SIMULTANEOUS MULTI-DESIGN POINT APPROACH TO GAS TURBINE ON-DESIGN CYCLE ANALYSIS FOR AIRCRAFT ENGINES

A Thesis
Presented to
The Academic Faculty

by

Jeffrey Scott Schutte

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
May 2009

SIMULTANEOUS MULTI-DESIGN POINT APPROACH TO GAS TURBINE ON-DESIGN CYCLE ANALYSIS FOR AIRCRAFT ENGINES

Approved by:

Dr. Dimitri Mavris, Advisor
Professor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Daniel P. Schrage
Professor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Brian German
Assistant Professor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Dr. Jimmy C. M. Tai
Research Engineer II
Aerospace Systems Design Lab
Georgia Institute of Technology

Dr. Richard Gaeta
Georgia Tech Research Institute
Georgia Institute of Technology

Scott Jones
Aerospace Engineer
NASA Glenn Research Center

Date Approved: 4/01/2009

To my wife Stacey and daughter Kristen

ACKNOWLEDGEMENTS

There are many people I would like to acknowledge for their support, guidance and encouragement throughout my dissertation process. First of course is my advisor, Professor Dimitri Mavris, for providing the opportunity to pursue my thesis. His leadership, work ethic, tirelessness, savvy and overall philosophy have provided me with an excellent role model and mentor.

I would like to thank the members of my thesis advisory committee; again Professor Mavris, Dr. Jimmy Tai for his knowledge and support throughout the development of the Multi-Point Design procedure, and Dr. Brian German whose enthusiasm for my work was contagious. Their guidance on the technical aspects of my work was valuable, as well as their contribution to the framing of my proposal and focus on the truly important aspects of my work.

I would also like to thank the members of my reading committee including Professor Daniel Schrage, Dr. Richard Gaeta from GTRI and Mr. Scott Jones from the NASA Glenn Research Center. After working on multi-point design for so long, it was nice to get a new perspective to help mold the final version of my thesis.

Next, I would like to express my thanks to all of the member past and present of the EDS team especially Dr. Michelle Kirby, Peter Barros, Keith Becker, Scott Wilson, Eric Hendricks, Paul Brett and Russell Young and Jorge deLuis and many others who have collaborated with me, not only for multi-point design but for all of the EDS development. In that regard, I would also like to thank Mr. Russ Denney for allowing me to unexpectedly pop in with random cycle questions.

Finally, I would like to thank my wife Stacey whose journey to her own dissertation has coincided with mine. This journey is over but a new one is just beginning. And thanks to my daughter Kristen who in the midst of chaos provides perspective on the really important things in life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ARP	Aerospace Recommended Practice
BPR	Bypass Ratio
CCD	Central Composite Design
CDS	Cycle Design Space
CdTh	Nozzle Discharge Coefficient
CMPGEN	Compressor Map Generation Program
CO_2	Carbon Dioxide
c_p	Specific Heat (ft-lbf/lbm-°R)
CPE	Component Performance Estimates (Parameter)
CPR	Compressor Pressure Ratio
CTS	Cycle Technology Space
CV	Nozzle Velocity Coefficient
DoE	Design of Experiments
DPMM	Design Point Mapping Matrix
DV	Design Variable
Fan_Deff	Fan Delta Adiabatic Efficiency
FAR	Fuel to Air Ratio
FLENTS	Future Large Engine/Nacelle Technology Study
FOM	Figures of Merit
FPR	Fan Pressure Ratio
γ	Ratio of Specific Heats
h	Specific Enthalpy (ft-lbf/lbm)
η_B	Burner Efficiency
η_C	Compressor Adiabatic Efficiency

η_R	Inlet Pressure Recovery
η_t	Cycle Thermal Efficiency
η_T	Turbine Adiabatic Efficiency
HC	Hydrocarbons
HHD	High Hot Day
HP	High Pressure (shaft)
HPCDP	High Pressure (Spool) Component Design Point
HPC	High Pressure Compressor
HPCPR	High Pressure Compressor Pressure Ratio
HPT	High Pressure Turbine
HPX	Horsepower Extraction
IR	Infra Red
ISA	International Standard Atmosphere
JAFE	Joint Advanced Fighter Engine
JSF	Joint Strike Fighter
LCT	Large Commercial Transport
LHC	Latin HyperCube
LHS	Left Hand Side (NPSS solver dependent term)
LP	Low Pressure (shaft)
LPC	Low Pressure Compressor
LPC_Deff	Low Pressure Compressor Delta Adiabatic Efficiency
LPCDP	Low Pressure (Spool) Component Design Point
LPCPR	Low Pressure Compressor Pressure Ratio
LPT	Low Pressure Turbine
LPT_Deff	Low Pressure Turbine Delta Adiabatic Efficiency
LTO	Landing and Take Off
$\dot{m}$	Mass Flow Rate (lbm/s)

MDP	Multiple Design Point
N	Shaft Speed (rpm)
Nc	Corrected Shaft Speed (rpm)
NASA	National Aeronautics and Space Administration
NFA	New Fighter Aircraft
NLA	New Large Aircraft
NOx	Nitrogen Oxides
NPSS	Numerical Propulsion System Simulation
OEO	One Engine Out
OPR	Overall (Compressor) Pressure Ratio
P	Pressure (psi)
PERF-R	Performance Requirement
$\dot{Q}$	Heat per Unit Mass Flow Rate (BTU-s/lbm)
RCS	Radar Cross Section
RHS	Right Hand Side (NPSS solver dependent term)
R&TD	Requirements and Technology Definition
SAE	Society of Automotive Engineers
SLS	Sea Level Static
SLSI	Sea Level Static Installed
SLSU	Sea Level Static Uninstalled
SOA	State of the Art
SFC	Specific Fuel Consumption
SPD	Single Point Design
T	Temperature (°R)
T3	Compressor Exit Temperature (°R)
T4	Combustor Exit Temperature (°R)
T41	Turbine Inlet Temperature (°R)

TKO	Take Off
TLIM	Technology Limit
TOC	Top of Climb
TOFL	Take Off Field Length (ft)
TOGW	Take Off Gross Weight (lb)
TSFC	Thrust Specific Fuel Consumption (lbm/lbf-hr)
T/W	Thrust to Weight Ratio
θ	Ratio of Temperature to Reference Temperature
Utip	Compressor Blade Tip Speed (ft/s)
Vmix	Mass Averaged Mixed Jet Velocity
W	Mass Flow Rate (lbm/s)
$\dot{W}$	Work per Unit Mass Flow Rate (ft-lbf-s/lbm)
W/S	Wing Loading
Xeff	Turbine Cooling Effectiveness

SUMMARY

Gas turbine engines for aircraft applications are required to meet multiple performance and sizing requirements, subject to constraints established by the best available technology level, that are both directly and indirectly associated with the aerothermodynamic cycle. The performance requirements and limiting values of constraints that are considered by the cycle analyst conducting an engine cycle design occur at multiple operating conditions. The traditional approach to cycle analysis chooses a single design point with which to perform the on-design analysis. Additional requirements and constraints not transpiring at the design point must be evaluated in off-design analysis and therefore do not influence the cycle design. Such an approach makes it difficult to design the cycle to meet more than a few requirements and limits the number of different aerothermodynamic cycle designs that can reasonably be evaluated.

Engine manufacturers have developed computational methods to create aerothermodynamic cycles that meet multiple requirements, but such methods are closely held secrets of their design process. This thesis presents a transparent and publicly available on-design cycle analysis method for gas turbine engines which generates aerothermodynamic cycles that simultaneously meet performance requirements and constraints at numerous design points. Such a method provides the cycle analyst the means to control all aspects of the aerothermodynamic cycle and provides the ability to parametrically create candidate engine cycles in greater numbers to comprehensively populate the cycle design space. The cycle design space represents all of the candidate engine cycles that meet the performance requirements for a particular application from which a "best" engine can be selected.

This thesis develops the multi-design point on-design cycle analysis method labeled simultaneous MDP. The method is divided into three different phases resulting in an 11 step process to generate a cycle design space for a particular application. The first

phase is the requirements and technology definition phase which defines the engine cycle problem to be analyzed through the establishment of requisite performance requirements and technology rules at the different design points and determines the overall engine architecture. The second phase is the MDP setup phase which establishes a set of nonlinear equations by formulating a system of nonlinear equations at on-design mode from design rules that couple the design points, performance requirements, technology rules and design variables. The key to the method is the understanding of the coupling of the performance between the different design points. The equations are divided into three categories; user defined equations specified by the cycle analyst based on the chosen design rules, engine component matching relations to ensure that conservation of mass and energy is maintained at each of the design points, and constraint relations which determine the feasibility of the candidate engines. The third phase is the MDP execution phase which populates the cycle design space by parametrically varying cycle design variables and then simultaneously finds the solution to the entire set of nonlinear equations with a modified version of the Newton-Raphson solver. For a specific cycle design problem, the first two phases are only performed once and the third phase repeated for each unique combination of design variables to create the cycle design space.

Through implementation of simultaneous MDP, a comprehensive cycle design space can be created quickly for the most complex of cycle design problems. Furthermore, the process documents the creation of each candidate engine providing transparency as to how each engine cycle was designed to meet all of the requirements. The cycle analyst is intricately involved in the simultaneous MDP method using their knowledge and expertise in the first two phases to define and setup the cycle design space, but are removed from the more time consuming task of finding each design that meets all of the requirements. As this process is left to the solver, the computational efficiency of the Newton-Raphson solver allows for the creation of numerous candidate engines to comprehensively cover the cycle design space.

The simultaneous MDP method is demonstrated in this thesis on a high bypass ratio, separate flow turbofan with up to 25 requirements and constraints and 9 design points derived from a notional 300 passenger aircraft with a large civil transport engine. Five separate experiments are designed to test different aspects of the simultaneous MDP method. The experiments highlight the transformation of the design rules into a system of nonlinear equations to be solved using the modified Newton-Raphson solver. The sensitivity of the solver to its initial iterate necessitated the development of a systematic approach to the generation of the initial iterate for a particular cycle design space. To ensure the highest solver convergence success rate possible, a multi-design point repair algorithm was devised for feasible candidate engines that initially fail to converge to the solution.

CHAPTER 1

INTRODUCTION AND MOTIVATION

This thesis addresses the need for a gas turbine engine aerothermodynamic cycle design method that incorporates multiple design points. One of the most important attributes of a gas turbine engine is its aerothermodynamic cycle which defines the engine performance at all operating conditions. For gas turbine engines, the aerothermodynamic cycle is established through the process of on-design cycle analysis. Traditionally, on-design cycle analysis involves the calculation of the aerothermodynamic cycle at a single design point. By selecting values of design variables at the design point such as fan pressure ratio (FPR), overall compressor pressure ratio (OPR), bypass ratio (BPR) and combustor exit temperature (T4), the cycle analyst can perform the on-design cycle analysis calculations to create a single engine cycle. Design variables, by definition, are independent cycle parameters which the cycle analyst has complete authority to set to obtain the desired engine performance. Unique combinations of design variables result in different candidate engine cycles and engine geometries. On-design cycle analysis can be repeated for multiple combinations of design variables to form a CDS comprised of numerous candidate engine cycles. The process of creating a CDS for a particular aircraft or mission is called parametric engine cycle design.

Depending on the design point designated for on-design cycle analysis, aircraft engines may seldom operate at the design point during a mission. Therefore, it is important to evaluate the performance of the candidate engine cycle at conditions other than the design point through the process of off-design cycle analysis. Often there are performance requirements and constraints imposed on the aerothermodynamic cycle at off-design conditions. Each unique set of operating conditions can be thought of as defining a single dimension of the aerothermodynamic cycle. Including additional performance requirements and constraints at operating conditions other than the design point increases the dimensionality of the design problem considerably. If a feasible candidate engine exists, that is an aerothermodynamic cycle that meets the performance requirements and constraints at all conditions for a specific combination of design variables, the aerothermodynamic cycle can be set using a single design point in on-design cycle analysis. However, adding performance requirements and constraints in dimensions other than the one defined by the designated design point changes the relationship of the design variables. While the design variables are still independent when all dimensioned are considered, the design variables may no longer be independent along a single dimension, as demonstrated in the following example.

In performing on-design cycle analysis for a single point, a cycle analyst selects a cruise condition as the design point. A thrust requirement is placed at an off-design point, take off (TKO), and the cycle analyst wants the T4 at this condition to be held constant for all designs. During the single point on-design cycle analysis, the cycle analyst must vary the design T4, shown as the vertical axis in Figure 1, in order to satisfy the thrust requirement at the appropriate off-design T4.

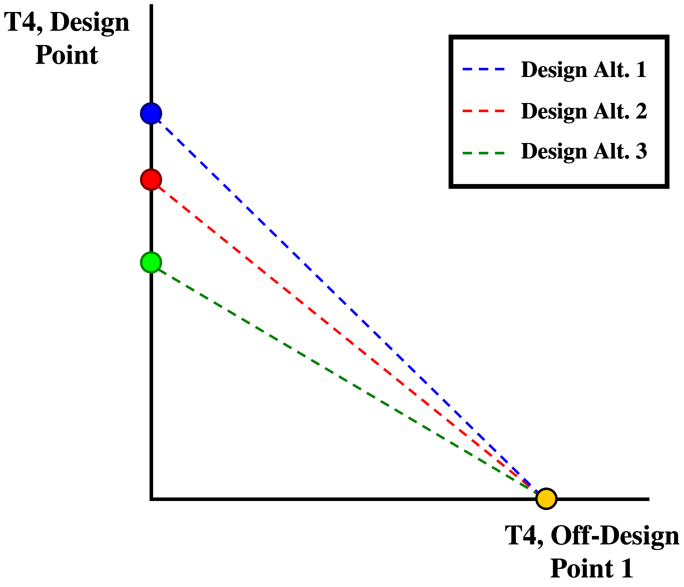


Figure 1: Example of Design variable Independence

Different combinations of the other design variables require different values of the design T4. Thus design T4 becomes a function of the other design variables in order to satisfy the off-design requirements at TKO for thrust and T4.

A single design point can still be used to set the cycle for design problems with requirements at different operating conditions, but the functional relationship between the design variables must be understood in order to find a satisfactory solution. On-design analysis using a single design point does not provide the information necessary to form these relations making it difficult to set the aerothermodynamic cycle and meet all of the performance requirements and constraints. The consequence of assuming the design variables are independent and using single point on-design cycle analysis is that the topography of the CDS morphs from the feasible topography. The feasible topography of the CDS is formed by the candidate engine cycles which satisfy all of the performance requirements and constraints for each combination of the design variables and an assumed level of technology. Selecting an engine for a particular aircraft mission from the CDS created using single point design can lead to an engine that is either suboptimal in its performance or an unrealizable design.

A method incorporating multiple design points is needed that can control the design variables at the appropriate operating conditions to maintain their independence and meet the performance requirements and constraints in all dimensions. In essence such a method would provide the necessary information to form relations between design variables along any single dimension. By using a multiple point design method for the aerothermodynamic cycle, a feasible CDS can be created to match the requirements constraints. A multiple design point method must be robust enough to handle several requirements and requirement types at various operating conditions for multiple engine architectures and applications. The method must be able to incorporate constraints, both those that directly impact the aerothermodynamic cycle and those that impact other aspects of the design such as mechanical limitations, to evaluate the feasibility of the engine. In addition, the method must be robust enough to ensure a candidate engine cycle can be found that meets all of the performance requirements and constraints for a wide range of design variables. Finally, the method must efficiently find the candidate engine cycle as the complexity of the cycle model and dimensionality of the design problem increase.

This thesis is divided into six chapters. The first chapter provides an introduction to the use of parametric cycle engine design and the motivation for developing multi-point cycle design methods. The second chapter provides background material necessary to elaborate on the desire for addressing multiple design points and to benchmark on-design analysis methods available in the literature today. The third chapter delineates the multi-point on-design cycle analysis method and the theory behind the methods development. The fourth chapter addresses the implementation of the method while the fifth chapter describes, .analyses and discusses the results from the five research experiments necessary to test and validate the method. The final chapter contains concluding remarks regarding the success of the method, lessons learned during the research and some potential future work not addressed in this document. Throughout this document, several formalized statements will be highlighted to identify and summarize the contribution that a particular section has to the overall research.

In Chapter 1, a **Problem Statement** will be developed to focus the motivation to a specific, well defined problem. **Observations** will be noted throughout Chapter 2 to summarize a statement of insight gained from literature, anecdotal evidence, or through experience working with cycle modeling that will a be a subject of poignant interest to the research. There will be a primary **Research Question**, based on the problem statement, which will provide the general impetus for the research. Secondary Research Questions are located at the end of Chapter 2 and act as specific queries related to one or more of the Observations. A **Hypothesis** is a statement providing a possible solution to one or more Research Questions that can be tested with **Research Experiments**. The

Research Experiments, outlined in Chapter 4 and discussed in Chapter 5, are the mechanism by which each hypothesis is tested and provide the information necessary to create a plan of action for completing the research.

The languages of the engineer and mathematician have been assimilated throughout this document. Often, even within a specialty language, terms and phrases can have different meanings to different groups of people. To help alleviate any confusion or misunderstanding of a terms usage, a list of different terms and their definition as utilized in this document is provided in Appendix A.

The next section of this chapter provides more detail about the motivation behind several groups' creation and exploration of a CDS through on-design cycle analysis and expounds some of the benefits gained by each group from such a process. Understanding how CDSs are used will help explain the need for multi-point design. Information on how parametric engine cycle design has been performed in the past will be discussed in Chapter 2, but the third section of this chapter will outline some criteria for creating a feasible CDS that meets all performance requirements and constraints.

1.1 Utilization of Parametric Cycle Design Space

There are many groups that have a desire to perform parametric engine cycle design studies of the aerothermodynamic cycle because of its importance to a gas turbine engine's performance and commercial success. These groups not only include engine manufacturers, but also developers of advanced cycle designs, technology specialists and forecasters and policy makers with regards to aircraft noise and emissions regulations. Each of these groups has disparate motives for performing parametric engine cycle design. Engine manufacturers are trying to find the aerothermodynamic cycle for a selected engine architecture that produces the "best" engine depending on the application and the accompanying Figures of Merit (FOM). Advanced cycle designers incorporate variations and combinations of different cycles to maximize performance and take advantage of the latest available technology. Technologists use parametric engine cycle design to determine how the optimum aerothermodynamic cycle shifts with changing technology and to better understand the contribution from the new technology to the improvements and/or degradations in engine performance. Aviation environmental regulators explore the CDS created during the parametric engine cycle design process in order to better understand the interdependencies between noise and emissions metrics and their impact on potential new stringency policies. The following sections provide examples about each of these groups use of a CDS when performing parametric engine cycle design studies of the aerothermodynamic cycle.

1.1.1 Engine Manufacturers

For an engine manufacturing company, the development of a new gas turbine engine for an aircraft application is a project with high associated risk, in terms of development costs, time to market, and technology capability. The development cost of such a project can be enormous with cost well in excess of 1 billion dollars[1]. The Economist reported that General Electric "spent $1m a day for four-and-a-half years to design the GE 90 aero-engine"[2]. Developing additional ratings of a current engine with increased thrust would usually cost around half of what would be required for an entirely new engine with a new aerothermodynamic cycle[3]. Therefore, the decision to create a new engine with a different aerothermodynamic cycle to meet increased performance requirements would only be made after long and careful consideration.

In addition to the associated costs, the time required to develop an engine can be extensive. Often the development of the engine requires more time than the development of the airframe[4]. For the F-22 advanced tactical fighter, the initial development of the engines was started in 1983 under the JAFE (Joint Advanced Fighter Engine) program even as the initial capability requirements for the advanced tactical fighter were still being decided[5].

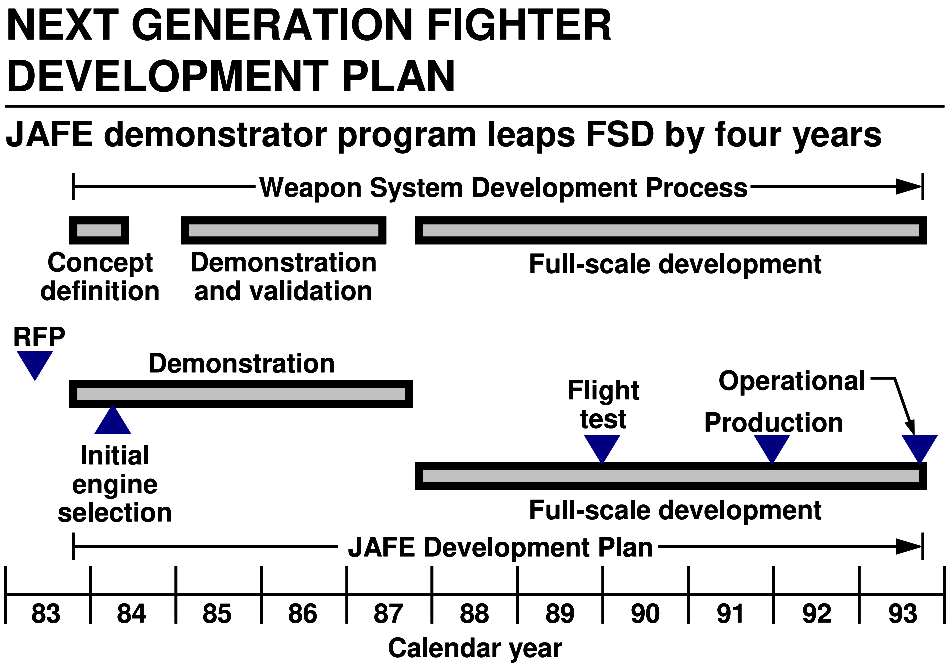


Figure 2: JAFE Development Plan [5]

The initial development plan, shown in Figure 2, called for a four year demonstration and validation phase followed by a six year full scale development phase. In actuality, while, the first development/evaluation flights occurred in 1990, the first F-22 was not delivered to the air force until 2003[6], almost 20 years after the initiation of the original development program.

Performance requirements were continually adjusted during the 20 year development of the engines for the F-22. One such change was the relaxation of the runway length requirement from 2,000ft to 3,000ft. This relaxation allowed the thrust reversers necessitated by the initial performance requirements to be dropped from the design, increasing the efficiency at which the aircraft could supercruise[6]. With such a long development time for the engine, estimates need to be made during the initial engine selection as to the level of technology that will be available at time of production. These estimates improve as the design progresses, but the risk is that the initial assumed technology level may not be achieved and the performance of the engine could be compromised. Therefore, when choosing the initial aerothermodynamic cycle from the CDS, the cycle analyst must consider robustness against changing requirements and technology levels as well as optimal performance. Improving technologies require more investment from the engine manufacturer as Jacquet and Seiwert propose "to assure development of robust, highly reliable systems. Methodology substantially more precise than traditional techniques is needed to enable superior strategic choices on engine configurations early in the development cycle, more specifically at the stage of assessing the feasibility of concepts[7]."

Engine manufacturers have generated structured development programs and design procedures to help mitigate the risks of cost, time and technology capability. A flowchart of a typical gas turbine engine design process from initial market research to production and field service is shown in Figure 3. This flowchart details some of the coupling of the design process elements that make the design of a gas turbine engine such an iterative process.

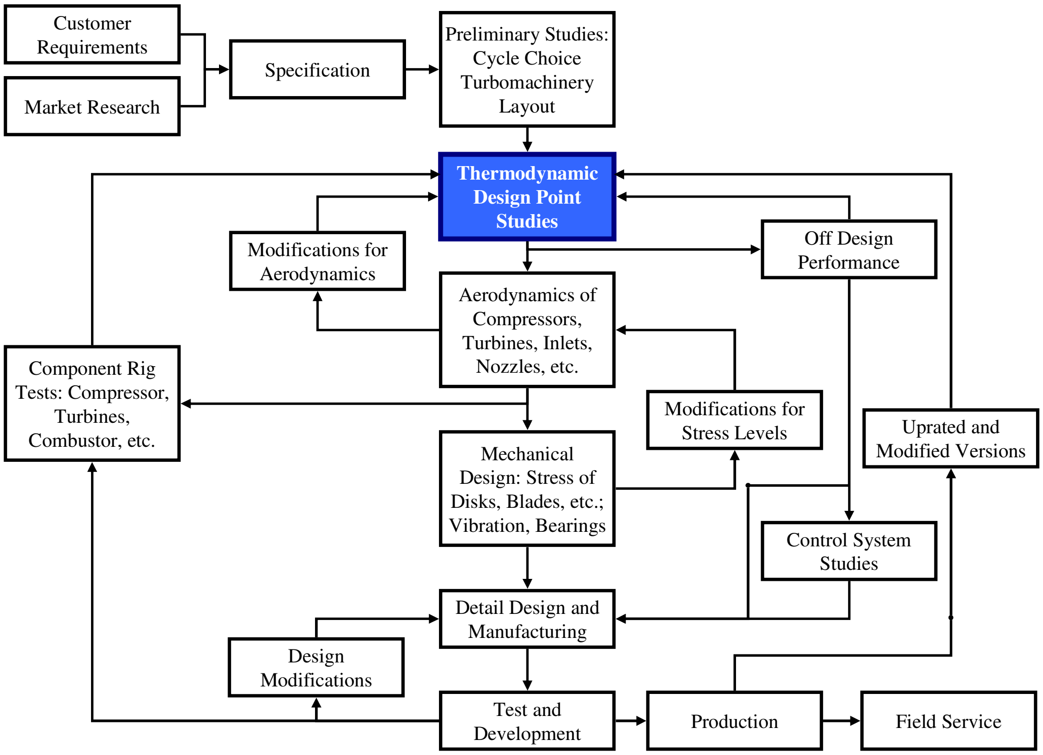

Customer Requirements
Market Research
Specification
Preliminary Studies: Cycle Choice Turbomachinery Layout
Thermodynamic Design Point Studies
Modifications for Aerodynamics
Off Design Performance
Aerodynamics of Compressors, Turbines, Inlets, Nozzles, etc.
Component Rig Tests: Compressor, Turbines, Combustor, etc.
Modifications for Stress Levels
Uprated and Modified Versions
Mechanical Design: Stress of Disks, Blades, etc.; Vibration, Bearings
Control System Studies
Detail Design and Manufacturing
Design Modifications
Test and Development
Production
Field Service

Figure 3: Typical Gas Turbine Design Procedure[8, 9]

Note the box labeled thermodynamic design point studies in Figure 3. This is the step in the design process where parametric engine cycle design is performed to generate the CDS. The two most commonly used thermodynamic performance metrics for air breathing engines are specific thrust and specific fuel consumption (TSFC or SFC) [10]. These performance metrics are determined exclusively by the aerothermodynamic cycle. From the thermodynamic design point studies in Figure 3, a “best” design is selected and then passed to other aspects of the design process. The studies may look like the example in Figure 4 for a supersonic fighter engine, which shows variations of subsonic cruise TSFC and the ratio of thrust required to augmented SLS thrust for a fighter engine create by parametrically varying the design variables FPR, compressor pressure ratio (CPR) and OPR.

