapter 6 for the sample simulations of the presented hierarchical control strategy.
Chapter 2

Aircraft Energy and Power Systems Modeling

Modern aircraft platforms are complex “systems of systems” that encompass multiple energy domains including high-performance mechanical and thermodynamic engines, electrical power systems and components, environmental thermal systems, and fluid power components. Previously, these systems were designed, tested, and optimized in isolation from one another, which provided acceptable operation. As modern and next-generation aircraft are becoming increasingly integrated, the need for co-design of integrated systems arises.

In system design it would be ideal to be able to compare integrated platform designs and then down-select designs prior to constraining the design and hardware configurations. Since aircraft are multi-million and multi-billion [8] dollar development projects, a modeling toolset is necessary to allow multiple designers and engineers from varying technical backgrounds to collaborate on the physical layout, operation, and optimization of an aircraft design.

The following sections highlight the development of a toolset designed to model the multi-physical systems of an aircraft. Dynamic models of the thermal power components are presented, while Appendix A contains detailed explanations of the integration with MATLAB/Simulink and mathematical models for hydraulic and electrical power systems. Throughout this thesis the different energy domains will be represented graphically using the color code in Table 2.1.

<table>
<thead>
<tr>
<th>Thermal</th>
<th>Mechanical</th>
<th>Electrical</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[-----]</td>
<td>[-----]</td>
</tr>
</tbody>
</table>

2.1. Fuel and Oil Thermal Management System

The fuel and oil thermal management system uses tanks, pumps, junctions, and heat exchangers in order to move mass and heat throughout the system and aircraft. In order to accomplish this, energy is supplied as an input to some components in the form of pneumatic
power (bleed air) or electrical power. These interconnections are relayed back to their respective sources in order to account for energy use in the fuel and oil system.

The following subsections detail the mathematical modeling of individual components and their inputs/outputs.

2.1.1. Fuel and Oil Tank

The primary purpose of an aircraft’s fuel tank is to store the required fuel for the mission; however, because of the large thermal capacitance of the fuel, it proves useful as a heat sink for thermal loads onboard the aircraft. Fuel can be routed through heat exchangers in order to absorb heat from other fluids such as hydraulic oil, used as a sink for thermal loads caused by inefficiencies of electrical components, or as a temporary storage for heat until it can be dissipated to the ambient.

It is therefore necessary for the fuel and oil tank model to track the time-varying fuel temperature, ullage (entrapped-air) temperature, and fuel mass. Mathematical modeling of the fuel tank is based upon conservation of energy and mass and heat transfer is captured between the fuel, ullage, tank walls, and internal heat loads.

2.1.1.1. Mathematical Model

Mass of the fluid in the tank is determined using conservation of mass,

\[ m_f = \int (m_{in} - m_{out}) \, dt \]  

(2.1.1)

where \( m \) is the flow rate in and out of the tank, and \( m_f \) is the mass of the fluid in the tank.

The rate of change of the fuel temperature is a function of the heat transfer between the fuel and each wet section of wall, the fuel being added to the tank, heat transfer between the fuel and the ullage, and heat loads. Mathematically this is presented as,

\[ m_f C_{p,f} \dot{T}_f = \sum_{i=1}^{n} h_{w,i} A_{w,i} (T_{wall,i} - T_f) + m_{in} C_{p,in} (T_{in} - T_f) + h_{fg} A_{fg} (T_g - T_f) + Q_{load} m_f C_{p,f} T_f \]  

(2.1.2)

where \( T \) is the temperature, \( C_p \) is the fluid specific heat, \( n \) is the number of walls, \( h_w \) is the heat transfer coefficient for a wet wall, \( A_w \) is the wet area of the wall, \( h_{fg} \) is the heat transfer coefficient between the fuel and ullage, \( A_{fg} \) is the area between the fuel and ullage, and \( Q_{load} \) is
the heat transfer rate of internal thermal loads. The subscripts \( f \) and \( g \) denote fluid and ullage, respectively.

The rate of change of the ullage temperature is a function of the heat transfer between each dry section of wall and between the fuel and the ullage. Mathematically this is represented by,

\[
\rho_g V_g C_p g \dot{T}_g = \sum_{i=1}^{n} h_{d,i} A_{d,i} (T_{wall,i} - T_g) + h_{fg} A_{fg} (T_f - T_g)
\]  

(2.1.3)

where \( \rho \) is the density, \( V \) is the volume, \( h_d \) is the heat transfer coefficient for the dry wall, and \( A_d \) is the area of the dry wall.

### 2.1.1.2. Component Inputs and Outputs

The fluid tank model acts solely as a thermal and fluid component. All inputs and outputs are in the thermal/fluid energy domain as shown in Figure 2.1.

Inputs to the fluid tank contain any fluid characteristics including mass flow rates, fluid temperatures, and fluid pressures. The ambient conditions of the aircraft are typically supplied via the mission profile and help in the calculation of the fluid tank pressure. The model assumes that the fluid tank is pressurized at a constant pressure above ambient. The final input to the fluid tank is the addition of any heat loads, which are accounted for in (2.1.2).

Fluid flow out of the tank contains fluid temperature, pressure, and flow rate signals. The flow rate out is commanded by the pump located upstream. In Simulink, this flow rate is supplied as an input to the fluid tank model. Other outputs include the fuel mass remaining in the tank and the ullage temperature.

![Figure 2.1. Fluid Tank Input and Output Energy Domains](image)

### 2.1.2. Fuel and Oil Pump

The pump model determines a mass flow rate of fluid being pumped from a source through the use of pump performance maps. The pump model does not contain internal dynamic states and is based off of static relationships and look up tables.
2.1.2.1. Mathematical Model

The mathematical model for the fuel pump is based upon first principles equations and widely available pump performance maps. The necessary maps require the relationship between the pump’s rotational speed, pressure differential, flow rate, and efficiency.

Mass flow rate is calculated as,

\[ \dot{m} = \dot{v} \rho \]  

(2.1.4)

where \( \rho \) is the density of the fluid being pumped and the volumetric flow rate, \( \dot{v} \), is determined using a pump performance curve that relates the pressure differential across the pump and the rotational speed of the pump,

\[ \dot{v} = f(\Delta P, \omega) \]  

(2.1.5)

Pump performance curves typically are published in the form of Figure 2.2 where multiple pump speed curves are given as a relationship between discharge flow rate and pressure differential across the pump. As the pump speed increases, the curves shift up and right. Normally pump curves use units of head (feet or meters) instead of pressure, but for a given fluid the conversion is trivial.

\[ T_{out} = T_{in} + \frac{\dot{Q}_{pump}}{\dot{m}C_p} \]  

(2.1.6)
\[
\dot{Q}_{\text{pump}} = \Delta P \dot{v} \left( \frac{1}{\eta} - 1 \right)
\] (2.1.7)

Pump efficiency is determined from an operating map similar to that of Figure 2.3 where the efficiency curves are plotted over the speed curves of Figure 2.2. This can be decomposed into a map relating the pressure differential, flow rate, and the pump’s rotational speed,

\[\eta = f(\Delta P, \omega, \dot{m}).\] (2.1.8)

![Discharge Flow Rate vs. Pressure Differential with efficiency curves](image)

**Figure 2.3. Pump Efficiency Curve**

The pump model has the ability to be coupled with a component model such as an AC motor, that provides the input torque to the pump; otherwise, the electrical power consumed by the pump is estimated by,

\[P_{\text{elec}} = \frac{1}{\eta} \tau \omega\] (2.1.9)

\[\tau = \frac{\Delta P \dot{v}}{2\pi \omega}\] (2.1.10)

where \(\eta\) is the pump overall efficiency, \(\Delta P\) is the pressure differential across the pump, \(\dot{v}\) is the volumetric flow rate through the pump, and \(\omega\) is the rotational speed of the pump. If a power factor \((pf)\) is applied, then the reactive power of the pump can be calculated,

\[Q = P \sqrt{1 - pf^2} \sqrt{pf^2}.\] (2.1.11)
2.1.2.2. Component Inputs and Outputs

Input and output energy domains for the fluid pump are shown graphically in Figure 2.4. Inputs to the fluid pump include temperature and pressure of the fluid upstream of the pump (i.e. fuel tank being pumped from) and pressure of the fluid downstream. To maintain causality in the system, the components connected to the pump must calculate a pressure while the pump will calculate a mass flow rate. The other input to the pump is a shaft speed in RPM. Typically fuel and oil pumps are driven by AC motors and thus should be connected to an external motor model.

Output from the model includes fluid flow rate through the pump and fluid temperature and pressure at the outlet. If an external motor is not attached to the pump, then electrical power is output from the pump as well.

![Fluid Flow (m, T, P) → Fluid Flow (m, T, P)](image)

**Figure 2.4. Fuel Pump Input and Output Energy Domains**

2.1.3. Stochastic Heat Loads

Often an aircraft’s thermal system will be subjected to unpredictable heat loads, albeit with some statistical behavior. These can be caused by cabin systems, ambient conditions, or system inefficiencies. The stochastic heat load models give the ability to model these loads and their effect on temperature. Since the loads are typically unknown, the model uses a normally distributed white noise signal. In implementation this signal is determined using a fixed seed so that the load has repeatability in simulation.

2.1.3.1. Mathematical Model

The model applies a temperature increase to a fluid flow and uses a normally distributed white noise as the load. The temperature of the flow out is found by integrating the white noise signal and adding it to the input flow temperature,

\[ T_{out} = T_{in} + \int v(t) \]  

(2.1.12)

Integration has a lower saturation value of zero to prevent the removal of heat from the flow.
2.2. Environmental Control System

The purpose of the environmental control system (ECS) is to condition the air that is supplied to the passenger cabin and cargo hold, and for the pressurization of cabin zones that cannot be at ambient pressures. Additionally, the ECS is responsible for dissipating heat generated by auxiliary systems, such as those components in electronic bays.

A typical ECS consists of several pneumatic air conditioning kits (PACKs) also known as air cycle machines (ACMs) that use a series of heat exchangers, compressors, and turbines to take bleed air at ~150°C and ~250kPa and condition to human comfort levels. An ECS also contains mixing junctions, valves, and fans that help move air throughout a cabin.

The following sections will detail the modeling of ECS components such as the turbine, compressor, heat exchanger, mixing junction, and a cabin thermal zone. Appendix A contains the additional models that are coupled to the ECS.

2.2.1. Thermal Zone/Cabin

The thermal zone operates as a passenger cabin, cockpit, cargo bay, or electronics bay. It is a 1-dimensional (lumped parameter) model, whereby the properties of the air inside the cabin are considered to be uniform throughout the zone. Pressure and temperature inside the zone are derived using conservation of mass and energy. Passengers inside the cabin are considered a source of sensible heat, and heat transfer due to radiation and kinetic friction are accounted for.

2.2.1.1. Mathematical Model

By treating the thermal zone model as a large control volume, the equations of conservation of mass and energy are used to determine the mass of air in the cabin and its pressure and temperature. In an ECS the air is typically routed three ways and in the following equations they will be denoted as: supply air \((SA)\), recirculation air \((RA)\), and exhaust air \((EA)\). The supply air comes from the PACK or ACM and is typically mixed with a fraction of the recirculation air which comes from the cabin. In order to keep the aircraft cabin pressurized and the air clean, a portion of the cabin air is exhausted from the aircraft.

Total energy balance of the thermal zone is given by,

\[
\dot{Q}_{\text{cab}} = \dot{Q}_{\text{in}} + \dot{Q}_{\text{gen}} - \dot{Q}_{\text{out}}.
\]  

(2.2.1)
This is broken up into four individual equations,

\[
\dot{Q}_{\text{cab}} = m_{\text{cab}} C_{p,\text{cab}} \frac{dT_{\text{cab}}}{dt} + C_{p,\text{cab}} T_{\text{cab}} \frac{dm_{\text{cab}}}{dt} \quad (2.2.2)
\]

\[
\dot{Q}_{\text{in}} = \left(\dot{m} C_p T\right)_{\text{SA}} + \dot{Q}_{\text{solar}} + \dot{Q}_{\text{kh}} \quad (2.2.3)
\]

\[
\dot{Q}_{\text{gen}} = n_{\text{pax}} q_{\text{pax}} \quad (2.2.4)
\]

\[
\dot{Q}_{\text{out}} = \left(\dot{m} C_p T\right)_{\text{RA}} + \left(\dot{m} C_p T\right)_{\text{EA}} \quad (2.2.5)
\]

where \(m_{\text{cab}}\) is the mass of the air in the thermal zone, \(C_{p,\text{cab}}\) is the specific heat of the air in the thermal zone, \(\dot{T}_{\text{cab}}\) is the time rate of change of the cabin air temperature, \(\dot{Q}_{\text{solar}}\) is the heat transfer rate due to solar radiation, \(\dot{Q}_{\text{kh}}\) is the heat transfer rate due to kinetic heating of the fuselage skin, \(n_{\text{pax}}\) is the number of passengers, and \(q_{\text{pax}}\) is the heat transfer rate per passenger, and \(\dot{m} C_p T\) is the mass flow rate, specific heat, and temperature of each air flow.

Kinetic heating, also known as aerodynamic heating, is produced by the movement of air over the fuselage slowing down in the boundary layer. As the fluid’s velocity approaches zero, its kinetic energy is converted into heat. Often the outer skin of a fuselage can reach temperatures of 100°C at subsonic speeds. While the leading edges of the aircraft are subjected to higher skin temperatures, most of the fuselage skin is away from the leading edge and is subjected to lower temperatures termed recovery temperature [9]. The recovery temperature can be estimated as a function of the aircraft’s Mach number,

\[
T_{\text{rec}} = T_{\text{amb}} \left(1 + 0.18 M^2 \right), \quad (2.2.6)
\]

where \(T_{\text{amb}}\) is the ambient temperature and \(M\) is the Mach number of the aircraft. The contribution of kinetic heating to the cabin temperature is,

\[
\dot{T}_{\text{kh}} = \frac{T_{\text{rec}} A_{\text{fus}} U}{m_{\text{cab}} C_{p,\text{cab}}} \quad (2.2.7)
\]

where \(A_{\text{fus}}\) is the area of the fuselage and \(U\) is the overall heat transfer coefficient of the fuselage skin.
The final source of heating comes from solar radiation. Since an aircraft is mostly opaque the transmission radiation is assumed to be zero, and the absorbed heat flux by the top half of the aircraft can be estimated as a function of the aircraft fuselage’s absorptivity \( \alpha_{\text{fus}} \),

\[ G_{\text{abs, top}} = \alpha_{\text{fus}} G \]  

(2.2.8)

where the total irradiation is \( G \). Total irradiation can be easily calculated as a function of latitude and time of day. Typical values for the absorptivity of an aircraft fuselage are between 0.2-0.5 [10]. The lower surface of the aircraft is heated by radiation reflected from the ground,

\[ G_{\text{abs, bot}} = \alpha_{\text{fus}} (1 - \alpha_{\text{grd}}) G \]  

(2.2.9)

where \( \alpha_{\text{grd}} \) is the absorptivity of the ground. Therefore, the total absorptivity and the contribution to cabin heating is,

\[ G_{\text{abs}} = G_{\text{abs, top}} + G_{\text{abs, bot}} \]  

(2.2.10)

\[ \dot{T}_{\text{solar}} = \frac{A_{\text{fus}} G_{\text{abs}}}{m_{\text{cab}} C_{p,\text{cab}}} \]  

(2.2.11)

Finally, (2.2.1) can be simplified to the following 1D transient differential equation for determining the temperature of the cabin air,

\[ \dot{T}_{\text{cab}} = \left( \frac{\dot{m}_{\text{SA}} C_{p,\text{SA}}}{m_{\text{cab}} C_{p,\text{cab}}} \right) T_{\text{SA}} - \left( \frac{\dot{m}_{\text{SA}} C_{p,\text{cab}}}{m_{\text{cab}} C_{p,\text{cab}}} \right) T_{\text{cab}} + \dot{T}_{\text{solar}} + \dot{T}_{\text{kh}} + \left( \frac{n_{\text{pax}} \dot{Q}_{\text{pax}}}{m_{\text{cab}} C_{p,\text{cab}}} \right). \]  

(2.2.12)

For the previous calculations, the mass of the air inside of the thermal zone is calculated by,

\[ m_{\text{cab}} = \int (\dot{m}_{\text{SA}} - \dot{m}_{\text{RA}} - \dot{m}_{\text{EA}}) dt. \]  

(2.2.13)

Cabin pressure can be determined using the assumption of the ideal gas law and the calculated cabin air mass,

\[ P_{\text{cab}} = \frac{m_{\text{cab}} R_{\text{air}} T_{\text{cab}}}{V_{\text{cab}}}. \]  

(2.2.14)

2.2.1.2. Component Inputs and Outputs

Input and output energy domains for the thermal zone model are shown in Figure 2.5. Inputs to the model include the fluid flow rates and temperatures, ambient conditions, thermal loads from other systems, and mission parameters such as Mach number and ambient pressure. The
thermal zone model will output conditions such as temperature and pressure and the properties of the recirculation air. In order to pressurize the zone, a control loop is required to adjust the mass flow rate in and out of the thermal zone.

Fluid Flow ($\dot{m}, T, P$) → Fluid Flow ($\dot{m}, T, P$)

Thermal Loads ($\dot{Q}$) → Cabin Conditions ($T, P$)

Mission Conditions ($M, T$) → Cabin Conditions ($T, P$)

**Figure 2.5. Thermal Zone Input and Output Energy Domains**

### 2.2.2. Heat Exchanger

Most aircraft utilize air-air heat exchangers in the ACM in order to condition the air to a desirable temperature for the cabin. Since these heat exchangers have the potential to vary widely between platforms, the presented model is a dynamic 1D lumped parameter model [11] intended to be scalable for multiple applications. More details of the model scalability can be found in Appendix A.

For aircraft, weight and size of components is an important factor in the design of systems. Although there are several types of compact heat exchangers on the market, offset strip-fim heat exchangers are often used by industries that require lightweight, high-performance heat exchangers. An example of this configuration is shown in Figure 2.6. The mathematical model presented in the following section is based upon this configuration.

**Figure 2.6 One Layer of an Offset Strip-Fin Heat Exchanger [11]**
2.2.2.1. Mathematical Model

The geometric parameters of the offset strip-fin heat exchanger are shown in Figure 2.7. Where \( s \) is the transverse spacing or free flow width, \( h \) is the free flow height, \( t \) is the fin thickness, and \( l \) is the fin length.

![Figure 2.7 Geometric Parameters of the Offset Strip-Fin Heat Exchanger [11]](image)

Using these dimensions, and the above figure, the hydraulic diameter can be defined as,

\[
D_h = \frac{4shl}{2(sl + hl + th) + ts}.
\] (2.2.15)

The correlation [12] for the friction factor is in the form of a power-law and given by,

\[
f = K_1(Re)^{a_1} \left(\alpha \right)^{a_2} \left(\delta \right)^{a_3} \left(\gamma \right)^{a_4},
\] (2.2.16)

where \( Re \) is the Reynolds number and \( K_1, a_1, a_2, a_3, a_4 \) are power law coefficients depending on if the flow is laminar or turbulent. The remaining terms are defined using geometric ratios,

\[
\alpha = \frac{s}{h}
\] (2.2.17)

\[
\delta = \frac{t}{l}
\] (2.2.18)

\[
\gamma = \frac{t}{s}
\] (2.2.19)

The friction factor is used to determine the mass flow rate through the heat exchanger. In order to do so, the definition of the average Fanning friction factor (2.2.20) is used to convert the
mean fluid velocity into mass flow rate (2.2.21). It is assumed that the mass of fluid inside the
heat exchanger is incompressible, and thus mass does not change with time. By rearranging
(2.2.20) and (2.2.21) the mass flow rate can be solved for,

\[ f = \frac{\Delta P}{2} \frac{D_h}{L} \frac{1}{\rho u_m^2} \] (2.2.20)

\[ u_m = \frac{\dot{m}}{\rho A} \] (2.2.21)

\[ \dot{m} = A \sqrt{\frac{\rho \Delta P D_h}{2 f L}}, \] (2.2.22)

where \( \Delta P \) is the pressure drop across the heat exchanger, \( L \) is the total length in the direction of
the flow, \( \rho \) is the density of the fluid moving through the heat exchanger, and \( A \) is the total
free flow area. By substituting in (2.2.16), the mass flow rate can be written as,

\[ \dot{m} = A \left[ \frac{1}{K_1 \mu^{-a_1} \alpha \beta \gamma \delta} \frac{\rho \Delta P D_h^{(1-a_1)}}{2 L} \right]^{\frac{1}{(2+a_1)}}, \] (2.2.23)

where \( \mu \) is the fluid viscosity.

The convective heat transfer is then calculated as,

\[ h = j \text{Re} \left( \frac{\mu C_p}{K_f} \right)^{1/3} \left( \frac{K_f}{D_h} \right) \] (2.2.24)

where \( j \) is the Colburn factor, \( K_f \) is the fluid thermal conductivity, and \( C_p \) is the fluid specific
heat at constant pressure. The Colburn factor can be determined using [12] for laminar and
turbulent flow regions.