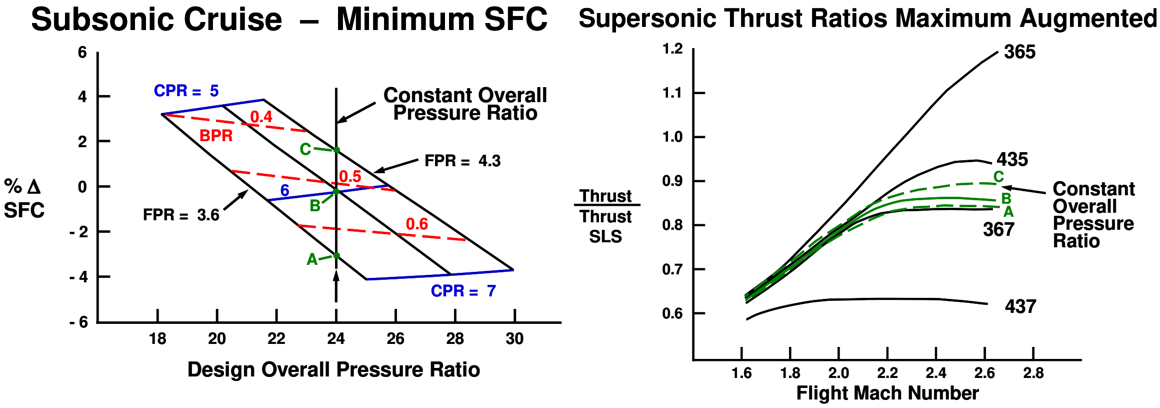


Figure 4: Example of Parametric Study for Fighter Engine[11]

The initial thermodynamic design point studies from Figure 3 corresponds to the initial engine selection milestone from Figure 2 and highlights the importance of parametric engine cycle design. As all future development of a new engine is dependent upon the initial engine cycle selection, it is imperative that the thermodynamic design point studies provide accurate and feasible results. In addition, these studies are never complete, but are continuously evolving as changing requirements and better performance estimates for the engine components become available. The initial selection of the aerothermodynamic cycle is therefore critically important in the development of a new engine as it establishes the engine performance capability and influences all other aspects of the engine design.

1.1.2 Development of Advanced Cycle Designs

Developers of advanced cycle designs construct variations of the Brayton cycle to maximize a gas turbine engine's performance. Some recent examples of these variations include the JSF dual-cycle engine[12], turbofan engines with interstage turbine burners[13, 14] and variable cycle engines[15-17]. Often these variations are made possible by improvements in technology or new aircraft design requirements. In the case of an interstage turbine burner study shown in Figure 5, a new design variable was created for control by the cycle analyst, the interstage burner exit temperature Tt4.5. The study created a CDS along the single dimension of Tt4.5 to evaluate its performance in both specific thrust and TSFC.

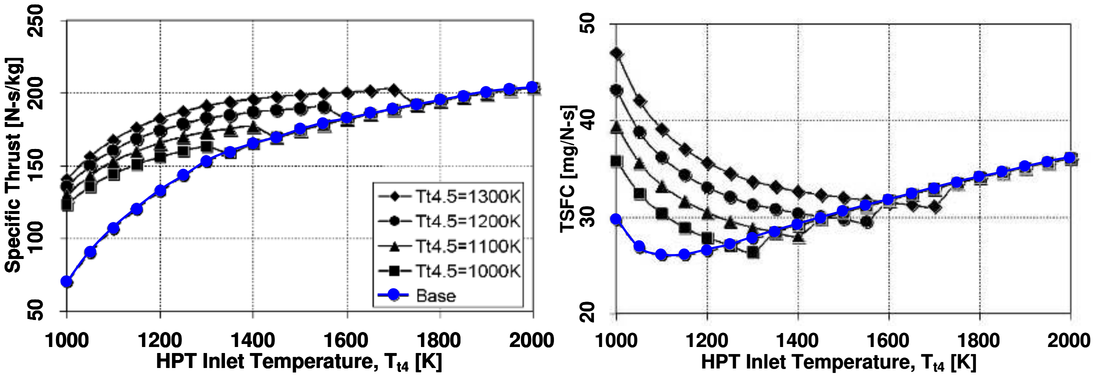


Figure 5: Example of Parametric Study for Interstage Turbine Burner[13]

1.1.3 Technologist and the Cycle Technology Space

Although related to the previous examples, technologists have a distinct motivation for the use of parametric engine cycle design. Technologists are not focused on the selection of the best design for a particular application, but are intent on quantifying the effects of improved technologies on future engine performance. Therefore, technologists create a cycle technology space (CTS). The CDS and CTS are closely related. The CDS is created by parametrically varying the cycle design variables. The CTS is created by parametrically varying both the cycle design variables and technology parameters. Thus the CDS is a single instance of the CTS with the technology parameters fixed to a particular technology level. By changing the technology parameters to different fixed values, a different CDS can be constructed representing a new technology level. The CTS is therefore composed of an infinite number of CDSs. To this end, a base engine cycle design with current state of the art (SOA) technology is usually established as a reference. The performance of this baseline is then compared to CDSs created with advanced technology. Advanced engine technologies can both positively and negatively impact the aerothermodynamic cycle performance. Improvements in technology parameters such as component efficiencies or material maximum allowable temperatures can improve the cycle performance. Other technologies, such as those focused on reducing gaseous emission or improving acoustic characteristics [18-21], can have a negative impact on the aerothermodynamic cycle performance. An example of a turbine material technology[22] study is shown in Figure 6. The metric of uninstalled SFC is compared for the CTS created by varying the turbine inlet temperature (T41) and OPR.

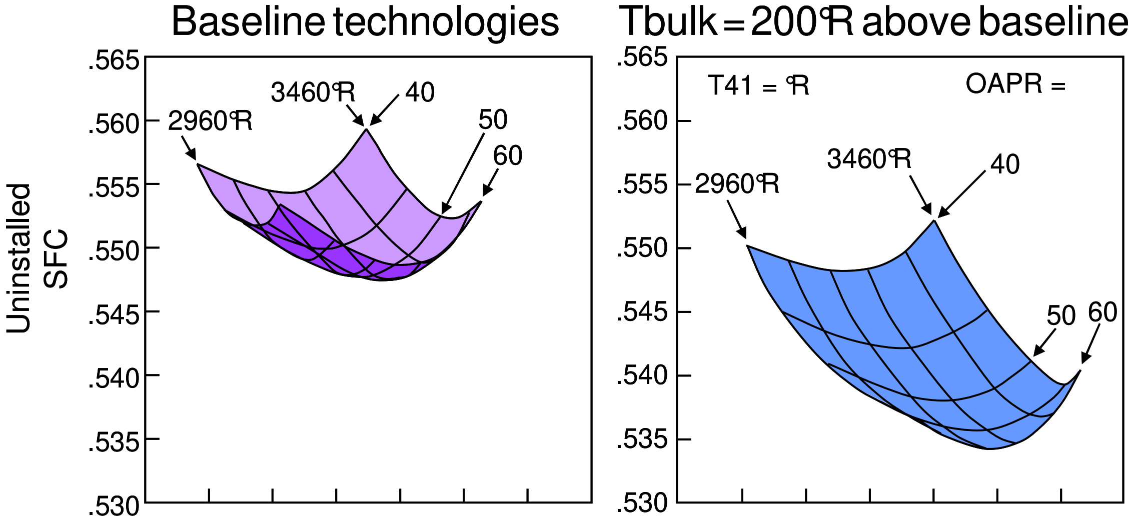


Figure 6: Cycle Technology Space Composed of Two CDSs[22]

The new technology not only improves uninstalled SFC by shifting the entire space downward, but changes the shape of the CDS itself. This results in the lowest uninstalled SFC for the baseline and advance turbine material technology residing at different values of the design variables. In the example, the minimum uninstalled SFC for the baseline study occurred at an OPR of 54, while the advanced technology minimum uninstalled SFC occurred at an OPR of 60. There was a 2.4 % reduction in minimum uninstalled SFC between these two "best" candidate engine cycles. Had the SOA OPR been used to calculate the "minimum" uninstalled SFC for the advanced technology CDS, the benefit would only have been 1.9%. This illustrates the requisite, when performing a technology study, to create and explore the entire CDS to correctly assess the technology impact to the engine's performance.

1.1.4 Environmental Policy Making

Aviation environmental policy makers also have a need to explore the CDS. According to Boeing's 2007 market outlook[23] reflected in Figure 7, almost 80% of the aircraft in service by 2026 will be new aircraft.

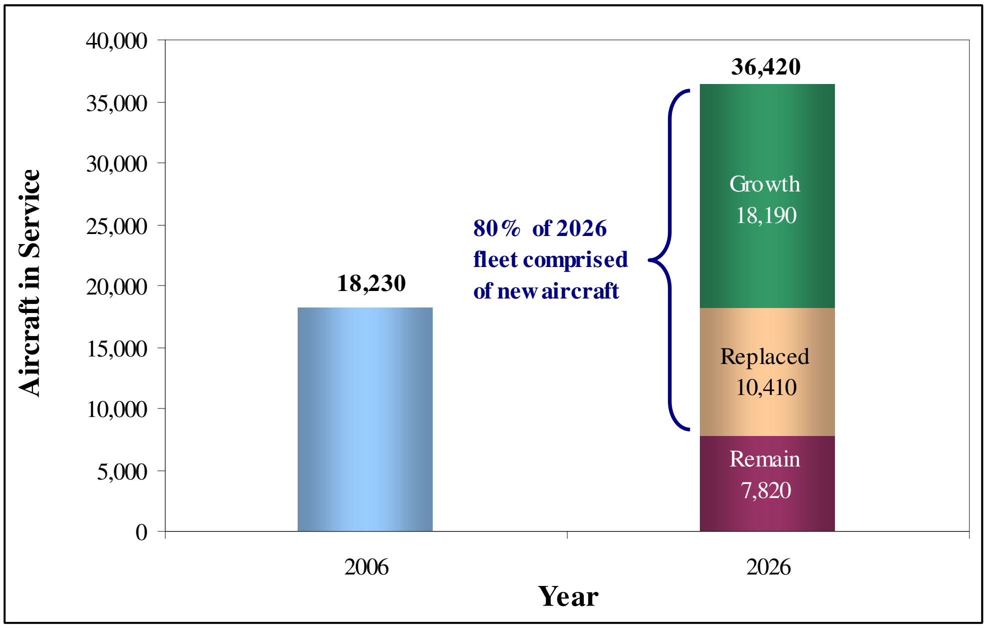


Figure 7: Boeing Marketing Projection of World Fleet[23, 24]

These new aircraft will fall under increasing stringency levels for both noise and emissions. In order to set new stringency levels, it is important to understand the interdependencies between noise and emissions due to the aerothermodynamic cycles through the creation of a CDS[25]. The process of setting stringency levels is as much a political exercise as it is a technical exercise. Policymakers may question the accuracy of the interdependencies produced from a CDS based on technology assumptions for aircraft 20 years in the future. However, Kraemer and King not that even though

> policymakers and analysts are sometimes cynical about the accuracy of models' estimates, they nonetheless support model use because they believe that if they do not use models and argue in numerical terms, their opponents will. In politics, "some numbers beat no numbers every time.[26]"

An example of such a CDS is shown in Figure 8, which shows the interdependencies for NOx emissions, cumulative noise margin and fuel burn.

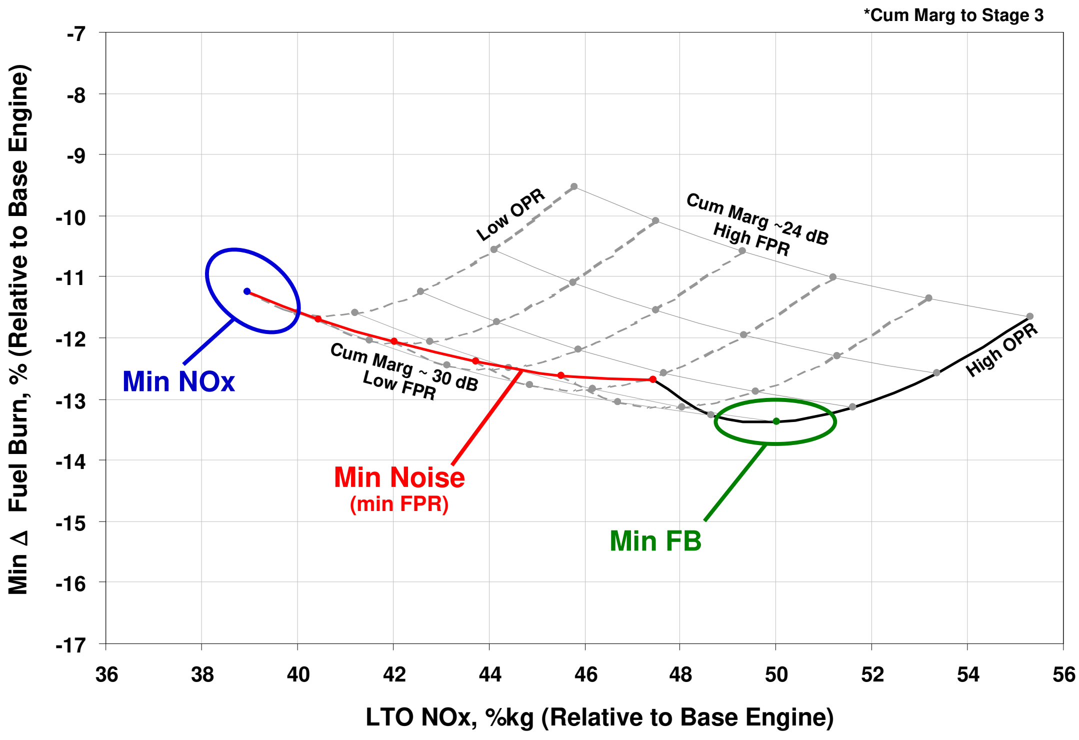


Figure 8: Example CDS for Environmental Policy[27]

Shakariyants et al write that “today’s civil airplane and engine manufacturers realize the tie between the environmental impact of air transport and sustainable growth of the industry. They continue developing technologies aimed to minimize the fuel burn and pollutant concentrations in the engine exhaust[28].” By better understanding the tradeoffs that must be made between these metrics, policy makers can make more informed decisions regarding new stringency regulations.

1.2 Criteria for Parametric Cycle Design

The goal for a cycle analyst performing parametric engine design is to compare feasible candidate engine cycles for a particular application. A candidate engine cycle represents a single engine in the CDS characterized by a specific combination of cycle design variables. As the definition implies, the cycle analyst wants to meet the feasibility criteria in order to make a valid comparison of candidate engine cycles. The CDS criteria ensure that each candidate engine cycle meets all of the performance requirements and constraints. The first criterion is that a consistent technology level be maintained throughout the CDS. The second criterion is that each engine in the CDS must achieve a consistent performance capability defined by a set of performance requirements. This leads to the problem statement that this thesis addresses.

Problem Statement: Develop an on-design cycle analysis method for gas turbine engines which generates engine aerothermodynamic cycles that meet performance requirements and constraints at multiple design points.

Such a method should allow the cycle analyst to parametrically generate candidate engine cycle designs that are consistent in technology level and performance capability while allowing the cycle analyst the freedom to control certain aspects of the aerothermodynamic cycle at various operating conditions. Adherence to these criteria will result in a feasible CDS. This leads to the question; how does one adhere to these criteria?

In order to maintain a consistent technology level, a set of technology rules must first be established. Technology rules describe how the technology parameters change as a function of the design variables. These rules can be divided into two categories; component performance estimates, and technology limits. Component performance estimates contains functions, tables or maps which estimate component technology parameters for a given technology level as a function of the design variables. This category would include items such as compressor efficiencies as a function of compressor pressure ratio. Component performance estimates does not require the component technology parameters to be the same throughout the CDS. Properly modeled, the component performance parameters are usually estimated as functions of the flight conditions and or design variables. Technology limits are constraints established by the technology level which can not be exceeded. Constraints are usually fixed regardless of the flight conditions or design variables. Constraining values for differing technology limits can be found at a variety of different operating conditions. Technology limits include items such as turbine inlet temperature limits.

Each candidate engine cycle achieves a consistent performance capability by meeting a set of cycle performance requirements. Cycle performance requirement are specific requisites defining the expected performance characteristics of the aerothermodynamic cycle throughout the operational envelope. Like the technology rules, the performance requirements can also be divided into two categories, performance targets and performance limits. The difference between the two categories is in how the performance requirement is treated during the on-design analysis. A performance target is matched to an exact value and can be considered as an equality constraint, where as the performance limit establishes a minimum performance requirement that is considered as an inequality constraint. It is possible that within the CDS certain performance requirements can switch from a target to a limit or a limit to a target based on the severity of each requirement. The most severe performance requirements become the targets while the remaining requirements are categorized as limits and usually set minimum levels of performance exceeded by the candidate engine cycle. The most typical performance requirements are net thrusts at different operating conditions.

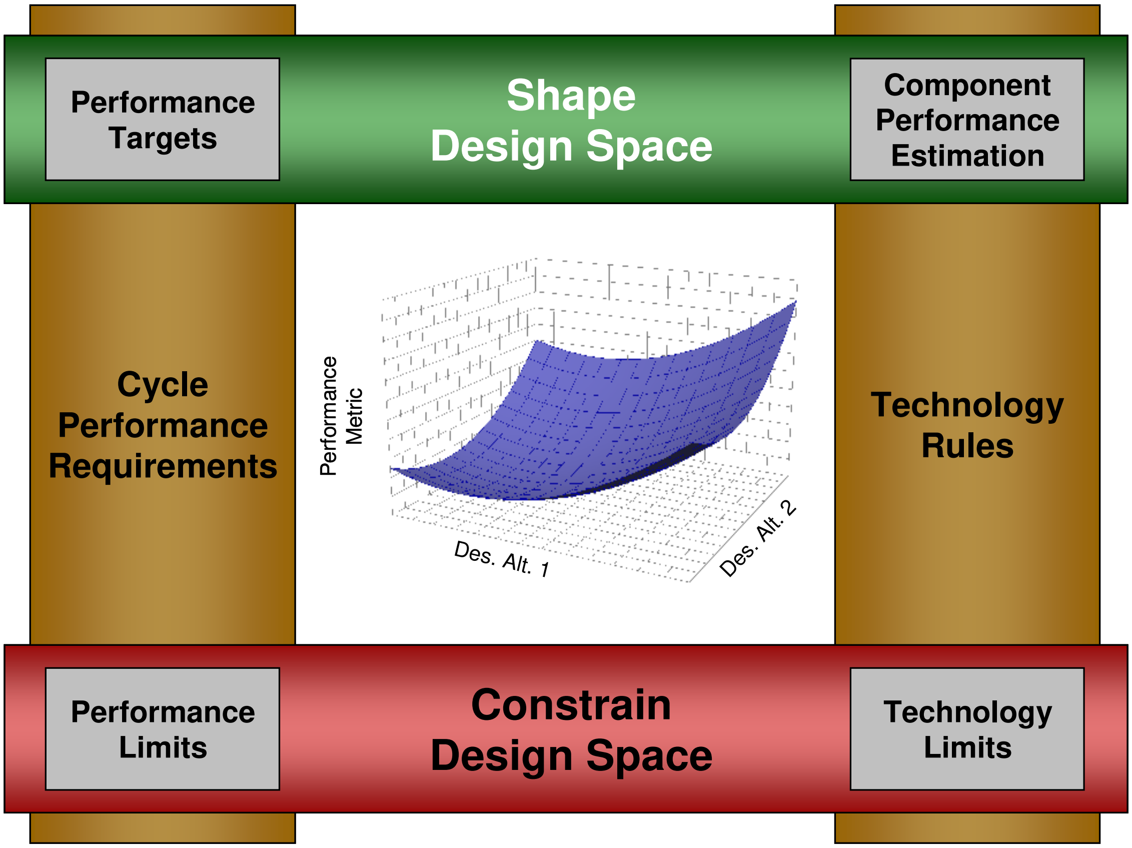


Figure 9: Effect of Technology Rules and Performance Requirements on CDS

Together with the technology rules, the performance requirements shape and constrain the CDS as depicted in Figure 9. The performance limits from the cycle performance requirements and technology limits from the design rules define the boundaries of the feasible CDS while the performance targets and component performance estimates shape the space within those boundaries.

CHAPTER 2

BACKGROUND INFORMATION

This chapter provides the background information necessary to understand the technical challenges associated with creating a feasible CDS. The chapter begins by briefly reviewing the ideal Brayton cycle for aircraft gas turbine engines, as this is the aerothermodynamic cycle to be analyzed by the simultaneous MDP method. The review will be followed by a section devoted to the creation of the CDS through the parametric engine cycle design process. The third section will provide background on the characteristics of the different design points to be considered and how the requirements and constraints that must be met by the aerothermodynamic cycle are established. The next section includes a benchmarking of the cycle design methods available in the literature. From the background information, additional research questions will be formed to more narrowly define the problem and generate a set of hypotheses..

In order to meet the criteria for producing the CDS the aerothermodynamic cycle must be evaluated at multiple design points. This leads to the primary research question to be addressed by this work that will focus the background information in this chapter.

Primary Research Question: How can an on-design cycle analysis method be developed that incorporates all desired cycle performance requirements and constraints at different design point and generates aerothermodynamic cycles in a robust and efficient manner to populate the cycle design space?

The method must be robust enough to handle several requirements and requirement types at various operating conditions for multiple engine architectures and applications. The method must be able to incorporate constraints, both those that directly impact the aerothermodynamic cycle and those influenced by the aerothermodynamic cycle that impact other aspects of the design, to evaluate the feasibility of the engine. In addition, the method must be robust enough to ensure a candidate engine cycle can be found that meets all of the requirements and constraints for a wide range of design variable values. Finally, the method must efficiently find the candidate engine cycle with higher complexity cycle models and increasing number of requirements.

2.1 The Ideal Brayton Cycle

The Brayton power cycle is the aerothermodynamic cycle model used for gas turbine engines. A discussion of the Brayton cycle and its use in aerospace applications can be found in multiple sources[9, 29-32]. Roth and Mavris[33, 34] also look at the cycle from the standpoint of work availability. A schematic of the Brayton cycle for a turbojet is shown in Figure 10. The cycle is divided into 5 elements: a diffuser, compressor, combustor, turbine and nozzle.

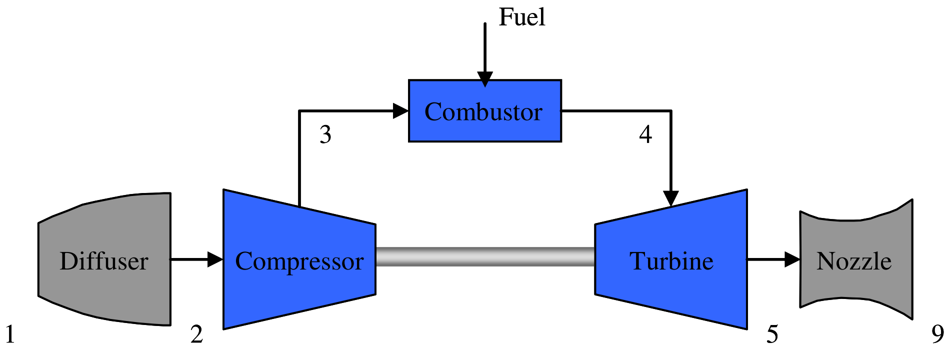


Figure 10: Brayton Cycle Turbojet Schematic

The diffuser decelerates the incoming flow relative to the engine. This process increases the pressure at the compressor inlet through the ram effect. The compressor further compresses the air to a high pressure. The combustor raises the temperature of the air by mixing the air with fuel and combusting the gas mixture. The gas mixture exits the combustor at a high pressure and temperature. The turbine expands the gas and extracts work. For aircraft applications, only the work required to drive the compressor and any accessories is extracted by the turbine. The remaining expansion of the gas occurs in the nozzle which accelerates the flow to a high exit velocity providing the propulsive force.

The cycle process is often shown in a T-S or temperature – entropy diagram. The ideal Brayton cycle for a turbojet is shown in Figure 11. The ideal Brayton cycle makes the following assumption about each of the processes:

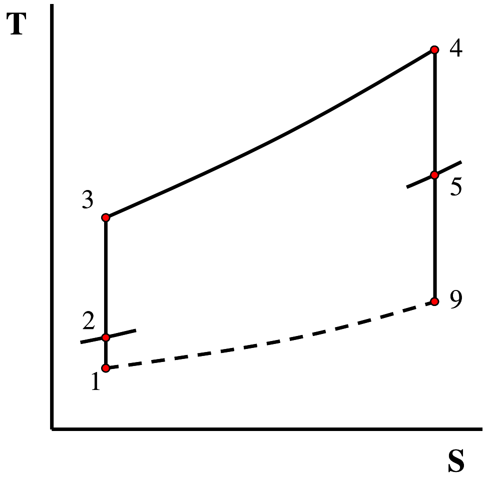


Figure 11: T-S Diagram for Ideal Brayton Cycle

- Each element is analyzed as a control volume at steady-state
- Process 1-2 (diffuser) is an isentropic compression
- Process 2-3 (compressor) is an isentropic compression
- Process 3-4 (combustor) is an isobaric heat addition
- Process 4-5 (turbine) is an isentropic expansion
- Process 5-9 (nozzle) is an isentropic expansion

There are two fundamental concepts of gas turbine performance that can be understood by analyzing the ideal Brayton cycle; the cycle thermal efficiency and maximum specific work. Looking at the gas generator only (compressor, burner, and turbine) the thermal efficiency is defined as the ratio of net work produced to the heat addition in the combustor:

$$\eta_t = \frac{\dot{W}_{net}}{\dot{Q}_{in}} = \frac{\dot{W}_{net}/\dot{m}}{\dot{Q}_{in}/\dot{m}} \tag{1}$$

Assuming a calorically perfect gas, expressions for the mass specific net work and heat addition can be found resulting in the thermal efficiency in the following form:

$$\frac{\dot{W}_{net}}{\dot{m}} = \frac{\dot{W}_t}{\dot{m}} - \frac{\dot{W}_c}{\dot{m}} = (h_4 - h_5) - (h_3 - h_2) = c_p[(T_4 - T_5) - (T_3 - T_2)] \tag{2}$$

$$\frac{\dot{Q}_{in}}{\dot{m}} = h_4 - h_3 = c_p(T_4 - T_3)$$

$$\eta_t = 1 - \frac{T_5 - T_2}{T_4 - T_3}$$

With some final rearranging and taking note of the isentropic flow relation, the thermal efficiency becomes:

$$\left(\frac{P_3}{P_2}\right)^{(\gamma-1)/\gamma} = \frac{T_3}{T_2} = \frac{T_4}{T_5} \tag{3}$$

$$\eta_t = 1 - \left(\frac{1}{P_3/P_2}\right)^{(\gamma-1)/\gamma} \tag{4}$$

The first fundament concept, obtained from examination of equation 4, is that the thermal efficiency increases with increasing compressor pressure ratio. This relationship is shown in Figure 12 for a gas with $\gamma = 1.4$. The relationship may seem to imply that the highest pressure ratio possible would be desired to increase thermal efficiency. However, this is not the case as illustrated by the second fundamental concept.

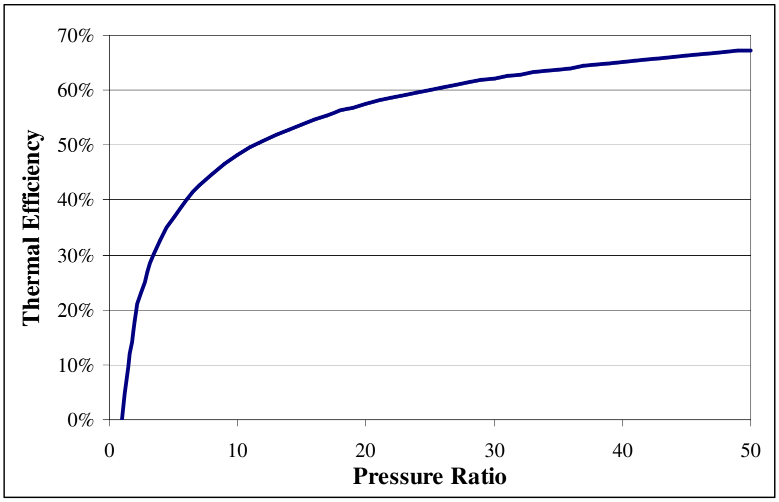


Figure 12: Thermal Efficiency of Ideal Brayton Cycle

For an ideal Brayton cycle the compressor and turbine inlet temperatures can be fixed based upon a given flight condition and maximum turbine material temperatures. In this instance an expression for the maximum specific work can be derived. By taking equation (2 and taking the derivative with respect to T_3:

$$\frac{d\dot{W}_{net}/\dot{m}}{dT_3} = c_p\left[\frac{dT_2}{dT_3} - \frac{dT_3}{dT_3} + \frac{dT_4}{dT_3} - \frac{dT_5}{dT_3}\right] = 0$$

$$0 - 1 + 0 - \frac{dT_5}{dT_3} = 0$$

$$-\frac{dT_5}{dT_3} = 1$$

Using the relations in equation 3 for isentropic flow shows:

$$T_5 = \frac{T_2 \cdot T_4}{T_3}$$

$$\frac{dT_5}{dT_3} = -\frac{T_2 \cdot T_4}{T_3^2}$$

$$T_3 = \sqrt{T_2 \cdot T_4}$$

$$\left(\frac{T_3}{T_2}\right)_{\text{max work}} = \sqrt{\frac{T_4}{T_2}} \tag{5}$$

Inserting equation 5 into equation (2 and rearranging, the maximum specific work can be given as:

$$\frac{\dot{W}_{net}}{\dot{m}} = c_p T_2 \left(\sqrt{\frac{T_4}{T_2}} - 1 \right)^2 \tag{6}$$

Equation 5 shows that for a given T2 and T4 there is an optimum compressor temperature ratio, and thus pressure ratio, that will provide the maximum specific work. This consideration is important for gas turbines intended for vehicle applications as the mass flow must be kept small to reduce the engine weight[32]. Therefore it is better to operate at a pressure ratio that maximizes specific work rather than one that increases thermal efficiency[35].

In order to facilitate communication regarding aerospace propulsion systems, the Society of Automotive Engineers (SAE) has published an Aerospace Recommended Practice (ARP) 755B that creates a common station designation for various engine architectures[36]. This station designation will be used throughout the remainder of this document. The station designations are used to locate the thermodynamic properties of the flow within the engine. An example of the station numbering for a separate flow turbofan engine is shown in Figure 13.

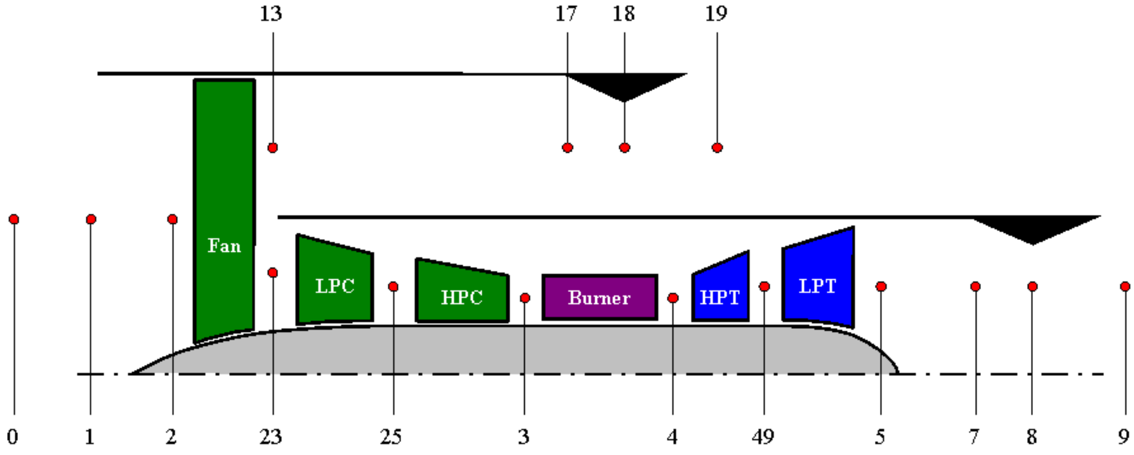


Figure 13: Example of Station Designation

The station numbering for the primary gas flow is listed in Table 1. To designate bypass streams, a numeric is placed in front of the station number signifying the number of the bypass stream. The numbering begins at 1 for the innermost bypass stream. For example, the core or primary stream nozzle throat station is designated 8, while the bypass stream nozzle throat is designated 18. Intermediate stations are designated by adding a numeric after the station numbers. The numbering increases in the direction of the flow. For example, the station between the Low Pressure Compressor (LPC) and High Pressure Compressor (HPC) is usually designated station 25.

Table 1. Numbering Station Designation

Number	Description
0	Free Stream
1	First station of interest to engine manufacturer
2	First compressor front face
3	Last compressor discharge or burner entrance
4	Burner discharge or first turbine entrance
5	Last turbine discharge
6	For mixer or afterburner
7	Engine exhaust nozzle interface
8	Exhaust nozzle throat
9	Exhaust nozzle discharge

2.2 Parametric Engine Cycle Design

Historically, the design of an aircraft engine begins with a study of the complete engine using an aerothermodynamic cycle analysis[37]. In Chapter 1, the parametric engine cycle design, (also referred to as propulsion system analysis[4]), was defined as the overall process which evaluates multiple candidate engine cycles for a particular aircraft or mission. This process can be decomposed into two sub processes as illustrated in Figure 14, cycle analysis, which contains both the on-design cycle analysis and the off-design cycle analysis and cycle selection which includes the flow path analysis, aircraft mission analysis, cost analysis and overall selection method[29, 38, 39]. The results of the cycle analysis process, in the form of candidate engine cycles that populate a CDS, are incorporated as inputs into the cycle selection process.

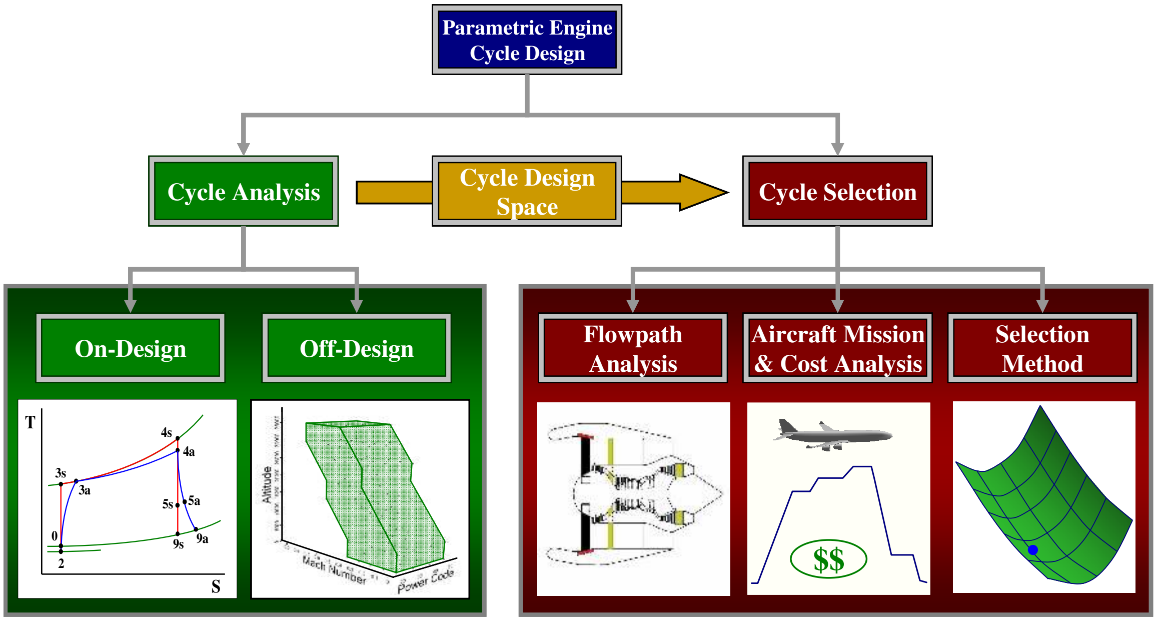


Figure 14: Parametric Engine Cycle Design Decomposition

Stricker notes the importance of the different elements of the cycle selection process which extend beyond the cycle analysis calculations when he writes:

> The advent of the computer has made early examination of numerous propulsion characteristics possible…In the early years of computers, engine selection was based primarily on cycle trade studies and the design engineer’s experience. Other elements such as installed performance, flowpath, and weight have to be put off for the detailed design part of the overall engine development process. This could result in the selection of an engine configuration, which was not fully optimized. In the worst case, the selected engine could not satisfy the aircraft requirements, necessitating a costly and time-consuming redesign[38].

2.2.1 Cycle Analysis Process

Cycle analysis is the process of determining thermodynamic performance for a particular engine as defined by a set of cycle design variables and technology parameters representing a specific technology level[9]. Design variables[10] are defined as independent cycle parameters that the cycle analyst has complete authority to set and are also referred to as cycle design parameters[30] or cycle design choices[9]. The design variables include parameters such as FPR, OPR, T4 and BPR for a separate flow turbofan. Cycle analysis determines the effect of each engine component on the thermodynamic properties of the working fluid as it flows through the engine[9]. Oates states that the objective of cycle analysis is to obtain estimates of the thermodynamic performance parameters in terms of design limitations, flight conditions and design choices[10].

Cycle analysis can be further subdivided into on-design analysis and off-design analysis. On-design analysis, also called design point analysis[30] or parametric cycle analysis[9], sets the aerothermodynamic cycle and sizes the engine to meet the cycle performance requirements at the design point. Changes in the values of the design variables will result in different engine geometries[30] representing a "rubber engine"[9, 40]. By parametrically varying the design variables a set or matrix of candidate engine cycles can be created to populate the CDS[41]. Off-design analysis[8, 10, 30] (also called engine performance analysis[9]) evaluates the thermodynamic performance of a fixed engine cycle for all conditions within the operational envelope. The operational envelope is the combination of all operating conditions and throttle settings within which the engine would be expected to operate and consists of three elements. The first element is composed of the ambient environmental conditions such as pressure, temperature, density, humidity and the speed of sound. Grouped together these parameters are referred to as the environmental envelope. The second element of the operational envelope consists of the flight velocity usually in terms of flight Mach number. The range of possible flight Mach numbers coupled with the environmental envelope form the flight envelope. In addition to the flight envelope, the throttle setting of the engine often referenced in terms of a power code is the third element that completes the composition of the operational envelope.

2.2.2 Cycle Selection Process

Cycle selection is the process of selecting from a CDS the "best" engine or engines based on a set of mission figures of merit. The determination of the appropriate engine cycles within the CDS for a particular mission must be done at the vehicle level as engine level metrics such as TSFC are inadequate for a comparison of the performance on the total system[22]. Therefore the cycle selection process is an integrated approach that uses candidate engine cycles from the CDS created from the cycle analysis process with flowpath analysis to determine the engine's weight and drag characteristics. The engine is then coupled to an airframe or multiple airframes and evaluated through vehicle mission analysis for a set of FOM[7, 29, 39]. The mission analysis can include evaluation of capabilities such as fuel burns, payloads, take off field lengths, noise, emissions and observability[42]. Additional analysis is performed to evaluate the different costs of the system[39]. Two examples of the entire parametric engine design process are shown in Figure 15 and Figure 16. In the first method, the parametric design process was focused on affordable capability. Thus the important FOM were the unit and life costs for the vehicle. The cycle analysis process is executed in step 1 of the method depicted in Figure 15, while steps 2-5 encompass the cycle selection process. Both methods initiate with the definition of the requirements and include flowpath analysis, mission analysis and cost analysis in order to select the appropriate aerothermodynamic cycle for the engine to produce the best vehicle performance.

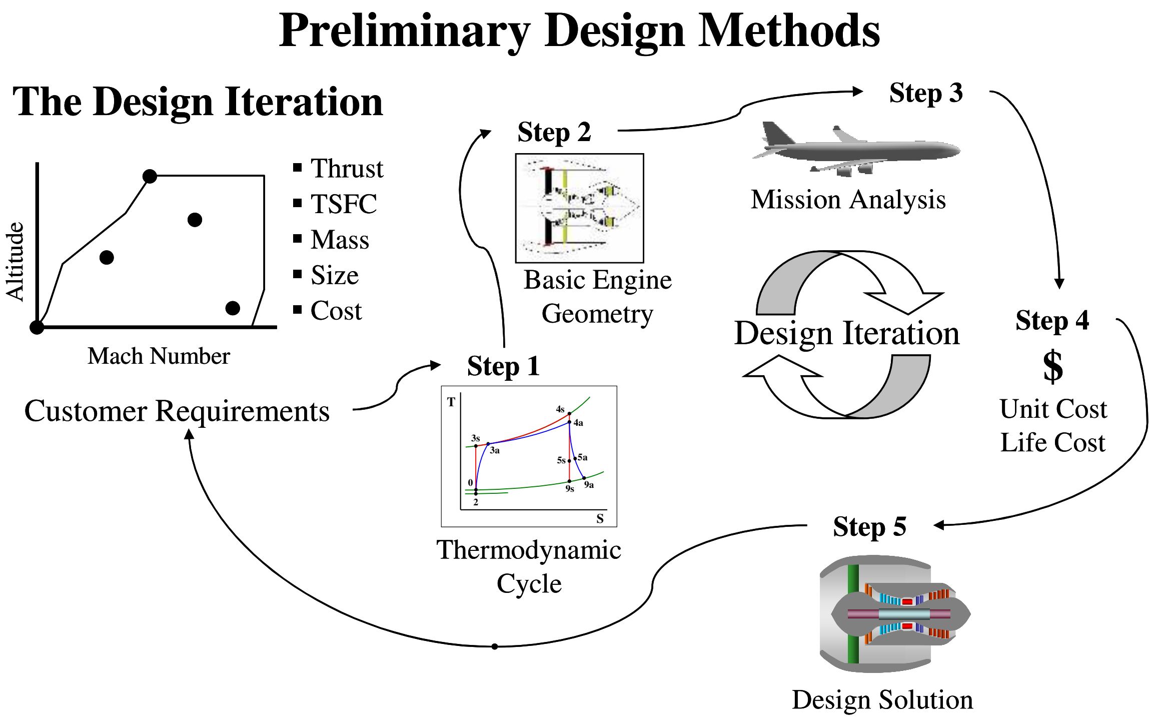


Figure 15: Parametric Cycle Design Process for Affordable Capability[39]

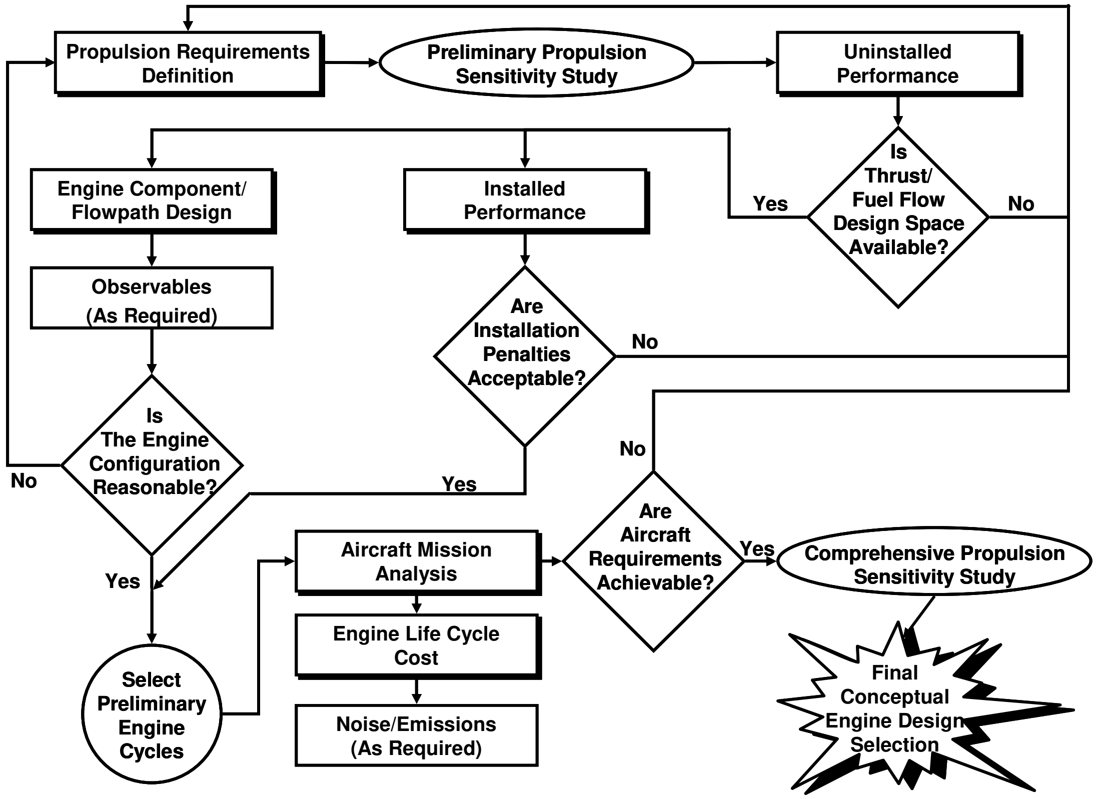

Propulsion Requirements Definition
Preliminary Propulsion Sensitivity Study
Uninstalled Performance
Is Thrust/ Fuel Flow Design Space Available?
Yes
No
Engine Component/ Flowpath Design
Installed Performance
Observables (As Required)
Are Installation Penalties Acceptable?
No
Yes
Is The Engine Configuration Reasonable?
No
Yes
No
Are Aircraft Requirements Achievable?
Yes
Aircraft Mission Analysis
Engine Life Cycle Cost
Noise/Emissions (As Required)
Select Preliminary Engine Cycles
Comprehensive Propulsion Sensitivity Study
Final Conceptual Engine Design Selection

Figure 16: Engine Conceptual Design Process [38]

A list of potential FOMs [29] for commercial aircraft applications [22, 27, 29, 41, 43] is shown in Table 2. Clearly military applications would have a different set of FOM, listed in Table 3, [4, 29, 39, 43] from which to select the best engine for the defined mission.

Table 2: Typical Figures of Merit for Commercial Applications

FOM Category	FOM Examples
Cost	Initial Investment, Direct Operating Cost, Cost per Seat Mile
Emissions	NO_x, CO_2, HC
Noise	Certification Points
Integration	Nacelle Length and Diameter
Reliability	Maintenance Intervals, Unscheduled Maintenance
Performance	Fuel Burn, Range

Table 3: Typical Figures of Merit for Military Applications

FOM Category	FOM Examples
Cost	Initial Investment, Unit Cost, Life Cycle Cost
Observability	RCS, IR signature
Integration	Vectoring
Reliability	Non-recoverable In-flight Shutdowns, Extended Twin Operations
Performance	Thrust to Weight, Range, Cruise Speed

The selection method compares the different candidate engines based on their FOM. The selection method can vary based on the goals of the engine designer. The selection method may use optimization techniques for an optimum engine design[38], probabilistic techniques for a robust design[44, 45] or perhaps generate a Pareto frontier of designs to explore trades between the FOM based on a weighting of the importance of each FOM[25, 46]. No matter which method is chosen, in order to properly compare the different candidate engine cycles within the CDS, each candidate engine must satisfy the feasibility criteria discussed in Chapter 1 when created during on-design analysis.

Healy et al [42] parametrically varied engine design variables such as CPR, T4, BPR as well as engine size to create a CDS for a commercial engine study. This CDS was then evaluated with several aircraft versions based on airframe parameters such as wing loading (W/S), aspect ratio, thickness-to-cord ratio, and airplane thrust-to-weight ratio (T/W). The selection of the engine cycle and airframe combination was based on the FOM of aircraft range, takeoff gross weight (TOGW), takeoff field length (TOFL), thrust margin, and specific excess power. Jacquet and Seiwert[7] also performed parametric engine cycle design to study potential engine to augment the CFM-56 family, but divided the cycle analysis even further by first selecting an optimum core configuration and then refining the low pressure system before evaluating the overall propulsion system and aircraft performance. Their overall process for parametric engine design is shown in Figure 17.

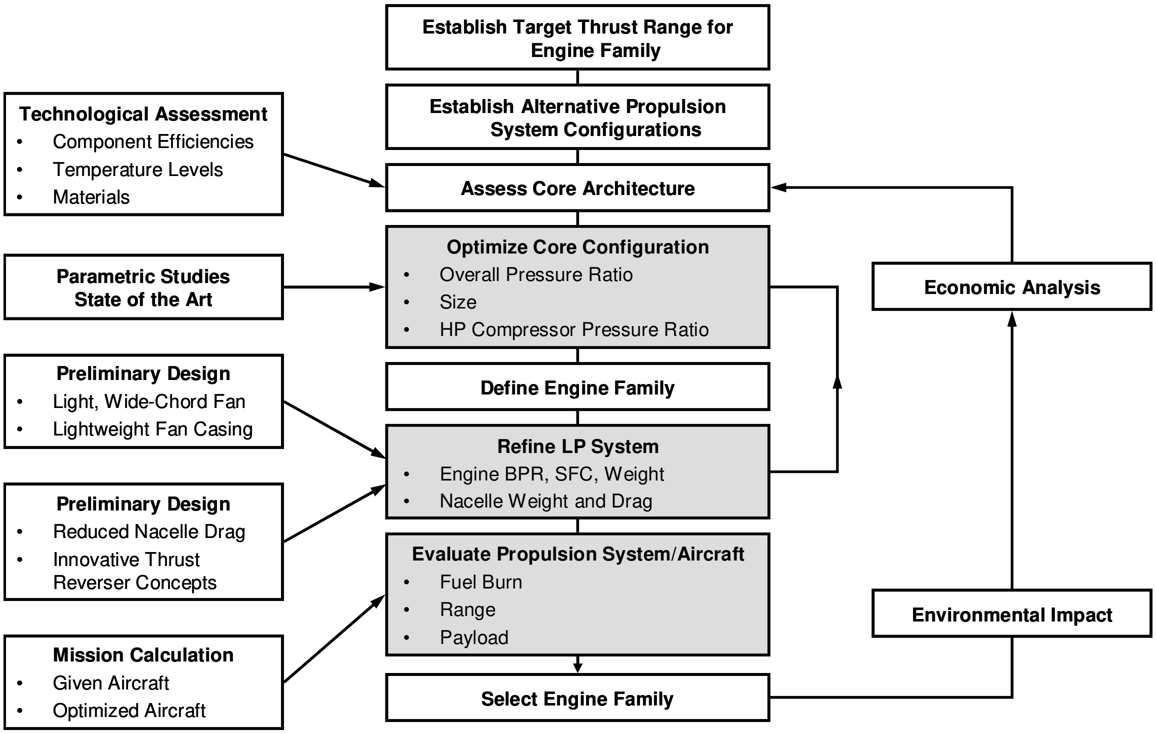


Figure 17: Parametric Engine Design Process for Commercial Engine Study[7]

Notice that the selected engine family was based on the FOM of fuelburn, range and payload. In addition, the first step of the process was to define performance requirements by establishing a target thrust range. As mentioned earlier, the first step of the cycle design process shown in both Figure 15 and Figure 16 was to establish performance requirements from a defined mission. The process of developing requirements and constraints for aircraft engines is discussed in the following section.

2.3 Development of Requirements and Constraints for Aircraft Engines

Chapter 1 discussed how a set of technology rules and cycle performance requirements must be established and followed in order to meet the feasibility criteria of a consistent technology level and performance capability when creating a CDS. The technology rules and performance requirements can originate from a variety of sources including airframe manufacturers, engine aerodynamic and mechanical designers, marketing research, customers such as passenger airlines, global freight shipping or the military and the cycle analyst [8, 27]. A flow chart with the decomposition of the technology rules, performance requirements and design rules from their potential sources is shown in Figure 18.