Heat transfer will occur in the fin through five different types of resistance as shown in
Figure 2.8, including convection and conduction. Convection from the fluid to the plate/fin base
is represented as \( R_1 \), convection from the fluid to the fin is \( R_2 \), conduction through the fin base is
\( R_3 \), conduction through the plate thickness is \( R_4 \), and conduction through the fin is \( R_5 \). Resistances \( R_2 \) and \( R_5 \) can be combined into a single resistance, while the other resistances can
be defined mathematically as functions of the geometry, total number of fins in the line \( \left( N_f \right) \).
the fin thermal conductivity \((K_n)\), and the plate thermal conductivity \((K_p)\). After defining the equations for each resistance and building a thermal circuit, an equivalent thermal resistance \(R_{eq}\) is calculated. The details of this derivation can be found in Ref. [11], and are recreated here,

\[
R_1 = \frac{1}{hN_fs} \quad (2.2.25)
\]

\[
R_3 = \frac{t}{K_nN_fs} \quad (2.2.26)
\]

\[
R_4 = \frac{t_p}{2K_pN_fsL} \quad (2.2.27)
\]

\[
R_2 + R_{5*} + R_5 = \frac{1}{2h^2N_f \varepsilon_f} , \quad (2.2.28)
\]

where \(\varepsilon_f\) is the efficiency of the fins. The equivalent thermal resistance for the hot and cold passages is represented as,

\[
R_{eq} = R_1 \left( \frac{R_1 R_5}{R_1 + R_5} + R_3 \right) \left( \frac{R_1 R_5}{R_1 + R_5} + R_3 + R_1 \right)^{-1} + R_4. \quad (2.2.29)
\]

---

The last part of the model involves the heat exchange between the cold and hot fluids via the core mass. A graphical representation of this problem is shown in Figure 2.9 and contains a hot
fluid control volume, core mass, and cold fluid control volume. Mass flow rate is represented as \( \dot{m} \), temperature is represented as \( T \), heat transferred is represented as \( q \), and the energy accumulation in the core mass is \( dE_{ac}/dt \). Hot and cold is denoted by \( h \) and \( c \) respectively, in and out is denoted by \( i \) and \( o \) respectively, and \( cm \) denotes the core mass.

![Figure 2.9. Core Heat Exchange Model [11]](image)

Heat lost by the hot fluid due to the contact with the surface of the core mass is given by (2.2.30). It should be noted here that all developments are based on mean temperature, as defined in (2.2.31).

\[
\dot{m}_h C_p (T_{hi} - T_{ho}) = \frac{1}{R_h} (\bar{T}_h - \bar{T}_{cm}) \quad (2.2.30)
\]

\[
\bar{T}_h = \frac{T_{hi} + T_{ho}}{2} \quad (2.2.31)
\]

The use of mean temperatures allows the heat transfer, and therefore the core mass temperature to be obtained only as a function of the inlet temperatures of the fluids, and not dependent on the fluid outlet temperature, which is to be determined. By isolating the hot outlet temperature in (2.2.31) and rearranging (2.2.30), the mean hot side temperatures can be determined. A similar process is performed for the cold side,

\[
\bar{T}_h = \bar{T}_{cm} + \frac{2 R_h \dot{m}_h C_p (T_{hi} - \bar{T}_{cm})}{1 + 2 R_h \dot{m}_h C_p}, \quad (2.2.32)
\]

\[
\bar{T}_c = \bar{T}_{cm} + \frac{2 R_c \dot{m}_c C_p (T_{ci} - \bar{T}_{cm})}{1 + 2 R_c \dot{m}_c C_p}. \quad (2.2.33)
\]

Equations (2.2.32) and (2.2.33) present the mean cold and hot fluid temperatures as a function of the mean core mass temperature and known parameters, such as mass flow and
equivalent thermal resistances \((R_h \text{ and } R_c)\). The transient response of the core energy and its temperature is derived as,

\[
\frac{dE_{ac}}{dT} = m_c C_{p,cm} \frac{dT_{cm}}{dT} = \left[ \frac{2 \dot{m}_h C_p (T_{hi} - T_{cm})}{1 + 2 R_h \dot{m}_h C_p} + \frac{2 \dot{m}_c C_p (T_{ci} - T_{cm})}{1 + 2 R_c \dot{m}_c C_p} \right].
\]  
(2.2.34)

2.2.2.2. Component Inputs and Outputs

The input and output energy domains for the heat exchanger are shown in Figure 2.10. Since the only purpose of the heat exchanger is the transfer of heat from one fluid to another, it solely deals with the thermal energy domain.

Inputs to the model are fluid flow rates, temperatures, and pressures of the hot and cold fluids. The outputs will be flow rates, temperatures, and pressures of both fluid streams.

![Figure 2.10. Heat Exchanger Input and Output Energy Domains](image)

2.2.3. Mixing Junction

The mixing junction calculates the output fluid temperature and pressure of fluid streams that merge together. Pressure in the mixing junction is used by upstream and downstream components to determine the flow rate in and out of the mixing junction.

2.2.3.1. Mathematical Model

Mass of the air in the mixing junction may be very small, but it is necessary for the calculation of mixing junction pressure. Mass is determined by the conservation of mass,

\[
m_{junc} = \int (\dot{m}_{in} - \dot{m}_{out}) \, dt.
\]  
(2.2.35)

The outlet temperature is determined using conservation of energy,

\[
T_{out} = \frac{\sum_{i=1}^{n} \dot{m}_i C_{p,i} T_i}{\dot{m}_{out} C_{p,out}}
\]  
(2.2.36)
where \( n \) is the number of flows into the junction, \( \dot{m} \) is the flow rate of each flow, \( C_p \) is the specific heat at constant pressure for each flow, and \( T \) is temperature. The outlet specific heat is determined as a weighted average of each inlet flow based on inlet mass flow rate.

Mixing junction pressure can be determined using the assumption of the ideal gas law and the calculated junction air mass,

\[
P_{junc} = \frac{m_{junc} R_{air} T_{out}}{V_{junc}}
\]

(2.2.37)

where \( V_{junc} \) is the volume of the junction, and \( R_{air} \) is the universal gas constant for air.

2.2.3.2. Component Inputs and Outputs

The mixing junction deals only with the thermal energy domain. Inlet flows provide mass flow rates and temperatures. The only outputs are the result of the calculations from the previous section for fluid temperature, pressure, and flow rate.

2.2.4. Compressor and Turbine

The compressor and turbine operate together to move air through the heat exchangers before the air is sent to the mixing junction. The turbine is powered by air exiting the secondary heat exchanger of an ACM. The turbine rotates a shaft that is connected to the compressor; therefore, the rotational speed of the shaft can be derived as a power balance between the turbine and compressor. The mathematical models for these two coupled components is presented in the following section.

2.2.4.1. Mathematical Model

The compressor and turbine combination is essentially a turbocharger, and thus the governing equations for this system are derived from Ref. [13][14]. The following first order differential equation is used to determine the rotational speed \( (N) \) of the shaft relative to the power balance between the turbine \( (P_t) \) and the compressor \( (P_c) \),

\[
\frac{dN}{dt} = \left( \frac{60}{2\pi} \right)^2 \frac{P_t - P_c}{JN}
\]

(2.2.38)

where \( J \) is the inertia of the turbocharger. Speed is represented in RPM.
By assuming the process of expanding the flow through the turbine is isentropic, the inlet and outlet pressure and temperature can be related. Since this process is not truly isentropic because of enthalpy losses across the turbine, an isentropic efficiency term \( \eta_t \) is included,

\[
T_{out} = \eta_t T_{in} \left( \frac{P_{out}}{P_{in}} \right)^{\frac{\gamma-1}{\gamma}}
\]  

(2.2.39)

where \( \gamma \) is the ratio of specific heats for air (typically \( \sim 1.3 \) for hot fluid flow through a turbine). In (2.2.39) the inlet fluid conditions are supplied via the exit of the upstream component and the outlet pressure is determined by the pressure downstream of the turbine.

The power delivered by the turbine can be calculated as a function of (2.2.39),

\[
P_t = \dot{m}_t C_p \eta_t T_{t,in} \left( 1 - \left( \frac{P_{out}}{P_{in}} \right)^{\frac{\gamma-1}{\gamma}} \right)
\]  

(2.2.40)

where \( \dot{m}_t \) is the mass flow rate through the turbine, \( T_{t,in} \) is the inlet temperature to the turbine, and \( C_p \) is the specific heat of the fluid moving through the turbine.

While the turbine provides power, the compressor will consume more power than it delivers to the flow due to inefficiencies. The total power consumed by the compressor is a function of the pressure ratio across the compressor and can be calculated by,

\[
P_c = \dot{m}_c C_p T_{in} \frac{1}{\eta_c} \left( \frac{P_{out}}{P_{in}} \right)^{\frac{\gamma-1}{\gamma}} - 1
\]  

(2.2.41)

where \( \dot{m}_c \) is the mass flow rate through the compressor, \( C_p \) is the specific heat of the fluid moving through the compressor, \( T_{in} \) is the inlet temperature to the compressor, and \( \eta_c \) is the compressor efficiency.

The compressor efficiency and pressure ratio are determined through 2D maps that can be defined functionally as,

\[
\eta_c = f \left( \frac{\dot{m}_c \sqrt{T_{in}}}{P_{in}}, \frac{N}{\sqrt{T_{in}}} \right)
\]  

(2.2.42)
\[ \frac{P_{out}}{P_{in}} = g \left( \frac{\dot{m} \sqrt{T_{in}}}{P_{in}}, \frac{N}{\sqrt{T_{in}}} \right). \]  

(2.2.43)

### 2.2.4.2. Component Inputs and Outputs

The input and output energy domains for the turbine and compressor can be seen in Figure 2.11 and Figure 2.12, respectively.

Input to the turbine includes thermal and mechanical energy. The thermal inputs are air flow rate, temperature and pressure while the mechanical input is the power consumed by the compressor. Outputs include air flow rate, temperature, and pressure in the thermal energy domain and a mechanical shaft speed.

The compressor accepts thermal and mechanical power inputs. The thermal input includes the air flow rate, pressure, and temperature. Mechanical power input is the shaft speed which is supplied as an output from the turbine model. Output from the compressor model also spans the thermal and mechanical energy domains. A fluid flow rate, pressure, and temperature are output in addition to the mechanical power consumed by the compressor.

![Figure 2.11. Turbine Input and Output Energy Domains](image1)

![Figure 2.12. Compressor Input and Output Energy Domains](image2)
Chapter 3

Variable Fidelity Modeling

Computer based modeling and simulation is heavily relied upon in multiple engineering fields throughout the phases of system design and development. With the continual advancement of computational power and modeling techniques available to engineers, the automotive and aerospace industries are seeing a steady increase in the use of models for rapid design iterations and prototyping, which helps reduce the development time and improves product quality [15][16]. The National Aeronautics and Space Administration (NASA) recently highlighted a greater need in the aerospace field for large-scale, high-fidelity models to be used throughout the design of systems [17].

High-fidelity models are advantageous to low-fidelity models because of their ability to better capture dynamics of a true physical system which helps facilitate improved system design. However, this comes at a trade off with computational speed. As models become more detailed and complex, the computational time required to simulate the operation of the system can increase dramatically relative to a lower fidelity model. For example, finite volume models use discretized control volumes to solve algebraic systems of equations. If the model is discretized into $m$ volumes with $k$ algebraic equations per volume, the equations could be arranged in a matrix $A$ of dimension $n \times n$ where $n = mk$. Then the equation $Ax = y$ could be solved for the unknown parameter vector $x$ given knowledge of inputs and outputs to the control volumes arranged in $y$. The time complexity of this calculation grows exponentially depending upon the algorithm used for inversion. For example, if using Gauss-Jordan elimination the time complexity would be $O(n^3)$ or as low as $O(n^{2.373})$ using the Williams method [18].

Due to this relationship between accuracy and computational speed, a system modeler will find himself battling with two competing objectives given a fixed amount of computational resources: 1) explore large regions of a design space and conduct multiple iterations of a system design or 2) explore a limited region of design space in greater detail for in-depth system design. This is similar to operating along the trade-off curve shown in Figure 3.1. The first objective could be achieved by using a low-fidelity model, denoted with the circle (●), to quickly cycle
through designs of a system using low-fidelity models that execute rapidly, but give results that are not as accurate as high-fidelity models. Alternatively, by using the high-fidelity model which is more accurate but computationally slow, represented as the square (■), the system designer could focus on fewer system designs than with the lower fidelity models, but at an increased accuracy.

The goal of variable fidelity modeling is to allow the system modeler to operate off of this tradeoff curve (★), and sacrifice a little accuracy for an increase in computational speed.

3.1. Relationship between Accuracy, Speed, and Fidelity

Before developing a method for varying the fidelity of a model during simulation, it is necessary to understand what causes model state and output inaccuracies as well as the effect on computational speed. First we define several terms,

1. **Fidelity** – the extent to which a model can replicate the actual physical event. The degree to which a model reflects the behavior of a real system [19][20]. Fidelity is a characteristic of the model.

2. **Accuracy** – in a modeling and simulation context, is the degree to which a set of parameters, outputs, or variables within a simulation conform exactly to reality or some chosen standard [21]. Accuracy is a characteristic of the simulation.

3. **Computational Speed** – mathematically represented as \( \frac{t_{\text{sim}}}{t_{\text{real}}} \), or the ratio of simulation length to elapsed time. Qualitatively this is the ratio of simulation length to time waiting for the simulation to finish.

4. **Computational Cost** – mathematically represented as \( \frac{t_{\text{real}}}{t_{\text{sim}}} \). Computational cost is the inverse of computational speed. It is desirable to minimize cost, which maximizes speed.

![Figure 3.1. Trade-off between computational speed and accuracy for high (■), low (●), and variable (★) fidelity models](image)
Differentiating between high and low-fidelity models is also important. If a physical system is realistically and fully described as a set of $n$ dynamic equations,

$$\dot{x}_i = f_i(x) \forall i \in N = [1, n], \quad (3.1.1)$$

then a high-fidelity model can be represented as one that contains some subset of those equations,

$$\dot{x}_j = f_j(x) \forall j \in H \subset N. \quad (3.1.2)$$

Furthermore, a low-fidelity model may be defined such that it contains a subset of the high-fidelity model dynamics,

$$\dot{x}_k = f_k(x) \forall k \in \Lambda \subset H. \quad (3.1.3)$$

It should be noted that low-fidelity as defined here is what is considered for this thesis; however, it is possible for a model containing a subset of the realistic dynamic equations to not be a subset of the high-fidelity model ($\Lambda \subset N \setminus H$), and also be referred to as low-fidelity. Ideally, models exhibiting these types of dynamics would not be used interchangeably. An example would be a high-fidelity model of a spacecraft in orbit and a low-fidelity model of a spacecraft in atmospheric flight. Both models may belong to the true dynamics of the spacecraft, but the two models would not be used interchangeably.

The relationship between (3.1.1), (3.1.2), and (3.1.3) is shown graphically in Figure 3.2. The largest area describes the true (real) dynamics (3.1.1) of a system being modeled. The high-fidelity model (3.1.2) may be capable of capturing a significant portion of this dynamic space, while the low-fidelity model (3.1.3) captures a smaller subset of the true dynamics, which is also a subset of the high-fidelity dynamics. In this situation, the low-fidelity model is assumed to be less accurate at estimating true phenomena outside of its dynamic space.

Figure 3.2. Dynamic space of high and low-fidelity models and a simulation (white line) that switches at the boundaries
Knowing the simulation space and the dynamic system being investigated, the model can by systematically switched between high and low-fidelity mode. In a simulation setting (Figure 3.2 – white line), a switch from high to low fidelity would occur as the simulation enters the dynamical space that the low-fidelity model is capable of capturing (■), and temporarily enters high-fidelity mode whenever the simulation leaves the dynamical space of the low-fidelity model (●). The difficulty lies in determining where these boundaries lie in the dynamical system space.

It often turns out that in the field of dynamical system modeling and simulation, the principles of inertia hold true. A system at steady state will remain at steady state unless it is perturbed, and a system in transient will remain in transient unless it is hindered. Luckily, the latter statement is naturally taken care of by properties of physics such as friction and damping. The former statement raises the subject of perturbations, and often in dynamical modeling and simulation these perturbations come in the form of exogenous system signals such as reference or disturbance changes. These exogenous signals greatly influence the transients in a system and the model fidelity directly influences how a model responds to these perturbations.

The following two sub-sections provide numerical examples of how model fidelity directly impacts simulation accuracy and computational speed. Examples are given using a finite volume heat exchanger model and finite volume pipe model from the Air Force Research Laboratory (AFRL) Transient Thermal Modeling and Optimization (ATTMO) toolbox [22]. This toolbox is based upon the vapor compression system modeling toolbox, Thermosys [23] which was developed at the University of Illinois at Urbana-Champaign (UIUC) in the Alleyne Research Group (ARG). While the evaporator and pipe models provide an excellent demonstration for this problem, it should be noted that not all systems will behave in an identical manner.

### 3.1.1. Effect of Model Fidelity on Simulation Accuracy

Figure 3.3 graphs the time history of a high and low-fidelity heat exchanger model where the high-fidelity model has a factor of five more volumes than the low-fidelity model. Both models are subjected to the same disturbance in the inlet conditions of the refrigerant flow. Two system outputs are plotted: primary flow (refrigerant) pressure at the exit of the heat exchanger (Figure 3.3 top), and secondary flow (air) temperature at the exit of the heat exchanger (Figure 3.3 bottom). It is observed that during the disturbance events, both high and low-fidelity heat exchangers track similar transients, but arrive at different steady state values. This error in
steady state values can be detrimental to system design and performance, especially in the situation where a heat exchanger model such as the one presented is coupled to a room model that has integrator or first order dynamics. The error between the high and low-fidelity models will get propagated to the coupled system and error will accumulate over the course of the simulation.

Figure 3.3. Comparison of the steady-state error between high and low-fidelity model outputs

In Figure 3.4 a different trend is represented by the pipe model. Similar to the heat exchanger model, the time history of a high and low-fidelity pipe model is graphed where the high-fidelity model has a factor of five more volumes than the low-fidelity model. In this simulation, the temperature of the inlet flow to the pipe is stepped up to 20°C at 15 seconds. The outlet temperature of the pipe is plotted. Both the high and low-fidelity models arrive at the same steady state value, but the transient responses are much different. The high-fidelity model has a delay before the temperature of the outlet flow begins to increase, which is physically intuitive given that the inlet flow has to propagate through the pipe. The low-fidelity model sees
a change in exit temperature almost instantaneously, which would only make sense if the pipe were extremely short, at which point both models should predict the same behavior.

![Graph showing temperature vs. time for high and low fidelity models.](image)

**Figure 3.4. Comparison transient response error between high and low-fidelity model outputs**

### 3.1.2. Effects of Model Fidelity on Computational Speed

Variable time step solvers provide an excellent solution to decreasing computational cost by allowing the simulation to take larger steps during steady state conditions, but decrease the step size when needed, typically during transient events. However, the maximum step size is sometimes limited by other system components or control systems that require a fixed update rate, and with a greater number of simulation computations, high-fidelity models would still drastically reduce computational speed.

Figure 3.5 shows how increasing modeling fidelity, in this case by increasing the number of volumes (y-axis), the computational speed (x-axis) decreases significantly. The speed difference between a model with 10 volumes and 100 volumes is a factor of five.

![Graph showing simulation speed and level of fidelity.](image)

**Figure 3.5. Comparison of simulation speed and level of fidelity**
3.2. Modeling Framework

It has been shown that both model fidelity and transient events affect the accuracy and computational speed of a model. Given that transient events are the result of changing exogenous signals, and the fact that the system modeler is responsible for defining those exogenous signals, a supervisory logic can be developed to analyze the signals and determine when the simulated system should switch to a high-fidelity mode. Since it is desired to capture the full transient event of the system subjected to the changing exogenous signal, it is necessary to switch to high-fidelity mode prior to the signal affecting the system. Because of this requirement, the supervisory logic responsible for determining model fidelity must receive the exogenous signals before the simulated system. Therefore the following model framework is proposed:

1. Exogenous signals \( (r(t), d(t)) \) are sent to a supervisor.

2. The supervisor is responsible for analyzing the exogenous signals and determining the level of fidelity in the simulated system model.

3. Exogenous signals are delayed being sent to the simulated system model by some time delay, \( z^{-n} \), that is determined heuristically as a function of the simulated system time constant and required time for the supervisor to analyze the exogenous signals.

Graphically this is shown in Figure 3.6, where \( r(t) \) and \( d(t) \) are fed to the supervisor, but delayed being sent to the simulated system by the time delay \( z^{-n} \). The supervisor is responsible for switching the level of fidelity by passing a high/low signal to the simulated system. The following section gives a detailed explanation of the supervisor algorithm for determining level of model fidelity during simulation.

![Figure 3.6. Signal flow of reference and disturbance signals through the supervisor and time delay to the simulated system](image-url)
3.3. Supervisor Design

The supervisor consists of three parts, Figure 3.7, with two variables, $K_{\text{filter}}$ and $t_{\text{dwell}}$, used for tuning the frequency and duration of high-fidelity simulation. The input exogenous signal, $d$, is fed through the filter. Output of the filter, $y_1$, is multiplied by $K_{\text{filter}}$ and the product is related to a threshold via discrete logic. This output, $y_2$, is passed to a module that coordinates whether the simulated model is in high-fidelity mode as a function of $t_{\text{dwell}}$. Output of the supervisor is denoted as $y_{\text{supervisor}}$.

![Figure 3.7. Supervisor signal flow](image)

The following sections describe in detail the purpose and structure of each part of the supervisor. Boolean logic is used by the supervisor to denote high or low-fidelity mode. An output of zero corresponds to low-fidelity mode and a one corresponds to high-fidelity mode.

3.3.1. Filter

The filter’s purpose is to determine if the exogenous signal is a significant transient event such that when it affects the system a switch to high-fidelity would be beneficial in capturing the true system response. In general, the filter should relate the exogenous signal’s magnitude change and rate of change to a threshold, such that if the output of the filter is greater than that threshold, then the simulated model should be in high-fidelity mode.

For the work presented herein, expected exogenous signals come in the form of ramps and steps. As such, a bandpass filter (3.3.1) is selected because of its output for each of these signals, which are shown in Figure 3.8. It is ideal to have the output of the filter return to zero after a step change occurs because once the system is subjected to a step change, it will reach steady state value and remain relatively constant. For a ramp exogenous signal, the filter output should be non-zero because the system will be experiencing transients during the ramp.

$$
Y_1(s) = \frac{s}{s^2 + 2\xi\omega_n + \omega_n^2} d(s)
$$

(3.3.1)
3.3.2. $K_{\text{filter}}$ and Discrete Logic

Output from the filter is proportional to the input exogenous signal and needs to be scaled relative to a threshold. For exogenous signals that are significant enough to trigger a switch to high-fidelity mode, it is ideal to scale the output of the filter to be greater than one for these events, and less than one for exogenous signals that should not require a switch to high-fidelity mode. By scaling the filter output, it gives the supervisor the flexibility to determine which exogenous signals are significant, and thus $K_{\text{filter}}$ provides a tuning parameter to determine the sensitivity of the supervisor to exogenous signals.
Output from the discrete logic is provided by (3.3.2). If the product of $K_{\text{filter}}$ and the filter output is greater than one, then the logic says the model should be in high-fidelity mode by passing out a one. If the product is less than one, then the logic says the model should be in low-fidelity mode and indicates this by passing a zero. This logic is demonstrated graphically in Figure 3.9.