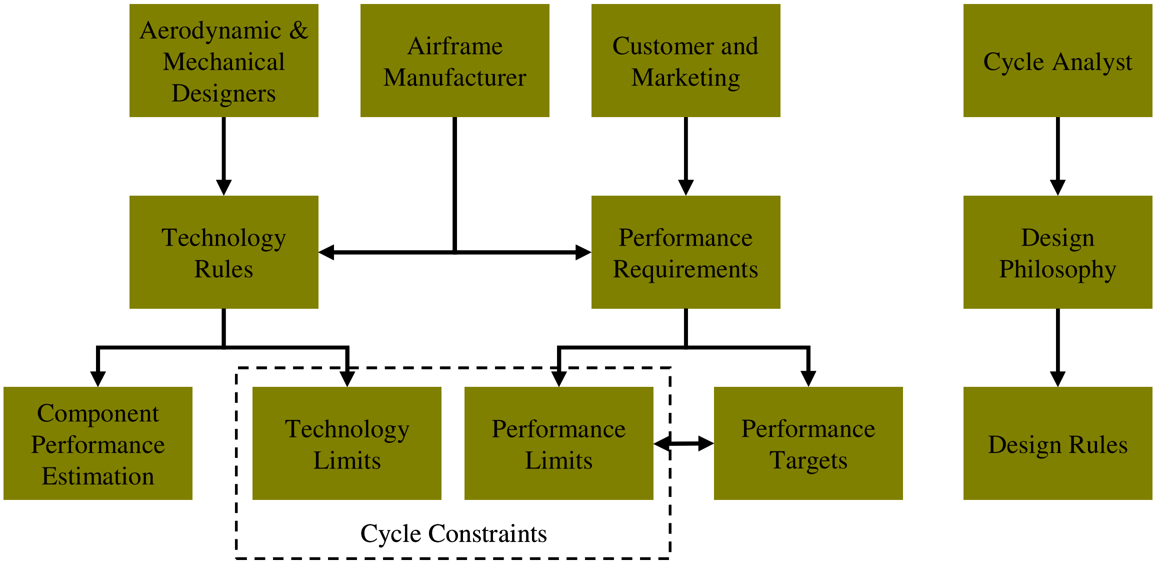


Figure 18: Decomposition of Technology Rules, Performance Requirements and Design Rules

Cycle performance requirements are specific requisites defining the expected performance characteristics of the cycle throughout the operational envelope[30]. Typical performance requirements include thrust, mass flow or mass flow ratio, rotational speed and flight Mach number [41, 47]. Cycle performance requirements can be divided into two categories, performance targets which are regarded as equality constraints and performance limits which act as inequality constraints. Whether a cycle performance requirement is a target or a limit is dependent on its severity relative to the values of the design variables.

Technology rules describe how the technology parameters change as a function of the design variables. These rules can be divided into two categories, component performance estimates and technology limits. Component performance estimates contains functions, tables or maps which estimate component technology parameters for a given technology level as a function of design variables. Component performance estimates do not require the component technology parameters to be a fixed value throughout the CDS. Properly modeled, the component performance parameters are usually estimated as functions of the flight conditions and design choices[48]. Examples of typical component performance estimates parameters for gas turbine engines are listed in Table 4[9, 30].

Table 4: Typical Component Performance estimates Parameters

Component Performance estimates	Notation
Compressor efficiencies	η_C
Turbine efficiencies	η_T
Burner efficiencies	η_B
Nozzle velocity and discharge coefficients	CV, CdTh
Duct and burner pressure drops	$\Delta P/P$
Compressor tip speeds	Utip
Inlet pressure recovery	η_R
Turbine cooling effectiveness	Xeff

Technology limits are constraints established by the technology level which can not be exceeded. Constraints are usually fixed regardless of the flight conditions or design variables. Constraining values for differing technology limits can be found at a variety of different operating conditions. A list of possible technology limits that may be considered during cycle analysis is shown in Table 5[41].

Table 5: Possible Technology Limits for Consideration During Cycle Analysis

Technology Limit	Notation
Pressure and temperature extremes for engine inlet and casing	T1, P1
Compressor casing Δp limits	Ps3, Ps14
Compressor casing and cooling flow temperature limit	T3
Turbine inlet temperature limit	T41, T49
Augmentor liner temperature limit	T6, T8
Maximum rotor speed limit	N
Maximum pressures and loading limits on nozzle plug, flaps, and reversers	P8-Pamb
Exhaust nozzle area ratio limits for actuator and flap design	A9/A8
Minimum combustor Δp for turbine cooling backflow margin in cooling circuits	(P3-P4)/P3
Combustor and augmentor fuel flows limits for control design, for burner smoke, emissions, and stability limits and augmentor thrust jumps	WF3, WF6
Combustor and augmentor blowout margin	CBOM, ABOM
Gear box and shaft torque limits for turbine output and customer extraction power	HPX
Customer bleed rates, temperature and pressure limits	WB, TB, PB
Maximum corrected speed and flow of each compressor	N/√θ
Stall pressure ratio on each compressor for stall margin stack on operating line	P/Pstall
Turbine flow function limits which establish turbine area requirements	W41, W49
Turbine corrected speed, work and pressure ratio limits	N/√θ, ΔH/T, ΔP
Turbine exit mach number limit	M5
Augmentor inlet Mach number limit	M6
Exhaust nozzle area requirements for complete expansion and nozzle stability	A9/A8
Duct Mach limits for sizing ducts	M(x)
Fan diameter limits which bound lower limit of fan pressure ratio	Fan Diam

The previously discussed cycle requirements can occur at multiple operating conditions. These operating conditions are points within the operational envelope defined by some combination of environmental conditions, flight Mach number and throttle setting. These points are selected to correspond to key segments outlined in the vehicle mission profile. Wood and Pilidis[15] state that “the design of an aircraft engine is a compromise between the various operational requirements of the mission it is expected to fulfill. When engines are required to operate over a wide range of flight speeds and altitudes, this compromise becomes progressively more difficult. This is because to be able to achieve all the required mission criteria, the engine designer has to accept certain penalties at each operating point.” Therefore, the selection of the points to include in the cycle analysis is dependent upon the vehicle mission, which leads to the first key observation:

Observation 1: When designing a gas turbine aerothermodynamic cycle to match to an aircraft, performance requirements and limiting values of constraints considered during cycle analysis occur at multiple operating conditions.

This first observation provides motivation to the incorporation of multiple design points into the cycle analysis process, such that the cycle analyst can create feasible engine aerothermodynamic cycles. A list of some of the common operating points for commercial/transport applications and military applications are shown in Table 6 [22, 30, 41]. All of the different operating conditions become possible design points for consideration by the cycle analyst, depending on the application or design problem. Requirements at each of these operating conditions define the engine performance envelope.

Table 6: Typical Key Operating Conditions

Commercial/Transport	Military
Component Aero Design Point (ADP)	Component Aero Design Point (ADP)
Take Off (TKO)	Take Off (TKO)
Top of Climb (TOC)	Top of Climb (TOC)
Cruise	Subsonic Cruise
High Hot Day Take Off (HHD)	Supersonic Cruise
One Engine Out (OEO)	Low Speed Maneuver
Noise Certification	High Speed Maneuver
	Loiter

2.4 Traditional Approach to Engine Cycle Design

The objective of this section is to review the traditionally employed methods for cycle analysis. As stated previously, there are two modes of cycle analysis, on-design analysis and off-design analysis. Traditionally, on-design analysis has been performed at a single design point to set the aerothermodynamic cycle. Thermodynamic performance at other operating conditions is evaluated in off-design analysis. Much of the available literature on cycle analysis resides in numerous texts whose foundations date back to the 1950's[49]. Saravanamuttoo writes:

> When the First edition of [Gas Turbine Theory, 5th ed., 2001] was written fifty years ago, the gas turbine was just becoming established as a powerplant for military aircraft. It took another decade before the gas turbine was introduced to civil aircraft, and this market developed so rapidly that the ocean liner was rendered obsolete … Despite the rapid advances in both output and efficiency, the basic theory of the gas turbine has remained unchanged [since the first edition][50].

In many of the texts, mention is made of the need for a cycle design to meet requirements at multiple operating conditions. However, for simplicity, a single design point approach is instituted and a process to meet all of the requirements at different operating conditions barely discussed. The single design point approach is employed because the focus of these texts is on providing a basic understanding of thermodynamic performance trends to cycle design variables for different engine architectures, rather than provide the tools necessary to design and match an engine to meet a set of performance requirements for a particular airframe. Furthermore, many of these texts consider applications for gas turbines other than aircraft propulsion such as power generation. In these cases, the single point design is entirely appropriate.

In his book, Cumpsty[51] designs engines to meet requirements for a "New Large Aircraft" (NLA) which was patterned after the initial requirements for the A380 and a "New Fighter Aircraft" (NFA) similar to the new eurofighter. Since the NLA was specified to have 4 high bypass ratio engines, he states that "the critical condition for sizing the engine is the top-of-climb" and that " engines giving adequate thrust at top-of-climb will … give ample thrust at take off under normal conditions". However he notes that most of the flight is spent in cruise and that "the design point, in the sense of lowest fuel consumption, should therefore correspond to the cruise condition." In his NLA example, Cumpsty uses Mach 0.85 at 31 kft as his start of cruise with a thrust of 75.1 kN to size the engine mass flow and assumes that the fan can be adequately sped up to provide the top of climb thrust of 84 kN. For his NFA example, Cumpsty independently designs the engine at three different design points, 11km at Mach 0.9, 1.5 and 2 and states that "the three design points are for distinct 'paper' engines and do not correspond to the operating conditions which the same engine would adopt at the different flying conditions". He makes no attempt to combine the distinct paper engine into a single engine design.

Mattingly[9] performs "parametric cycle analysis" on several engine architectures, including turbojets, turbofans and turboprops, in which he parametrically varies design variables to produce plots of aerothermodynamic cycle performance. However, he does not account for any cycle requirements in this process. His method follows the logic of creating a parametric CDS and then determining which designs meet the requirements as opposed to establishing the requirements up front and creating a CDS to meet those requirements. He does note the importance of adhering to technology rules by stating "the value of parametric cycle analysis depends directly on the realism with which the engine components are characterized" and gives as an example for compressors that "for the conclusions to be useful, a realistic variation of efficiency with total pressure ratio must be included".

Kerrebrock[4], besides including the standard cycle performance calculations, includes a chapter on propulsion system analysis. He states that the goal of propulsion system analysis is "to determine the best propulsion system for some application". He outlines a four step process for propulsion system analysis. They include the identification of FOM for which the system (engine and airframe) is to be judged, the definition of a model (architecture) for both an engine and airframe characterized by a set of parameters, the definition of a mission to evaluate the FOM against, and the determination of the optimum setting of the model parameters. He briefly discusses some segments of aircraft missions that would define the requirements, but makes no mention of how to meet them.

Cohen et al[8] states that "the designer of aircraft engines must recognize the differing requirements for take-off, climb, cruise, maneuvering, the relative importance of these being different for civil and military applications and for long- and short-haul aircraft". They give as an example the problem for a long range civil aircraft which must minimize SFC at cruise but meet thrust requirements at either take off or top of climb concluding that "the selection of design conditions [for aircraft gas turbines] is much more complex than for a land-based unit". In their example design point calculations are performed at SLS or cruise depending on the architectures, but no attempt is made to reconcile both design points.

Walsh and Fletcher [30] contend that for the initial cycle design, the operating condition where an engine will spend the most time has traditionally been chosen as the engine design point, but that other points such as some important high power conditions may be chosen. With regards to component design points they state that:

> For the concept design phase described here the component design points are usually at the same operating condition as the engine design point. In a detailed design phase however, this may not be true. For example in the detailed design phase an aero-engine fan may be designed at the top of climb, the highest referred speed and flow, whereas the engine design point would be cruise[30].

Walsh and Fletcher use design point diagrams based on the operating conditions of International Standard Atmosphere (ISA), 11,000 m at both Mach of 0.8 and Mach of 2.2 for turbojet and turbofan aircraft engines. They make no consideration for take off or top of climb requirements.

Clearly the selection of a single aerothermodynamic cycle design point for an aircraft engine is a difficult task to resolve with all of the requirements. Before discussing the details of incorporating the design point into cycle analysis, it is necessary to describe the engine cycle model.

2.4.1 The Engine Cycle Model

Parametric engine cycle design is almost invariably done on a computer[30]. The common mechanism for analyzing cycle performance is the engine cycle model. The engine cycle model is a mathematical model of the aerothermodynamics of the internal flow of the engine used to determine the performance data for a given flight condition when the component characteristics and fuel flows are known[41]. A mathematical representation of each component of the engine is created based on its impact to the thermodynamic properties[52]. However, the properties are measured at stations in between each element. At each station thermodynamic parameters are calculated based on the performance of engine components upstream [30].

"The cycle model strictly adheres to the principles of mass and energy conservation so that the difference between the gas flow exiting a component and that entering a downstream component must be equal after accounting for scheduled bleeds, leakages, and the addition of fuel or water to the mass flow[41]." Additionally, there are mechanical constraints such as engine components operating on the same shaft rotating at the same speed[41]. The resulting engine cycle model not only provides the overall performance of the engine but provides the flow rates, pressures and temperatures at each station which are given to the engine component designers for calculation of forces, stresses, and thermal loads[41].

A good engine cycle model consists of the following items[41]:

- Thermodynamic properties of dry air, water vapor, fuels and products of combustion
- The aerothermodynamic characteristics of all engine components as functions of their independent variables
- Modifiers to component performance (Reynolds number effect, tip clearance effects, variable geometry)
- Parasitic losses (friction, bearings, pumps, coolers)
- Turbine cooling flows and air leakage (flanges, seals, inlets)
- Customer horsepower extraction bleed and inlet recovery schedules
- Operational philosophy for off-design performance

2.4.2 On-Design Analysis Using Single Point Design

Engine cycle design starts with the on-design analysis but the final selection of an engine for a particular application is based on its off-design performance over the entire aircraft mission during the cycle selection process[29]. Choosing a single design point to set the thermodynamic cycle can be a difficult proposition. Sanghi [53] writes that "the assumption of engine design point should be such that the chosen engine cycle provides the best performance over the entire flight envelope of specified mission and operates satisfactorily." Should the point be chosen to meet some thrust requirement at a flight condition that the engine will rarely if ever operate at during a mission, or should it be a point that it will operate at most of the time such as at cruise? Even the concept of a single cruise design point is misleading as the engines will in fact operate over a range of points at different altitudes and thrust settings as the aircraft burns fuel and reduces weight throughout the mission, thus reducing the required thrust to maintain the same flight Mach number. There are additional reasons for considering other operating conditions as the design point. For instance, sea level static could be chosen,, "the reason being that it is the sea level static at which the engine is manufactured and tested, and a common design point permits comparison of various engine designs[53]."

Why bother with performing the on-design analysis if it is the performance in off-design that determines the desirability of the engine? First, off-design analysis can not be conducted until the design point and the size of the engine have been chosen by some means. On-design analysis is much less tedious to calculate than off-design analysis, often providing mathematical optima which can be directly exploited[29]. Furthermore, the entire aerothermodynamic cycle, no matter how complicated the performance requirements, constraints and complexity of the cycle model, can always be defined by a single design point allowing easy categorization of the different candidate engine cycles.

Observation 2: The cycle analyst would find it difficult to create a feasible CDS using single point design method (SPD) as the proper value of certain cycle assumptions would be unknown a priori. However, any cycle design created using a multi-design point method (MDP) can be duplicated by SPD since the appropriate assumptions would then be known and any aerothermodynamic cycle can be created with a single design point.

The advantage to single point on-design cycle analysis is that, unlike off-design analysis, it is a direct calculation[29]. The thermodynamic effect of each component on the working fluid is calculated starting at the inlet of the engine and working through, component by component, to the exit at the nozzles. The thermodynamic properties of the working fluid exiting one component become the inlet properties of the next component until the inlet and exit thermodynamic properties are known for all components. The on-design cycle analysis calculations for gas turbine engines will vary based on the engine configuration and therefore the combination of components that make up the engine, but the effect of each component type can be calculated in the same manner regardless of the architecture. Some typical gas turbine engine components that can be included in the on-design cycle analysis are the inlets, compressors including fans, turbines, combustors, ducts, flow splitters, mixers, shafts, and nozzles[9]. Each of these elements has a set of equations associated with them that describe their effect on the working fluid. The general process for on-design analysis for turbojets and turbofans can be found in several texts, [4, 8, 9, 48] to list a few. It should be noted that the accuracy of the on-design cycle analysis depends on the assumptions that accompany these equations regarding the performance of each component. This background information leads to a third key observation:

Observation 3: MDP methods are intended to ensure the feasibility of all cycle designs for a particular application. MDP methods are not intended to improve the accuracy of the performance prediction from the cycle analysis. The accuracy of the performance prediction is dependent upon the fidelity of the cycle model and the quality of the component performance maps.

2.4.3 Off-Design Cycle Analysis

The data generated during the on-design cycle analysis are an input for off-design analysis[53]. Off-design cycle analysis evaluates the thermodynamic performance of a fixed engine as operational conditions change within the operational envelope according to atmospheric properties, flight Mach number and engine throttle setting often expressed as a power code [4, 8, 30, 41, 54]. The off-design performance follows from the conditions that are fixed by the choice of engine design point for on-design analysis, flight conditions, power setting, and the area settings of variable geometry features like the exhaust nozzle and the variable stators in compressors[53].

The performance of each component is no longer specified as in on-design cycle analysis, but determined from engine component performance maps scaled around the design point [4, 8, 30, 41, 54]. However, for a given flight condition, the operating location on each map is unknown and an iterative process is required to determine the engine cycle performance. “The components must be ‘matched’ to determine the pressure ratio, airflow, rotor speed, [and] efficiency[54].” The almost universal form of a steady state off-design model is a thermodynamic matching model[30]. The solution to the iterative process must satisfy continuity - the mass flow leaving one component must equal the mass flow entering the next component - and conservation of energy which balances the work done by the compressors and extracted from the turbines[52, 54].

There are two basic methods described in the literature for finding the solution; nested loops with fixed point iterations (serial nested loops) and Newton's method[30].

As the name implies, nested loops with fixed point iteration creates a series of nested loops as shown in the flowchart in Figure 19 for a separate flow turbofan engine. Each loop consists of an initial guess, represented as the blue input arrows, and a decision point represented as the green diamonds[30]. Each loop is intended to ensure either continuity or conservation of energy. Within each loop, the initial guess is adjusted until the criteria for the decision point is met and the process continues to the next decision point. However, finding the correct value of the initial guess for the next decision point may invalidate the results from the first decision point. Thus the process must continually repeat loops until all loops have successfully met their decision criteria. This method is good for developing an understanding of the off-design process but becomes numerically inefficient as the number of loops increase and therefore is not a good alternative for implementation into a cycle analysis code[30]. For this reason almost all cycle analysis codes utilize a form of Newton's method for their off-design analysis[30].

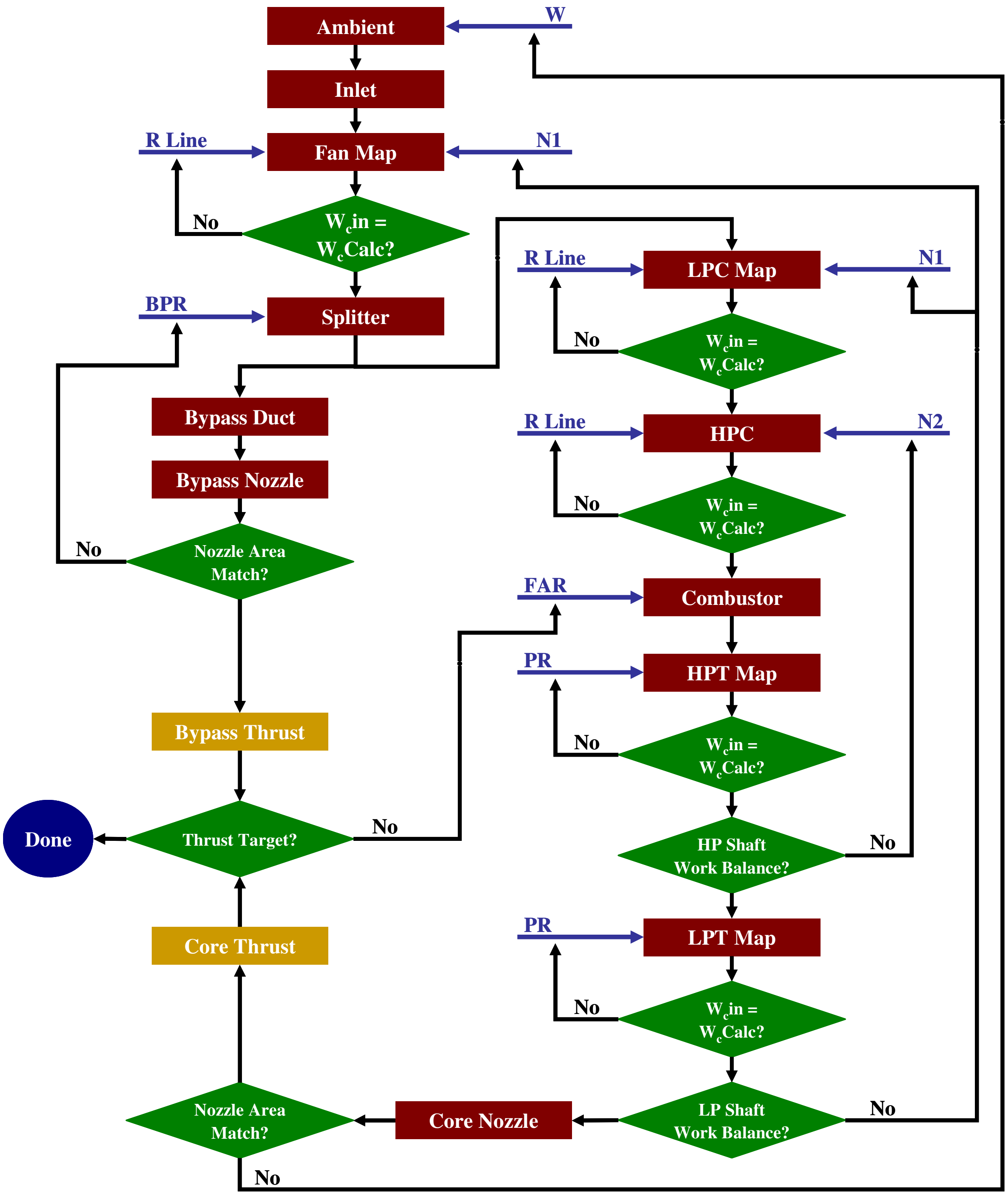


Figure 19: Off-Design Nested Loop for Separate Flow Turbofan

For off-design cycle analysis using Newton's Method[55], the nested loops are converted to a system of nonlinear equations[30]. The problem is setup as a vector of independent variables, x, which are iterated upon by the solver to find the roots of a vector of dependent condition functions, F(x) as show in equation 7.

$$F(x) = 0 \tag{7}$$

The steps for using Newton's method to balance the cycle are[30]:

Step 1: Setup independents (x's) and dependents condition

Step 2: Choose initial iterate values of x's designated x0

Step 3: Calculate engine cycle as in on-design

Step 4: Calculate the error term of the dependent conditions, F(x), and test for convergence

Step 5: Create Jacobian, F′(x)

Step 6: From error and Jacobian calculate new values of x's

The solver method begins by establishing the same number of independent variables and dependent conditions. As with the nested loops, an initial iterate is chosen for the independent variables. The cycle calculations are then performed as a direct calculation as in the on-design analysis. The error term, the value of F(x), is calculated and tested for convergence against a tolerance established by the cycle analyst. If convergence is not met, the Jacobian matrix is calculated. The Jacobian matrix is the matrix of all first-order partial derivatives of the dependent condition functions. A new vector of independent variables is created using the Newton step as defined in equation.

$$X_{n+1} = X_n + F'(X_n)^{-1} F(X_n) \tag{8}$$

Steps 3-6 of the process are then repeated with the new independent variables until the cycle analyst selected convergence tolerance is met.

2.4.4 Sequential Single Point Design

Few methods for performing multi-design point cycle analysis are available in the literature. In 1986, as part of a NASA program to define optimum turbofan engine cycles for the year 2000 designated as Future Large Engine/Nacelle Technology Study (FLENTS), Steinmetz and Wagner developed a technique to match thrust requirements at two design points[22]. This method, which is designated as Sequential SPD, is outlined in the flowchart shown in Figure 20. It is a combination of SPD on-design cycle analysis and off-design analysis. The method differs from SPD in that the process to define the cycle is now iterative, where as with SPD the cycle was defined directly. Sequential SPD was developed to perform a parametric study composed of the cycle design variables maximum cycle T41, OPR (designated OAPR), and BPR as well as the architectural choices of mixed versus separate flow exhaust and geared versus direct drive fans. In addition, the study included varying levels of technology in the form of component efficiencies and turbine cooling requirements.

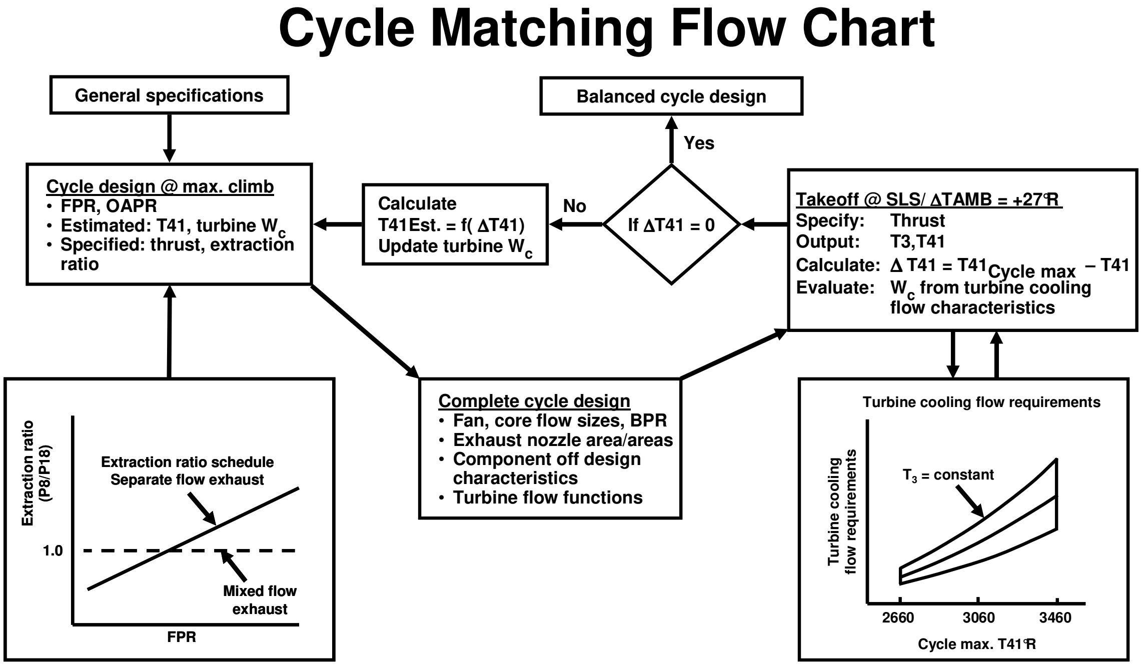

Cycle Matching Flow Chart
General specifications
Balanced cycle design
Yes
Cycle design @ max. climb
• FPR, OAPR
• Estimated: T41, turbine W$_c$
• Specified: thrust, extraction ratio
Calculate
T41Est. = f(ΔT41)
Update turbine W$_c$
No
If ΔT41 = 0
Takeoff @ SLS/ ΔTAMB = +27°R
Specify: Thrust
Output: T3,T41
Calculate: Δ T41 = T41$_{Cycle\ max}$ − T41
Evaluate: W$_c$ from turbine cooling flow characteristics
Complete cycle design
• Fan, core flow sizes, BPR
• Exhaust nozzle area/areas
• Component off design characteristics
• Turbine flow functions
Extraction ratio (P8/P18)
Extraction ratio schedule
Separate flow exhaust
1.0
Mixed flow exhaust
FPR
Turbine cooling flow requirements
Turbine cooling flow requirements
T$_3$ = constant
2660
3060
3460
Cycle max. T41°R

Figure 20: Flowchart of Sequential Single Point Design Method for FLENTS[22]

The method was designed to handle thrust requirements at takeoff and maximum (top of) climb based on a long range, twin-engine commercial aircraft mission. In addition, a cruise condition and maximum cruise thrust were identified for comparison of different engine cycles. These requirements are listed in Table 7.

Table 7: FLENTS Engine Requirements

Altitude	Mach No.	ΔTAMB	Engine Rating	Thrust
39,000 ft	0.8	ISA + 18°F	Max. Climb	12,750 lbf
0 ft	0.2	ISA + 27°F	Takeoff	55,400 lbf
39,000 ft	0.85	ISA	Max. Cruise	11,800 lbf

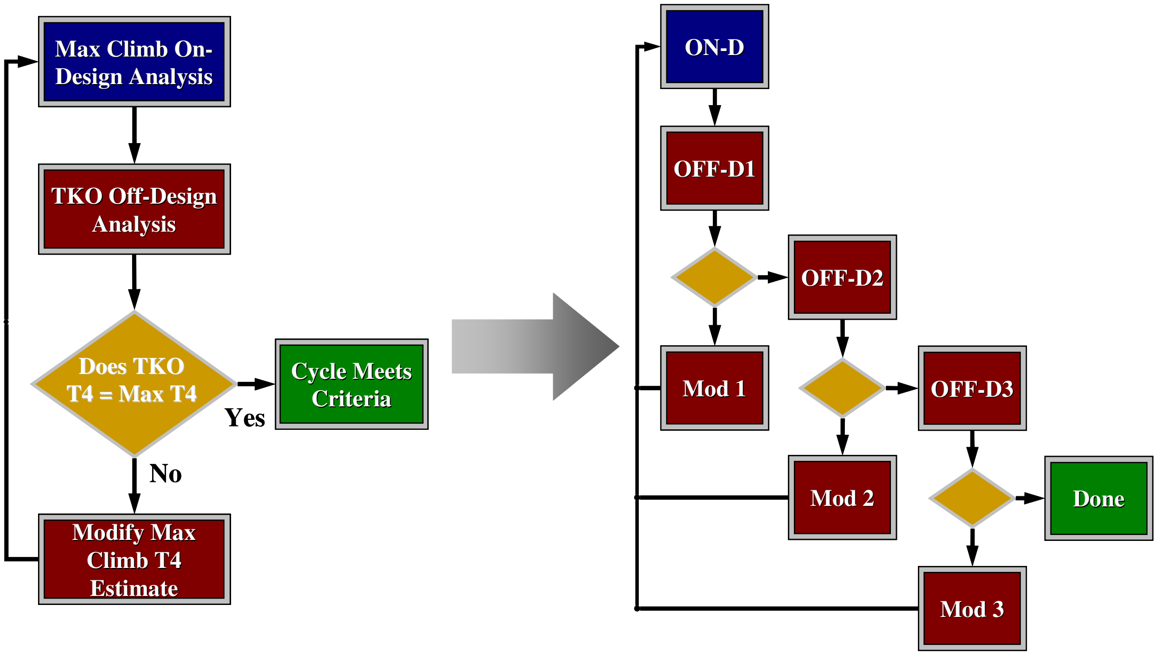
In this method, the cycle maximum T41 and required turbine cooling were defined at takeoff while the cycle design variables and component efficiencies were defined at maximum climb. The first step of the method was to perform a cycle design at maximum climb specifying FPR, OPR, thrust and extraction ratio. The extraction ratio, the ratio of P8/P18, was previously optimized for each value of FPR with a separate flow exhaust creating the schedule shown in Figure 20. Values of T41 and turbine cooling flows were estimated in order to complete the cycle single point design calculations from which the fan and core airflows, FAR, and BPR were determined. Next the cycle was analyzed in off-design analysis at takeoff. Only the thrust was specified at takeoff. From the off-design analysis, T3 and T41 were calculated. With T3 and T41, the turbine cooling flow requirements were updated based on the schedule shown in Figure 20. A new value for T41 at maximum climb was calculated as a function of the maximum cycle T41, takeoff T41 and the previously estimated maximum climb T41. With the new estimates of maximum climb T41 and turbine cooling flow, the cycle single point design was recalculated at maximum climb. This process repeated until the T41 at takeoff matched the maximum cycle T41. The result was a candidate engine cycle that had a balanced cycle design by achieving both thrust targets at maximum climb and takeoff, and matching the maximum cycle T4 at takeoff.

One drawback to this method is that it requires an understanding of how choices at the design point affect off-design performance. Thus the estimate of maximum climb T41 was a function of maximum cycle T41, takeoff T41 and the previously estimated maximum climb T4. This function must be determined prior to the analysis or some technique must be employed to numerically determine the estimate. This leads to a fourth observation:

Observation 4: Because the performance of the design points is coupled, any change that affects the cycle performance at one of the design points, changes the performance at all of the other design points.

Therefore, one technical challenge was to develop an MDP method that does not requires an understanding of how choices at the design point affect off-design performance.

Another issue with sequential SPD is that the method does not scale well with increasing requirements and design points. This is illustrated in Figure 21. For two design points such as in the FLENTS study, a single iteration loop exists with one decision point. However, if additional design points or requirements are added, a series of nested loops are created as in the off design analysis. These design points could be for additional thrust requirements or component design points. By creating a series of nested loops, the computational efficiency of the method is seriously degraded[30]. This is due to the fact that the method is continuously converging on partial solutions, and must repeatedly reconverge on internal loops every time an external loop is checked because of the coupling that exists between the different design points. Therefore, another important technical challenge is to develop a MDP method that can efficiently incorporate as many design points as necessary.



Single Loop

Nested Loop

Figure 21: Transition to Nested Loops for Increased Number of Design Points

2.5 Research Questions and Hypotheses

The primary research question was stated previously at the beginning of this chapter to focus the direction of the background research.

Primary Research Question: How can an on-design cycle analysis method be developed that incorporates all desired cycle performance requirements and constraints at different design point and generates aerothermodynamic cycles to comprehensively populate the cycle design space in an efficient manner?

This question focuses on two primary ideas: the importance of examining as many design points as necessary in the on-design analysis to evaluate and adjust the design to meet all of the cycle requirements and the need to comprehensively but efficiently populate a CDS. An overall hypothesis to address the primary research question is proposed based of the observations made in the background research:

Overall Hypothesis: A method that ***simultaneously*** satisfies the performance requirements and constraints for all design points can comprehensively populate the cycle design space in an efficient manner.

A set of three detailed research questions has been composed from the primary research question and the observations made throughout the background information. For each research question an additional hypothesis has been devised. The more detailed research questions and hypotheses are the guiding principles for the creation of the method in Chapter 3.

The first research question addresses the differences between MDP methods and SPD methods with no attempt made to meet requirements at operating conditions other than the design point. The focus of the research question is on the performance of the resulting engines.

Research Question 1: What are the differences in the CDS as a result of using an MDP cycle analysis method versus a SPD cycle analysis method to generate the space?

This question focuses on the results of the cycle analysis in the form of the CDS. Will the CDSs be different depending on the cycle analysis method? If so, what drives them to be different and is one CDS more correct than the other? An attempt to answer this question leads to hypothesis 1:

Hypothesis 1: The topography and feasible boundaries of the SPD generated CDS differ from the MDP CDS because many of the assumptions required by the SPD method at the design point are unknown functions of the design variables and therefore cannot be correctly estimated prior to analysis.

The MDP method creates a feasible CDS, one in which every candidate engine cycle meet all of the performance requirements and constraints. Furthermore, if the design range of design variables extends too far, MDP identifies the boundaries of the design variables where all every candidate engine cycles become infeasible. The SPD created CDS will result in candidate engines that fall into one of the following categories:

1. The candidate engine cycle does not meet all performance requirements and constraints and is therefore infeasible.
2. The candidate engine cycle meets all performance requirements and constraints and is a feasible cycle design.
3. The candidate engine cycle exceeds the performance requirements and is penalized in engine performance at other operating conditions.

Also, SPD will not determine the boundaries of the CDS which will be established by the range of the design variables and values of the design point assumptions.

The second research question addresses the method for including multiple design points simultaneously as proposed by the overall hypothesis.

Research Question 2: How can the on-design cycle analysis approach be developed in order to account for multiple design points simultaneously?

This question focuses on the core issue with developing an MDP method and leads to several additional questions. How does one link the different design points together?

How are constraints incorporated into the method to determine feasibility of the candidate engine cycle? What are the biggest challenges to incorporating as may design points as necessary? The second hypothesis states:

Hypothesis 2: An approach will simultaneously meet multiple design points by formulating a system of nonlinear equations at on-design mode from design rules that couple the design points, performance requirements, technology rules and design variables.

The design rules remove the technical challenge of understanding completely the coupling of the performance at the different design points and provide a systematic method to include as many design points as necessary. The third research question addresses the convergence of the system of equations created with the design rules in the simultaneous MDP method

Research Question 3: Can simultaneous MDP find all of the feasible candidate engines to form a multidimensional cycle design space composed of design variables and technology parameters? What is the tradeoff in the simultaneous MDP method's efficiency to ensure finding all of the feasible designs?

Essentially, the premise of research question three is that a converged solution should be found for all combinations of design variables, if the candidate engine cycle represents a feasible design. This premise leads to additional questions; Is there a practical limit on the number of performance requirements or design points that can be handled using simultaneous MDP? What is the trade off in efficiency to ensure 100% convergence for all combinations of feasible design variables? Are any additional steps required to ensure robustness against changes to the technology parameters to evaluate different technology levels? The hypothesis for research question three states that:

Hypothesis 3: Simultaneous MDP can find the solution for a high percentage of candidate engine cycles within the CDS generated from a wide range of design variables and technology levels while maintaining a high level of efficiency.

As such, any feasible candidate engine cycle can be found in a short amount of time allowing for comprehensive coverage of the CDS. The final research question addresses the improvements of the simultaneous MDP methods to the other MDP methods. The next chapter will formulate the simultaneous MDP method and develop its mathematical formulation. The research questions and hypotheses will serve as guides for the theoretical development of the method. The research experiments, delineated in Chapter 4, provide the data, presented in Chapter 5, necessary to test and prove or disprove the stated hypotheses.

CHAPTER 3

RESEARCH FORMULATION

Chapter 2 discusses the need for a cycle design method that creates a CDS of gas turbine candidate engine cycles. The objective of the method is to shape the design space such that all of the candidate engine cycles are feasible. This objective can be accomplished by fulfilling the criteria of consistent technology and performance capability for each candidate engine cycle.. Also, the deficiencies of the traditional method have been discussed with regards to design problem dimensionality resulting in the desire for a new method that can simultaneously meet *all* of the performance requirements and constraints for a given technology level. In this chapter, the mathematical formulation of a method called simultaneous Multi-Point Design is developed to address this objective.

There are five categories of parameters incorporated into the multiple design point process. Each parameter category is assimilated within the simultaneous MDP process in a different manner and a review of each is beneficial before outlining the entire process. The first category of parameters is the *cycle design variables*. These are the parameters that are systematically varied to create the CDS. Traditionally, these would be variables that defined the performance at the single design point such as BPR, FPR, OPR and T4. For multiple design points, the design variables can be a scalar, vector or array. Scalar variables must specify which design point the cycle design variable is to be set. Furthermore, component performance can no longer be parametrically varied by a single parameter such as FPR. Instead the component is defined by a performance map represented as a vector or array which must be parametrically varied for different designs. This requires a method be in place to parametrically generate component performance maps – a necessity if the predicted performance at the different design points is to be accurate. (Note that this requirement would be true for off-design analysis of an aerothermodynamic cycle generated with SPD.)

The next category of parameters is the *operating conditions* that define the different design points. The operating conditions are composed of the external environment in which the engine is operating including the atmospheric properties such as pressure, temperature, density, and humidity and the flight Mach number. To allow for a consistent comparison of performance, atmospheric properties are usually specified as deviations from a standard atmosphere at a given altitude[56].

The third and fourth categories of parameters are the *cycle independent parameters* and *cycle dependent functions*. The cycle independent parameters are variables controlled by the modified Newton-Raphson solver to achieve the desired performance targets specified by the dependent functions. Each of these parameters is associated with a specific design point. The dependent functions can be either defined by the user from the design rules or by the engine match relations required to ensure laws of continuity and conservation of energy.

The final category of parameters is *constraints*. Constraints are limits placed on the aerothermodynamic cycle which may not be exceeded by the candidate engine cycle. The constraints are composed of the technology limits and performance limits. Constraints may be specified for a specific operating condition or apply to all operating conditions. Each constraint will be related to a one of the dependent functions parameter. Violation of a constraint will result in a relaxing of the dependent function parameter until the constraint is satisfied.

This chapter provides an overview of the simultaneous MDP method. Included is a discussion of the three general phases that comprise the simultaneous MDP method. This discussion is intended to highlight the flow of information and thought process behind the simultaneous MDP method. Additionally, details regarding the execution of the modified Newton-Raphson solver, such as the use of bracketing limits on the independent parameters and Jacobian estimation techniques are included in this chapter. A step by step procedure for implementation of simultaneous MDP is provided at the end of the chapter.

3.1 Simultaneous Multi-Design Point Overview

The flow of information for the simultaneous MDP method, shown in Figure 22, can be classified into three general phases. The first phase is the Requirements and Technology Definition (R&TD) phase which establishes the design problem to be addressed by the cycle analyst and sets the level of technology to be incorporated in the CDS. The second phase in the process is the MDP Setup phase which organizes the information from the R&TD phase and prepares the candidate engine cycle for analysis. The third phase is the MDP Execution phase which performs the cycle analysis for a different combination of design variables to generate the CDS.

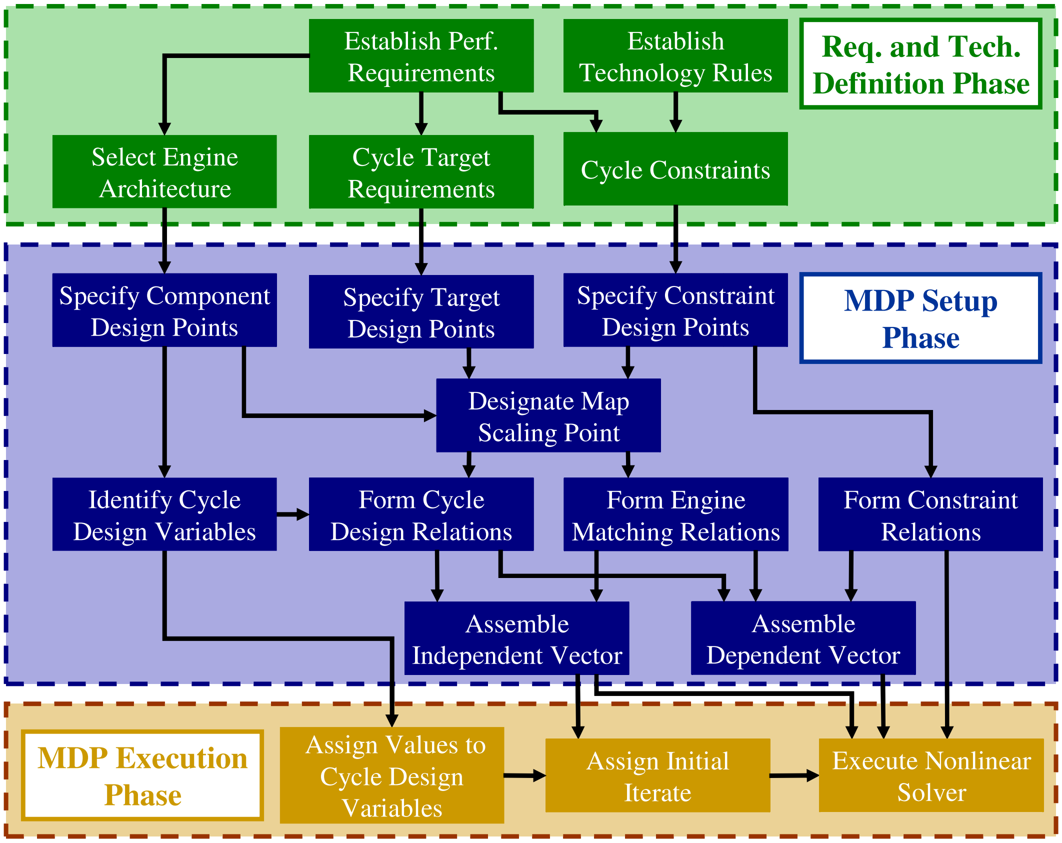

Establish Perf. Requirements
Establish Technology Rules
Req. and Tech. Definition Phase
Select Engine Architecture
Cycle Target Requirements
Cycle Constraints
Specify Component Design Points
Specify Target Design Points
Specify Constraint Design Points
MDP Setup Phase
Designate Map Scaling Point
Identify Cycle Design Variables
Form Cycle Design Relations
Form Engine Matching Relations
Form Constraint Relations
Assemble Independent Vector
Assemble Dependent Vector
MDP Execution Phase
Assign Values to Cycle Design Variables
Assign Initial Iterate
Execute Nonlinear Solver

Figure 22: Process Flowchart for Simultaneous MDP Method

3.1.1 Requirements and Technology Definition Phase

The performance requirements and technology rules are established in the R&TD phase. Stricker states:

> Perhaps the most critical phase of any development process is right at the beginning – the definition of requirements. An over constrained or poorly defined set of requirements can lead the design team on a wild goose chase focusing on the wrong criteria…For a successful design, a clearly defined set of requirements right up-front is critical [38].

The cycle performance requirements are specific requisites defining the expected performance characteristics of the cycle throughout the operational envelope. The classic cycle performance requirements are thrust requirements at different operating conditions, e.g. take off and top of climb. Additional performance requirements that may be considered include mass flow or mass flow ratio, rotational speed and flight Mach number. Often these requirements are developed from an aircraft/engine system constraint analysis. "In many respects, the requirements definition phase is a mini-conceptual design process[38]." The reader is directed to references [41, 57] for more details on constraint analysis. The cycle performance requirements are divided into two categories, performance targets and performance limits. Performance targets are incorporated into the method as equality constraints and categorized as cycle target requirements. Performance limits are included as inequality constraints and form part of the cycle constraints. While the cycle analyst must initially categorize each performance requirement as either a target or constraint, the method has the capability to exchange requirements between these groups based on the requirement's severity. To illustrate this point, a fixed cycle turbofan or turbojet with no afterburner has the ability to control two parameters, combustor fuel flow or exit temperature and engine mass flow, in order to generate thrust. Therefore, the cycle analyst can only match two thrust requirements for a candidate engine cycle. Any additional thrust requirements must be treated as performance limits. At these conditions the thrust requirements must be exceeded in order to maintain consistent performance capability. For a specific set of design variables, the method identifies the most severe thrust requirement with regards to each control parameter and labels it as a target requirement. All remaining thrust requirements would then be regarded as cycle constraints. The actual mechanism for exchanging performance requirements is created in the MDP Setup phase and performed automatically during simultaneous MDP execution.

In addition, the performance requirements will allow the high level selection of engine architecture. It is necessary to specifying the components that make up the engine in order to create the cycle model and define all of the necessary component technology parameters established by the technology rules. However, specific architectural details of the individual components such as stage count that are not incorporated into the cycle model are not required.

As discussed in Chapter 1, the technology rules describe how the technology parameters change as a function of the cycle design variables. There are two categories of technology rules, component performance estimates and technology limits. Component performance estimates parameters are functions, tables or maps which estimate component technology parameters for a given technology level as a function of design variables. Typical technology parameters have been discussed in Chapter 2 and are listed in Table 4. The technology parameters are imbedded within the individual components included in the cycle model. The technology limits are constraints established by the technology level which can not be exceeded by the aerothermodynamic cycle. Constraining values for differing technology limits can be found at a variety of different operating conditions and are categorized by the method as cycle constraints along with the performance limits. Typical technology limits were listed in Table 5.

3.1.2 Multi-Design Point Setup Phase

The MDP Setup phase begins by assessing the information from the definition phase and identifying all of the design points of interest. Each set of unique operating conditions forms a different design point. This includes different design points for individual components to provide a means of parametrically varying the component performance maps, the operating conditions specified for each target requirement and any operating condition in which a cycle constraint is most likely to be violated. From the complete collection of design points, one design point must be designated as the component map scaling point. At this design point, the component performance maps will be scaled to attain the necessary mass flow. To facilitate identification of all of the design points, it is helpful to create a Design Point Mapping Matrix (DPMM) as shown in Figure 23.

Design Points		Design Point 1	Design Point 2	Design Point 3	Design Point 4
Comp. Map Scaling Point		**X**			
Design Alternatives (DA)	DA 1	**X**			
	DA 2	**X**			
	DA 3		**X**		
	DA 4				**X**
Performance Requirements (PERF-R)	PERF-R 1	**X**			
	PERF-R 2		**X**		
	PERF-R 3			**X**	
	PERF-R 4				**X**
Component Performance Estimation (CPE)	CPE 1	**X**			
	CPE 2	**X**			
	CPE 3			**X**	
	CPE 4			**X**	
Technology Limits (TLIM)	TLIM 1				**X**
	TLIM 2		**X**	**X**	
	TLIM 3		**X**	**X**	
	TLIM 4		**X**	**X**	

Figure 23: Design Point Mapping Matrix

The DPMM links the design points with the design variables, performance requirements, component performance estimates, technology limits and the component map scaling point. Not only does the DPMM identify the design points, but it also assists in conversion of the design rules into the engine cycle model.

The selection of the engine architecture and the creation of a set of design rules will determine the cycle design variable available to the cycle analyst to create the different dimensions of the CDS. Design rules establish how the design points, performance requirements, technology rules and design variables are linked together. The design rules are dependent on the design philosophy of the cycle analyst. Multiple sets of design rules can exist for the same design problem. For a single design point, the performance requirements of interest can be calculated directly for a given set of cycle design variables. However, for multiple design points, a system of nonlinear equations must be solved in order to determine the performance of the engine for a given set of design variables. This system of equations results from the coupling of the design points as shown in Figure 24. Thus the design of the aerothermodynamic cycle becomes the solution of the system of nonlinear equations.

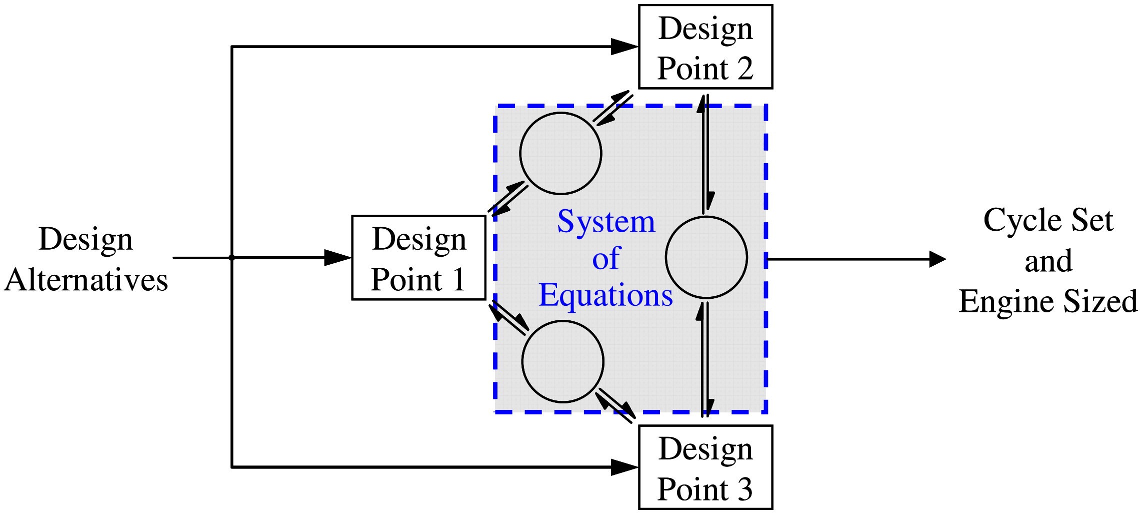


Figure 24: Formulation of Simultaneous MDP System of Equations

Simultaneous MDP uses design rules that link the design points, performance requirements, technology rules and design variables into a system of nonlinear equations. The equations can be divided into three categories; cycle design relations, engine matching relations and constraint relations. The *cycle design relations* are created by selecting cycle variables for use as independent parameters controlled by the modified Newton-Raphson solver to satisfy the target requirements. This is done in a manner such that independent parameter x is linked to achieve target requirement y so that each target requirement is guaranteed to be a function of at least one independent parameter. The linked independent and dependent parameters can be specified at the same or different design points, but the independent parameters must be unique. Two dependent parameters cannot be linked to the same independent parameter. This does not mean that dependent parameters cannot be functions of more than one independent parameter, as indeed most dependents will be a function of multiple independent parameters. Rather, in the setting up of the cycle design relations, a unique independent parameter will be assigned to control the value of the dependent parameter by the modified Newton-Raphson solver.

As with off-design cycle analysis, for the simultaneous MDP process, the component performance is not specified by a single parameter but by a map (or array) which describes the performance of each component usually as a function of dimensionless or quasi-dimensional parameters[8, 9, 30, 41]. Because the location on each map is unknown for each operating condition, an iterative process is required to determine the engine cycle performance. This process must satisfy the physical laws of continuity and conservation of energy [8, 9, 30, 41]. Therefore, additional equations for component matching are added to the system of nonlinear equations created from the design relations to ensure mass flow and power balance at every design point. This second set of equations form the *engine (component) matching relations*. The number of equations added from the engine matching relations is dependent upon the number of components with performance maps, the total number of shafts, and the number of design points. From the design and engine relations a vector of independent parameters and a vector of dependent functions are assembled and, along with the design variables become three of the four required inputs passed to the modified Newton-Raphson solver in the MDP Execution phase.