![Graphs](image)

**Figure 3.9.** Scaling by $K_{\text{filter}}$ of the filter output for the step (a) and ramp (c) and the corresponding discrete logic for the step (b) and the ramp (d)
The relationship between the filter output for a step response and the scaling by $K_{filter}$ can be seen in Figure 3.9a. For the length of time that the product of $K_{filter}$ and the filter output is greater than one, the output of the logic (3.3.2) is one, as shown in Figure 3.9b. A similar trend is presented for the input of a ramp in Figure 3.9c and Figure 3.9d.

### 3.3.3. Coordination of Fidelity Mode and $t_{dwell}$

The final part of the supervisor is the determination of fidelity mode and the application of the dwell time.

Level of fidelity is determined using the output $y_2$ and the dwell time. If $y_2$ is non-zero then the simulated system should be operating in high-fidelity mode because a significant exogenous signal has affected the system or is about to affect it. Since $y_2$ will return to zero after the filter output drops below one, the dwell time is used to keep the model in high-fidelity mode long enough to capture the full transient event. Once the simulation has progressed far enough, the model will be switched back to low-fidelity mode. This is shown mathematically in (3.3.3) where the time at which $y_2$ switches from zero to one is denoted as $t_{switch}$ and the current simulation time is denoted as $t_{sim}$. Graphically this logic is shown in Figure 3.10, as well as the example for the step and ramp inputs in Figure 3.11.

$$
y_2 = \begin{cases} 
1 & |K_{filter}y_1| \geq 1 \\
0 & |K_{filter}y_1| < 1 
\end{cases} \tag{3.3.2}
$$

$y_2 = \begin{cases} 
1 & y_2 = 1 \\
1 & t_{sim} < t_{dwell} + t_{switch} \quad \& \quad y_2 = 0 \\
0 & t_{sim} \geq t_{dwell} + t_{switch} \quad \& \quad y_2 = 0 
\end{cases} \tag{3.3.3}$

---

**Figure 3.10. Application of dwell time to level of fidelity switching logic**
3.4. **Effect of $K_{\text{filter}}$ and $t_{\text{dwell}}$ on Computational Cost and Accuracy**

The two tunable parameters $K_{\text{filter}}$ and $t_{\text{dwell}}$ provide the ability to determine the frequency and length of time spent in high-fidelity mode. Intuitively, by increasing $K_{\text{filter}}$ the model will switch to high-fidelity for smaller exogenous signals, and increasing $t_{\text{dwell}}$ will result in remaining in high-fidelity mode for longer periods of time.
Determining the effect these variables have on computational cost and accuracy can be achieved by simulating the model for a range of $K_{\text{filter}}$ and $t_{\text{dwell}}$. As an example, a finite volume heat exchanger model from the ATTMO toolbox [22] will be used. The high-fidelity model consists of 25 volumes while the low-fidelity model consists of five volumes. Error accumulation is defined as the magnitude of the difference between the high-fidelity model and the switched-fidelity model integrated over the length of the simulation. This is mathematically represented as,

$$e_{\text{acc}} = \int_0^{t_f} |y_{\text{high}} - y_{\text{switched}}| \, dt$$

(3.4.1)

where $y_{\text{high}}$ is the output of the high-fidelity model over the full simulation, $y_{\text{switched}}$ is the output from the switched-fidelity model over the full simulation, and $t_f$ is the simulation length.

Two outputs from the evaporator model will be used for calculation of accumulated error: primary flow (refrigerant) exit pressure and secondary flow (air) exit temperature.

The full range of simulations cover the range of $0.1 \leq K_{\text{filter}} \leq 10$ and $5 \text{ sec.} \leq t_{\text{dwell}} \leq 100 \text{ sec.}$. For $K_{\text{filter}} = 0.1$ the model would switch for only $5\%$ of exogenous signals, compared to switching for all exogenous signals if $K_{\text{filter}} = 10$. For a $t_{\text{dwell}} = 5 \text{ sec.}$ the model has completed less than one time constant, while for $t_{\text{dwell}} = 100 \text{ sec.}$ the system has gone through 5 or 6 time constants.

### 3.4.1. Influence of $K_{\text{filter}}$ and $t_{\text{dwell}}$ on Accumulated Error

The effect of $K_{\text{filter}}$ and $t_{\text{dwell}}$ on accumulated pressure error can be seen in Figure 3.12 and on accumulated temperature error in Figure 3.13. Similar trends occur in both variables. For high values of $K_{\text{filter}}$ and $t_{\text{dwell}}$ the error accumulation is very small. This is a direct result of the fact that the simulation is operating in high-fidelity mode for extended periods of time, and since the high-fidelity model is assumed more accurate, the error accumulated is minimal. Alternatively the error accumulation is very large for small values of $K_{\text{filter}}$ and $t_{\text{dwell}}$. In these situations the model is executing the inaccurate low-fidelity model for extended periods of time leading to large error accumulation.
Figure 3.12. Effect of $K_{\text{filter}}$ and $t_{\text{dwell}}$ on the error accumulation in the primary flow exit pressure

Figure 3.13. Effect of $K_{\text{filter}}$ and $t_{\text{dwell}}$ on the error accumulation in the secondary flow exit temperature
3.4.2. Influence of $K_{\text{filter}}$ and $t_{\text{dwell}}$ on Computational Cost

The computational cost of a model is heavily affected by the system specifications. The simulations presented herein were conducted on system with an Intel Xeon E31225 processor at 3.10GHz, 8GB of physical memory, Windows 8.1, and MATLAB 2012a.

The effect of $K_{\text{filter}}$ and $t_{\text{dwell}}$ on computational cost is plotted in Figure 3.14. For high values of $K_{\text{filter}}$ and $t_{\text{dwell}}$ the computational cost is very high, meaning the simulation speed is very slow. It is also observed that the minimal computational cost comes when switching to high-fidelity mode infrequently, which is for low values of $K_{\text{filter}}$.

![Figure 3.14. Effect of $K_{\text{filter}}$ and $t_{\text{dwell}}$ on computational cost](image)

3.4.3. Optimal Trade-off between Computational Cost and Accuracy

In [24] a multi-variable optimization problem is formulated to determine the optimal values of $K_{\text{filter}}$ and $t_{\text{dwell}}$ to minimize error accumulation and computational cost. That analysis is recreated here.
Given the inverse relationship between error accumulation and computational cost as a function of $K_{\text{filter}}$ and $t_{\text{dwell}}$, the following cost function if formulated,

$$J = \lambda \left[ \sum_{i=1}^{n} \gamma_i e_{\text{acc},i} \right] + \left( 1 - \lambda \right) \frac{t_{\text{real}}}{t_{\text{sim}}}$$

(3.4.2)

where $\lambda$ is the weighting term, $\gamma$ is the scaling factor for each accumulated error signal, $n$ is the number of error signals being analyzed, $t_{\text{real}}$ is the elapsed time, and $t_{\text{sim}}$ is the simulation length.

The MATLAB function `fmincon` from the optimization toolbox [25] is used to find the argument of the minimum (argmin) of the cost function, where the arguments are $K_{\text{filter}}$ and $t_{\text{dwell}}$.

Given that the cost function is in two arguments, the value of the cost function can be plotted in 2D, as shown in Figure 3.15. The minimal cost occurs for a $K_{\text{filter}} = 0.3$ and $t_{\text{dwell}} = 80s$.

**Figure 3.15.** Cost associated with different values of $K_{\text{filter}}$ and $t_{\text{dwell}}$
Using these values for $K_{\text{filter}}$ and $t_{\text{dwell}}$ the finite volume model is simulated in a switched-fidelity framework and subjected to a wide range of disturbances. The result is a decrease of 56% in computational cost compared to the high-fidelity model (Figure 3.16). Quantitatively this means that for a 1000 second simulation, the high-fidelity model would take approximately 100 seconds, the low-fidelity model would take approximately 15 seconds, and the switched-fidelity model would take approximately 50 seconds to complete the simulation. Additionally, the switched-fidelity model has a 76% decrease in primary flow pressure error (Figure 3.17 top), and 69% decrease in secondary flow temperature error (Figure 3.17 bottom).

Figure 3.16. Computational speed for low, high, and switched-fidelity models

Figure 3.17. Accumulated pressure and temperature errors between the low and switched-fidelity models
3.5. Variable Fidelity of Closed Loop Dynamical Systems

The previous section described the application of the variable fidelity modeling to a single system component. In the following subsections the application of the variable fidelity framework to a dynamical system is investigated.

3.5.1. System Description

For the case study presented in this section, a vapor compression system (VCS) is dynamically modeled and simulated in the Simulink® environment using the Thermosys toolbox [23] from the University of Illinois at Urbana-Champaign (UIUC). This toolbox uses a modular approach wherein each component is modeled independently. The heat exchangers use a lumped parameter moving boundary approach to model the three refrigerant fluid zones in the condenser (superheat, two-phase, subcooled) and the two refrigerant fluid zones in the evaporator (two-phase, superheat). For high-fidelity modes of simulation, finite volume models of the heat exchangers are used, where the heat exchanger is divided into \( N \) different volumes. In this analysis \( N = 25 \). The heat exchanger models calculate the system pressures and the compressor and valve models calculate the refrigerant mass flow rates.

The VCS is modeled using the traditional four components with four actuator inputs: compressor speed, electronic expansion valve opening, and heat exchanger fan speeds (Figure 3.18). Maximum cooling capacity is 1.4kW, representing a relatively small VCS (comparable to a window unit). The system uses R-134a, has tube-and-fin heat exchangers, and a \( \frac{1}{2} \) horsepower reciprocating compressor. The model is calibrated using a UIUC experimental system.

The air being blown over the evaporator, referred to as the secondary flow, is used to cool a thermal zone, whose dynamics are defined by,

\[
T_{zone} = \frac{1}{C} \int \left( \dot{Q}_{in} + \dot{Q}_{dist} - \dot{Q}_{out} \right) dt
\]  

where \( \dot{Q} \) is the heat transfer rate in, out, and of the disturbance. The thermal capacity, \( C \), is set as 15J/K. This represents a relatively small thermal zone that is comparable to an electronics bay on board an aircraft or automobile.

Control of the simulated system consists of two PI controllers that independently control the temperature in a thermal zone, \( y_1 \), and the evaporator superheat, denoted as \( y_2 \) and defined in
(3.5.2) as the difference between the outlet and saturation temperatures in the evaporator. Compressor speed, \( u_1 \), is used to regulate the temperature of the thermal zone and EEV opening, \( u_2 \), is used to regulate superheat.

\[
T_{\text{super}} = T_{e,\text{out}} - T_{e,\text{sat}}.
\]

Figure 3.18. Four component vapor compression system configuration

This setup replicates systems found in aerospace and automotive applications that demand a tight tolerance on zone temperatures and experience sudden changes in thermal loads due to rapidly changing electrical loads. These sudden fluctuations in the reference set points or disturbances will quickly affect the small thermal zone, causing the temperature to change. Therefore, for successful controller and component design, it is necessary to accurately account for the transients in the system, and thus the need for high-fidelity models.

3.5.2. Supervisor Setup for Variable Fidelity System Modeling

Knowing the disturbance, \( d(t) \), and reference signals, \( r(t) \), prior to simulation of the model, the supervisor is set up to sample these signals in order to determine if a switch in fidelity is necessary. The signals are delayed being sent to the system by \( z^{-n} \), giving the supervisor time to determine if a switch is necessary. In the event that the supervisor signals a switch to high-fidelity mode, then the delay also provides enough time to settle out switching transients in the high-fidelity model.
Within the grey box of Figure 3.19, the high and low-fidelity models are represented as $g_1(t)$ and $g_2(t)$, while the remaining system dynamics and controller are represented as $h(t)$. The supervisor controls the switching of the input $u(t)$ and the output $y(t)$. Output from the system, $y$, is either from the output of the low-fidelity system model, $y_l$, or output from the high-fidelity system model, $y_h$. When the model is initialized, both high and low-fidelity models are switched on to allow initial condition transients to settle and the system to come to steady state. Then the supervisor switches the system to low-fidelity mode and is operating in the top process of Figure 3.20. This continues until output from the supervisor switches from 0 to 1.

![Figure 3.19. Configuration of Reference and Disturbance Signal Propagation to Supervisor and System](image)

When the supervisor output switches from 0 to 1 (top decision block of Figure 3.20), the time is denoted as $t = t_{\text{switch}}$, and the high-fidelity model is initialized, but $y_{\text{sys}} = y_l$ for
$t \in [t_{\text{switch}}, t_{\text{switch}} + z^{-n})$, depicted as the third process block of Figure 3.20. Because the disturbance or reference change has not yet affected the system, the output can remain as $y_l$ so that switching transients in $y_h$ are not captured in the system outputs. Once the delay has passed and $t = t_{\text{switch}} + z^{-n}$, then $y_{\text{sys}} = y_h$ for $t \in [t_{\text{switch}} + z^{-n}, t_{\text{switch}} + z^{-n} + t_{\text{dwell}}]$, where $t_{\text{dwell}}$ is the dwell time determined from Section 3.4. At $t = t_{\text{switch}} + z^{-n} + t_{\text{dwell}}$, the supervisor output will return to zero, $y_{\text{supervisor}} = 0$, and the simulation would exit the bottom decision loop of Figure 3.20.

When executing in high-fidelity mode, the low-fidelity model continues to execute and be subjected to the same input as the high-fidelity model so that the low-fidelity model does not need to be reset. At the switch back to low-fidelity mode, model outputs are adjusted.

![Switched-fidelity logic tree](image_url)

**Figure 3.20. Switched-fidelity logic tree**
3.5.3. Simulation Case Study for Variable Fidelity System Modeling

The vapor compression system as described previously is implemented into the switched-fidelity framework as shown in Figure 3.21.

The supervisor monitors the thermal zone reference temperature setpoints, \( r(t) \), and the thermal zone heat load disturbances, \( d(t) \). Both signals are delayed being sent to their respective system components by a delay of 10 seconds, as determined by heuristic methods previously described. The controller, \( C(t) \), receives the delayed reference set point and the thermal zone receives the delayed disturbance. Output from the supervisor controls the switching of inputs and outputs from the high and low-fidelity evaporators and condensers as per the previous section description.

In Figure 3.21 the gray box encompasses the entire system model. Thermal zone temperature is passed to the controller in addition to the delayed reference signal. In order to regulate thermal zone temperature and evaporator degree of superheat, the controller passes actuator inputs to the compressor and EEV. Output from the evaporator secondary fluid flow cools the thermal zone.

![Figure 3.21. Four component vapor compression system configuration](image)

*Figure 3.21. Four component vapor compression system configuration*
Exogenous reference and disturbance profiles are designed as shown in Figure 3.22. The reference temperature is being changed for the thermal zone and is controlled by adjusting the speed of the compressor. The disturbance heat load affects the thermal zone and has to be rejected by the VCS. By monitoring the temperature of the thermal zone, the controller adjusts the speed of the compressor and EEV opening to regulate temperature.

![Reference Temperature and Disturbance Heat Load Profiles](image)

Figure 3.22. Reference temperature (top) and disturbance heat load (bottom) profiles for simulation case study

A comparative study between independent high and low-fidelity system models shows that the level of model fidelity results in differing control signals being sent to the compressor and EEV. There exists a difference in transients as well as steady state bias, and this is demonstrated in Figure 3.23. By only using the low-fidelity model for controller design, it is clear that the resulting controller would not operate as it would if it were designed with the high-fidelity model.

Figure 3.24 depicts the effect of regulation by the controller on the two input actuator signals. The normalized $u_1$ corresponds to the compressor speed and is the control action for regulating
the room temperature, represented as \( y_1 \) in Figure 3.25. The normalized \( u_2 \) corresponds to the EEV opening and is the control action for regulating superheat in the VCS, represented as \( y_2 \) in Figure 3.25. Inputs are normalized by actuator saturation values. For the first input, it is observed that the low-fidelity model has different transient time constants and has a large bias during steady state after transient responses to disturbances. Comparatively, the second input has a larger steady state offset and improved transient response. In the case of both inputs, the switched system is capable of tracking the high-fidelity model with much better precision.

The overarching goal of the controller in all three models (high, low, switched) is to regulate the zone temperature, \( y_1 \), and the evaporator superheat, \( y_2 \). The zone temperature should track the reference temperature (top graph of Figure 3.22) and regulate the superheat at 10°C. Even though controller inputs in Figure 3.24 differ between the three modeling approaches, Figure 3.25 shows how both control objectives are still met during each of the three simulation configurations, with the exception of transients and the effect of random heat loads in the thermal zone.

Outputs from each of the three simulations are discretized at the same time sample for the entirety of the simulation. The root-mean-square error (3.5.3) in the actuator input values is calculated comparing the switched and low-fidelity model inputs, \( \hat{u} \), to the high-fidelity model inputs, \( u \), for \( n \) samples. The results for the two input signals are presented in Table 3.1.

\[
RMSE = \sqrt{\frac{\sum_{t=1}^{n} (\hat{u} - u)^2}{n}}
\]  

(3.5.3)

<table>
<thead>
<tr>
<th>Input:</th>
<th>( u_1 )</th>
<th>( u_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low-fidelity</td>
<td>54.9</td>
<td>3.05</td>
</tr>
<tr>
<td>Switched-fidelity</td>
<td>6.68</td>
<td>0.70</td>
</tr>
</tbody>
</table>

For \( u_1 \) it is observed that the switched-fidelity model is capable of capturing the same system phenomena as the high-fidelity model much better than the low-fidelity model. This is apparent
by the order of magnitude difference between RMS errors. Cross-comparing signals in Figure 3.24, it is clear that the difference between the high-fidelity and low-fidelity control inputs for $u_1$ is much greater than it is between the high-fidelity and switched-fidelity control inputs.

A similar case is made for $u_2$ as it is observed that the switched-fidelity model is capable of better matching the high-fidelity model actuator inputs. However, the relative difference is much closer than $u_1$, and can be observed by Figure 3.25.

The trade off with utilizing a high-fidelity model during simulation is the increased computational cost, in this case the execution time for the simulation. Figure 3.26 compares the time ratios for the three different simulation configurations. Each time is relative to the fastest executing model, the low-fidelity model. It is observed that the high-fidelity model requires 182% more time to execute the same simulation as the low-fidelity model. However, if a switched-fidelity framework is implemented, the computational cost is 37% more than the low-fidelity model.

![Figure 3.23. Normalized error between high and low-fidelity system inputs $u_1$ and $u_2$](image-url)
Figure 3.24. Normalized system inputs $u_1$ and $u_2$ time history for simulation case study

Figure 3.25. Tracking of system outputs $y_1$ and $y_2$ for simulation case study
Figure 3.26. Comparison of time ratios for low, high, and switched-fidelity models
Chapter 4
Model Predictive Control

4.1. Overview

Model predictive control (MPC) is a receding-horizon optimal control framework which uses a dynamic model of a system to predict the response of the system to an input. By solving a finite-time horizon, open-loop, optimal control problem using the current measured state of the system, MPC determines a sequence of optimal control decisions that minimize the specified cost function over the prediction horizon. The first element of this sequence is then applied to the system and the process is repeated at discrete intervals. This causes the prediction horizon to be continually shifted forwards and for this reason MPC is referred to as receding horizon control. Although this process is not the most optimal, it has the ability to provide very good results and is widely used in industrial applications [26]. The stability and robustness of MPC has been thoroughly researched and is available in literature [27][28].

Model predictive control is a prominent method of control because of its ability to consider hard constraints on inputs, outputs, and states of a system, perform multi-input multi-output (MIMO) control, and utilize a large array of cost functions. Both linear and nonlinear MPC formulations and solvers are available and documented in literature [28][29][30], but only linear MPC will be considered for the work in this thesis because of its ability to approximate the anticipated system behavior while having significantly lower computational costs compared to nonlinear MPC.

The concept of model predictive control and the receding-horizon framework is demonstrated in Figure 4.1. At the current time \( k \), the finite-horizon control problem consists of \( N_p \) time samples each of size \( \Delta t \), where \( N_p \) is known as the prediction horizon. For all steps in the prediction horizon, the model states and outputs are predicted as a function of the predicted control inputs. Similarly, the control horizon \( N_u \) represents the number of future time steps for which control decisions are decided when solving the optimization problem. Note that
\( N_u \leq N_p \), and in the event that \( N_u < N_p \) then the predicted input remains constant after the control horizon, or,

\[
u(k+N_u) = u(k+N_u + 1) = \ldots = u(k+N_p).
\] (4.1.1)

**Figure 4.1. A discrete model predictive control scheme**

### 4.2. Basic Formulation

A basic MPC formulation utilizes a discrete system model of the form,

\[
x[k+1] = Ax[k] + Bu[k] + Vd[k]
\]

\[
y[k] = Cx[k] + Du[k] + Wd[k]
\] (4.2.1)

where the states \( x \in \mathbb{R}^{n_x} \), the outputs \( y \in \mathbb{R}^{n_y} \), the inputs \( u \in \mathbb{R}^{n_u} \), and the disturbances \( d \in \mathbb{R}^{n_d} \). The matrices \( A, B, C, D, V \), and \( W \) are selected to properly represent the system dynamics and outputs while being of appropriate dimensions. By using this model, the future states \( x[k+j|k] \) can be predicted given a future control sequence \( u[\cdot|k] \) and the current state.
The generic MPC formulation then uses these future states and control sequences to solve the following control problem,

\[
\min_{u[\cdot | k]} \ J = \sum_{j=k}^{k+N_p-1} f (x[j | k], y[j | k], u[j | k])
\]

subject to

\[
\begin{align*}
\forall k \in U,
\forall j \in \mathcal{X},
\forall j \in \mathcal{Y},
\forall j \in \mathcal{Y},
\end{align*}
\]

\[
x[k + j | k] = A x[k + j - 1 | k] + B u[k + j - 1 | k] + V d[k + j - 1 | k]
\]

where \( u[\cdot | k] \) is the set control inputs over the length of the control horizon, and \( U, \mathcal{X}, \) and \( \mathcal{Y} \) are the sets of admissible control inputs, state values, and system outputs, respectively. These admissible sets are what allows MPC controllers to consider hard constraints when solving the optimization problem.