The final required input is the *constraint relations* for each of the cycle constraints. The constraint relations are linked to one of the cycle design relations whose independent parameter will be used to meet the constraint should it be violated. In essence, this relation instructs the modified Newton-Raphson solver how to proceed when a constraint is violated. It may be necessary to create additional design relations for the purpose of having the correct dependent function available to couple to a constraint. Also, more than one constraint can be coupled to a cycle design relation. Should more than one constraint be violated at the same instance, the solver adjusts the independent parameters to meet the most stringent constraint [58].

3.1.3 Multi-Design Point Execution Phase

As stated above, the third phase of the simultaneous MDP process, MDP Execution requires four inputs from the MDP Setup phase, a vector of design variables, a vector of independent parameters, a vector of dependent functions and the constraint relations. The first step of the MDP Execution phase is to assign values to each of the cycle design variables. Each unique set of design variables represents a different candidate engine. After the design variables have been assigned, initial values for each of the independent parameters must be assigned forming the initial iterate. The final step is to execute the modified Newton-Raphson solver to arrive at the solution to the system of nonlinear equations. When the solution is found the engine cycle is set and the engine sized to meet all of the cycle design criteria. While the execution seems simple, it is in fact extremely difficult[59]. The next section is devoted entirely to this subject.

3.2 MDP Solver Theory: Modified Newton-Raphson

Before beginning the detail formulation of the system of nonlinear equations, it is necessary to establish the nomenclature listed in Table 8.

Table 8: Simultaneous MDP Nomenclature

Symbol	Description
α	Design variable
β	Design point operating condition
x	Cycle independent parameter
X	Vector of cycle independent parameters
X_0	Initial iterate
X^*	Vector of cycle independent parameters that solves system of equations
y	Cycle dependent parameter
$\hat{y}$	Cycle dependent parameter calculated value
$f(X)$	Cycle dependent function
$F(X)$	Vector of dependent functions
g	Constraint parameter
$z(X)$	Constraint function
$Z(X)$	Vector of constraint functions
d	Number of design points
k	Number of design variables
u	Number of user defined independents and dependents
m	Number of engine matching independents and dependents
l	Number of constraints
ε	Error tolerance
Subscripts	Description
i	Index for vector of independent parameters, dependent parameters or dependent functions
ref	Reference value for dependent parameter
n	iteration
Superscripts	Description
U	User defined independent parameter or dependent function
M	Engine matching defined independent parameter or dependent function
D	Design point index

The notation of an uppercase symbol signifies a vector while a lower case variable signifies an element within the vector.

3.2.1 Choice of Solver Method

One of the most commonly occurring problems in the field of applied mathematics is the finding of roots to a set of nonlinear equations [60]. The method selected for solving, or finding the roots of the set of nonlinear equation within the simultaneous MDP method is a modified version of the Newton-Raphson method. The Newton-Raphson method is a widely used technique for solving systems of nonlinear equations. It is the solver algorithm employed in many gas turbine modeling software packages for analysis of off-design performance [30, 58, 61]. In addition to off-design analysis, the Newton-Raphson method has been applied to transient modeling of gas turbines [62, 63], calibration of engine modeling to experimental data through modification factors [64, 65], and sensor diagnostics especially in terms of engine performance deterioration [66, 67]. Along with its capabilities for finding roots, the familiarity of the Newton-Raphson method with engine cycle modelers and its availability in gas turbine software tools makes it an ideal choice for incorporation into the simultaneous MDP method.

3.2.2 Newton-Raphson Formulation within Simultaneous MDP

Through the design rules and engine matching relations, the multiple design points create a system of (u+m) coupled nonlinear equations of the form

$$Y = y(x, \alpha, \beta, g) \tag{9}$$

Where U is the number of user defined equations and m is the number of engine matching equations. There are no good, general methods for solving systems of nonlinear equations[59]. The solution to the set of nonlinear equations is found iteratively using a modified version of the Newton Raphson method. This method requires a transformation of the set of equations to the standard form[59] of

$$F(x,\alpha,\beta,g)=0 \tag{10}$$

Where F is also a vector of length $(u+m)$ composed of the transformed cycle dependent parameters into dependent functions whose solution is zero and can be shown with individual elements as:

$$F=\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{u+m} \end{bmatrix}$$

There are three different methods in which the transformation of the cycle dependent parameters may be formulated; as an absolute transformation, as a relative transformation, and as a relative to a reference transformation[58]. The transformations are shown in equations 11 - 13:

1) Absolute: $$f_i(x, \alpha, \beta, g) = \hat{y}_i(x, \alpha, \beta, g) - y_i \tag{11}$$

2) Relative: $$f_i(x, \alpha, \beta, g) = \frac{\hat{y}_i(x, \alpha, \beta, g) - y_i}{y_i} \tag{12}$$

3) Relative to reference: $$f_i(x, \alpha, \beta, g) = \frac{\hat{y}_i(x, \alpha, \beta, g) - y_i}{y_{i,ref}} \tag{13}$$

The variable $\hat{y}$ represents the calculated value of the dependent parameter from the cycle analysis program, y represents the target value for the dependent parameter and y_{ref} is a reference value established by the cycle analyst. In most cases, the relative transformation in equation 12 would be used instead of the absolute term as it allows for a uniform setting of tolerances[58]. In some instances where the target value of y is near zero, a reference term may need to be included in the transformation as in equation 13 to prevent division by a small number and allow the method to converge to a solution[58].

The final solution will be determined by satisfying some predetermined convergence criteria based on the residual term [59, 68]. A tolerance for the transformed terms is required to allow the solver to converge to an adequate solution without overtaxing the computational requirements. The standard form with an error term is shown in equation 14:

$$F(x, \alpha, \beta, g) = 0 \Rightarrow F(x, \alpha, \beta, g) \leq \varepsilon \tag{14}$$

Where ε is a vector of length $(u+m)$ composed of the individual tolerance values that the error terms must reach for convergence.

$$\varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_{u+m} \end{bmatrix} \tag{15}$$

The Newton-Raphson method solves the system of equations by successively calculating new values of the independent parameters until the system of equation is satisfied or the method is stopped for exceeding a maximum number of specified iterations[59]. The independent parameters, and dependent parameters, can be divided into two subclasses. The first subclass, designated with a superscript U, is the cycle analyst specified independent parameters set up according to the chosen design rules to meet the performance targets of each of the design points. The second subclass, designated with a superscript M, are cycle variables selected to ensure the laws of continuity and conservation of energy between the different components for each of the design points are satisfied by matching the different components. In addition, each of the independent parameters can also be directly related to one of the design points. A matrix detailing the notation of the subclasses is shown in Figure 25.

	Design Point 1	Design Point 2	Design Point 3	Design Point D
User Defined	$x_i^{U,1}$	$x_i^{U,2}$	$x_i^{U,3}$	$x_i^{U,D}$
Engine Matching	$x_i^{M,1}$	$x_i^{M,2}$	$x_i^{M,3}$	$x_i^{M,D}$

Figure 25: Matrix of Subclasses of Cycle Independent Parameters

The independent parameters can be organized into a vector X, also of length (u+m), by sub class so that

$$X^U = \begin{bmatrix} x_1^U \\ x_2^U \\ \vdots \\ x_u^U \end{bmatrix} \quad X^M = \begin{bmatrix} x_1^M \\ x_2^M \\ \vdots \\ x_m^M \end{bmatrix} \quad X = \begin{bmatrix} X^U \\ X^M \end{bmatrix} \quad X = \begin{bmatrix} x_1^U \\ \vdots \\ x_u^U \\ x_{u+1}^M \\ \vdots \\ x_{u+m}^M \end{bmatrix} \tag{16}$$

The Newton-Raphson method determines new iterations of the independent parameters using equation 17, referred to as the Newton step[59, 69-71]

$$X_{n+1} = X_n + F'(X_n)^{-1}F(X_n) \tag{17}$$

Where X_{n+1} is the new value of the independent vector, X_n is the current value of the independent vector, $F(X_n)$ is the current value of the dependent function vector and $F'(X_n)^{-1}$ is the inverse of the Jacobian matrix. This method can be graphically explained[59, 70] for a one-dimensional problem as in Figure 26.

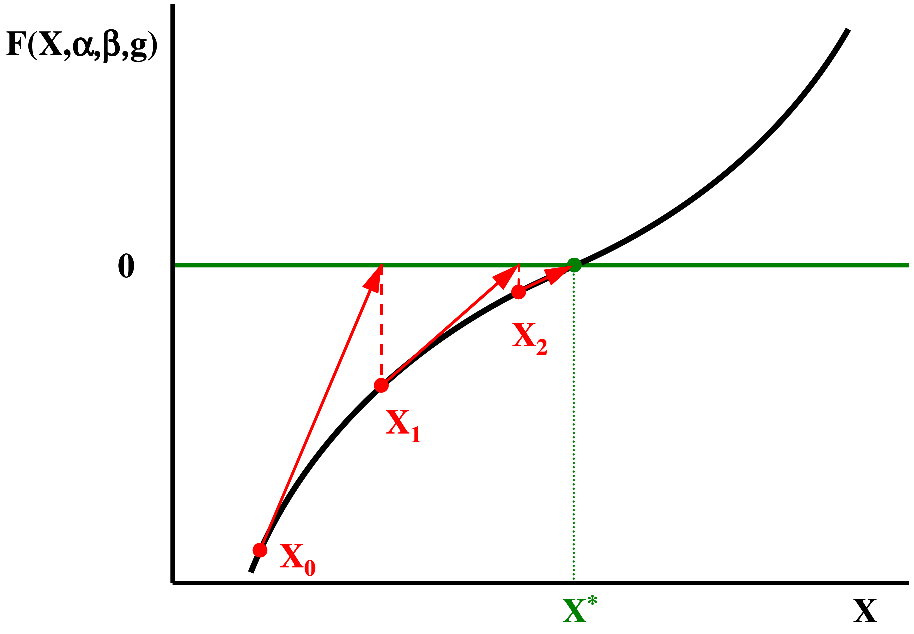


Figure 26: Newton-Raphson Method for One-Dimensional Problem

At the initial iterate X_0, a local model of the function $F(X,\alpha,\beta,g)$ is created, in this instance a linear model with slope $F'(X_0,\alpha,\beta,g)$ tangent to the curve at X_n. The root of the local model is found and becomes the next iteration of the independent variable, X_1. Each of the subsequent iterations will find the root of a new local model until the method converges to the solution, X^*. Expanding to multi-dimension problems works in the same manner with the Jacobian matrix used to create the local model instead of F'.

The Jacobian is an $(u+m)$ by $(u+m)$ matrix defined as[59, 70]:

$$F'(X) = \frac{\partial(F)_i}{\partial(X)_j}(X) \tag{18}$$

for $i = 1$ to $u+m$ and $j = 1$ to $u+m$.

To better understand the coupling of the nonlinear set of equations, the Jacobian matrix from equation 18 can be divided into different submatrices, first for the cycle analyst selected and engine matching selected parameters as shown in equation 19,

$$F'(X) = \left[\begin{array}{c|c} \dfrac{\partial F^{U}}{\partial X^{U}} & \dfrac{\partial F^{U}}{\partial X^{M}} \\ \hline \dfrac{\partial F^{M}}{\partial X^{U}} & \dfrac{\partial F^{M}}{\partial X^{M}} \end{array}\right] \tag{19}$$

and second by the different design points as shown in equation 20.

$$F'(X) = \left[\begin{array}{cc|c|cc} \dfrac{\partial F^{U,1}}{\partial X^{U,1}} & \dfrac{\partial F^{U,1}}{\partial X^{M,1}} & \cdots & \dfrac{\partial F^{U,1}}{\partial X^{U,D}} & \dfrac{\partial F^{U,1}}{\partial X^{M,D}} \\ \dfrac{\partial F^{M,1}}{\partial X^{U,1}} & \dfrac{\partial F^{M,1}}{\partial X^{M,1}} & \cdots & \dfrac{\partial F^{M,1}}{\partial X^{U,D}} & \dfrac{\partial F^{M,1}}{\partial X^{M,D}} \\ \hline \vdots & \vdots & \ddots & \vdots & \vdots \\ \hline \dfrac{\partial F^{U,D}}{\partial X^{U,1}} & \dfrac{\partial F^{U,D}}{\partial X^{M,1}} & \cdots & \dfrac{\partial F^{U,D}}{\partial X^{U,D}} & \dfrac{\partial F^{U,D}}{\partial X^{M,D}} \\ \dfrac{\partial F^{M,D}}{\partial X^{U,1}} & \dfrac{\partial F^{M,D}}{\partial X^{M,1}} & \cdots & \dfrac{\partial F^{M,D}}{\partial X^{U,D}} & \dfrac{\partial F^{M,D}}{\partial X^{M,D}} \end{array}\right] \tag{20}$$

The Jacobian matrix must be nonsingular, therefore the cycle analyst must select the independents and dependents so that every column and every row in the Jacobian matrix has at least 1 nonzero term. The diagonal terms highlighted in blue represent the partial derivatives of the user defined dependents with respect to the user defined independents and the partial derivative of the engine matching dependents with respect to the engine matching dependents within the same design point. The highlighted red terms are the cross partial derivatives between the user defined and engine matching relations. The off diagonal submatrices represent the cross partial derivatives between design points. One advantage of simultaneous MDP is that this coupling is automatically taken accounted for in the off-diagonal terms of the Jacobian matrix. Many of these terms will be zero indicating no coupling, but the non-zero terms not only create the coupling but serve as a way to document the coupling of the system of equations.

3.2.3 Newton-Raphson Convergence Characteristics

Assuming the Jacobian matrix is nonsingular, the successful convergence of the Newton-Raphson method is dependent upon how close the initial iterate, X_0, is to the actual solution X^*[69-71]. It cannot be overemphasized how crucially convergence depends on having a good first guess for the initial iterate, especially for multidimensional problems[59]. This dependence on X_0, is demonstrated in Figure 27.

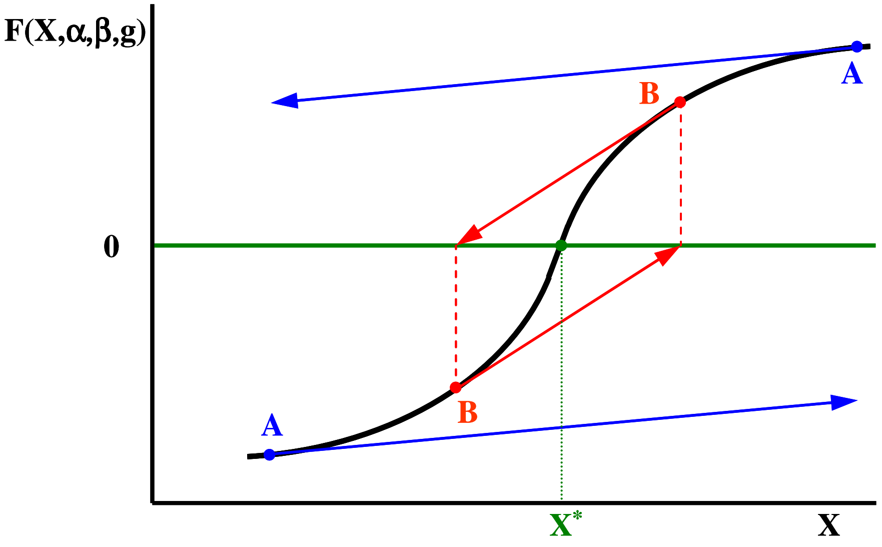


Figure 27: Initial Iterate Criteria for Convergence of Newton-Raphson Method

If X_0 is not close enough to X^*, such as points A in Figure 27, then successive iterations of X with diverge from X^*. If X_0 was placed at point B, successive iterations would not diverge from X^*, but would get no closer and the method would result in an infinite loop. However, if X_0 were closer to X^* than point B, than the method would converge rapidly to X^*. When X_0 is close enough to X^*, the convergence rate for Newton-Raphson method has been shown to be quadratic[69-71]. This results in the error of each subsequent iteration being roughly squared and essentially doubles the correct number of significant digits for each iteration. From this discussion it is shown that the initial iterate x_0 can not haphazardly be chosen but must be chosen intelligently to ensure convergence to the solution.

There are many modification to the Newton-Raphson method sometimes called Quasi-Newton-Raphson methods[59, 72]. The modifications usually attempt to address one of two issues, computational expense and convergence robustness for x_0[69-71]. The first category of modifications is utilized to reduce the computational expense associated with calculating the Jacobian matrix. While the Newton-Raphson method converges quadratically [70], quite a few model passes are required in order to calculate the Jacobian matrix for each iteration[59]. To fully calculate the Jacobian matrix, $u+m+1$ passes through the model must be made. Calculating the full Jacobian can become computationally expensive as the number of design points and requirements increase. The number of cycle analyst selecting parameters, u, is proportional to the number of cycle requirements. The number of engine matching selected variables, m, is proportional to the number of design points minus one times the number of matching components. While the standard Newton-Raphson method recalculates the Jacobian matrix after each iteration, there are several techniques which calculate an update to the Jacobian matrix with the information from a single model pass without recalculating the entire matrix. For simultaneous MDP, the full Jacobian matrix is not calculated unless the convergence rate slows beyond an established threshold. Instead, an update to the Jacobian matrix is estimated each iteration based on Broyden’s method [58, 69, 70, 72, 73]. Broyden’s method falls within a class of methods referred to as secant methods as each reduces to the secant method in one dimensional problems[59]. Broyden’s method was the first of these methods and still appears to be the best[59]. Broyden’s method reduces the convergence rate such that more iterations are required, but will require far fewer model passes to converge to a solution [59, 74, 75].

The second variation to the Newton-Raphson method is to apply a slower converging method that can work towards X^* even when X_0 is not close, called a globally convergent method[70]. A global method is one that will converge from almost any starting value for the initial iterate[59]. The method switches to the Newton-Raphson method once X_n becomes close enough to X^*. Often, this means deriving new initial iterates for cases where X_0 is not close enough to X^*[59]. Finally, the method should be controlled such that for each iteration, X_n does not get outside the best bracketing bounds obtained for that iteration[59]. For the simultaneous MDP method, limits will be placed on the allowable change between X_n and X_{n+1}. An illustration of this technique and its benefits is shown in Figure 28.

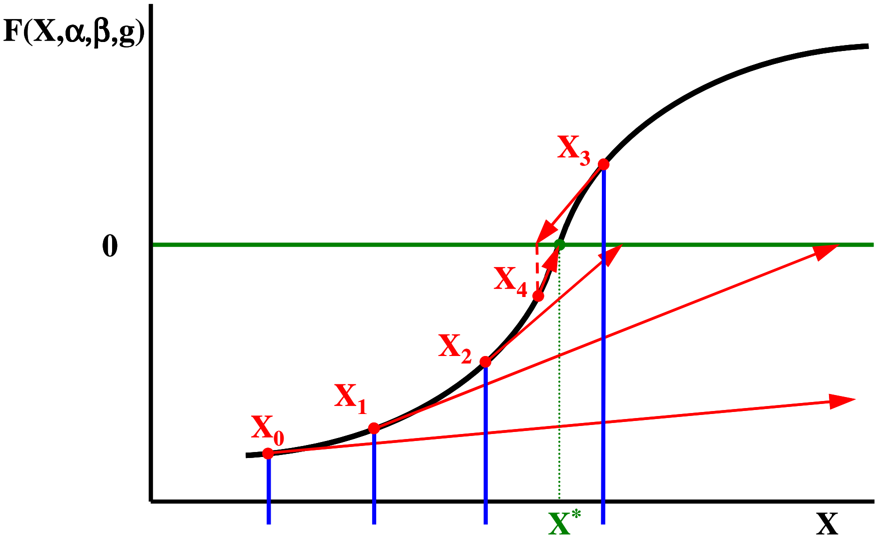


Figure 28: Limits Placed on Allowable Change in X to Improve Convergence

Here the choice of X_0, which would have diverged using the standard Newton-Raphson method is limited in it change to X_1. This process continues until X_3 which is close enough to X^* that the regular Newton-Raphson method can be used to reach X^*. This allows simultaneous MDP to have a higher probability of convergence to X^* but at the expense of more model passes.

Before moving to the incorporation of constraints into the MDP method, a comparison of the convergence rates for sequential SPD and simultaneous MDP is in order. The order of convergence of the iterative sequence is P if P is the largest positive number such that[71, 76]:

$$\lim_{\mathrm{n}\to\infty} \frac{\left|\varepsilon_{n+1}\right|}{\left|\varepsilon_{n}\right|^{P}} = C \tag{21}$$

If P = one the convergence is called linear where as if P = two the convergence is called quadratic. It is also possible to have non-integer values of P. A linear convergence means that the error is always reduced by a constant. A quadratic convergence means the error of each successive iteration is proportional to the square of the previous iteration error.

Sequential SPD uses the iterative method fixed point iteration(FPI) to find the solution. In general, fixed point has linear convergence[71]. The pure Newton-Raphson method has quadratic convergence[71]. However, due to the modifications incorporated into the Newton-Raphson method, namely step limits on the independent parameters and the use of Broyden's method to update the Jacobian, the order of convergence will be less than quadratic but still much faster than linear convergence. This convergence rate is sometimes referred to as superlinear[70].

3.2.4 Incorporation of Constraints into Simultaneous MDP

Constraint incorporation into the simultaneous MDP method in many ways resembles the dependent parameters. In fact, a constraint is nothing more than an auxiliary dependent that only becomes active when the constraint is violated. The first step is to define the constraint, usually within the technology rules, such as a turbine temperature limit. As with the dependent parameters, a transformation is required into a constraint function using the one of the following three methods[58]:

1) Absolute: $$z_i(x, \alpha, \beta) = \hat{g}_i(x, \alpha, \beta) - g_i \tag{22}$$

2) Relative: $$z_i(x, \alpha, \beta) = \frac{\hat{g}_i(x, \alpha, \beta) - g_i}{g_i} \tag{23}$$

3) Relative to reference: $$z_i(x, \alpha, \beta) = \frac{\hat{g}_i(x, \alpha, \beta) - g_i}{g_{i,\,ref}} \tag{24}$$

Each constraint function is then linked to a dependent function. Should the constraint be violated, this linkage specifies how the performance of the engine will change in order to meet the constraint. One or more constraint functions can be linked to a single dependent function. In each case, the severity of the constraint determines when it becomes active.

During the MDP Execution phase when the solver is calculating the Jacobian matrix for the dependent functions, the solver also calculates the partial derivative of each constraint, Z', as a function of the independent parameters using equation 25,

$$Z'(X) = \frac{\partial(Z)_i}{\partial(X)_j}(X) \tag{25}$$

for $i = 1$ to l and $j = 1$ to $u+m$. When a constraint is violated, it is designated as active. The partial derivative for the constraint function is placed within the Jacobian matrix and the linked dependent function is removed[58]. In the case of more than one constraint being violated at the same time that are both linked to the same dependent function, then the most severe will be used in the Jacobian matrix. The dependent function will no longer be satisfied. This method is ideally suited for performance targets and limits such as thrust at different operating conditions. Multiple thrust targets can be linked together and the most severe will automatically be included in the Jacobian matrix and matched while the other thrust targets will be exceeded by the solver.

3.3 Simultaneous MDP Procedure

While a process flowchart of the simultaneous MDP method showing the general flow of information was provided in Figure 22, a detailed step by step procedure is listed in this section that can be utilized to implement the simultaneous MDP method for any gas turbine engine. Each of the experiments utilizing simultaneous MDP, Experiments 2 through 5, follow this procedure.

Step 1. Identify Cycle Requirements and Constraints: Identify the performance requirements that the aerothermodynamic cycle design must obtain when performing parametric engine cycle design. In addition, determine cycle constraints that will be imposed during parametric engine cycle design. Definition of the requirements and constraints must include the value of the specific metric and a complete description of operating condition for which it applies. For some constraints which apply to all operating conditions, one or more operating conditions must be defined that represent the most likely conditions for which the constraint may be violated.

Step 2. Select Engine Architecture and Create Cycle Model: Select the type of engine to be analyzed and identify all of the necessary engine components based on the cycle requirements and constraints. Create the base engine cycle model, including all of the necessary component performance estimates. There is no difference between the base engine cycle model for simultaneous MDP and a model that would be used for SPD.

Step 3. Identify Available Cycle Design Variables and Technology Parameters: Based on the engine architecture, identify the cycle design variables available for the cycle analyst to control during the cycle selection process. Also identify any technology parameters that may be varied to create a CTS.

Step 4. Establish Technology Rules: Determine all of technology parameters, component performance estimates and technology limits, that are functions of the cycle design variables and establish functional relations that can be evaluated by the cycle model. Incorporate into the model the ability to update the technology parameters, performance estimates and technology limits for each case.

Step 5. Specify the Design Points and Designate Map Scaling Point: Identify all of the operating condition to be included as design points for the simultaneous MDP method. The design points are derived from the performance requirements, constraints and engine architecture. Designate one of the design points as the map scaling design point. Incorporate the design points into the engine cycle model.

Step 6. Create Design Point Mapping Matrix and Incorporate into Cycle Model: Create the DPMM and link the design points with the design variables, performance requirements, component performance estimates and technology limits. Incorporate the information from the DPMM into each of the design points previously added to the engine cycle model.

Step 7. Construct System of Nonlinear Equations: This is the step that can have the most variation in the process, the creation of the system of nonlinear equations. Form the cycle design, engine matching and constraint relations as described in Chapter 3. From each relation, identify the independent and dependent parameters and assemble into the solver independent and dependent vectors. In addition, identify any constraint relations that must be linked to a dependent parameter. Add the independent vector, dependent vector and linked constraints to the cycle model.

Step 8. Assign Values to Cycle Design Variables: Determine the range of the design variables that establish the CDS. Populate the CDS with as many combinations of design variables as desired, with each combination, contained as a single case, representing a unique engine design.

Step 9. Assign Initial Values to Solver Independent Vector: Based on the range of the design variables assign values for the initial iterate. The technique for assigning values is discussed in Chapter 4.

Step 10. Solver Setup: Assign all values necessary for the execution of the solver including but not limited to error tolerance for convergence, step limits on the solver independent parameters, convergence criteria for full generation of the Jacobian matrix versus an update of the matrix using Broyden's method, iteration limits and full Jacobian generation limits.