It is ideal to solve this control problem with the cost function being written only as a function of the initial state \( x[k | k] \) and the control sequence over the horizon \( u[\cdot | k] \), which is possible by using the discrete model of the system (4.2.1) to express all future states and outputs of the system. This process is shown for an initial state \( x[k | k] \) at sample \( k \), for control and prediction horizons of \( N_u \) and \( N_p \), respectively, where the disturbance is assumed constant over the length of the prediction horizon,

\[
\begin{align*}
x[k + 1 | k] &= A x[k | k] + B u[k | k] + V d[k | k] \\
x[k + 2 | k] &= A^2 x[k | k] + A B u[k | k] + B u[k + 1 | k] + (A + I) V d[k | k] \\
&\vdots \\
x[k + N_u | k] &= A^{N_u} x[k | k] + \sum_{i=0}^{N_u-1} (A^{N_u-i-1} B u[k + i | k]) + (A^{N_u-1} + \ldots + A + I) V d[k | k] \\
&\vdots \\
x[k + N_p | k] &= A^{N_p} x[k | k] + \sum_{i=0}^{N_p-1} (A^{N_p-i-1} B u[k + i | k]) + (A^{N_p-1} + \ldots + A + I) V d[k | k]
\end{align*}
\]

This predicted system response can be represented in a lifted form,

\[
X = T x[k] + S U + R d[k],
\]

where
\[ X = [x[k+1] \ x[k+2] \ \cdots \ x[k+N_p]]^T \in \mathbb{R}^{(n_x \times N_p)} \]  
\[ U = [u[k+1] \ u[k+2] \ \cdots \ u[k+N_p]]^T \in \mathbb{R}^{(n_u \times N_u)} \]  
\[ T = [A \ A^2 \ \cdots \ A^{N_p}]^T \in \mathbb{R}^{(n_x \times N_p)} \]  
\[ R = [V \ (A+I)V \ \cdots \ (A^{N_p-1} + \cdots + A + I)V]^T \in \mathbb{R}^{(n_y \times n_y \times n_d)} \]  
\[ S = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ A^{N_u-1}B & \cdots & AB & B \\ \vdots & \vdots & \vdots & \vdots \\ A^{N_p-2}B & \cdots & \cdots & \sum_{i=N_u+1}^{N_p} A^{N_p-i}B \\ A^{N_p-1}B & \cdots & A^{N_p-N_u+1}B & \sum_{i=N_u}^{N_p} A^{N_p-i}B \end{bmatrix} \in \mathbb{R}^{(n_x \times n_u \times n_y \times n_d)} \]  

Note that the summation terms in (4.2.9) are the result of the control input remaining constant after the length of the control horizon.

The outputs of the system in lifted form can be represented as,
\[ Y = PX = \begin{bmatrix} y[k+1] \\ y[k+2] \\ \vdots \\ y[k+N_p] \end{bmatrix} \in \mathbb{R}^{(n_y \times N_p)} \]  
\[ P = \begin{bmatrix} C & 0 & \cdots & 0 \\ 0 & C & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & C \end{bmatrix} \in \mathbb{R}^{(n_y \times N_p \times n_x \times N_p)} \]  

Now the cost function from (4.2.2) can be expressed as,
\[ J = U^T H U + F^T U, \]  

(4.2.12)

where \( H \in \mathbb{R}^{(n_u \cdot N_u) \times (n_u \cdot N_u)} \), \( F \in \mathbb{R}^{(n_u \cdot N_u)} \) are positive definite weighting matrices and are functions of \( T, S, R, P, x[k | k] \), and \( d[k] \).

The constraints on the states and actuators can be represented in the lifted form as,

\[ X_{\min} \leq X \leq X_{\max} \]  

(4.2.13)

\[ U_{\min} \leq U \leq U_{\max}. \]  

(4.2.14)

4.3. MPC Algorithm Summary

Using the structure from the previous section, the basic MPC problem can be solved. The general algorithm used for MPC can be summed up as,

1. At time \( k \) measure the current state \( x[k | k] \),
2. Obtain the optimal control sequence \( u[\cdot | k] \) by solving (4.2.2),
3. Apply \( u[k] = u[k | k] \) (the first control input from the sequence is applied at the current time),
4. Wait for the time update, \( k := k + 1 \),
5. Repeat from step 1.

While this is a high-level explanation of how model predictive control can be implemented, there are multiple methods for determining and implementing solutions to the MPC problem that are discussed in literature [31].
Chapter 5
Hierarchical Energy and Power Management

In complex systems such as aircraft, there is a natural hierarchy to the system architecture. From the overall vehicle down to an individual component, an aircraft is composed of multiple levels of systems and subsystems, each doing its part to achieve an overall operating goal. This type of structure is shown graphically in Figure 5.1. At the highest level, the vehicle parameters and flight parameters directly influence how the overall vehicle will behave. Often these parameters are a function of the mission phase and are therefore known throughout the mission. The system-level introduces the major systems of an aircraft, namely the electrical power system (EPS), the thermal management system (TMS), and the flight control system (FCS). This thesis focuses on the TMS which are typically made up of vapor compression systems (VCSs), air cycle machines (ACMs), and fuel loops. Each of these subsystems is made up of components such as pumps, valves, fans, and compressors, which are capable of being actuated. The physical level captures the necessary signals, such as AC motor signals, for actuating these components.

Figure 5.1. Physical hierarchy of aircraft systems

By utilizing the natural hierarchy of aircraft systems, a hierarchical control strategy can be developed and implemented. Optimization can occur at each level but with different time-scales,
knowledge of the system, and the objectives being considered. Additionally, information should be allowed to flow up and down through the entire platform, but also side-to-side between systems or subsystems with similar objectives.

The proposed method in this thesis follows that of Table 5.1. The vehicle level monitors the overall vehicle state knowledge such as Mach number, altitude, location, and mission phase. Using a graph of the entire platform and simple models of each system, the vehicle level controller can coordinate system behavior by prioritizing the objectives of each system and setting an energy budget that each system must adhere to. Therefore, the top level of the control algorithm is responsible for overall vehicle energy consumption, but updates the optimization problem at the slowest time constant so that long periods of the mission can be taken into consideration.

System-level controllers communicate with the vehicle level to determine their respective energy budgets. As an example, the TMS budget could consist of periods of low power consumption with no heat dissipation (due to system constraints, stealth operation, etc.) or periods of unconstrained operation where it may be desirable to remove as much heat as possible for upcoming disturbances. The system-level controller then becomes responsible for coordinating setpoints within the system while operating within constraints. For the TMS this consists of determining where heat should be moved or stored within the platform. The system-level controller updates on the second slowest time constant. This allows it to receive updates from the vehicle level controller, implement them into system-level control decisions, and allow the subsystems to reach their steady state behavior. Ultimately this prevents fighting between the optimization levels.

Coordination at the subsystem-level occurs when multiple systems are required to tackle an objective set by the system-level controller. The subsystem-level controller is designed to optimize the operation of each subsystem while achieving the desired output as set by the system-level controller. This requires knowledge of the subsystem performance characteristics so that the controller can determine appropriate setpoints. Updates at this level occur faster than the system-level.

With the subsystem-level determining an operational set point, the component level determines the optimal actuator positions that achieve the set point. This can be done with pre-
determined lookup tables or models of the components. In the TMS this would consist of determining the optimal compressor speed and valve opening for the VCS in order to provide the desired amount of cooling. As such, the update rate of the components must be faster than the subsystem-level control so that delays in the system are negated.

The physical level is responsible for the actual signals being sent to the actuators, such as electrical PWM signals, switch positioning, voltages/current, etc. These signals can come from detailed models of the actuators or lookup tables, and will update at the fastest time constant in the control architecture.

<table>
<thead>
<tr>
<th>Table 5.1. Proposed 5-level hierarchical control strategy for aircraft</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Level</strong></td>
</tr>
</tbody>
</table>
| Vehicle | • Vehicle & mission state  
• Coordinate systems | Simple static models | 60+ sec |
| System | • Coordinate system setpoints  
• Operate within constraints  
• Monitor future events/disturbances | Simple first order dynamic models | 10-60 sec |
| Subsystem | • Optimize subsystem operation  
• Determine operational model of individual subsystems | Subsystem performance curves | 1-10 sec |
| Component | • Determine optimal setpoints for components | Simplified models or lookup tables | 0.1-1 sec |
| Physical | • Servo-level commands for adjusting component set-points | Detailed model of actuators or lookup tables | 0.01-0.1 sec |

5.1. Proposed Two-Level Hierarchical Control Strategy

In this thesis, the focus is on system and subsystem-level control strategies. Only the thermal system will be investigated, so it is assumed that the energy budget provided to the thermal system is uncapped for the entire mission. However, the system-level controller can easily be adapted to include current and future constraints on total energy consumption.

The following sections will refer to the thermal management units of an aircraft. These are the cooling systems responsible for the removal of heat from each thermal zone. Each cooling system’s ability to remove heat, in watts, is referred to as cooling capacity, represented as \( \dot{Q} \).
5.1.1. System-level Control Using Model Predictive Control

The system-level controller is formulated as a model predictive control optimization problem, because of MPC’s ability to use a system model and prediction horizon to minimize a cost function. As the prediction horizon moves through time, the optimization problem is solved recursively so that new control decisions can be made based on current states and constraints.

For the system-level controller of the thermal system, linear models are used to predict temperatures throughout the aircraft for the major thermal zones such as the cabin, electronics bays, cargo bays, and fuel tanks,

\[
T_{zone,i}[k + \tau_{s,1}] = \alpha T_{zone,i}[k] + \frac{\beta \tau_{s,1}}{C_{zone,i}} \left[ \dot{Q}_{in}[k] - \dot{Q}_{out}[k] \right]_{zone,i}
\]

(5.1.1)

where \( T_{zone,i} \) is the temperature in the \( i \)-th zone, \( \tau_{s,1} \) is the sample rate of the system-level controller, \( C_{zone,i} \) is the thermal capacitance of the \( i \)-th zone, and the quantity \( \dot{Q}_{in}[k] - \dot{Q}_{out}[k] \) is the net heat flow through the \( i \)-th zone. The coefficients \( \alpha \) and \( \beta \) are the first order system coefficients which can be determined using the time constant \( \tau_{sys} \) of a continuous system,

\[
\alpha = e^{-\tau_{s,1}/\tau_{sys}},
\]

(5.1.2)

\[
\beta = -\tau_{sys} \left( e^{-\tau_{s,1}/\tau_{sys}} - 1 \right).
\]

(5.1.3)

In the thermal system, the net heat flow through each thermal zone is a function of thermal loads and disturbances (\( \dot{Q}_L \)) and the controllable thermal management units (\( \dot{Q}_S \)). As such, the entire system can be modeled as a single state space representation,

\[
X[k + \tau_{s,1}] = AX[k] + BU[k] + VD[k]
\]

(5.1.4)

\[
X[k + \tau_{s,1}] = \begin{bmatrix} T_{zone,1}[k + \tau_{s,1}] \\ T_{zone,2}[k + \tau_{s,1}] \\ \vdots \\ T_{zone,n_z}[k + \tau_{s,1}] \end{bmatrix}
\]

\[
X[k] = \begin{bmatrix} T_{zone,1}[k] \\ T_{zone,2}[k] \\ \vdots \\ T_{zone,n_z}[k] \end{bmatrix}
\]

(5.1.5)
where \( n_z, n_s, \) and \( n_l \) are the number of thermal zones, thermal sources, and disturbances, respectively. The \( D \) vector contains the exogenous loads and disturbances. The \( U \) vector contains the decision variables that will be optimized in the MPC problem.

The selection of the \( A \) matrix is straightforward,

\[
A = \begin{bmatrix}
\alpha_1 & 0 & \cdots & 0 \\
0 & \alpha_2 & 0 & \vdots \\
\vdots & \vdots & \ddots & 0 \\
0 & \cdots & 0 & \alpha_{n_z}
\end{bmatrix};
\]

however, the \( B \) and \( V \) matrices must be selected appropriately to match the system configuration between thermal zones, sources, and disturbances. These matrices should be a function of the system-level sample rate and capacitances of each zone. Section 6.3 provides an example of these matrices given a system configuration.

The cost function for the system-level controller seeks to meet a performance objective, \( J_{\text{perf}} \), while minimizing energy consumption, \( J_{\text{eff}} \). An example of this in the thermal system would be to regulate the temperature of thermal zones, \( J_{\text{perf}} \), while minimizing the cooling capacity provided by each of the thermal management units, \( J_{\text{eff}} \). An example of such a cost function is,

\[
J = J_{\text{perf}} + J_{\text{eff}} = \sum_{i=0}^{N_p} \sum_{j=1}^{n_z} \hat{Q}_{S_j}^2(k+i) + \sum_{i=0}^{N_p} \sum_{j=1}^{n_z} \left( T_{\text{zone},j}^* - T_{\text{zone},j}(k+i) \right)^2
\]

where the efficiency objective attempts to minimize the summation of all cooling capacities provided by the system over the prediction horizon \( (N_p) \), while maintaining the temperatures of the thermal zones near the set points, \( T_{\text{zone},j}^* \). This form of performance objective imitates proportional control of the zone temperatures. It may be desired to incorporate integral control.
to drive steady state temperature errors to zero. In which case, (5.1.8) can be adjusted to include the summation of the temperature error,

\[
J = \sum_{i=0}^{N_p} \sum_{j=1}^{n_x} \hat{Q}^2_{S_i} [k+i] + \sum_{i=0}^{N_p} \sum_{j=1}^{n_x} \left( T_{zone,j}^* [k] - T_{zone,j} [k+i] \right)^2 \\
+ \sum_{i=0}^{N_p} \sum_{j=1}^{n_x} \left( \sum_{m=0}^{0} \left( T_{zone,j}^* [k-m] - T_{zone,j} [k+i-m] \right)^2 \right)
\]

(5.1.9)

Since the MPC is discrete, the integral control is implemented by using \( m \) previous values of the temperature deviation from the set point. The value for \( m \) can vary throughout time in order to “reset” the integral term of the cost function.

The final step of the system-level controller is to solve the MPC problem and implement the first control decision in the sequence. The system-level controller for the thermal system determines the desired cooling capacity that should be delivered to each thermal zone, and where that heat should be dumped. An example of such an MPC problem with upper and lower constraints on the temperatures and cooling capacities is,

\[
\text{minimize} \quad J \\
\text{subject to} \quad T_{min, zone,i} \leq T_{zone,i} [k] \leq T_{max, zone,i} \quad \forall i \& k \\
\hat{Q}_{min, Si} \leq \hat{Q}_{Si} [k] \leq \hat{Q}_{max, Si} \quad \forall i \& k
\]

In this situation, it is assumed that \( D[k] \) through \( D[k + N_p] \) are known at sample \( k \), allowing the MPC controller to preview the upcoming disturbances.

### 5.1.2. Subsystem Operating Mode Selection Using Mixed Integer Quadratic Programming

The goal of the subsystem-level controller is to determine the optimal resource allocation from each thermal management unit in order to meet the demands set by the system-level controller. For the thermal system, this optimal combination could include turning some thermal management units to on/off/stand-by modes. Because of these specific modes, the subsystem-
level controller is formulated as a mixed integer quadratic programming (MIQP) problem. The major difference between MIQP and MPC is the introduction of binary decision variables into the optimization problem. These binary variables can be used for determining operating modes of each thermal management unit.

An additional goal of the subsystem-level controller of the thermal system is to minimize the total power consumption of each thermal management unit. In order to do so, the controller needs to know the relationship between power consumption and cooling capacity for each thermal management unit. Often these systems will have quadratic operating curves similar to that of Figure 5.2 where the continuous operating region requires substantially more power input for each incremental amount of cooling capacity. Additionally, the system can operate in a stand-by mode with little power consumption and minimal to no cooling capacity being delivered.

Mathematically this relationship can be represented as a function of the three distinct points in Figure 5.2. For each \(i\)-th thermal management unit the cooling capacity, \(\dot{Q}_i\), delivered and the corresponding rate of work (power), \(\dot{W}_i\), can be determined,

\[
\dot{Q}_i = (\dot{Q}_{3,i} - \dot{Q}_{2,i})u_{c,i} + (\dot{Q}_{2,i} - \dot{Q}_{1,i})u_{b,i} + \dot{Q}_{1,i}
\]

\[
\dot{W}_i = (\dot{W}_{3,i} - \dot{W}_{2,i})u_{c,i}^2 + (\dot{W}_{2,i} - \dot{W}_{1,i})u_{b,i} + \dot{W}_{1,i}
\]

where \(u_{c,i} \in [0,1]\) and \(u_{b,i} \in \{0,1\}\) are continuous and Boolean decision variables. Note that this relationship only holds true for systems with quadratic operating curves. If the system is
represented by some other convex relationship, then the quantities \(( \dot{Q}_{3,i} - \dot{Q}_{2,i} ) u_{c,i}\) and 
\(( \dot{W}_{3,i} - \dot{W}_{2,i} ) u_{c,i}^2\) would need to be replaced accordingly.

With these models of the individual thermal management units, the optimal control problem can be formulated. Since the goal of the subsystem-level controller is to determine the most efficient allocation of resources among the available thermal management units, given a desired cooling capacity, the cost function can be designed with a performance term, \( J^{\text{perf}} \), that helps to meet the desired cooling demanded, and an efficiency term, \( J^{\text{eff}} \), that attempts to minimize power consumption,

\[
\text{minimize } J = \gamma \left( N_p \cdot \dot{Q}_{\text{tot,des}} - \sum_{i=0}^{N_p} \sum_{j=1}^{n_i} \dot{Q}_{S_j} [k + i] \right)^2 + (1 - \gamma) \sum_{i=0}^{N_p} \sum_{j=1}^{n_i} \dot{W}_{S_j} [k + i]
\]

subject to \( u_{b,i} \in \{0,1\} \ \forall i \)
\( 0 \leq u_{c,i} \leq u_{b,i} \ \forall i \)

where \( \gamma \) is the weighting factor between \( J^{\text{perf}} \) and \( J^{\text{eff}} \). \( \dot{Q}_{\text{tot,des}} \) is the total cooling capacity desired by the system-level controller, and \( \dot{W}_{S_j} \) is the rate of power consumption for the \( j \)-th thermal management unit. Note that in the case where \( u_{b,i} = 0 \) then \( u_{c,i} = 0 \) must hold so that the system would be operating in stand-by mode.

One final concern of the subsystem-level controller is switching frequency, or having the thermal management units cycle between continuous and stand-by mode. This behavior is undesirable because it can reduce the life of components, decrease system response time, and cause noticeable swings in the cooling supplied. As such, a penalty on switching can be added to the cost function,

\[
J = \gamma \left( N_p \cdot \dot{Q}_{\text{tot,des}} - \sum_{i=0}^{N_p} \sum_{j=1}^{n_i} \dot{Q}_{S_j} [k + i] \right)^2 + (1 - \gamma) \sum_{i=0}^{N_p} \sum_{j=1}^{n_i} \dot{W}_{S_j} [k + i] + \gamma_{\text{switch}} \sum_{i=0}^{N_p} \sum_{j=1}^{n_i} \Delta u_{b,i}
\]

(5.1.13)
Chapter 6

Case Study Using an Aircraft Thermal System

To show the capabilities of the hierarchical control strategy presented in the previous chapter, a case study using an aircraft thermal system is conducted.

6.1. Example System Configuration

In order to accurately represent a real-world application of the proposed controller, the example system consists of two thermal zones, two thermal management units, three thermal loads, and two thermal sinks. Figure 6.1 shows the layout of the example system. From left to right, the system consists of two thermal loads that affect the temperature of the passenger cabin. One load simulates a constant heat load, while the second simulates a pulse load that could come from electronics or on-board systems. An ACM and VCS are capable of removing heat from the passenger cabin and dumping it to the fuel tank via heat exchangers (HX). The fuel tank is subjected to a heat load that varies throughout the mission. Finally the ACM and VCS dump additional heat to ram air heat exchangers and the fuel tank dumps heat to the bypass duct air.

![Diagram of the example system layout](image)

**Figure 6.1.** Example system layout containing three loads, two thermal zones, two thermal management units, and two thermal sinks
The ACM and VCS require additional loops and valves in order to route heat to the fuel tank and ram air. For the ACM this can be achieved with parallel loops, one with an air-liquid heat exchanger to dump heat to the fuel tank, and another loop with an air-air heat exchanger to dump heat to the ambient air. The VCS could utilize multiple condensers either in parallel or series.

With this thermal system configuration, the system-level controller becomes responsible for determining how much heat should be removed from the passenger cabin and fuel tank. The subsystem-level controller is responsible for determining the amount of heat each thermal management unit removes from the passenger cabin and the ratio of that heat that goes to the fuel tank. The system-level controller is unaware that multiple subsystems are capable of removing the heat from the passenger cabin.

### 6.2. Mission Profile and System Parameters

Aircraft operate in a wide range of environments during a typical mission profile. As such, many subsystem parameters are a function of the mission phase and directly affect the performance of each system. Those limitations are taken into consideration in the constraints of the optimization problem.

For the case study in this thesis, a two-hour mission profile (Figure 6.2) was designed consisting of six phases: ground and taxi operations, high powered takeoff, climb, cruise, descent, landing and taxi. Since the longest portion of a mission is the cruise phase, increasing the total mission length will have little effect on how different the controller performs.

![Figure 6.2. Mission profile for simulation case study](image-url)
All values selected for thermal loads and system parameters are scaled relative to each other and are not representative of a real platform. As such, units for $\dot{Q}$ and $\dot{W}$ are not presented.

Throughout the mission the fuel tank and aircraft cabin are subjected to loads mimicking the layout shown in Figure 6.1. On the ground an aircraft will experience larger thermal loads due to warmer conditions than at altitude. Once the aircraft is airborne the loads typically drop off, but may spike at times throughout the mission. The time histories for the thermal loads used in the simulation case study are shown in Figure 6.3. Shading boundaries correspond to the different mission phases of Figure 6.2. Note that the load peaks during the cruise phase are designed to exceed cooling capabilities of the thermal management system, and is addressed in later sections.

![Graphical representation of heat loads into the cabin and fuel tank for the simulation case study](image)

**Figure 6.3. Profile of heat loads into the cabin and fuel tank for the simulation case study**

The thermal management system has physical limitations on the amount of heat that it can dissipate into the ambient air ($\dot{Q}_{amb}$), the fuel tank ($\dot{Q}_{tank}$), and via ram air ($\dot{Q}_{ram}$). These limitations are a function of the mission phase, and follow the logic that it is much more difficult to reject heat to the ambient air at ground level than it is at 35,000 feet. Figure 6.4 provides a graphical representation of the maximum $\dot{Q}$ that can be sent to the ambient air, fuel tank, and ram air during each mission phase. Table 6.1 provides the data in tabular form. Recall from Figure 6.3 that at times during the mission loads may exceed cooling capacity capability.

In addition to physical constraints on the system, there are comfort constraints placed on the passenger cabin. These are shown graphically in Figure 6.5 and in tabular form in Table 6.1.
The nominal temperature value is used for regulation of cabin temperature. The fuel tank temperature has a maximum of 45°C for the entire mission profile. It is assumed that the fuel pumped into the aircraft at the beginning of the mission is pre-cooled to 15°C.

![Figure 6.4: Maximum $\dot{Q}$ capable of being dumped via ambient air, fuel tank, and ram air](image)

![Figure 6.5: Nominal temperature and upper/lower constraints of the cabin for mission phase](image)

Table 6.1. Constraints on the thermal system during sample mission

<table>
<thead>
<tr>
<th></th>
<th>Max $\dot{Q}_{\text{tank}}$</th>
<th>Max $\dot{Q}_{\text{amb}}$</th>
<th>Max $\dot{Q}_{\text{ram}}$</th>
<th>Max $T_{\text{cabin}}$</th>
<th>Min $T_{\text{cabin}}$</th>
<th>Max $T_{\text{tank}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground/Taxi</td>
<td>10</td>
<td>6</td>
<td>5</td>
<td>24</td>
<td>20</td>
<td>45</td>
</tr>
<tr>
<td>Takeoff</td>
<td>10</td>
<td>6</td>
<td>5</td>
<td>25</td>
<td>20</td>
<td>45</td>
</tr>
<tr>
<td>Climb</td>
<td>10</td>
<td>6</td>
<td>5</td>
<td>23</td>
<td>20</td>
<td>45</td>
</tr>
<tr>
<td>Cruise</td>
<td>12</td>
<td>10</td>
<td>8</td>
<td>21.5</td>
<td>20</td>
<td>45</td>
</tr>
<tr>
<td>Descent</td>
<td>10</td>
<td>8</td>
<td>7</td>
<td>23</td>
<td>20</td>
<td>45</td>
</tr>
<tr>
<td>Landing/Taxi</td>
<td>10</td>
<td>6</td>
<td>6</td>
<td>25</td>
<td>20</td>
<td>45</td>
</tr>
</tbody>
</table>
Recall from section 5.1.2 that the thermal management units are assumed to function optimally along a quadratic operating curve like that of Figure 6.6. The configuration being used for the simulation case study (Figure 6.1) requires an adaptation to that operating curve in order to account for loops dissipating heat to the fuel tank and ambient air.

![Diagram](image)

**Figure 6.6. Tradeoff between cooling capacity and power consumption for thermal management unit**

The approach for the new curves assumes that each thermal management unit can dissipate heat in isolation to the fuel tank or ambient air with an optimal performance curve similar to Figure 6.6. When both loops are active, then the summation of those individual performance curves yields the combined operational performance curve. For this work, it is assumed that the systems will operate along these curves, but ultimately it will be the component level controllers responsible for ensuring that is the case. In preparation for that, seven operation regions are defined. The modes are as follows:

1. Heat dissipation to ambient air only
2. Heat dissipation to the fuel tank only
3. Heat dissipation to the fuel tank only or ambient air only
4. Heat dissipation to ambient air only, or some combination of ambient air and fuel tank
5. Heat dissipation to the fuel tank only, or some combination of ambient air and fuel tank
6. Heat dissipation to the fuel tank only, ambient air only, or some combination of both
7. Heat dissipation to the fuel tank and ambient air
The performance regions are presented for the vapor compression system and air cycle machine in Figure 6.7. Minkowski addition defines the summed regions. The individual points used for the creation of these operating curves are provided in tabular form in Table 6.2.

![Figure 6.7. Operating regions and performance curves for the VCS and ACM](image)

### Table 6.2. Performance parameters for the ACM and VCS during sample mission

<table>
<thead>
<tr>
<th></th>
<th>VCS To Tank</th>
<th>VCS To Ambient</th>
<th>ACM To Tank</th>
<th>ACM To Ambient</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\hat{\dot{Q}}_1)</td>
<td>0.1</td>
<td>0.1</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>(\hat{\dot{Q}}_2)</td>
<td>1.2</td>
<td>3.0</td>
<td>1.2</td>
<td>3.0</td>
</tr>
<tr>
<td>(\hat{\dot{Q}}_3)</td>
<td>6.0</td>
<td>8.0</td>
<td>3.0</td>
<td>8.0</td>
</tr>
<tr>
<td>(\hat{\dot{W}}_1)</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>(\hat{\dot{W}}_2)</td>
<td>2.0</td>
<td>5.0</td>
<td>2.5</td>
<td>6.0</td>
</tr>
<tr>
<td>(\hat{\dot{W}}_3)</td>
<td>2.5</td>
<td>6.0</td>
<td>2.0</td>
<td>5.0</td>
</tr>
</tbody>
</table>

### 6.2.1. Dynamic Model of Thermal System

A continuous linear model is used to represent the simulated temperature dynamics of the cabin and fuel tank. Thermal capacitance of the fuel tank, \(C_{\text{tank}} = 2000\), and thermal capacitance of the cabin, \(C_{\text{cab}} = 800\), are selected to represent the magnitude difference between the capacitance of a fuselage and fuel tank. The air in the fuselage has a thermal capacitance that is much lower than that of fuel. The heat loads from Figure 6.3 are represented by...
with the cabin being subjected to the first two loads. Currently only integrator dynamics are considered for the continuous time model.

A continuous-time state space model is used to represent the system dynamics,

\[
\begin{bmatrix}
\dot{T}_{\text{tank}} \\
\dot{T}_{\text{cab}}
\end{bmatrix} =
A 
\begin{bmatrix}
T_{\text{tank}} \\
T_{\text{cab}}
\end{bmatrix}
+ B
\begin{bmatrix}
\dot{Q}_{\text{tank}} \\
\dot{Q}_{\text{amb}} \\
\dot{Q}_{\text{ram}} \\
\dot{Q}_{L1}
\end{bmatrix}
+ W
\begin{bmatrix}
\dot{Q}_{L1} \\
\dot{Q}_{L2} \\
\dot{Q}_{L3}
\end{bmatrix},
\]

(6.2.1)

Where,

\[
A = 
\begin{bmatrix}
0 & 0 \\
0 & 0
\end{bmatrix}
\]

\[
B = 
\begin{bmatrix}
1/C_{\text{tank}} & 0 & -1/C_{\text{tank}} \\
-1/C_{\text{cab}} & -1/C_{\text{cab}} & 0
\end{bmatrix}
\]

(6.2.2)

\[
W = 
\begin{bmatrix}
0 & 0 & 1/C_{\text{tank}} \\
1/C_{\text{cab}} & 1/C_{\text{cab}} & 0
\end{bmatrix}
\]

6.3. System-level MPC Formulation

The system-level controller is based upon the framework laid out in Section 5.1.1. Sample code is provided in Appendix C where it is programmed in MATLAB/Simulink [7] using the Yalmip toolbox [32].

The system-level controller is responsible for determining the heat transfer rates out of the cabin and into either the fuel tank or ambient air. It also determines how much heat should be dissipated from the fuel tank via ram air. The objective is to minimize total cooling capacity while keeping temperatures within constraints. Temperature can also be regulated to a set point. A single formulation of the cost function is required to handle both scenarios.

The model approximates the temperatures in the fuel tank and cabin using a discrete state-space model,

\[
x[k + \tau_s] = Ax[k] + Bu[k] + Wd[k]
\]

(6.3.1)

Where,

\[
x[k] = 
\begin{bmatrix}
T_{\text{tank}}[k] \\
T_{\text{cab}}[k]
\end{bmatrix};
\quad u[k] = 
\begin{bmatrix}
\dot{Q}_{\text{tank}}[k] \\
\dot{Q}_{\text{amb}}[k] \\
\dot{Q}_{\text{ram}}[k] \\
\dot{Q}_{L1}[k]
\end{bmatrix};
\quad d[k] = 
\begin{bmatrix}
\dot{Q}_{L2}[k] \\
\dot{Q}_{L3}[k]
\end{bmatrix}
\]

(6.3.2)
\[ A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \]

\[ B = \begin{bmatrix} \frac{\tau_s}{C_{\text{tank}}} & 0 & -\frac{\tau_s}{C_{\text{tank}}} \\ -\frac{\tau_s}{C_{\text{cab}}} & -\frac{\tau_s}{C_{\text{cab}}} & 0 \end{bmatrix} \]  \hspace{1cm} (6.3.3)

\[ W = \begin{bmatrix} 0 & 0 & \frac{\tau_s}{C_{\text{tank}}} \\ \frac{\tau_s}{C_{\text{cab}}} & \frac{\tau_s}{C_{\text{cab}}} & 0 \end{bmatrix} \]

with a sample time, \( \tau_s = 60 \text{sec} \), tank capacitance, \( C_{\text{tank}} = 2000 \), and cabin capacitance, \( C_{\text{cab}} = 800 \). The state matrix is augmented to include accumulated error and set point for each temperature. Each state-space matrix is also augmented to include the four additional states,

\[
\begin{bmatrix}
T_{\text{tank}}[k + \tau_s] \\
T_{\text{cab}}[k + \tau_s] \\
\sum \Delta T_{\text{tank}} [k + \tau_s] \\
\sum \Delta T_{\text{cab}} [k + \tau_s] \\
T_{\text{tank, setpoint}}[k + \tau_s] \\
T_{\text{cab, setpoint}}[k + \tau_s]
\end{bmatrix} = \bar{A}
\begin{bmatrix}
T_{\text{tank}}[k] \\
T_{\text{cab}}[k] \\
\sum \Delta T_{\text{tank}} [k] \\
\sum \Delta T_{\text{cab}} [k] \\
T_{\text{tank, setpoint}}[k] \\
T_{\text{cab, setpoint}}[k]
\end{bmatrix} + \bar{B}
\begin{bmatrix}
\dot{Q}_{\text{tank}}[k] \\
\dot{Q}_{\text{amb}}[k] \\
\dot{Q}_{\text{ram}}[k] \\
\dot{Q}_{L1}[k] \\
\dot{Q}_{L2}[k] \\
\dot{Q}_{L3}[k]
\end{bmatrix} + \bar{W}
\begin{bmatrix}
\tau_s/C_{\text{tank}} \\
-\tau_s/C_{\text{tank}} \\
\tau_s/C_{\text{cab}} \\
-\tau_s/C_{\text{cab}} \\
0_{4 \times 3}
\end{bmatrix}
\]  \hspace{1cm} (6.3.4)

\[
\bar{A} = \begin{bmatrix}
0.99 & 0 & 0_{2 \times 2} & 0_{2 \times 2} \\
0 & 0.99 & 0_{2 \times 2} & 0_{2 \times 2} \\
0_{2 \times 2} & 0_{2 \times 2} & I_2 & -I_2 \\
0_{2 \times 2} & 0_{2 \times 2} & -I_2 & I_2
\end{bmatrix}
\]  \hspace{1cm} (6.3.5)

\[
\bar{B} = \begin{bmatrix}
\tau_s/C_{\text{tank}} & 0 & -\tau_s/C_{\text{tank}} \\
-\tau_s/C_{\text{cab}} & -\tau_s/C_{\text{cab}} & 0 \\
0_{4 \times 3} & 0_{4 \times 3}
\end{bmatrix}
\]  \hspace{1cm} (6.3.6)

\[
\bar{W} = \begin{bmatrix}
0 & 0 & \tau_s/C_{\text{tank}} \\
\tau_s/C_{\text{cab}} & \tau_s/C_{\text{cab}} & 0 \\
0_{4 \times 3} & 0_{4 \times 3}
\end{bmatrix}
\]  \hspace{1cm} (6.3.7)

Constraints for the system-level controller focus only on the whole system performance, and ignore the individual thermal management unit performance curves. As such, the decision variables are constrained (6.3.8) by the performance characteristics of Figure 6.4 depending upon
the current mission phase, and the temperature states are constrained (6.3.9) as per Figure 6.5 and Table 6.1. A slack variable is added to the constraints for the fuel tank temperature and cabin temperature. This variable allows the MPC solver to adjust constraints and keep the problem feasible when constraint violations are an issue due to initial conditions or disturbances. The augmented states are not constrained as they only affect the regulation of temperature.

\[
\begin{bmatrix}
\dot{Q}_{\text{tank,min}}[k] \\
\dot{Q}_{\text{amb,min}}[k] \\
\dot{Q}_{\text{ram,min}}[k]
\end{bmatrix} \leq \begin{bmatrix}
\dot{Q}_{\text{tank}}[k] \\
\dot{Q}_{\text{amb}}[k] \\
\dot{Q}_{\text{ram}}[k]
\end{bmatrix} \leq \begin{bmatrix}
\dot{Q}_{\text{tank,max}}[k] \\
\dot{Q}_{\text{amb,max}}[k] \\
\dot{Q}_{\text{ram,max}}[k]
\end{bmatrix}
\] (6.3.8)

\[
\begin{bmatrix}
T_{\text{tank,min}}[k] \\
T_{\text{cab,min}}[k]
\end{bmatrix} \leq \begin{bmatrix}
\sum \Delta T_{\text{tank}}[k] \\
\sum \Delta T_{\text{cab}}[k]
\end{bmatrix} \leq \begin{bmatrix}
T_{\text{tank,max}}[k] + T_{\text{slack}} \\
T_{\text{cab,max}}[k] + T_{\text{slack}}
\end{bmatrix}
\] (6.3.9)

The objective function focuses on minimizing the total cooling capacity delivered by the system while maintaining constraints and regulating temperatures,

\[
J = \sum_{i=0}^{N_p} \sum_{j=1}^{n_z} \dot{Q}_j^z[k+i] + \sum_{i=0}^{N_p} \sum_{j=1}^{n_z} (T_{\text{zone},j}^*[k] - T_{\text{zone},j}[k+i])^2
\]  
\[= \sum_{i=0}^{N_p} \sum_{j=1}^{n_z} \left( \sum_{m=0}^{n_z} (T_{\text{zone},j}^*[k-m] - T_{\text{zone},j}[k+i-m])^2 \right)
\] (6.3.10)

In the simulation case study, the prediction horizon, \(N_p = 10\), the total number of thermal management units, \(n_z = 2\), the number of thermal zones, \(n_z = 2\), and \(m\) is adjusted to the current value of \(k\) whenever \(T_{\text{zone},j} \leq T_{\text{zone},j}^*\), effectively resetting the accumulated error term.

The objective function is rewritten in matrix form,

\[
J = \sum_{i=0}^{N_p} \left( x[k+i]^T C x[k+i] + u[k+i]^T R u[k+i] - F x[k+i] \right)
\] (6.3.11)
where \( x[k] \) is the state vector, \( u[k] \) is the decision variable vector, \( C & R \) are penalty matrices for states and constraints, respectively, and \( F \) is a linear penalty on the states.

With the soft constraints on cabin and fuel tank temperatures, an extra term is added to (6.3.11) that penalizes the magnitude of the slack variable to prevent large constraint violations,

\[
J = \sum_{i=0}^{N_p} \left( \begin{array}{c} x[k] \end{array} \right)^T C \left( \begin{array}{c} x[k] \end{array} \right) + u[k]^T R u[k] - F x[k] + \gamma_{\text{slack}} T_{\text{slack}} \left( \begin{array}{c} x[k] \end{array} \right)^T T_{\text{slack}} \left( \begin{array}{c} x[k] \end{array} \right) \right) \]  

(6.3.12)

6.3.1. Tunable Parameters

The system-level controller as formulated above contains the following tunable parameters that will be investigated later in this chapter:

- \( N_p \) - prediction horizon for the system-level controller
- \( M \) - preview length of disturbances
- \( \gamma_{\text{slack}} \) - weighting on the slack variable that allows temporary constraint violations
- \( C \) - weighting matrix on model states
- \( R \) - weighting matrix on decision variables

6.4. Subsystem-Level MIQP Formulation

The subsystem-level controller is based upon the framework laid out in Section 5.1.2. Sample code is provided in Appendix C where it is programmed in MATLAB/Simulink [7] using the Yalmip toolbox [32].

The subsystem-level controller is responsible for determining the optimal mode of operation and set points for the thermal management units. The objective is to minimize the power consumed by the thermal management units while meeting the cooling capacity requirements set by the system-level controller.

Subsystem-level control uses models of the thermal management units that were presented in Figure 6.7 and Table 6.2. The cooling capacity output and power consumption is defined mathematically as,

\[
\dot{Q}_i = \left( \dot{Q}_{3,i} - \dot{Q}_{2,i} \right) u_{c,i} + \left( \dot{Q}_{2,i} - \dot{Q}_{1,i} \right) u_{b,i} + \dot{Q}_{1,i} \\
\dot{W}_i = \left( \dot{W}_{3,i} - \dot{W}_{2,i} \right) u_{c,i}^2 + \left( \dot{W}_{2,i} - \dot{W}_{1,i} \right) u_{b,i} + \dot{W}_{1,i} 
\]  

(6.4.1)
where $u_{c,i} \in [0,1]$ and $u_{b,j} \in \{0,1\}$ are continuous and Boolean decision variables. Given this relationship, the following coefficients are defined for use in the optimization,

$$
\begin{align*}
\alpha_1 &= \left( \dot{Q}_3 - \dot{Q}_2 \right) \text{ VCS} \\
\beta_1 &= \left( \dot{Q}_2 - \dot{Q}_1 \right) \text{ Tank} \\
\alpha_2 &= \left( \dot{Q}_3 - \dot{Q}_2 \right) \text{ ACM} \\
\beta_2 &= \left( \dot{Q}_2 - \dot{Q}_1 \right) \\
\alpha_3 &= \left( \dot{Q}_3 - \dot{Q}_2 \right) \text{ VCS} \\
\beta_3 &= \left( \dot{Q}_2 - \dot{Q}_1 \right) \text{ Ambient} \\
\alpha_4 &= \left( \dot{Q}_3 - \dot{Q}_2 \right) \text{ ACM} \\
\beta_4 &= \left( \dot{Q}_2 - \dot{Q}_1 \right)
\end{align*}
$$

(6.4.2)

The top four coefficients correspond to the VCS and ACM performance curves for dissipating heat in the fuel tank, while the bottom four coefficients correspond to dissipating heat to the ambient. Similar coefficients are defined for the power consumption,

$$
\begin{align*}
\delta_1 &= \left( \dot{W}_3 - \dot{W}_2 \right) \text{ VCS} \\
\varepsilon_1 &= \left( \dot{W}_2 - \dot{W}_1 \right) \text{ Tank} \\
\delta_2 &= \left( \dot{W}_3 - \dot{W}_2 \right) \text{ ACM} \\
\varepsilon_2 &= \left( \dot{W}_2 - \dot{W}_1 \right) \\
\delta_3 &= \left( \dot{W}_3 - \dot{W}_2 \right) \text{ VCS} \\
\varepsilon_3 &= \left( \dot{W}_2 - \dot{W}_1 \right) \text{ Ambient} \\
\delta_4 &= \left( \dot{W}_3 - \dot{W}_2 \right) \text{ ACM} \\
\varepsilon_4 &= \left( \dot{W}_2 - \dot{W}_1 \right)
\end{align*}
$$

(6.4.3)

Since the Boolean and continuous variables naturally constrain the system states by restricting operation between the upper and lower points of the performance curves, the subsystem-level controller only utilizes constraints on decision variables. The continuous decision variable is squeezed by the Boolean variable allowing for continuous operation of the
system when the Boolean variable is 1, and restricting the system to stand-by mode when the Boolean variable is 0,

\[ u_{b,i} \in \{0,1\} \quad \forall i \]
\[ 0 \leq u_{c,i} \leq u_{b,i} \quad \forall i \]

(6.4.4)

The objective function is then formulated to minimize error between the system-level controller’s \( \dot{Q}_{tot,des} \) and what is delivered by the subsystem-level controller. Additionally, the objective function seeks to minimize the total power consumption,

\[
J = \gamma \left( N_p \cdot \dot{Q}_{tot,des} - \sum_{i=0}^{N_p} \sum_{j=1}^{n_s} \dot{Q}_{S_j} [k+i] \right)^2 + (1 - \gamma) \sum_{i=0}^{N_p} \sum_{j=1}^{n_s} W_{S_j} [k+i] \]

(6.4.5)

In the event that the system experiences rapid switching, a penalty on switching mode can be implemented,

\[
J = \gamma \left( N_p \cdot \dot{Q}_{tot,des} - \sum_{i=0}^{N_p} \sum_{j=1}^{n_s} \dot{Q}_{S_j} [k+i] \right)^2 + (1 - \gamma) \sum_{i=0}^{N_p} \sum_{j=1}^{n_s} W_{S_j} [k+i] + \gamma_{switch} \sum_{i=0}^{N_p} \sum_{j=1}^{n_s} \Delta u_{b,i} \]

(6.4.6)

In the simulation case study, the prediction horizon, \( N_p = 6 \), and the total number of thermal management units, \( n_s = 2 \).