Step 11. Execute Solver Algorithm and Save Cycle Design Parameters: Execute the solver algorithm to determine the solution for the particular combination of design variable values. The algorithm may include techniques for repairing any case that initially fails to converge. Once a solution has been found, the final step is to save all of the necessary information about each of the engine components so that the engine can be evaluated in off-design for operating conditions other than the design points.

CHAPTER 4

IMPLEMENTATION

Five research experiments have been identified to test the hypotheses established in Chapter 3. The experiments are performed on the sample problem of a high bypass ratio separate flow turbofan created to meet the requirements of a notional commercial 300 passenger aircraft. Different experiments will incorporate different levels of complexity from a problem with just two design points in experiment 1 to one with nine design points in experiment 5, dependent upon the level complexity required to test the specific hypothesis. The different levels of complexity were incorporated through inclusion of the different design points in the model. This chapter details the implementation of the simultaneous MDP method described in section 3.3 for each of the experiments, but especially experiments 2 through 5. This chapter identifies the common elements of each step within the five experiment of the simultaneous MDP method. Any differences in the implementation of the simultaneous MDP method for the experiments is noted in the experimental setup sections in Chapter 5.

4.1 MDP Requirements

The first step of the simultaneous MDP method is to identify the performance requirements and constraints. The basis for simultaneous MDP is the need to meet multiple requirements at different design points. To this end, a total of nine requirements, listed in Table 9 have been developed for the LCT engine. In some instances, a specific value for a requirement may not be fully established before the cycle analysis need to be performed. For example the aircraft drag at TOC may not be set as aerodynamic and weight estimates are finalized by the airframe manufacturer. In this case, the requirement can be varied across a range of likely values. For the five experiments, it was assumed that all requirements were known before the analysis. One of the advantages of the simultaneous MDP process is the ability to capture changes to the requirements and properly adjust the CDS.

Table 9. Cycle Performance Requirements for Experiment 2 through 5

Name	Value	Description	Comment
TOC1 Thrust	16,025 lbf	Minimum thrust at top of climb (1) condition	Determines mass flow schedule
TOC2 Thrust	19,600 lbf	Minimum thrust at top of climb (2) condition	Determines mass flow schedule
TKO Thrust	75,000 lbf	Minimum thrust at take off condition	Determines T4 schedule
HHDTKO Thrust	68,000 lbf	Minimum thrust at high hot day take off condition	Determines T4 schedule
Cutback Vmix	1050 ft/sec	Max. average mixed jet velocity at cutback condition for noise	Cutback maximum Vmix to meet noise certification
Maximum Fan Diameter	128 in	Maximum diameter of fan allowable for wing clearance	Provides a minimum constraint for FPR
Maximum T4	3275 to 3450 °R	Maximum allowable combustor exit temperature	Partially determines cooling flow requirements
Maximum T3	1700 to 1850 °R	Maximum HPC compressor exit temperature (Material constraint)	Provides a maximum constraint for OPR
SLSI Thrust	SLSU Thrust	SLS flat rated thrust must match 100% thrust	Determines maximum SLS thrust

The first two requirements are thrust requirements at two different TOC conditions. These requirements could apply to the same aircraft or derive from multiple aircraft that a manufacturer is attempting to service with a single engine design. The thrust requirement of the higher altitude TOC point is set to 16025 lbf and the thrust of the lower altitude TOC point is set to 19600 lbf. The second two requirements are thrust requirements at different takeoff conditions. The sea level TKO thrust requirement is set to 75,000 lbf and the high altitude, hot day TKO point is set to 68,000 lbf of thrust. For these four thrust requirements, the most stringent operating conditions will depend on the values of the design variables and technology parameters.

The next requirement is a limit on the maximum value of the mass averaged mixed jet exhaust velocity, termed Vmix. Vmix is one of the dominant noise sources from modern turbofan engines[90]. Limiting its maximum value will help ensure an engine design capable of meeting noise certification The maximum value of Vmix is set to 1050 ft/s (approximately M 0.95) as both a reasonable jet velocity [21] and a value that provided some constraint to the design space without eliminating all feasible space. Vmix is calculated using equation 26 where V is the jet velocity and W the mass flow.

$$V_{mix} = \frac{V_{Byp\_Nozzle} * W_{Byp\_Nozzle} + V_{Core\_Nozzle} * W_{Core\_Nozzle}}{W_{Byp\_Nozzle} + W_{Core\_Nozzle}} \tag{26}$$

The sixth requirement is a limit placed on the maximum allowable fan size that could be accommodated under the wing of the aircraft. In order to incorporate this requirement, an assumption is made on the fan inlet hub to tip ratio of 0.3 which allows the calculation of the fan diameter along with the fan area set by the fan specific flow.

The next requirement is the maximum combustor temperature T4. A range of values is explored within the design space from 3275 to 3450 °R. The eighth requirement is the maximum HPC exit temperature. Normally, this would be a single value based on the last stage blade and vane material temperature characteristics. For these experiments, a range of 1700 to 1850 °R is placed on T3 simulating different material technologies and allowing the feasible design space to be expanded to accommodate more candidate engine cycles.. The final requirement is the installed SLS thrust at flat rated temperature which must match the rated engine thrust, based on SLS uninstalled thrust at 100% fan speed without violating maximum T4. If this requirement can not be met then the rated thrust must be set at an SLS uninstalled condition with less than 100% fan speed. This requirement helps set the conditions under which emissions testing for the engine will be performed and could be used to constrain the CDS based on combustor requirements for meeting emission certifications. Not all requirements are included with each experiment. The details of which requirements for the different experiments are given in the research experiment descriptions in section 3.3.

4.2 LCT Engine Architecture and Cycle Model

The second step of the simultaneous MDP method is the selection of the engine architecture and the creation of a base engine cycle model. The cycle model was created using the NASA gas turbine cycle simulation program Numerical Propulsion System Simulation (NPSS) [58, 77-84]. This program is capable of performing cycle design and steady state cycle off-design performance prediction as well as other capabilities such as transient cycle off-design performance prediction which are not required for this work. There are many features of NPSS that make it a suitable tool for implementing MDP cycle analysis. Chief among these are the built in object–oriented programming language and a sophisticated solver based on the Newton-Raphson method that allows for constrained solutions[58]. Object oriented programming allows for efficient handling of the different design points by treating them as separate instances of a class of engine model object, called an assembly in NPSS notation[58]. As many design points as desired can be created by simply instancing additional assemblies. The creation of a large system of nonlinear equations to link the different design points would not be valuable without a solver capable of finding the solution to the system. The design of the aerothermodynamic cycle is essentially finding the solution to the system of nonlinear equations; without a solution, there is no cycle design. Many of the design points are included to constrain the aerothermodynamic cycle design; therefore the solver must have the ability to solve a constrained design problem. Any cycle analysis program that contains the features of object-oriented programming and constrained solution capability would also be a good candidate for implementation of MDP cycle analysis. However, without these features, implementation of MDP cycle analysis would be extremely difficult to setup and highly inefficient to execute. Additional features of NPSS that will be utilized include model and design point definition through input files, a thermodynamic gas properties package that can be accessed by the solver, and flexible report generation not only for the design of the cycle but for the performance of the modified Newton-Raphson solver.

The architecture chosen to meet the requirements of the LCT engine is a high bypass ratio separate flow turbofan. The engine architecture consists of a single stage fan and multiple stage LPC driven by a multiple stage LPT. A fixed core (HPC and HPT are fixed for both performance and geometry) is maintained for all engine design variations consisting of a ten stage HPC and a two stage HPT. The NPSS engine cycle model is assembled from a collection of interconnected components or engine elements based on the selected engine architecture. The NPSS model for the LCT engine includes the following elements: ambient flight conditions, inlet, fan, splitter, LPC, HPC, fuel start, burner or combustor, HPT, LPT, core nozzle, bypass nozzle, LP and HP shafts as well as various connecting ducts and bleed ports. To determine the required turbine cooling flow, the subroutine coolit is included in the model. The subroutine determines the chargeable and nonchargeable cooling flows for both the HPT and LPT based on the temperature of the flow and the thermal properties of the blade and stator materials. A schematic showing the connection of the different engine elements and their flow station designation is shown in Figure 29. The base NPSS cycle model file used for all experiments is located in Appendix C.

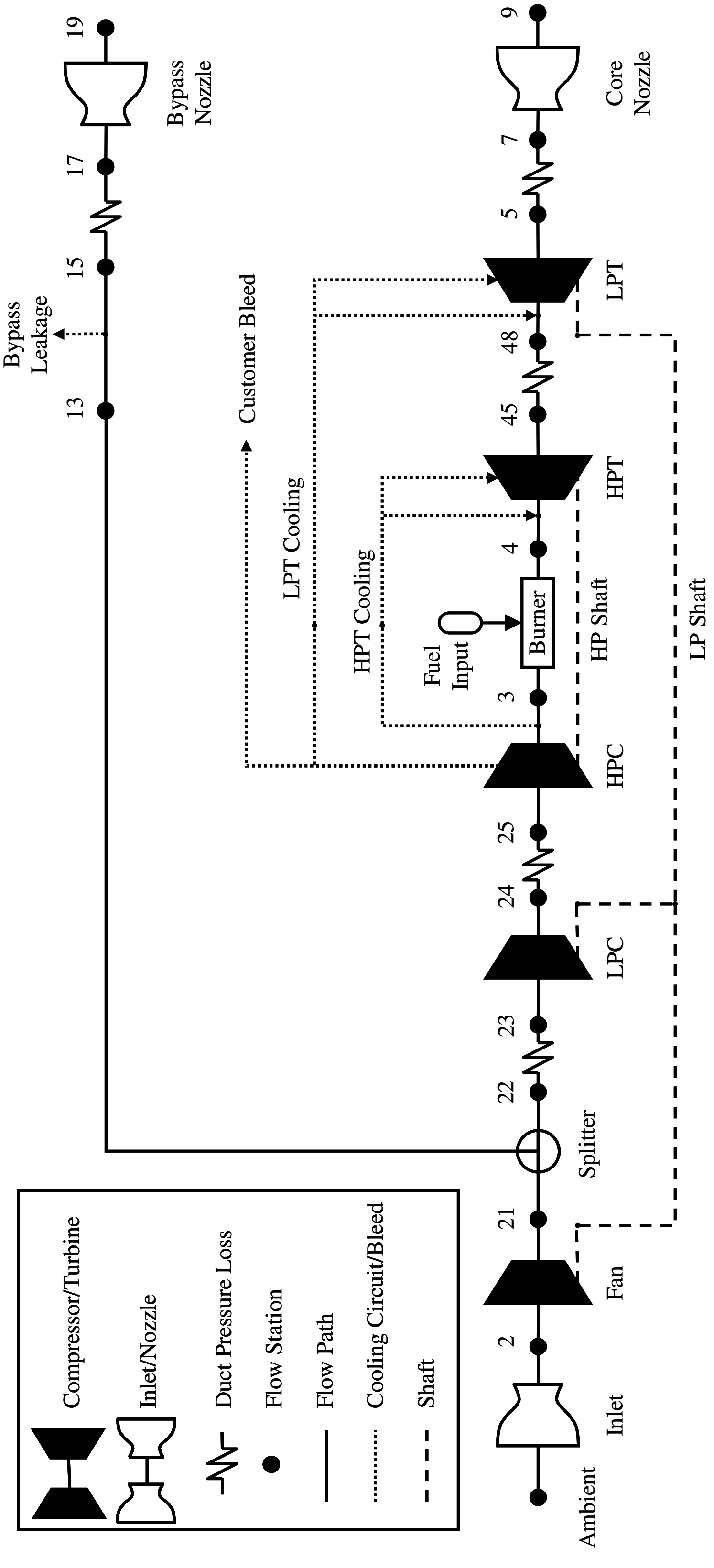


Figure 29. NPSS Model Schematic

4.3 Design Variables and Technology Parameters

The design variables and technology parameters that define the CDS and CTS, listed in Table 10, were identified as part of the third step of the simultaneous MDP method. There are five design variables and 7 technology parameters included in the experiments. The five design variables are the fan pressure ratio (P21/P2), the overall pressure ratio (P3/P2), the nozzle extraction ratio (P18/P8), the combustor exit temperature and the TOC maximum fan overspeed. The seven technology parameters include the maximum HPC exit temperature, fan adiabatic efficiency delta to equation 29, LPC adiabatic efficiency delta to equation (30, adiabatic efficiency delta for the LPT, and three material temperatures; the HPT first vane temperature, HPT blade and vane temperature, and the LPT blade and vane temperature.

Table 10. Design Variables and Technology Parameters for Exp. 2 through 5

Name	Description	Design Point	Units
FPR	Fan Pressure Ratio (P21/P2)	LPCDP	-
OPR	Overall Pressure Ratio (P3/P2)	LPCDP	-
EXTR	Extraction Ratio (P18/P8)	LPCDP	-
T4_Max	Maximum Combustor Exit Temp.	All	°R
TOC_NcMap	Maximum Fan Corrected Speed	TOC1 & TOC2	Nc/100%Nc
T3_Max	Maximum HPC Exit Temp.	All	°R
Fan_Deff	Fan Efficiency Delta	LPCDP	-
LPC_Deff	LPC Efficiency Delta	LPCDP	-
LPT_Deff	LPT Efficiency Delta	LPCDP	-
HPT_1VT	HPT 1st Vane Surface Temperature	TKO & HHDTKO	°R
HPT_Temp	HPT Blade and Vane Surface Temperature	TKO & HHDTKO	°R
LPT_Temp	LPT Blade and Vane Surface Temperature	TKO & HHDTKO	°R

The variables change with each experiment both in the variables incorporated and the range of values depending on the objective of the experiment. As with the requirements, the details of which design variable and technology parameter are included as well as the range of values for each variable are given in the research experiment descriptions in section 3.3.

4.4 Technology Rules

Unlike SPD on-design cycle analysis, where a single value is required for component performance estimates, simultaneous MDP requires the use of component performance maps at on-design in order to ensure that the performance of each design point obeys the physical laws of continuity and conservation of energy. Furthermore, the performance of some components is dependent upon the value of the design variables. The following section describes the implementation of the fourth step of the simultaneous MDP method, the establishment of the technology rules by providing the performance maps for each NPSS engine component and details how the performance changes with regard to the design variables.

4.4.1 Inlet

For the inlet, the only variable component performance estimate captured by the LCT engine cycle model is the inlet pressure recovery. The pressure recovery [85] is shown in Figure 30 as a function of flight mach number and therefore is dependent upon each design point's Mach number as defined in the MDP Setup phase.

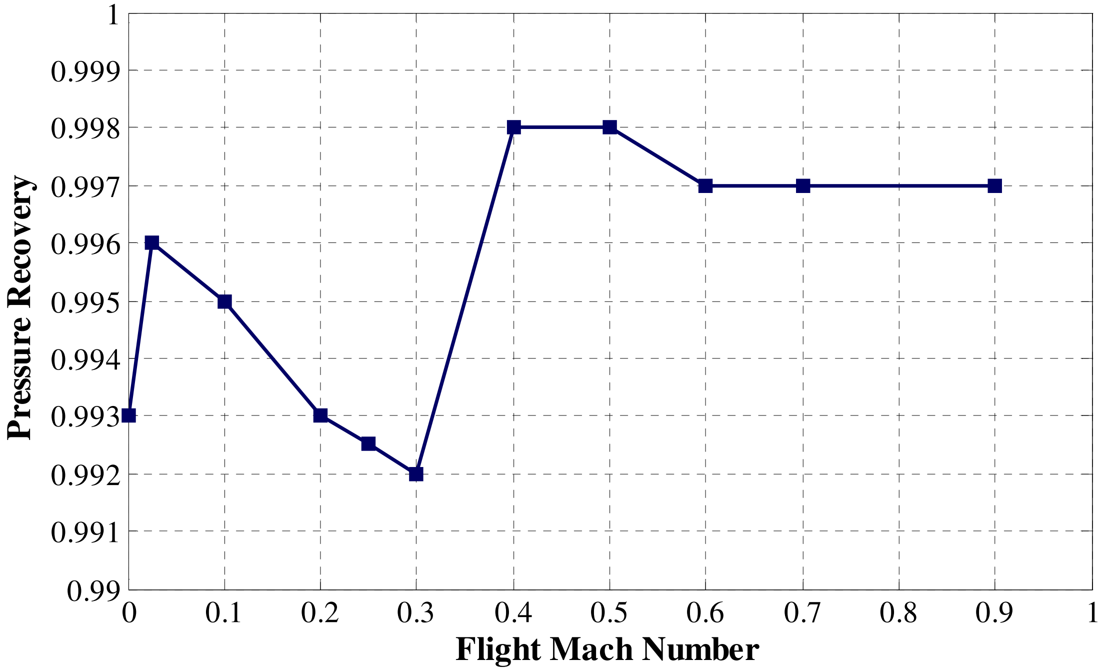


Figure 30. Inlet Pressure Recovery versus Flight Mach Number

4.4.2 Fan

For the fan the necessary component performance estimates are the tip speed, adiabatic efficiency, and fan specific flow. The first two metrics are dependent upon the design variable FPR[48, 86, 87]. The fan tip speed is calculated using the polynomial regression shown in equation 27. The fan polytropic efficiency is calculated next from the previously calculated fan tip speed in equation 28. The fan polytropic efficiency is then converted to adiabatic efficiency for the fan map design point using equation 29. Within the fan adiabatic efficiency equation is a delta efficiency term, Fan_Deff, which allows for changes in the fan technology level.

$$\text{Fan\_Utip} = -508.171 * \text{FPR}^2 + 2968.66 * \text{FPR} - 2208 \qquad (27)$$

$$\text{Fan\_Poly} = -1.07547e^{-8} * \text{Fan\_Utip}^2 - 4.582011e^{-6} * \text{Fan\_Utip} + 0.93 \qquad (28)$$

$$\text{Fan\_Adiab} = \frac{\left(\text{FPR}^{0.28555}\right) - 1}{\left(\text{FPR}^{0.28555/\text{Fan\_Poly}}\right) - 1} + \text{Fan\_Deff} \tag{29}$$

The fan tip speed and fan adiabatic efficiency are plotted in Figure 31. The fan tip speed goes from a value of 620 ft/s at a FPR of 1.2 to 1750 ft/s at a FPR of 2. The fan design point adiabatic efficiency drops from a maximum value of 0.92 at a FPR of 1.2 to a minimum value of 0.88 at a FPR of 2.

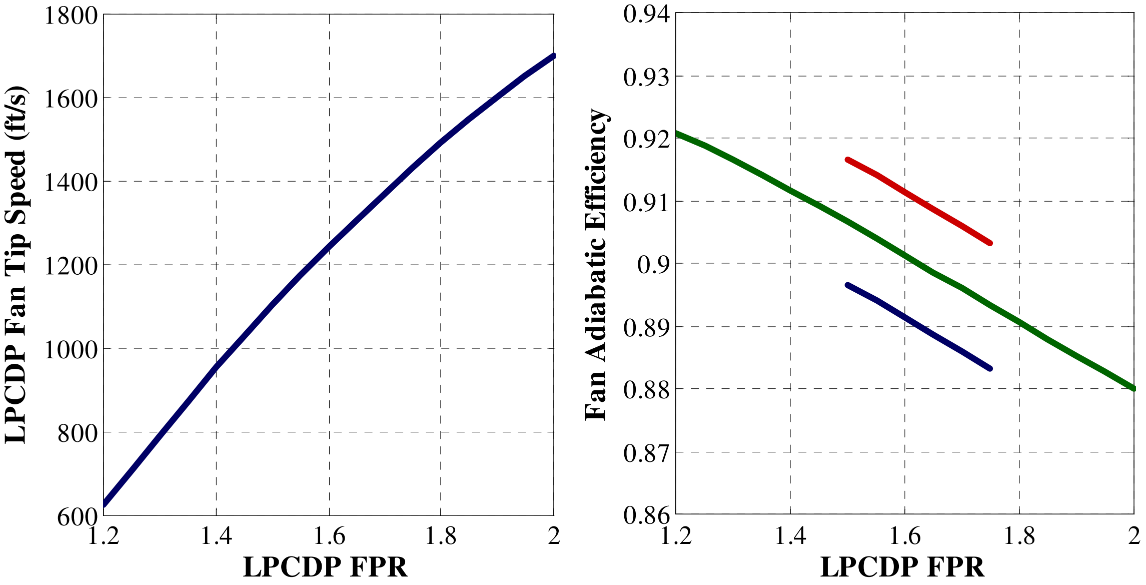


Figure 31. Fan Tips Speed and Fan Design Adiabatic Efficiency versus FPR

The base, unscaled fan map is shown in Figure 32 and includes the location of the fan design point, peak efficiency at 100% speed, which is used by NPSS to scale the base map for efficiency. The base map was generated using the compressor map generation program CMPGEN[88], and is scaled for efficiency, pressure ratio and corrected flow for each candidate engine cycle by NPSS. An assumption is also made for the fan inlet hub to tip value which is set to 0.3 for use with the LP shaft speed calculations in equation 31. The final component performance estimate, fan specific flow is held constant for all candidate engine cycles at 43.3 lbm/s/ft^2 at LPCDP.

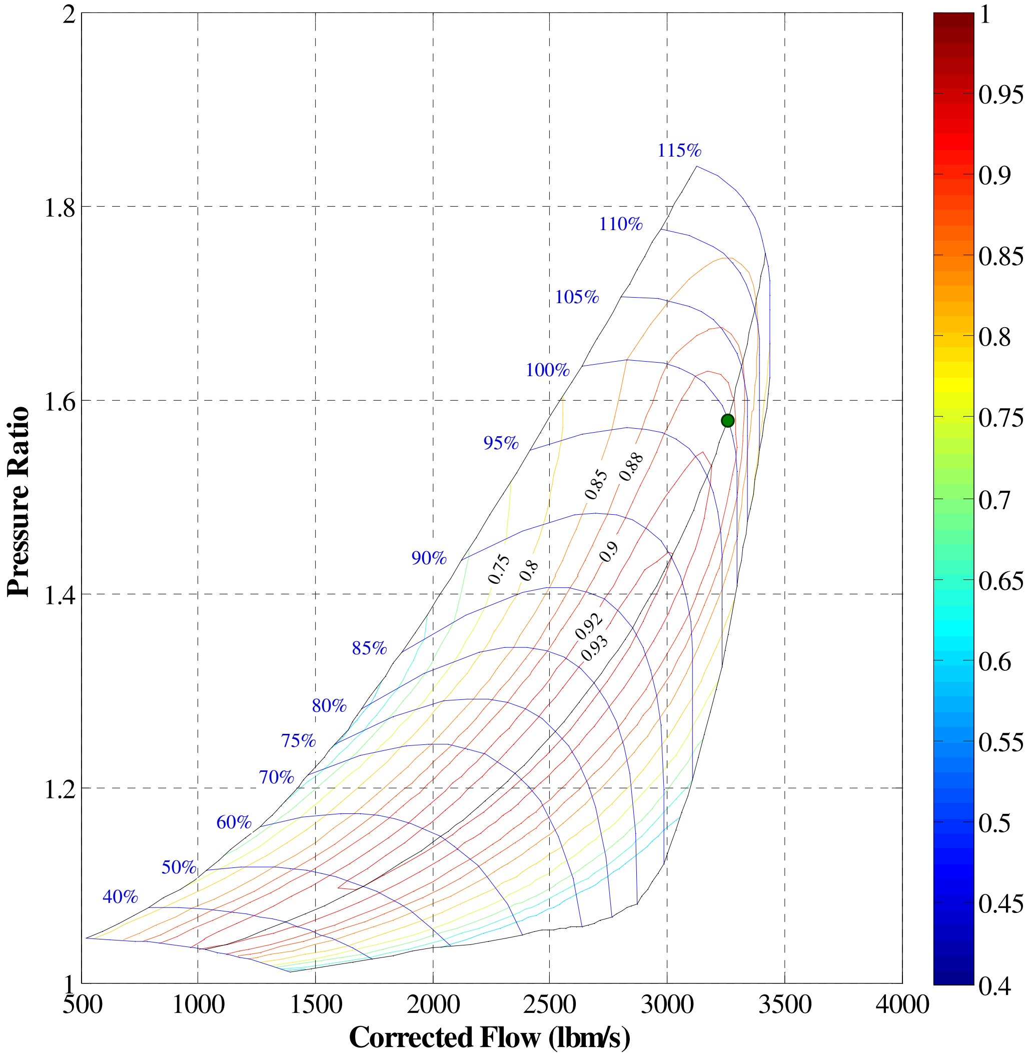


Figure 32. Base Fan Performance Map

One attribute to note about the construction of the compressor maps are the Rlines. These are reference lines used in the generation of the compressor maps as shown in Figure 33. The solver uses the Rlines, along with corrected flow as the independent parameters for defining the location of an operating condition on the compressor map. The pressure ratio, corrected speed and adiabatic efficiency are set by the choice of Rline and corrected flow.