### 6.4.1. Tunable Parameters

The subsystem-level controller as formulated above contains the following tunable parameters that will be investigated later in this chapter:

- \( \gamma \) - weighting factor between performance and efficiency objectives
- \( \gamma_{switch} \) - independent weighting factor on switching frequency
- \( N_p \) - prediction horizon for the subsystem-level controller

### 6.5. Simulation Results

The method proposed is intended to be scalable for multiple platforms and systems with no limitations on the magnitude of system parameters and loads. As such, the following simulations use values for system parameters and loads that are relative to each other; therefore, units are not presented for parameters and signals.
Results from four different simulations are presented. The first is a baseline simulation that shows the capability of the controller to regulate temperatures, maintain constraints, and minimize power consumption. Second, is a simulation showing the effect of preview and maintaining constraints. The third simulation looks at the effect of turning off regulation and only maintaining temperatures within constraints. The final simulation looks to decrease switching frequency and the effects caused by fewer mode switches.

There exist sixteen modes of operation at the subsystem-level given that the ACM and VCS can be in stand-by or continuous mode and have the ability to reject heat to the fuel tank and/or ambient air. Table 6.3 contains the mode, status and thermal sink for the ACM and VCS, and a designated color. Colors are used to easily distinguish the operating mode of thermal subsystem in the following results.

Table 6.3. Subsystem operational modes and color designations

<table>
<thead>
<tr>
<th>Mode</th>
<th>Color Designation</th>
<th>ACM Status/Thermal Sink</th>
<th>VCS Status/Thermal Sink</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Stand-by</td>
<td>Stand-by</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Ambient Air</td>
<td>Stand-by</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Stand-by</td>
<td>Ambient Air</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Ambient Air</td>
<td>Ambient Air</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Fuel Tank</td>
<td>Stand-by</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Fuel Tank &amp; Ambient Air</td>
<td>Stand-by</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>Fuel Tank</td>
<td>Ambient Air</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>Fuel Tank &amp; Ambient Air</td>
<td>Ambient Air</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>Stand-by</td>
<td>Fuel Tank</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>Ambient Air</td>
<td>Fuel Tank</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>Stand-by</td>
<td>Fuel Tank &amp; Ambient Air</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>Ambient Air</td>
<td>Fuel Tank &amp; Ambient Air</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>Fuel Tank</td>
<td>Fuel Tank</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>Fuel Tank &amp; Ambient Air</td>
<td>Fuel Tank</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>Fuel Tank</td>
<td>Fuel Tank &amp; Ambient Air</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>Fuel Tank &amp; Ambient Air</td>
<td>Fuel Tank &amp; Ambient Air</td>
</tr>
</tbody>
</table>
6.5.1. Baseline Simulation

Parameters for the baseline simulation are presented in Table 6.4 and (6.5.1). The weighting matrices in (6.5.1) correspond to the weighting matrices of the cost function for the system-level controller as described in (6.3.12).

<table>
<thead>
<tr>
<th>Table 6.4. Baseline case study optimization parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System-level Controller</strong></td>
</tr>
<tr>
<td>$N_p$</td>
</tr>
<tr>
<td>10 steps</td>
</tr>
<tr>
<td>(10 min.)</td>
</tr>
</tbody>
</table>

$$C = \begin{bmatrix}
10 & 0 & 0 & 0 & 0 & 0 \\
0 & 10 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 10 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix} \quad (6.5.1)$$

$$R = I_3$$

Figure 6.8 presents the time histories for the baseline simulation case. The top graph shows the heat transfer per unit time into the passenger cabin and fuel tank, and is a recreation of Figure 6.3. In the second graph is the commanded and delivered heat transfer per unit time to the fuel tank. The green line is the system-level command, while the red and blue lines are the subsystem commands to the VCS and ACM, respectively. Similarly, the middle graph presents the commanded and delivered heat transfer rate to the ambient air. The fourth graph shows the system-level command to the ram air loop that is used to cool the fuel tank. The final graph shows the total power consumption by the VCS, ACM, and ram air loop, based upon the performance curves of Figure 6.7. Note that the system operates within constraints and that the subsystem controller meets the desired setpoints of the system-level controller.

Figure 6.9 shows the time histories for cabin and fuel tank temperatures. The fuel tank temperature has a steady state bias because of the use of proportional-only regulation. Cabin temperature violates the constraints at several points in the mission because the system-level controller has no preview of upcoming constraint changes. With the incorporation of the slack
variable for temperature constraints, the problem does not become infeasible. Figure 6.10 presents the operational mode over the course of the mission, which has minor switching until the last 1500 seconds where there are multiple switches in a short period of time.

Figure 6.8. Performance of the baseline controller
Figure 6.9. Cabin and fuel temperature for the baseline controller

Figure 6.10. Operational mode throughout the mission for the baseline controller
6.5.2. Effect of Load Preview

This section looks at how system response and constraint violations are affected by previewing disturbances. Parameters for the simulation are presented in Table 6.5 and (6.5.2). Since in the baseline case there was a preview length of six minutes, this simulation looks at the case where there is no preview ($M = 0$) in order to compare preview and no preview. The weighting matrices in (6.5.2) correspond to the weighting matrices of the cost function for the system-level controller as described in (6.3.12).

Table 6.5. Case study optimization parameters with no preview of disturbances

<table>
<thead>
<tr>
<th>System-level Controller</th>
<th>Subsystem-level Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_p$</td>
<td>$M$</td>
</tr>
<tr>
<td>10 steps (10 min.)</td>
<td>0 steps (0 min.)</td>
</tr>
</tbody>
</table>

$$ C = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (6.5.2) $$

$$ R = I_3 $$

Figure 6.11 presents the time histories for the same signals as in the baseline case. While the system reacts differently from the baseline case at each disturbance change, the most significant differences occur just before 4000 seconds and just after 5000 seconds. The first disturbance exceeds the thermal system’s capability to instantaneously remove heat from the cabin. As such, the system-level controller hits the constraints on maximum $\dot{Q}$ being removed from the cabin and sent to the tank and ambient. Similarly, the ram air loop used to cool the tank hits the upper constraint. The subsystem-level controller responds by meeting the demand, but at the cost of a large spike in power consumption and the cabin temperature temporarily violates the upper constraint (Figure 6.12). Comparatively, when looking at the baseline case (Figure 6.8) the ability to preview the disturbances allows the thermal system to pre-cool the cabin so that when
the disturbance occurs the cabin temperature remains within the constraints, and requires less power consumption. This is also reflected in Figure 6.13 where the system enters mode 16 (VCS and ACM on, rejecting heat to the fuel tank and ambient) for an extended period of time compared to the baseline case.

Figure 6.11. Performance of the controllers with no system-level preview of disturbances
Figure 6.12. Cabin and fuel temperature with no preview of disturbances

Figure 6.13. Operational mode throughout the mission with no preview of disturbances
6.5.3. Effect of Regulation

This section looks at how system response and constraint violations are affected when the fuel tank and cabin temperatures are not regulated. Parameters for the simulation are presented in Table 6.6 and (6.5.3). The weighting matrices in (6.5.3) correspond to the weighting matrices of the cost function for the system-level controller as described in (6.3.12), and since there is no regulation of temperatures the weighting on states is zero.

Table 6.6. Case study optimization parameters with no regulation of temperatures

<table>
<thead>
<tr>
<th>System-level Controller</th>
<th>Subsystem-level Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_p$</td>
<td>$M$</td>
</tr>
<tr>
<td>10 steps (10 min.)</td>
<td>6 steps (6 min.)</td>
</tr>
</tbody>
</table>

$$C = 0_{6 \times 6}$$

$$R = I_3$$ (6.5.3)

Figure 6.14 presents the time histories for the same signals as in the baseline case. The most noticeable difference is that the fuel tank is used as a heat sink much more when its temperature is not being regulated. Since the initial temperature of the fuel tank is low, the system-level controller seeks to reject a large portion of heat to the tank instead of the ambient air. Additionally, the ram air loop is utilized less than in the baseline case. These two characteristics lead to an overall decrease in total power consumption. The tradeoff is that temperatures of the fuel tank and cabin are allowed to approach their upper constraints as shown in Figure 6.15. Similar to previous cases, the cabin temperature violates the constraints only when the constraints suddenly drop, due to the controller not previewing the constraints.

It is also important to note the response of the controller to the disturbance transients at 3800 seconds and 5200 seconds. For the former, the system-level controller pre-cools the cabin so that the upper temperature constraint is not violated due to the disturbance. Similarly at 5200 seconds, the ram air loop is increased (Figure 6.14) and heat from the cabin is routed away from the fuel tank, keeping the fuel tank temperature below the upper constraint.

The operational mode for the mission is shown in Figure 6.16. For the first 5000 seconds, only four different modes are required for operation due to the fact that the fuel tank temperature
is away from its upper constraint, so both the VCS and ACM are continually rejecting heat to it instead of having to switch on/off as in the baseline case.

**Figure 6.14.** Performance of the controllers with no regulation on fuel tank or cabin temperature
Figure 6.15. Cabin and fuel temperature with no regulation

Figure 6.16. Operational mode throughout the mission when not regulating temperatures
6.5.4. Effect of Switching Penalty

This section looks at the how system response and constraint violations are affected by penalizing switching frequency. Parameters for the simulation are presented in Table 6.7 and (6.5.4), where the difference from the baseline case is the penalty on switching frequency. The weighting matrices in (6.5.4) correspond to the weighting matrices of the cost function for the system-level controller as described in (6.3.12).

<table>
<thead>
<tr>
<th>System-level Controller</th>
<th>Subsystem-level Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_p$</td>
<td>$M$</td>
</tr>
<tr>
<td>10 steps</td>
<td>6 steps</td>
</tr>
<tr>
<td>(10 min.)</td>
<td>(6 min.)</td>
</tr>
</tbody>
</table>

$$
C = \begin{bmatrix}
10 & 0 & 0 & 0 & 0 & 0 \\
0 & 10 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 10 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
$$

$R = I_3$

Figure 6.17 presents the time histories for the same signals as in the baseline case. The most noticeable difference caused by a higher penalty on switching shows up during the last 1500 seconds of the simulation, where in the baseline case there were 22 switches compared to only 10 switches in this case study. This has a direct effect on the continuous cooling provided by the ACM as shown in the 3rd graph of Figure 6.17, where the ACM lags the cooling demand of the system-level controller. During these periods of lagging, the temperature in the cabin begins to increase (Figure 6.18) until the ACM switches to continuous cooling. However, even with the increased penalty on switching temperature of the fuel tank and cabin are well-regulated near the nominal values.

Figure 6.19 shows the direct effect of increased switching penalty on the operational mode over the length of the mission profile. Compared to Figure 6.10, there are substantially fewer switches.
Figure 6.17. Performance of the controllers with higher penalty on switching frequency
Figure 6.18. Cabin and fuel temperature with high penalty on switching frequency

Figure 6.19. Operational mode throughout the mission with high penalty on switching frequency
6.5.5. Total Power Consumed Comparison

The final comparison between the different controller formulations is the effect on total power consumption. Both the system and subsystem controllers attempt to reduce power consumption by minimizing cooling capacity supplied and operating the thermal management units efficiently, while meeting the performance objects. However, each formulation results in different values of total power consumption over the entire mission.

Figure 6.20 presents total power consumption over the course of the mission for each designed controller. It is clear from this figure that greatly penalizing switching frequency results in a larger total power consumption, whereas not regulating temperatures reduces power consumption the most. The difference between controllers with and without preview occurs during large transient disturbances where preview helps reduce total power consumption.

This observation is further supported in Table 6.8 where the integrated values of the total power consumption for each controller are presented. Typically these values would be in units of joules; however, relative magnitude is most important in this analysis. The largest difference in energy consumption is nearly 30% between the cases without regulation and less switching. This shows that less regulation and more switching can significantly improve efficiency, while maintaining performance requirements and observing constraints.

![Graph showing total power consumption over mission time for different controllers](image_url)

**Figure 6.20. Total power consumption over the course of the mission for each controller**
Table 6.8. Total work in the system over the course of the mission using each controller

<table>
<thead>
<tr>
<th></th>
<th>Baseline</th>
<th>Without Preview</th>
<th>Without Regulation</th>
<th>With Less Switching</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>82,600</td>
<td>83,500</td>
<td>72,400</td>
<td>93,400</td>
</tr>
</tbody>
</table>

6.6. Case Study Remarks

The previous sections highlight several functionalities of the proposed two-level hierarchical control strategy. The effects of preview, temperature regulation, and switching frequency are all shown to directly impact temperatures in each thermal zone and the total power consumption. Ideally each function could be appropriately tuned depending upon the aircraft, mission, and operating environment in order to achieve the objectives of the platform while minimizing energy consumption.

A major assumption of the above analysis is that the components of each thermal management unit will be capable of actuating the unit to deliver the cooling capacity designated by the subsystem-level controller. In the event that the thermal management unit does not achieve the cooling capacity required of it, then information would need to flow back up through the control hierarchy in order to make adjustments to control decisions at each level. This is ongoing work and is not addressed in this thesis.

Finally, it is important to note that the proposed control strategy is intended to be scalable across multiple platforms. The system parameters $\dot{Q}$ and $\dot{W}$ that are presented in this thesis can easily be adjusted for each thermal management unit and the overall system.
Chapter 7

Conclusion

7.1. Summary of Research Contributions

This thesis seeks to develop the groundwork for control design of large-scale systems that operate in a hierarchical structure with multiple time scales, specifically commercial and military aircraft. The large number of components and systems that make up these vehicles allow for a significant number of actuators that can fulfill control objectives.

First, component and system models are developed that represent the architecture of these systems. Many models are dynamic and capable of capturing the transient behavior of systems operating in the electrical, thermal, hydraulic, or pneumatic energy domains. The most important aspect of the toolset is the interconnection of energy domains. Electrical systems have thermal models that account for inefficiencies resulting in the generation of heat which the thermal system is then responsible for removing from the aircraft. All systems and components that require power eventually trace back to the engine which acts as the primary source of energy on board the aircraft. These interconnections lead to an overall better representation of the aircraft and help facilitate the development of controllers and improved system configuration.

Additionally, a modeling technique allowing a user to maintain the high accuracy levels of high-fidelity models while increasing computational speed is presented. The modeling technique is shown to improve component and system output accuracy which directly affects the closed loop control decisions, and improves the realism of controllers designed in simulation. The models and modeling techniques will be of considerable importance when multiple simulations are conducted for system and controller design, verification, and validation.

The second half of the thesis begins to look at developing controllers for these hierarchical systems. A method of hierarchical control that matches the hierarchical nature of the system is proposed. This five-tier structure looks at the aircraft from a vehicle level down to the physical actuation level, and will attempt to develop individual control methods for each level. This
thesis focused solely on the system and subsystem-levels, specifically the thermal management system.

At the system-level, a model predictive controller is developed for minimizing energy consumption while maintaining temperatures within constraints. The subsystem-level controller takes inputs from the system-level controller and minimizes a mixed integer quadratic programming problem in order to meet the demand of the system-level controller and minimize power consumption of each thermal management system.

Finally, the proposed system and subsystem-level controllers were developed to control a sample configuration of an aircraft thermal system consisting of a cabin, fuel tank, vapor compression system, air cycle machine, and ram air cooling loop. The objective was to maintain the temperature of the cabin within bounds while removing heat via the air cycle machine or vapor compression system and rejecting it to the fuel tank or ambient air. The temperature of the fuel tank was also required to remain within some bounds. Various functionalities of the controller were demonstrated in simulation to show that the proposed control methods were capable of meeting constraints, minimizing energy consumption, and utilizing each thermal management system in an efficient manner. A difference of 30% in energy consumption was shown to result from different configurations of the control strategy, meaning the proposed strategy could be used to reduce component size and weight, or used with current platforms to better handle loads.

7.2. Future Work

Future work will continue to develop the hierarchical control framework for large-scale systems. With only two levels currently being developed, plenty of work remains for the vehicle level, component level, and physical level of the hierarchy.

Simplified models have been used in this analysis with no discrepancies between models for simulation of the system dynamics and models for control decisions. Future work will add robustness for model uncertainty so that controllers are capable of achieving control objectives without having a 100% accurate model. The current work has also assumed that knowledge of disturbances is completely accurate, which will be addressed in future work.
Communication between the controllers is currently a top-down approach with no communication being transferred back to upper levels. In the event that a lower level controller determines it is incapable of achieving the command from an upper level controller, knowledge of that occurrence should be passed up. Future work will look at how to develop these communication paths and how controllers should update when infeasibility occurs at the lower level controllers.

Finally, additional work will look at asynchronous information flow through the control structure and how to update different controllers outside of their respective sample rates.
References


Appendix A

Additional Modeling

In addition to the thermal systems modeled in Chapter 2, models have been created for aircraft electrical and hydraulic systems. The mathematical equations for several of those systems are presented in this appendix.

The development of the following models was done in partnership with other students at UIUC. Acknowledgement must be given to the following students who played a fundamental role in developing the presented models: Srikanthan Sridharan (ECE), Subhabrata Banerjee (MechSE), and Tutku Buyukdegirmencı (ECE).

Electrical Power System

Electrical power system models are presented for the following components:

- Generator
- Exciter system
- Battery
- Inverter
- AC Loads
- Transformer
- Rectifier

Generator Mathematical Model

This acts as a standalone generator model. Common generator models assume either an infinite bus or a voltage supply connected to the machine terminals [33]; however, such an implementation would require the knowledge of line voltages and would violate the causality of the system. Therefore, a voltage-based model is created. This way, the load information would not be necessary while producing line voltage. An accurate implementation of this approach would require line transients such as time rate of change in the line inductance currents and voltage drop across the line reactors to be monitored. However, a singular perturbation approach presented in [34] is followed to mitigate this problem and optimize the system model details and
simulation speed. To achieve faster simulations, the models are implemented in a synchronous reference frame. Synchronous machine dynamic model is as follows:

\[
T_{do}' \frac{dE'_{d}}{dt} = -E'_{q} - (X_{d} - X'_{d}) \left[ I_{d} - \frac{X'_{d} - X''_{d}}{\left(X'_{d} - X_{ls}\right)^2} \left(\psi_{1d} + (X'_{d} - X_{ls})I_{d} - E'_{q}\right)\right] + E_{fd},
\]

(A.1)

\[
T_{qo}' \frac{dE'_{q}}{dt} = -E'_{d} + (X_{q} - X'_{q}) \left[ I_{q} - \frac{X'_{q} - X''_{q}}{\left(X'_{q} - X_{ls}\right)^2} \left(\psi_{2q} + (X'_{q} - X_{ls})I_{q} + E'_{d}\right)\right],
\]

(A.2)

\[
T_{do}'' \frac{d\psi_{1d}}{dt} = -\psi_{1d} + E'_{q} - (X'_{d} - X_{ls})I_{d},
\]

(A.3)

\[
T_{qo}'' \frac{d\psi_{2q}}{dt} = -\psi_{2q} - E'_{d} - (X'_{q} - X_{ls})I_{q},
\]

(A.4)

where \(X_{d}\) and \(X_{q}\) are the direct and quadrature axis per-unit reactances, respectively; \(X'_{d}\) and \(X'_{q}\) are the direct and quadrature axis per-unit transient reactances, respectively; \(X''_{d}\) and \(X''_{q}\) are the direct and quadrature axis per-unit subtransient reactances, respectively; \(T_{do}'\) and \(T_{qo}'\) are the direct and quadrature axis field winding per-unit transient time constants, respectively; \(T_{do}''\) and \(T_{qo}''\) are the direct and quadrature axis field winding per-unit sub-transient time constants, respectively. These quantities are defined as:

\[
X'_{d} = X_{d} - \frac{X_{md}^2}{X_{fd}}
\]

(A.5)

\[
X'_{q} = X_{q} - \frac{X_{mq}^2}{X_{1q}}
\]

(A.6)

\[
X''_{d} = X_{ls} + \frac{1}{\frac{1}{X_{md}} + \frac{1}{X_{jd}} + \frac{1}{X_{1d}}}
\]

(A.7)

\[
X''_{q} = X_{ls} + \frac{1}{\frac{1}{X_{mq}} + \frac{1}{X_{lq}} + \frac{1}{X_{2q}}}
\]

(A.8)

\[
T_{do}' = \frac{X_{fd}}{\omega_{f}R_{fd}}
\]

(A.9)
\[ T_{qo}' = \frac{X_{iq}}{\omega_s R_{1q}} \]  
(A.10)

\[ T_{do}'' = \frac{1}{\omega_s R_{1d}} \left( X_{1d} + \frac{1}{X_{sdl}} \right) \]  
(A.11)

\[ T_{qo}'' = \frac{1}{\omega_s R_{2q}} \left( X_{12q} + \frac{1}{X_{sdl}} \right) \]  
(A.12)

and the synchronous machine terminal voltages are:

\[ V_q = -\gamma (X_d'' + X_{TL}) I_d - (R_s + R_{TL}) I_q + \gamma \left( E_q' \frac{X_d'' - X_{ls}}{X_d' - X_{ls}} + \psi_{1d} \frac{X_d' - X_{ls}''}{X_d' - X_{ls}} \right) \]  
(A.13)

\[ V_d = \gamma (X_d'' + X_{TL}) I_q - (R_s + R_{TL}) I_d + \gamma \left( E_q' \frac{X_q'' - X_{ls}}{X_q' - X_{ls}} - \psi_{2q} \frac{X_q' - X_{ls}''}{X_q' - X_{ls}} \right) \]  
(A.14)

where \( R_{TL} \) and \( X_{TL} \) are the transmission line per-unit resistance and reactance, respectively; \( \gamma \) is the per-unit electrical frequency (typical base value is 377 rad/s), \( R_s \) is the synchronous machine per-unit stator resistance. Synchronous machine per-unit electromagnetic torque is,

\[ T_{EM} = \frac{X_d'' - X_{ls}}{X_d' - X_{ls}} E_q' I_q + \frac{X_d' - X_d''}{X_d' - X_{ls}} \psi_{1d} I_q + \frac{X_q'' - X_{ls}}{X_q' - X_{ls}} E_d' I_d - \frac{X_q' - X_q''}{X_q' - X_{ls}} \psi_{2q} I_d + (X_{q''} - X_q'') I_d I_q \]  
(A.15)

**Generator Inputs and Outputs**

The synchronous generator model has two electrical inputs, one mechanical input, one mechanical output, two electrical outputs and one thermal output. The electrical inputs are the line currents in the synchronous reference frame \((i_{dq0})\) and the field current \((i_{fd})\) and their units are amperes. These quantities are converted into per-unit and the base currents \(I_{BDQ}\) and \(I_{BFD}\) should be defined in the MATLAB workspace. The mechanical input is the shaft speed from the gearbox model and its unit is rad/s. The two electrical outputs are the synchronous frame line voltages \((v_{dq0})\) and field winding voltage \((v_{fd})\). The mechanical output is the shaft torque \((T_{em})\) and its unit is N-m. The thermal output is the generator power loss \((P_{loss})\) and it is defined in Watts. The per-unit synchronous machine model requires the user to define the base quantities.
Summarized inputs to the model are:

- \textbf{idq0} – Synchronous frame line current (Units – Ampere)
- \( \omega \) – Shaft speed (Units – rad/s)
- \textbf{ifd} – Field Current (Units – Ampere)

Outputs from the model are:

- \textbf{vdq0} – Synchronous machine line voltage (Units – Volt)
- \textbf{Te} – Electromagnetic shaft torque (Units – N-m)
- \textbf{vfd} – Field Winding Voltage (Units – Volt)
- \textbf{Ploss} – Power loss (Units – Watt)

**Exciter System Mathematical Model**

The synchronous machine described in the previous section requires a field current supply. Commonly, another wound-field or permanent magnet synchronous generator coupled to the main generator shaft is utilized to provide this current. The output terminals are rectified and directly connected to the main generator field terminals. This field current must be provided independent of the generator and should be controlled properly to regulate the generator terminal voltage. A battery provides the exciter-generator field current, and it is regulated through a dc/dc converter. A generic structural diagram is shown,

\[ \dot{V} = k_1 \left( V_{ref} - V_{line} (t) \right) \]  \hspace{1cm} (A.16)

\[ \dot{m} = k_2 \left( r - k_3 m \right) \]  \hspace{1cm} (A.17)

where

\[ V_{line} (t) = \sqrt{v_a^2 + v_q^2} \]  \hspace{1cm} (A.18)

**Figure A.1. Structural diagram of an exciter system**
Here $k_1$, $k_2$, and $k_3$ are controller gains, $r$ is an arbitrary variable and $m$ is the dc/dc converter duty ratio. The reference line voltage is $V_{\text{ref}}$ and measured line voltage is $V_{\text{line}}$.

The generator and exciter models are coupled in Simulink to form a single component model with a GUI (Figure A.2).

![GUI for synchronous generation unit with exciter](image)

**Figure A.2.** GUI for synchronous generation unit with exciter

**Battery Mathematical Model**

The mathematical model described in this section is used to implement a lithium-ion battery. The battery capacity is a function of the charging/discharging rates $i(t)$, temperature $T(t)$ and cycle number $n_{\text{cycle}}$ and a rate factor $f(i(t))$ which is a function of current. The rate factor is used to account for undesired side reactions with increase in current magnitude. The dynamic capacity of the battery represented by its state of charge (SOC), is a function of the abovementioned factors and given by the following expression.

$$SOC\left(i(t), T(t), n_{\text{cycle}}, t\right) = SOC_{\text{initial}} + \int f_1\left(i(t), T(t), n_{\text{cycle}}, f(i(t)), t\right) dt$$  \hspace{1cm} (A.19)

The battery is modeled using the notion of multiple scale time constants, each at the level of seconds, minutes and hours. In the electrical equivalent circuit, each time constant can be modeled as a resistance-capacitance combination, as shown in Figure A.3. Measurements of the circuit parameters are found using a battery testing apparatus and recording the test sequences and data corresponding to open circuit voltage ($V_{oc}$) and terminal voltage ($V_i$) versus SOC at room temperature. Each parameter (resistance and capacitance) in the model shown in Figure
A.3, is a nonlinear function of SOC. For a practically useable model, each parameter is represented as a polynomial function of the SOC up to sixth order given as

\[ R(or \ C)_{s,m,h} = A_0 + A_1SOC + A_2SOC^2 + \ldots + A_6SOC^6 \]  

(A.20)

![Figure A.3. Battery constants for multiple time scales](image)

The coefficients $A_0$-$A_6$ are obtained by a best-fit polynomial expression on the experimentally determined data points. From the equivalent circuit, the battery terminal voltage can be calculated as follows,

\[ V_s = V_{oc} - I \left( R_{series} + \left( R_s \parallel \frac{1}{sC_s} \right) + \left( R_m \parallel \frac{1}{sC_m} \right) + \left( R_h \parallel \frac{1}{sC_h} \right) \right) \]  

(A.21)

where $I$ refers to the series current flowing in the circuit. The various resistance-capacitance combinations ($R_s$-$Cs$, $R_m$-$C_m$, $R_h$-$C_h$) refer to the time constants corresponding to the second, minute and hour time scales.

**Battery Inputs and Outputs**

A high-level block diagram with inputs and outputs to the model is shown in Figure A.4. The internal blocks indicate the sequence of steps executed to compute the battery dc bus voltage output. Also, the total power loss in the battery pack is computed as the sum of power dissipated in various resistors of the electrical equivalent circuit. The net charge/discharge detection block determines whether the net effect is charging or discharging, depending on the difference between the magnitudes of charging and discharging currents. This information along with the current flowing in the circuit is used to compute the dynamic SOC, which enables the calculation of the terminal voltage using (5.3.19). A single cell lithium-ion battery can provide a nominal dc voltage of 3.81 V. By connecting a number of cells in series, a required dc bus voltage can be realized. Connection of parallel modules enables to increase the current capacity of the battery module.
The user inputs and outputs are given below. Inputs to the model are:

- **CTOUT** - Discharging current magnitude (Units – Ampere)
- **CTIN** - Charging current magnitude (Units – Ampere).

Outputs from the model are:

- **VDC** - Voltage output of the battery (Units – Volt)
- **PLOSS** - Power lost in the battery due to its internal resistances (Units – Watt)

The battery model requires the following variables to be defined in the GUI (Figure A.5),

- **Number of cells in each module** - Determines the series voltage of the entire string
- **Number of modules** - Determines the number of parallel battery modules, each module consisting of its string
- **Initial SOC** - Nominal initialization done between 0.9-1.0 (Range: 0-1).

![Battery GUI](image)
Inverter Mathematical Model

The battery voltage serves as the dc bus input voltage \( V_{dc} \) to the battery. The other input needed to control the three-phase inverter are switching (or modulating) functions. Since the inverter is an averaged model in \( dq0 \) frame, the switching functions \( q \) are also steady state averaged signals. When sinusoidal pulse width modulation method is used to control the inverter switches, the output L-L (rms) voltage of the inverter can be given as,

\[
V_{\text{L-L (rms)}} = \frac{q\sqrt{3}V_{dc}}{\sqrt{2}} \tag{A.22}
\]

Because of the reference frame transformation, the switching functions for the three-phase inverter are also constant values, instead of sinusoidal functions, so that the output voltages are constant values. The phase of the inverter output voltages with respect to the rest of the AC system can be changed by modifying the constant vector passed as input into the switching function block.

Inverter Inputs and Outputs

The user inputs and outputs are given below. Inputs to the model are:

- **DC bus voltage** – DC link voltage for the inverter obtained from the battery output voltage (Units – Volt)
- **Switching function** – A 1x3 vector signal with constant values passed as modulating functions to control the inverter output voltage
- **Inverter AC side current** – Current demand in \( dq0 \) frame (Units – Ampere)

Outputs from the model are:

- **\( dq0 \) output voltage** – Three-phase inverter output voltage in \( dq0 \) frame (Units – Volt)
- **DC link current** – Computed based on the inverter AC output current and passed as the dc bus current demand feedback to the battery (Units – Ampere)

- **Power loss** – Calculated as the conduction and switching losses in the inverter (Units – Watt)

The inverter model requires the following variables to be defined,

- **Rise time and fall time** – Depends on the specific type of semiconductor switch and can be obtained from the corresponding datasheets (Units – Second). This is useful to calculate switching losses.

- **On state Vce drop** - The voltage drop (ideally desired to be 0) across the switch during conduction is used to compute the conduction losses in the switch (Units – Volt)

- **Switching frequency** – PWM frequency that determines the number of times that each switch is turned on and off and is required to compute switching losses (Units - Hz)

- **Number of switches** – Refers to the number of semiconductor switches in the inverter. Usually, a two-level inverter configuration is used, which has six power switches.

![Figure A.7. Inverter GUI](image-url)
AC Loads Mathematical Model

Although the loads are divided as power, current and impedance loads, the generators require the dq0 axis currents drawn by each load. In general, the complex power $S$ can be expressed as,

$$ S = VI^* $$

(A.23)

where $V$ and $I^*$ are the voltage and complex conjugate of the current drawn.

For the power loads, the active ($P$) and reactive components ($Q$) of the complex power can be expressed as,

$$ P + jQ = (V_d + jV_q)(I_d - jI_q) $$

(A.24)

From the above equation, after separating the real and imaginary parts, the active and reactive currents can be given as,

$$ I_d = \frac{PV_d + QV_q}{V_d^2 + V_q^2}, \quad I_q = \frac{PV_q - QV_d}{V_d^2 + V_q^2} $$

(A.25)

For the impedance loads, the active and reactive currents can be given as,

$$ I_d = \frac{RV_d + XV_q}{R^2 + X^2}, \quad I_q = \frac{RV_q - XV_d}{R^2 + X^2} $$

(A.26)

The $dq$ currents drawn by the loads are computed using (A.22) and (A.23) and are passed as feedback signals to the corresponding generator bus.

Figure A.8. AC loads GUI
Transformer Mathematical Model

A simplified circuit diagram for the modeled transformer is shown in Figure A.9. The magnetic coupling is ideal and only copper losses are modeled. The electrical relation is,

\[
\frac{V_p - i_p R_p}{V_s - i_s R_s} = \frac{N_p}{N_s}
\]  

(A.27)

where \( V_p \) is the primary winding voltage, \( V_s \) is the secondary winding voltage, \( i_p \) is the primary winding current, \( i_s \) is the secondary winding current, and \( N_p \) and \( N_s \) are the primary and secondary windings number of turns, respectively.

![Figure A.9. Simplified circuit diagram of the modeled transformer](image)

Transformer Inputs and Outputs

The user inputs and outputs are given below. Inputs to the model are:

- **V_{in}** – Primary winding voltage input (Units – Volt)
- **I_{out}** – Secondary winding current output (Units – Ampere)

Outputs from the model are:

- **V_{out}** – Secondary winding output voltage (Units – Volt)
- **I_{in}** – Primary winding current (Units – Ampere)

The transformer model requires the following variables to be defined in the GUI (Figure A.11),

- **Primary Winding Resistance** – (Units – Ohm)
- **Secondary Winding Resistance** – (Units – Ohm)
- **Number of Turns in Primary**
- **Number of Turns in Secondary**

![Figure A.10. Transformer inputs and outputs](image)
Rectifier Mathematical Model

A voltage-based model is implemented and a dc-link inductor is assumed. The rectifier is modeled as:

\[
L \frac{di_{out}}{dt} = m \cdot V_{line} (t) - V_{out},
\]

(A.28)

\[
\theta = \arctan \left( \frac{v_q}{v_d} \right),
\]

(A.29)

\[
i_q = m \cdot i_{out} \cdot \sin (\theta),
\]

(A.30)

\[
i_d = m \cdot i_{out} \cdot \cos (\theta),
\]

(A.31)

where, \( V_{out} \) is the output dc voltage, \( i_{out} \) is the output dc current, and \( m \) is the modulation depth.

Rectifier Inputs and Outputs

The user inputs and outputs are given below. Inputs to the model are,

- \( V_{qd0} \) – Input line voltage in the synchronous reference frame (Units – Volt)
- \( \text{cont} \) – Converter modulation depth. It is a number between 0 and 1.
- \( V_{out} \) – Output dc voltage level (Units – Volt)
Outputs from the model are:

- \( i_{\text{out}} \) – Output dc current level (Units – Ampere)
- \( i_{qd0} \) – Input line current in the synchronous reference frame (Units – Ampere)

The rectifier model requires the following variable to be defined in the GUI,

- **Inductance** – DC link filter inductor (Units – Henry)

**Hydraulic Power System**

Hydraulic power system models are presented for the following components:

- Engine Driven Pump
- Fluid Reservoir
- Hydraulic Load

**Engine Driven Pump Mathematical Model**

The mathematical model for the engine driven pump is based upon first principle relationships between mass flow rate, pressure, density, and pump properties.

The mass flow rate through the pump is calculated,

\[
\dot{m} = \rho (DN - k_{leak} \Delta P)
\]  
(A.32)

where \( \rho \) is the fluid density in kg/m\(^3\), \( D \) is the pump displacement in m\(^3\)/rev, \( N \) is the pump shaft rotational frequency in rev/s, and \( \Delta P \) is the pressure differential in Pascals.

The leakage flow coefficients determined based upon the assumption that it is linearly proportional to the Hagen-Poiseuille coefficient,

\[
k_{leak} = \frac{k_{hp}}{\mu} \quad (A.33)
\]

\[
k_{hp} = \frac{DN_{\text{nom}}(1-\eta_v)\mu_{\text{nom}}}{\Delta P_{\text{nom}}} \quad (A.34)
\]

where \( \mu \) is the dynamic viscosity of the fluid in Pa·s, \( \eta_v \) is the volumetric efficiency of the pump, and the subscript \( nom \) represents nominal values.

The torque applied to the driving shaft of the pump where \( \eta_{mech} \) is the mechanical efficiency,

\[
T = \frac{Q\Delta P}{2\pi N\eta_{mech}} \quad (A.35)
\]
Engine Driven Pump Inputs and Outputs

The first input to the hydraulic fluid pump is the fluid bus containing flow rate, temperature, and pressure variables. Pressure upstream from the pump also has to be supplied as an input. To maintain causality in the system, the components connected to the pump must calculate a pressure while the pump will calculate a mass flow rate. The final input to the pump is a shaft speed in RPM. This RPM should be taken from the auxiliary gearbox of the engine. Output signals include a fluid flow bus containing flow rate, temperature, and pressure, in addition to the mechanical torque applied to the pump shaft.

![Diagram](image)

**Figure A.12. Engine driven pump inputs and outputs**

The GUI of the hydraulic engine driven pump provides the ability to input multiple parameters in order to specify the operation and efficiency of the pump.

![GUI](image)

**Figure A.13. EDP general GUI**

**Figure A.14. EDP Hagen-Poiseuille GUI**
**Fluid Reservoir Mathematical Model**

The mathematical model of the fluid reservoir is based upon first principles. The mass of the fluid in the reservoir is calculated as,

\[ m_{\text{fluid}} = \int \dot{m}_{\text{in}} - \dot{m}_{\text{out}} \]  \hspace{1cm} (A.36)

where \( \dot{m} \) is the flow rate of the hydraulic fluid and the subscripts denote into and out of the reservoir.

The rate of change in the reservoir fluid temperature is determined as a function the flow rate and temperature of the flow in and the fluid mass,

\[ \dot{T}_{\text{fluid}} = \dot{m}_{\text{in}} \left( T_{\text{in}} - T_{\text{fluid}} \right) / m_{\text{fluid}} \]  \hspace{1cm} (A.37)

where the \( T_{\text{in}} \) is the temperature of the fluid flow into the reservoir and \( T_{\text{fluid}} \) is the temperature of the fluid in the tank.

The rate of change in the air pressure of the reservoir is determined using input bleed air properties and mass of the air in the tank,

\[ \dot{P}_{\text{air}} = \dot{m}_{\text{in}} RTZ / V \]  \hspace{1cm} (A.38)

where \( \dot{m}_{\text{in}} \) is the flow rate of bleed air into the reservoir, \( R \) is the universal gas constant for air, \( T \) is the temperature of the bleed air into the reservoir, \( Z \) is the compressibility factor for air, and \( V \) is the volume of the air in the reservoir. The volume of the air changes in time with respect to how much volume of hydraulic fluid is within the reservoir.

**Fluid Reservoir Inputs and Outputs**

The first input to the hydraulic fluid reservoir contains flow rate, temperature, and pressure variables of the engine bleed air. The second input is the bypass return containing flow rate, temperature, and pressure variables. The mass flow rate flowing out of the fluid reservoir is the third input. The engine driven pump provides this mass flow rate as an output. The final input is the mass flow rate of the returning hydraulic fluid from the loads.

Output signals include a bleed air flow rate demand and output fluid flow bus. The bleed air demand is determined as a function of air pressure in the fluid reservoir and the desired set point.

The GUI of the hydraulic fluid reservoir provides the ability to input multiple parameters in order to specify the geometry and operation of the reservoir.
Hydraulic Load Mathematical Model

The hydraulic load model is reflective of mass flow rate through hydraulic actuators over the course of the mission profile. Dynamics of the actuators are not modeled as the power consumed by the actuators is a linear function of the mass flow rate through the actuators. As such, the hydraulic load model provides the user a selection of hydraulic loads which are located in a lookup table and contain stochastic flow rates that are scaled appropriately for each mission phase.

The hydraulic load model takes a fluid input bus from the bypass valve and the flight phase from the mission profile. The outputs are the return flow rate of hydraulic fluid flow and the power consumed by the hydraulic load. The power is determined by,

\[ P = \dot{m}P/\rho \]  

where \( \dot{m} \) is the mass flow rate of hydraulic fluid through the actuator, \( P \) is the pressure of the fluid, and \( \rho \) is the density of the fluid.

Simulink Graphical User Interfaces

The models presented in Chapter 2 have additional inputs that are defined via GUIs in Simulink. These GUIs are presented in the following figures.
Figure A.17. Fuel tank geometry GUI

Figure A.18. Fuel tank initial conditions GUI

Figure A.19. Fuel tank thermal GUI

Figure A.20. Fuel pump general GUI
Figure A.21. Fuel pump power GUI

Figure A.22. Heat loads GUI

Figure A.23. Cabin zone parameters GUI

Figure A.24. Cabin zone wall parameters GUI
Figure A.25. Cabin zone passengers GUI

Figure A.26. Cabin zone heat transfer GUI

Figure A.27. Air-air heat exchanger parameters GUI

Figure A.28. Air-air heat exchanger fin parameters GUI
Figure A.29. Mixing junction GUI

Figure A.30. Turbine GUI
Appendix B

Variable Fidelity Modeling Setup

This section presents the modeling details for simulating variable fidelity systems described in Chapter 3.5.

Figure B.1 shows the high-level layout of the simulation. Reference and disturbance signals are loaded from workspace variables and scaled appropriately. The High-Low Switch block contains the supervisory logic described in Chapter 3.3. The Controller actuates the VCS to maintain temperature in the room. Double clicking on the High-Low Switch block brings up a GUI (Figure B.2) for defining $K_{\text{filter}}$ and $t_{\text{dwell}}$ of the supervisory logic.

Figure B.1. Layout of the variable fidelity VCS model

Figure B.2. Supervisory logic GUI
The supervisory logic, as implemented in Simulink, can be seen in Figure B.3. The reference (ref) and disturbance (dist) signals are passed through their respective filters, then the absolute value is taken, and each signal is scaled by the respective $K_{\text{filter}}$. This value is compared to a threshold of one. The Hi/Lo Switch block (Figure B.4) is responsible for applying $t_{\text{dwell}}$. If either of the signals are one, then a switch will be triggered. The Switch block ensures that the trigger is equal to one for the first 200 seconds of the simulation in order to help initialize both models to improve switching transients at the first switch.

**Figure B.3. Supervisory logic**

The Detect Change block of Figure B.4 has an initial condition of zero. The Memory block has an initial condition of $-\text{hold\_time}$. The Switch has a threshold of $\approx 0$. The Saturation block has a lower limit of zero and an upper limit of infinity. The Gain has the value $1/\text{hold\_time}$.

**Figure B.4. Hi/Lo switch block diagram**

The VCS is a traditional four-component system with an EEV (Figure B.5), a compressor (Figure B.6), switched-fidelity evaporator, and switched-fidelity condenser. Signals have to be specially routed to maintain causality within the simulation. The following set of images are intended to help a user understand how these signals are routed and the correct input/output of each component.
The evaporator and condenser are capable of switching between a finite volume model and moving boundary model. The setup for switching is shown in Figure B.7 for the evaporator and Figure B.8 for the condenser. The Trigger input is from the High/Low Switch block of Figure B.1. Only outputs that are communicated between components are switched between models. The Sample and Hold blocks are set to trigger on falling edges. The addition and subtraction of model outputs between the finite-volume and moving boundary models is to ensure that outputs to other system components track what the finite-volume model would output even when it is not actively calculating outputs.
Figure B.7. Switched evaporator setup

Figure B.8. Switched condenser setup
Two PI controllers are used to monitor the temperature in the room and the superheat in the evaporator. Those controllers are shown in Figure B.9 and Figure B.10, respectively.

![Figure B.9. Controller framework for temperature regulation](image1)

![Figure B.10. Controller framework for superheat regulation](image2)
Appendix C

Hierarchical Control Code

Mission Definition

The following code is support code for the simulation studies in Chapter 6. This segment of code defines the mission profile and disturbance profile.

%% Mission profile
% Mission consists of 15 minutes on the ground/taxi; 5 minutes of high
% powered takeoff and climb; 15 minutes of climb; 65 minutes of cruise; 15
% minutes of descent; 5 minutes of landing and taxi
mssn_phase = [repmat(1,1,900), repmat(2,1,300), repmat(3,1,900), ...
    repmat(4,1,3900), repmat(5,1,900), repmat(6,1,300)];

%% Disturbance Profile
% mission time is 7200second - preview needs an extra 300 sec. at the end
% dist1 and dist2 are loads into the cabin, dist3 is load into the tank
dist(1,:) = [repmat(12,[1,600]), repmat(14,[1,900]), ...
    repmat(3,[1,3700]), repmat(2,[1,2300])];
dist(2,:) = [repmat(4,[1,800]), repmat(5,[1,1100]), repmat(7,[1,1880]),...
    repmat(25,[1,120]), repmat(7,[1,1500]), repmat(2,[1,2100])];
dist(3,:) = [repmat(17,[1,1400]), repmat(15,[1,900]), repmat(16,[1,500]),...
    repmat(14,[1,1500]), repmat(13,[1,1000]), repmat(21,[1,120]), ...
    repmat(13,[1,880]), repmat(14,[1,1200])];
Mission Profile Constraints

The following code is support code for the simulation studies in Chapter 6. This segment of code defines the mission phase specific operating parameters and constraints.

```matlab
%% Mission profile specification
% Mission phases:
% 1 - ground/taxi
% 2 - takeoff
% 3 - climb
% 4 - cruise
% 5 - descent
% 6 - landing
%

%% Constraints for all mission phases
% cabin and tank constraints
Tcab_min     = 20;          % min cabin temperture °C
Ttnk_min     = 0;           % min tank temperture °C
Ttnk_max     = 45;          % max tank temperture °C

% Cooling Capacity Operation
Qvcs_1       = .1;          % Standby Mode VCS cooling
Qvcs_2       = 1.2;         % Lowerbound for VCS continuous cooling
Qvcs_3       = 6;           % Upperbound for VCS continuous cooling
Qvcs_2a      = 3;           % Lowerbound for VCS continuous air cooling
Qvcs_3a      = 8;           % Upperbound for VCS continuous air cooling
Qacm_1       = .15;         % Standby Mode ACM cooling
Qacm_2       = 1.2;         % Lowerbound for ACM continuous cooling
Qacm_3       = 3;           % Upperbound for ACM continuous cooling
Qacm_2a      = 3;           % Lowerbound for ACM continuous air cooling
Qacm_3a      = 8;           % Upperbound for ACM continuous air cooling

% Power consumption
Wvcs_1       = .5;          % Standby Mode VCS power
Wvcs_2       = 2;           % Lowerbound for VCS continuous power
Wvcs_3       = 5;           % Upperbound for VCS continuous power
Wvcs_2a      = 2.5;         % Lowerbound for VCS continuous power
Wvcs_3a      = 6;           % Upperbound for VCS continuous power
Wacm_1       = .5;          % Standby Mode ACM power
Wacm_2       = 2.5;         % Lowerbound for ACM continuous power
Wacm_3       = 6;           % Upperbound for ACM continuous power
Wacm_2a      = 2;           % Lowerbound for ACM continuous power
Wacm_3a      = 5;           % Upperbound for ACM continuous power

%% Constraints during ground/taxi
% VCS, ACM, ram constraints
Qsys_min(1)   = 2;          % min Q dot of the thermal sys
```

124
Qsys_max_air(1) = 6; % max Q dot the thermal sys can dump to air
Qsys_max_tnk(1) = 10; % max Q dot the thermal sys can dump to tank
Qram_max(1) = 5; % max Q dot of the ram air HX
Qram_min(1) = 0; % min Q dot of the ram air HX
Tcab_max(1) = 24; % max cabin temperature °C
Tcab_nom(1) = 22; % nominal cabin temp °C

%% Constraints during takeoff
% VCS, ACM, ram constraints
Qsys_min(2) = 2; % min Q dot of the thermal sys
Qsys_max_air(2) = 6; % max Q dot the thermal sys can dump to air
Qsys_max_tnk(2) = 10; % max Q dot the thermal sys can dump to tank
Qram_max(2) = 5; % max Q dot of the ram air HX
Qram_min(2) = 0; % min Q dot of the ram air HX
Tcab_max(2) = 25; % max cabin temperature °C
Tcab_nom(2) = 22.5; % nominal cabin temp °C

%% Constraints during climb
% VCS, ACM, ram constraints
Qsys_min(3) = 2; % min Q dot of the thermal sys
Qsys_max_air(3) = 10; % max Q dot the thermal sys can dump to air
Qsys_max_tnk(3) = 10; % max Q dot the thermal sys can dump to tank
Qram_max(3) = 8; % max Q dot of the ram air HX
Qram_min(3) = 0; % min Q dot of the ram air HX
Tcab_max(3) = 23; % max cabin temperature °C
Tcab_nom(3) = 21.5; % nominal cabin temp °C

%% Constraints during cruise
% VCS, ACM, ram constraints
Qsys_min(4) = 1; % min Q dot of the thermal sys
Qsys_max_air(4) = 12; % max Q dot the thermal sys can dump to air
Qsys_max_tnk(4) = 12; % max Q dot the thermal sys can dump to tank
Qram_max(4) = 10; % max Q dot of the ram air HX
Qram_min(4) = 0; % min Q dot of the ram air HX
Tcab_max(4) = 21.5; % max cabin temperature °C
Tcab_nom(4) = 21; % nominal cabin temp °C

%% Constraints during descent
% VCS, ACM, ram constraints
Qsys_min(5) = 2; % min Q dot of the thermal sys
Qsys_max_air(5) = 8; % max Q dot the thermal sys can dump to air
Qsys_max_tnk(5) = 10; % max Q dot the thermal sys can dump to tank
Qram_max(5) = 7; % max Q dot of the ram air HX
Qram_min(5) = 0; % min Q dot of the ram air HX
Tcab_max(5) = 23; % max cabin temperature °C
Tcab_nom(5) = 21.5; % nominal cabin temp °C

%% Constraints during landing
% VCS, ACM, ram constraints
Qsys_min(6) = 2;  % min Q dot of the thermal sys
Qsys_max_air(6) = 6;  % max Q dot the thermal sys can dump to air
Qsys_max_tnk(6) = 10;  % max Q dot the thermal sys can dump to tank
Qram_max(6) = 6;  % max Q dot of the ram air HX
Qram_min(6) = 0;  % min Q dot of the ram air HX
Tcab_max(6) = 25;  % max cabin temperature °C
Tcab_nom(6) = 22.5;  % nominal cabin temp °C
System Definition

The following code is support code for the simulation studies in Chapter 6. This segment of code defines the models for use in simulating system dynamics and use in the MPC optimization.

```matlab
%% physical parameters
C_c        = 0800;    % Thermal capacitance of the cabin
C_t        = 2000;    % Thermal capacitance of the fuel tank
tau        = 10000;
Ts         = 60;      % sample time of the system-level model
alph       = exp(-Ts/tau);  % first order dynamics

%% System Model - discrete time model used by System-level controller
Asysd = alph * eye(2);
Bsysd = [ Ts/C_t   0
          -Ts/C_t
          -Ts/C_c -Ts/C_c   0
          Ts/C_c   Ts/C_c   0
        ];
Fsysd = [0        0
         Ts/C_t
         Ts/C_c  Ts/C_c  0
        ];
Csysd = eye(2);
Dsysd = zeros(size(Csysd,1),size([Bsysd Fsysd],2));

%% System Dynamics - actual continuous model used for simulation
Asyc = -1/tau*eye(2);
Bsysc = [ 1/C_t   0
          -1/C_t
          -1/C_c -1/C_c   0
          1/C_c   1/C_c   0
        ];
Fsysc = [0        0
         1/C_t
         1/C_c  1/C_c   0
        ];
Csysc = eye(2);
Dsysc = zeros(size(Csysc,1),size([Bsysc Fsysc],2));

clear C_c C_t alpha
```
System-level Code

The following code is for the simulation studies in Chapter 6. This is the system-level controller algorithm. It is intended to be used in a Simulink environment with the Interpreted MATLAB Function block. The block calls are as follows,

**MATLAB function:** sys_level([u(1) u(2)] , [u(3:20)] , [u(21) u(22)] , u(23), u(24))

**Output dimensions:** 4

**Sample time:** 60

The first two inputs to the block are current temperatures of the fuel tank and cabin, respectively. Inputs 3 to 20 are the disturbance loads at the current step and 300 seconds into the future. For each load there are 6 signals spaced out at 60 second increments. The first six signals should be for the first load into the cabin, the second six signals for the second load into the cabin, and the final six signals should be the load into the fuel tank. Inputs 21 and 22 are temperature set points for the fuel tank and cabin, respectively. Input 23 is a clock signal, and input 24 is the current mission phase.

Outputs from the block include the three control variables, $\dot{Q}_{\text{tank}}, \dot{Q}_{\text{amb}}, \dot{Q}_{\text{ram}}$ and the work consumed by the ram air loop.

```matlab
function y = sys_level(temp,dist,set_points,time,mp)

% persistent variables are stored in memory for each call of the function.
% Since the controller is defined and compiled during the first iteration,
% it needs to be stored in memory for future calls of the controller. The
% same is true for other variables that are used at each call of the func.
persistent controller x0 n_dist M prev_mp n_states n_inputs

% at the first call of the controller, initialize variables
if time == 0
    prev_mp = 1;
    n_states = 6; % number of states
    n_inputs = 3; % number of inputs
    n_dist = 3; % number of disturbances
    x0 = zeros(n_states,1); % initializing the state vector
    M = 6; % length of preview
end

% define the controller at the initial call, or re-define the controller if
% the mission phase changes (alternatively these can be precompiled)
if time == 0 || mp ~= prev_mp
    %% Setup variables
    sys_param(); % call system parameters
```

128
mission_prof_constraints();  % call mission constraints
prev_mp = mp;          % saving for next call of function

%% Augmented system
% defining the system with augmented temperature variables for the
% accumulation of error and deviations from the set point
A_bar = [Asysd zeros(2,4);  eye(2) eye(2) -eye(2);  zeros(2,4) eye(2)];
B_bar = [Bsysd; zeros(4,3)];
W_bar = [Wsysd; zeros(4,3)];
C_bar = [eye(2) zeros(2,4)];

%% define controller using yalmip
yalmip('clear')
N = 10;         % prediction horizon

% upper and lower contraints on inputs
ul = [Qsys_min(mp); Qsys_min(mp); Qram_min(mp)];
uu = [Qsys_max_tnk(mp); Qsys_max_air(mp); Qram_max(mp)];

% upper and low constraints on states
xl = [Ttnk_min; Tcab_min; -inf*ones(n_states-2,1)];
xu = [Ttnk_max; Tcab_max(mp); inf*ones(n_states-2,1)];

% declaring variables for YALMIP
u = sdpvar(repmat(n_inputs,1,N),ones(1,N));    % inputs
x = sdpvar(repmat(n_states,1,N+1),ones(1,N+1)); % states
d = sdpvar(repmat(n_dist,1,N),ones(1,N));     % disturbances
slack = sdpvar(repmat(n_states,1,N+1),ones(1,N+1)); % slack term

% penalty on states
C = zeros(n_states);
C(1,1) = 10;  C(2,2) = 10;               % P control
C(3,3) = 0;  C(4,4) = 10;               % I control
R = 1*eye(3);           % penalizing each Q_dot

% weighting term on the slack variable
gam_slack   = 1e3;

obj = 0;  con = [];    % initializing objective function and constraints

% update the objective function over the course of the pred. horizon
for k = 1:N
  % the linear term of the cost function is updated at each
  % prediction step to account for different temperature set points
  F = 2*[C(1,1)*x(k)(5)  C(2,2)*x(k)(6)  zeros(1,n_states-2)];

  % updating the objective function over the horizon
  obj = obj + x(k)'*C*x(k) + u(k)'*R*u(k) - F*x(k);

  % adding the slack variable for temperature soft constraints
  obj = obj + gam_slack*slack(k)'*eye(6)*slack(k);

  % linear system - only preview to M dist. and assume const. after
if \( k > M \)
\[
\text{con} = \{\text{con}, x(k+1) == A_{\text{bar}}x(k) + B_{\text{bar}}u(k) + W_{\text{bar}}d(M)\};
\]
else
\[
\text{con} = \{\text{con}, x(k+1) == A_{\text{bar}}x(k) + B_{\text{bar}}u(k) + W_{\text{bar}}d(k)\};
\]
end

% input constraints (all Q_dot within bounds)
con = \{\text{con}, 0 <= u(k) <= uu\};

% state constraints with slack term on temperatures
con = \{\text{con}, x_l <= x(k) <= x_u + \text{slack}(i,k)\};
end

% define the MPC
controller = optimizer(con, obj, [], \{x(1), \text{d}(1:M)\}, u(1));

d % resetting the "integral" term of the state matrix whenever temperature drops below the set point
if \( \text{temp}(2) < \text{set\_points}(2) \)
x0(4) = 0;
end

% updating the state matrix for the optimization
x0(1) = \text{temp}(1); % tank temperature
x0(2) = \text{temp}(2); % cabin temperature
x0(3) = x0(3) + \text{temp}(1) - \text{set\_points}(1); % tank accumulated error
x0(4) = x0(4) + \text{temp}(2) - \text{set\_points}(2); % cabin accumulated error
x0(5) = \text{set\_points}(1); % tank temp. setpoint
x0(6) = \text{set\_points}(2); % cab. temp. setpoint

% transform vector of disturbance preview to matrix
\text{dist} = \text{vec2mat}(\text{dist}, M);

% for \( i = 2:M \)
\text{dist}(:,i) = \text{dist}(:,1); % no preview
% end

\text{d} = \text{mat2cell}(\text{dist}, [\text{n\_dist}], [\text{ones}(1,M)]); % convert matrix to cell
\text{Inputs} = \text{cell}(1, \text{M}+1); % pre-allocate size of controller input cell
\text{Inputs}(1) = \{x0\}; % state vector as 1st cell
\text{Inputs}(2:2+\text{M}) = \text{d}; % disturbance preview as remaining cells

% pass the controller the inputs and get the optimized decision variables
\text{u\_MPC} = \text{controller}(\text{Inputs});

% calculating work consumed by the ram air loop
if u\_MPC(3) < 1e-4
\text{W} = 0;
else
\text{W} = 1.2 + ((u\_MPC(3)-1.2)^2)/12;
end
\text{y} = [u\_MPC; \text{W}]; % output of controller decisions and power consumed
**Subsystem-Level Controller Code**

The following code is for the simulation studies in Chapter 6. This is the subsystem-level controller algorithm. It is intended to be used in a Simulink environment with the *Interpreted MATLAB Function* block. The block calls are as follows,

- **MATLAB function**: `subsys_level([u(1) u(2)], u(3), u(4), u(5))`
- **Output dimensions**: 6
- **Sample time**: 10

Input 1 should be from the system-level controller commanding the amount of heat that should be dumped to the fuel tank. Input 2 should be from the system-level controller commanding the amount of heat that should be dumped to the ambient. Input 3 is a clock signal. Input 4 is a *Repeating Sequence Stair* that outputs integers from 1 to 6 at a sample rate of 10 seconds. Input 5 is the current mission phase.

Output 1 is the command to the VCS for amount of heat to dump to the tank. Output 2 is the command to the ACM for amount of heat to dump to the tank. Output 3 is the command to the VCS for amount of heat to dump to the ambient. Output 4 is the command to the ACM for amount of heat to dump to the ambient. Output 5 is the operational mode. Output 6 is the total power consumed.

```matlab
function y = subsys_level(currQdes,time,cntr,mp)

% persistent variables are stored in memory for each call of the function. % Since the controller is defined and compiled during the first iteration, % it needs to be stored in memory for future calls of the controller. The % same is true for other variables that are used at each call of the func. persistent controller N alpha1 beta1 alpha2 beta2 alpha3 beta3 alpha4 beta4 persistent Qvcs_tnk Qvcs_air Qacm_tnk Qacm_air u_MIQP Qvcs_1 Qacm_1 u_mode persistent power_con alpha5 alpha6 alpha7 alpha8 beta5 beta6 beta7 beta8 persistent Wvcs_1 Wacm_1

% at the first call of the function, the controller is defined if time == 0 % Setup Variables
sys_param(); % call system parameters
mission_prof_constraints(); % call mission constraints

% number of inputs corresponds to number of decision variables
n_inputs = 8;
u_ = zeros(8,1);
```

---

131
% Avoid explosion of internally defined variables in YALMIP
yalmip('clear')

% Optimization Problem
N = 6; % Prediction Horizon
gamma = .9; % Weighting factor between performance and efficiency
W_switch = 5; % Penalty on switching frequency

% state variable declared as a symbolic decision variable in YALMIP
Qdes = sdpvar(repmat(2,1,N),ones(1,N)); % Q_dot desired from sys lvl
u0 = sdpvar(repmat(8,1,1),ones(1,1)); % inputs from previous

% decision variable declared as a symbolic decision variable in YALMIP
u = sdpvar(repmat(n_inputs,1,N),ones(1,N));

% Objective function -- cooling
alpha1 = Qvcs_3 - Qvcs_2; % coefficient for uc, vcs
beta1 = Qvcs_2 - Qvcs_1; % coefficient for ub, vcs
alpha2 = Qacm_3 - Qacm_2; % coefficient for uc, acm
beta2 = Qacm_2 - Qacm_1; % coefficient for ub, acm

alpha3 = Qvcs_3 - Qvcs_2; % coefficient for uc, vcs
beta3 = Qvcs_2 - Qvcs_1; % coefficient for ub, vcs
alpha4 = Qacm_3 - Qacm_2; % coefficient for uc, acm
beta4 = Qacm_2 - Qvcs_1; % coefficient for ub, acm

% matrix containing coefficients for each system cooling capacities
H = [alpha1 beta1 alpha2 beta2 zeros(1,4);
     zeros(1,4) alpha3 beta3 alpha4 beta4];
G = blkdiag([alpha1 beta1],[alpha2 beta2],...%
            [alpha3 beta3],[alpha4 beta4]);

% operating points for the VCS and ACM
Q_low = [Qvcs_1 Qacm_1 Qvcs_1 Qacm_1]';
Q_high = [Qvcs_3 Qacm_3 Qvcs_3 Qacm_3]';
Q_stby = Q_low;

obj = 0; con = []; % initializing objective function and constraints

% input constraints - u_c exists in [0,1] and u_b exists in {0,1}
% if u_b is 0 then the corresponding u_c is squeezed to zero as well
for k = 1:N
  % boolean variables are 0 or 1
  con = [con, ismember(u{k}(2:2:end),[0 1])]; % u_b in {0,1}
  % continuous variables are between 0 and 1
  con = [con, 0 <= u{k}(1:2:end) <= u{k}(2:2:end)]; % u_c in [0,1]
  % cooling capacity must be within bounds
  con = [con, Q_low <= G*u{k} + Q_low <= Q_high]
end

% penalizing the difference between the desired Q dot over the horizon
% and what is commanded by the controller. Qvcs_1 and Qacm_1 are
% removed from this desired because they are added back later
obj = obj + gamma*((Qdes{1})'*N - H*u{1} - H*u{2}) ...
    - H*u{3} - H*u{4} - H*u{5} - H*u{6} - N*[Qvcs_1+Qacm_1; 0])' * ...
    ((Qdes{1})'*N - H*u{1} - H*u{2} - H*u{3}) ...
    - H*u{4} - H*u{5} - H*u{6} - N*[Qvcs_1+Qacm_1; 0]);

% OBJECTIVE FUNCTION -- work
alpha5 = Wvcs_3 - Wvcs_2; % coefficient for uc, vcs
beta5  = Wvcs_2 - Wvcs_1; % coefficient for ub, vcs
alpha6 = Wacm_3 - Wacm_2; % coefficient for uc, acm
beta6  = Wacm_2 - Wacm_1; % coefficient for ub, acm

alpha7 = Wvcs_3 - Wvcs_2; % coefficient for uc, vcs
beta7  = Wvcs_2a - Wvcs_1; % coefficient for ub, vcs
alpha8 = Wacm_3 - Wacm_2; % coefficient for uc, acm
beta8  = Wacm_2a - Wacm_1; % coefficient for ub, acm

% matrix and vector containing coefficients for each system cooling capacity
G1 = diag([alpha5 alpha6 alpha7 alpha8]);
G2 = [beta5 beta6 beta7 beta8];

% constraints on inputs to minimize the work consumed by the system
for k = 1:N
    % updating the objective function over the horizon
    obj = obj + (1-gamma) * (u{k}(1:2:end)' * G1 * u{k}(1:2:end) ... 
          + G2 * u{k}(2:2:end));
end

% penalizing switching from previous control decisions
obj = obj + W_swtch*(u{1} - u0)' * (u{1} - u0);

% penalizing future switches
for k = 2:N
    obj = obj + W_swtch*(u{k} - u{k-1})' * (u{k} - u{k-1});
end

% setting solver settings to include gurobi - MIQP solver
opts = sdpsettings('solver','gurobi');
% optimizer(Constraints,Objective,Options,Parameters,DecisionVariables)
controller = optimizer(con,obj,opts,[Qdes{1};u0],[u(1:end)]);

% This section executes each time the function is called at time > 0
% the MIQP controller uses a finite horizon control approach so the % controller is only executed on the first time step
if cntr == 1
    % u_ is the initial conditions from the previous control sequence
    % it is used to reduce switching frequency
    u_MIQP = controller([currQdes(1);currQdes(2)];u_1);

    % calculating Q & W

133
for i = 1:N
    u_ = u_MIQP{i};
    Qvcstnk(i) = alpha1*u_(1) + beta1*u_(2) + Qvcs_1;
    Qvcsair(i) = alpha3*u_(5) + beta3*u_(6);
    Qacmntnk(i) = alpha2*u_(3) + beta2*u_(4) + Qacm_1;
    Qacmair(i) = alpha4*u_(7) + beta4*u_(8);
    power_con(i) = alpha5*u_(1) + beta5*u_(2) + Wvcs_1;
    power_con(i) = alpha7*u_(5) + beta7*u_(6) + power_con(i);
    power_con(i) = alpha6*u_(3) + beta6*u_(4) + Wacm_1 + power_con(i);
    power_con(i) = alpha8*u_(7) + beta8*u_(8) + power_con(i);
    mode(i) = [0 8 0 4 0 2 0 1] * u_;
end
end

y = [Qvcstnk(cntr) Qacmntnk(cntr) Qvcsair(cntr) ... 
     Qacmair(cntr) mode(cntr) power_con(cntr)];