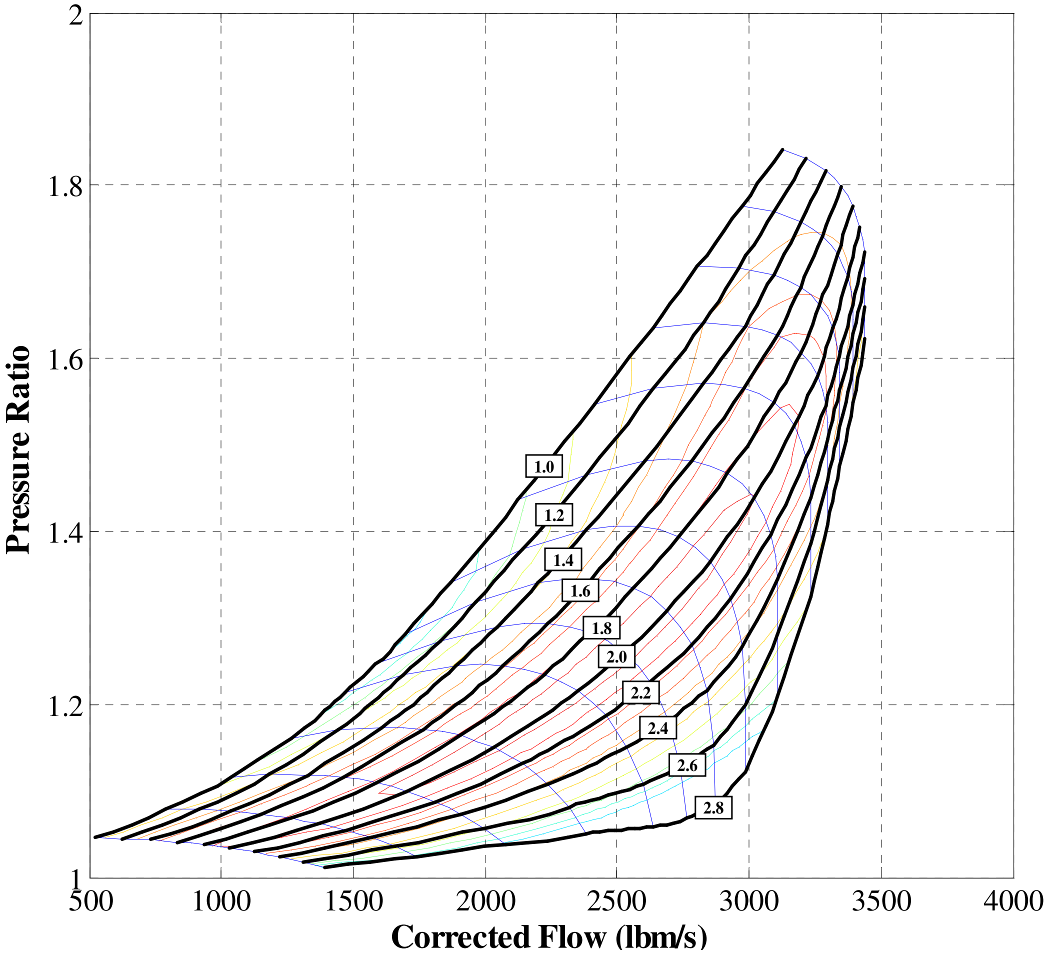


Figure 33. Compressor Map Rlines

4.4.3 LPC

The two component performance estimates required for the LPC performance are the LPC design point adiabatic efficiency as a function of LPCPR and the LPC specific flow. LPCPR in controlled by the design variables FPR and OPR. The equation for LPC adiabatic efficiency, given in equation (30, is a polynomial regression of data from CMPGEN, and is equivalent to a constant polytropic efficiency of 0.918 when the delta efficiency term LPC_Deff is 0.

$$\text{LPC\_Adiab} = -1.71831e^{-4} * \text{LPCPR}^3 + 2.42747e^{-3} * \text{LPCPR}^2 - 1.57814e^{-2} * \text{LPCPR} + 0.918597 + \text{LPC\_Deff} \qquad (30)$$

A plot of the LPC design point adiabatic efficiency for different levels of LPC_Deff is shown in Figure 34. The LPC specific flow is set to 26.5 lbm/s/ft^2 at LPCDP. No LPC equation for the shaft speed is calculated as the shaft speed is determined by the performance of the fan.

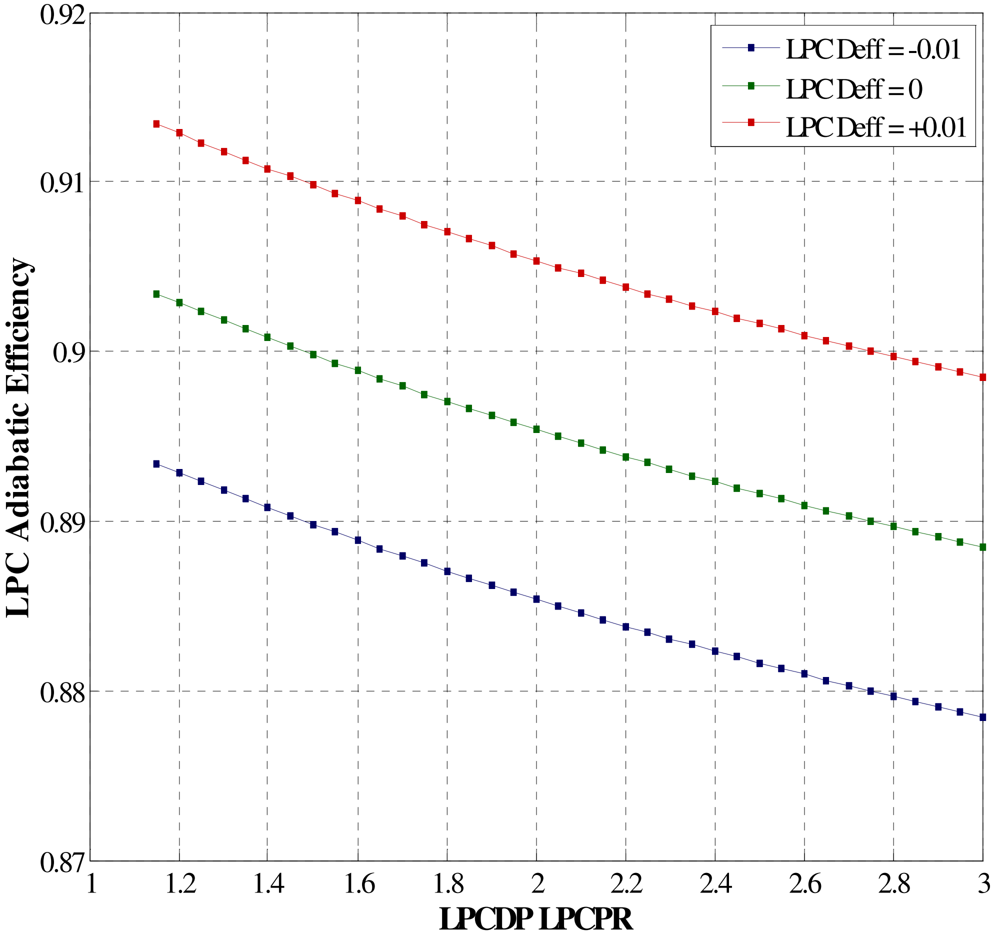


Figure 34. LPC Design Adiabatic Efficiency versus LPCPR

The base LPC performance map created by CMPGEN, with the LPC design point highlighted, is shown in Figure 35. As with the fan, NPSS scales the map for efficiency, pressure ratio and corrected flow.

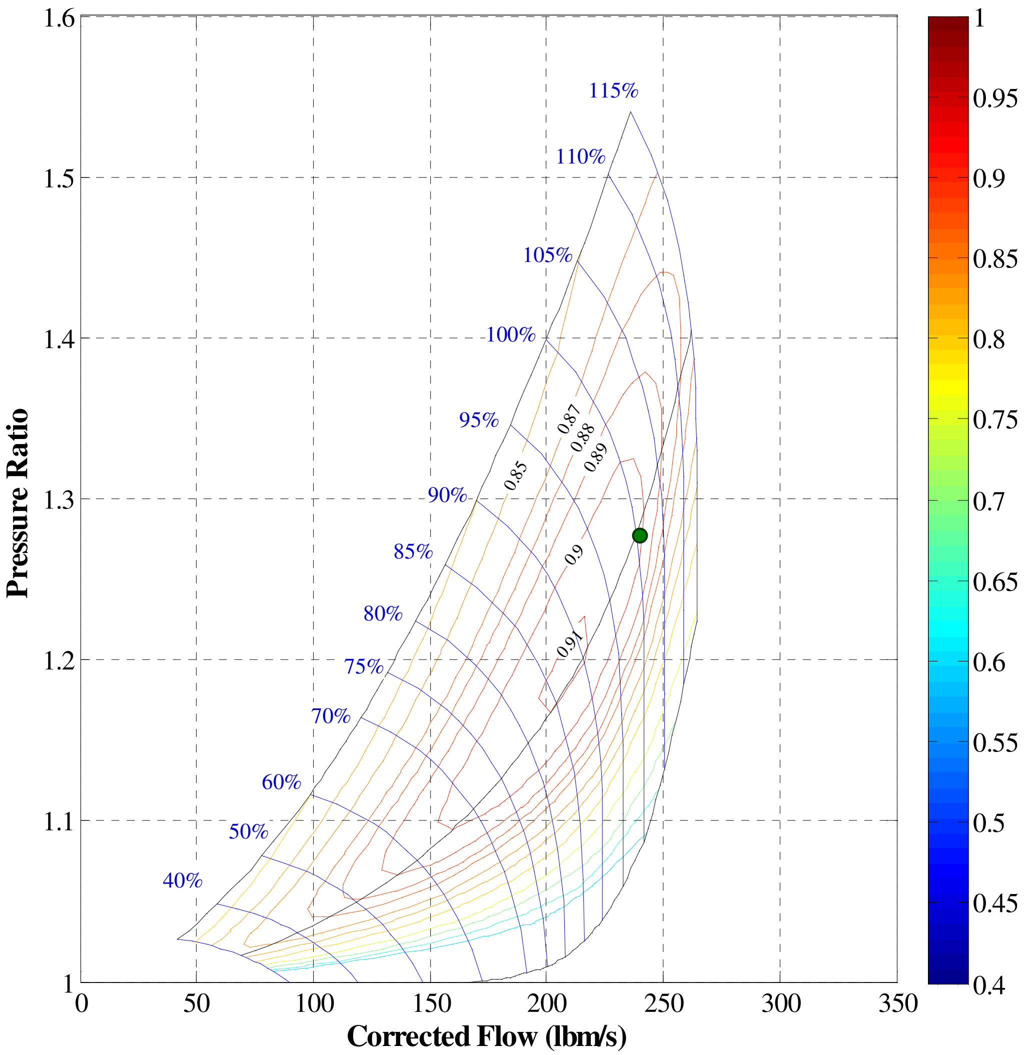


Figure 35. Base LPC Performance Map

4.4.4 HPC

The HPC is fixed both for performance and size. The HPC performance map was created in CMPGEN and is shown in Figure 36. The map remains constant for all candidate engine cycles. NPSS does not scale the map for efficiency, pressure ratio or corrected flow. Actually the scale factor for the HPC are included in the solver as dependent parameters and set to one. The HPC inlet flow area is fixed at 875 in^2 and the exit flow area to 100 in^2. The fixed geometry will result in different HPC inlet and exit flow conditions for the different candidate engine cycles. The HPC inlet hub to tip, required for the HP shaft speed calculation in equation 32, is set to 0.475.

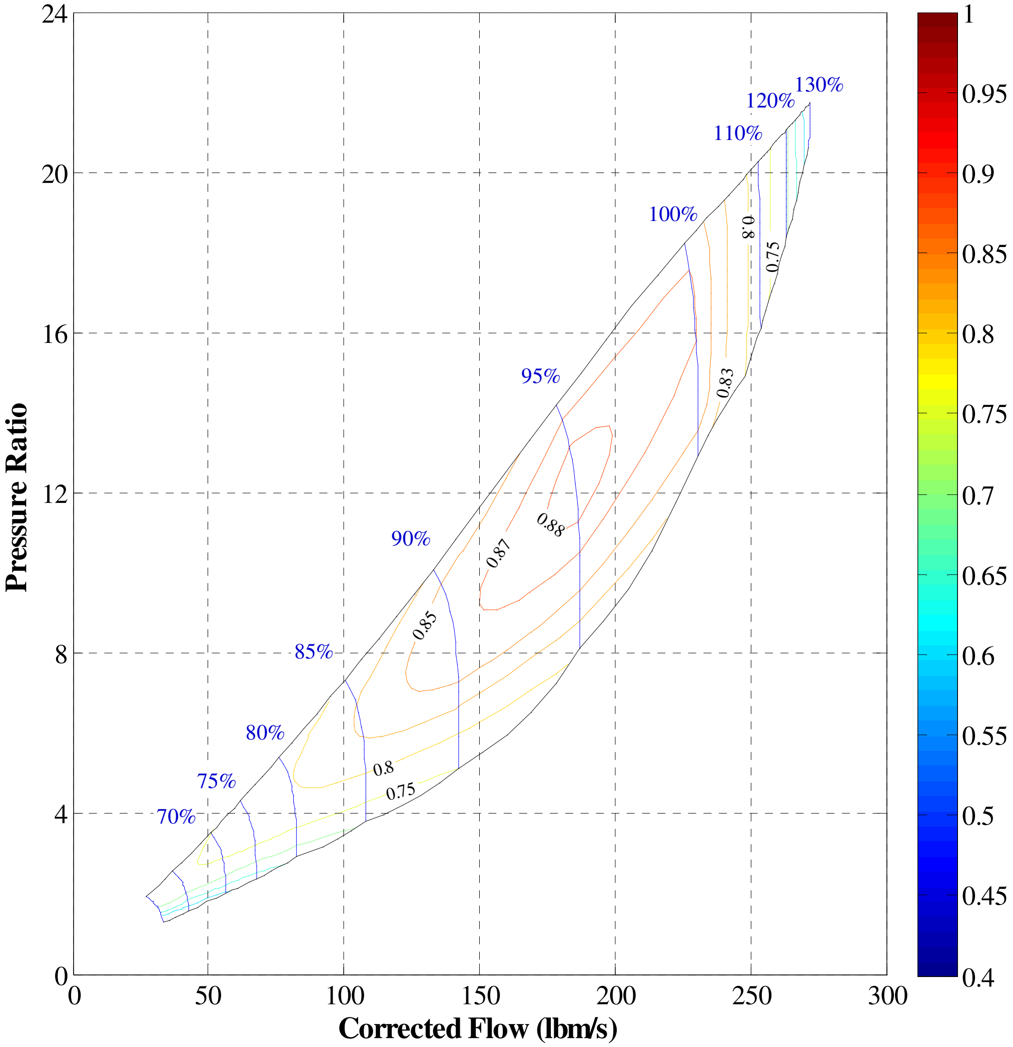


Figure 36. HPC Performance Map

4.4.5 HPT and LPT

Like the HPC, the HPT is also fixed for both performance and size. The HPT performance map is shown in Figure 37. Scale factors for the HPT are also include in the solver as dependent parameters and set to 1. The HPT inlet flow area is 425 in^2 and the exit area is 700 in^2. The LPT performance map is scaled by NPSS along with the fan and

LPC, based on the LPT adiabatic efficiency defined by the user. The base performance map for the LPT is shown in Figure 38.

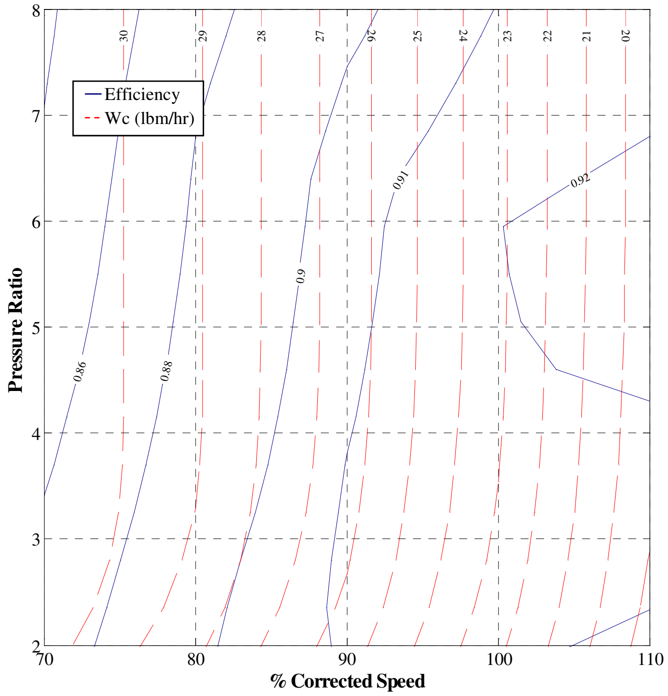


Figure 37. HPT Performance Map

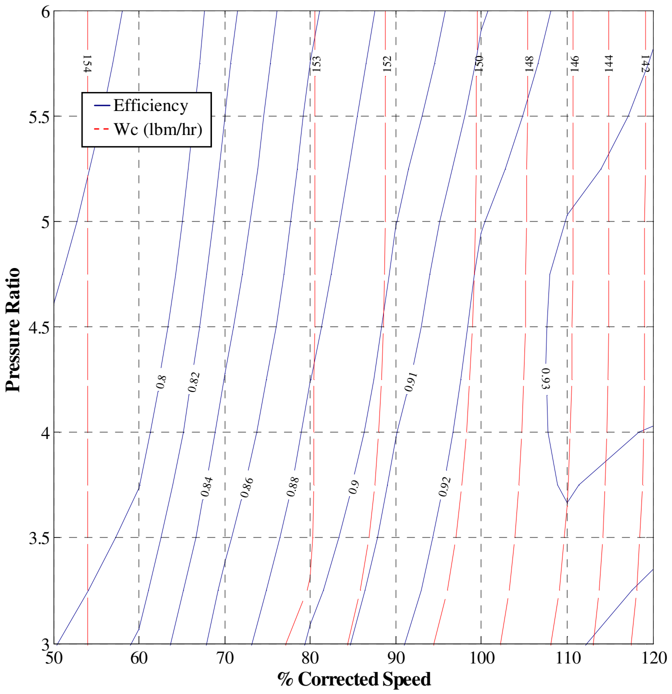


Figure 38. Base LPT Performance Map

4.4.6 Shafts

The shaft speeds for the LP shaft and HP shaft are shown in equation 31 and 32 respectively for the map scaling point. The LP shaft speed is dependent upon the area of the fan and fan tip speed. The HP shaft speed is dependent only on HPC_NcDes as the HPC hub to tip and area are constants. The 100% speed for the HPC is 8000 rpm.

$$\text{LP\_Nmech} = \text{Fan\_Utip} * \frac{60*12}{2\pi} * \sqrt{\frac{\pi*\left(1-\text{Fan\_HtoT}^2\right)}{\text{Fan\_Inlet\_Area}}} \tag{31}$$

$$\text{HP\_Nmech} = \text{HPC\_NcDes} * \frac{1325*60*12}{2\pi} * \sqrt{\frac{\pi*\left(1-\text{HPC\_HtoT}^2\right)}{\text{HPC\_Inlet\_Area}}} \tag{32}$$

4.4.7 Nozzles

The component performance estimates for the nozzles include the nozzle velocity coefficient[89] and discharge coefficient[89]. The velocity coefficient and discharge coefficient for the LCT engine are plotted in Figure 39 versus the nozzle pressure ratio. Both the core and bypass nozzle utilize the same performance maps.

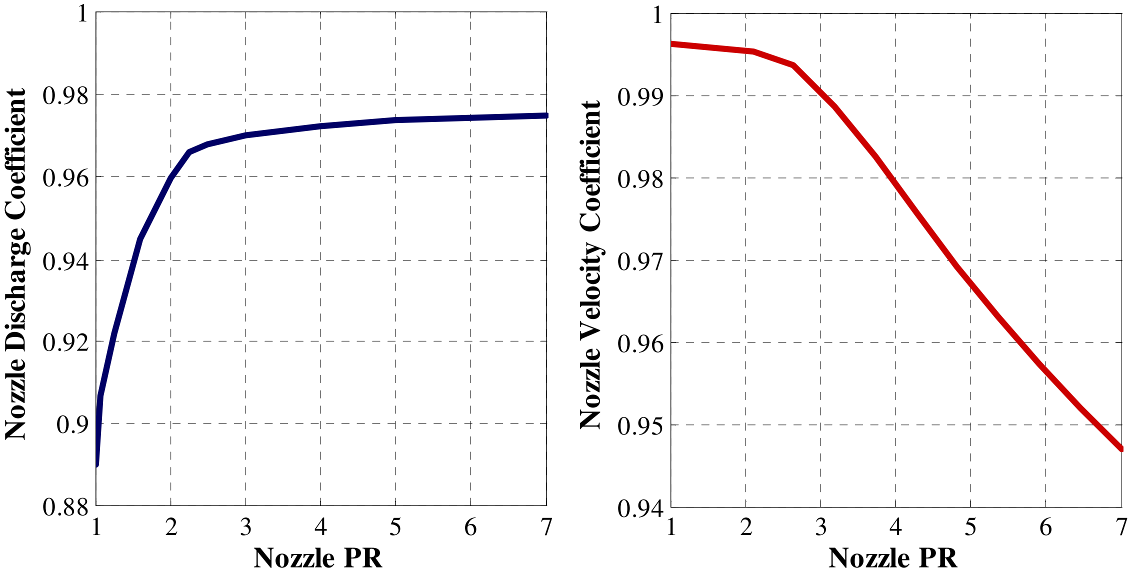


Figure 39. Nozzle Velocity Coefficient and Discharge Coefficient

4.5 Design Points

The next step in the simultaneous MDP method is the determination of the different design points. For the simultaneous MDP experiments a total of nine design points were identified for possible use within the MDP model. The operating conditions and specifications that define each design point are listed in Table 11. Not all of the experiments incorporate every design point.

Table 11. MDP Design Points for Experiments 2 through 5

Design Point	Mach	Altitude	Delta Temp.	Other Specifications
LPCDP	0.8	35,000 ft	0	Fan NcDes = 100%
HPCDP	0	0	0	HPC NcDes= f(LPCDP OPR)
TOC1	0.85	39,000 ft	0	Fn = 16,025 lbf
TOC2	0.85	35,000 ft	0	Fn = 19,600 lbf
TKO	0.25	0 ft	+27 °F	Fn = 75,000 lbf
HHDTKO	0.25	5000 ft	+27 °F	Fn = 67,000 lbf
Cutback	0.25	2000 ft	+18 °F	Fn = 60,000 lbf
SLSI	0	0	+27 °F	Fn = SLSU thrust, Includes customer bleed, installation effects and Horsepower extraction
SLSU	0	0	0	Fan NcDes = 100%, No customer bleed, installation effects or Horsepower extraction

The first design point is the reference point for the LP shaft components, the Low Pressure Component Design Point (LPCDP). This point is used to incorporate the technology rules for the fan and LPC. In general, there should be a reference point for the turbomachinery components on each shaft. The LPCDP is specified at Mach 0.8, at an altitude of 35,000 ft and ISA ambient temperature. The speed of the engine LP shaft is specified as 100% corrected speed. This design point is also used to locate the cruise operating line for the fan and LPC. By adjusting the Rline value for the LPCDP, the cruise operating line can be shifted as shown in Figure 40. The Rline value chosen for LPCDP was 2 for both the fan and LPC which places the cruise operating line slightly to the right of the peak efficiency line. Finally, the LPCDP is designated as the map scaling point for all experiments.

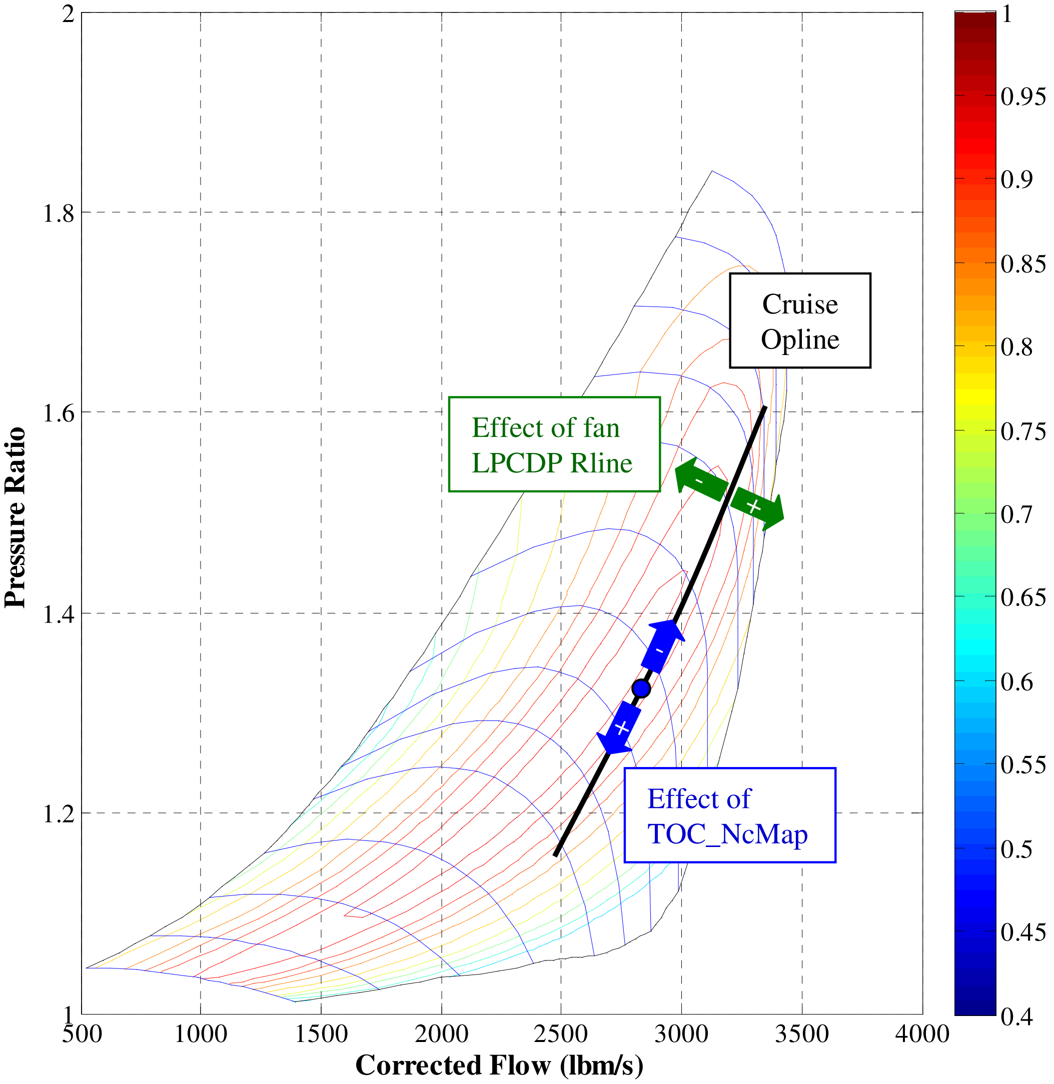


Figure 40. Illustration of Rline and TOC_NcMap on Cruise Operating Line

The next design point is the reference point for the HP shaft, the High Pressure Component Design Point (HPCDP). This design point is used to ensure that the engine core remains fixed for all engine designs. The reference point chosen was sea level static ISA, similar to a standard engine test condition. However, the engine speed or desired thrust value must also be specified at this design point. Unfortunately, a single reference speed will not suffice for the different engines within the CDS. A chart of HPCDP fan corrected speed as a function of the design variable OPR for different values of HPCDP HPC corrected speed, shown in Figure 41, illustrates why one value will not perform adequately. As OPR is increased, the fan corrected speed is reduced for the same HPC corrected speed. Operating the fan at extremely high or low speeds can result in efficiency and robustness issues for the solver. This results in the requirement that the HPCDP design point specification become a function of the design variable OPR. As the design point is only a reference point, it is better to adjust the HPC corrected speed to maintain a good operating condition for the fan than to maintain an arbitrary HPC corrected speed. Therefore, a target fan corrected speed was chosen of 70% and the corresponding HPC_Nc found for different OPRs.

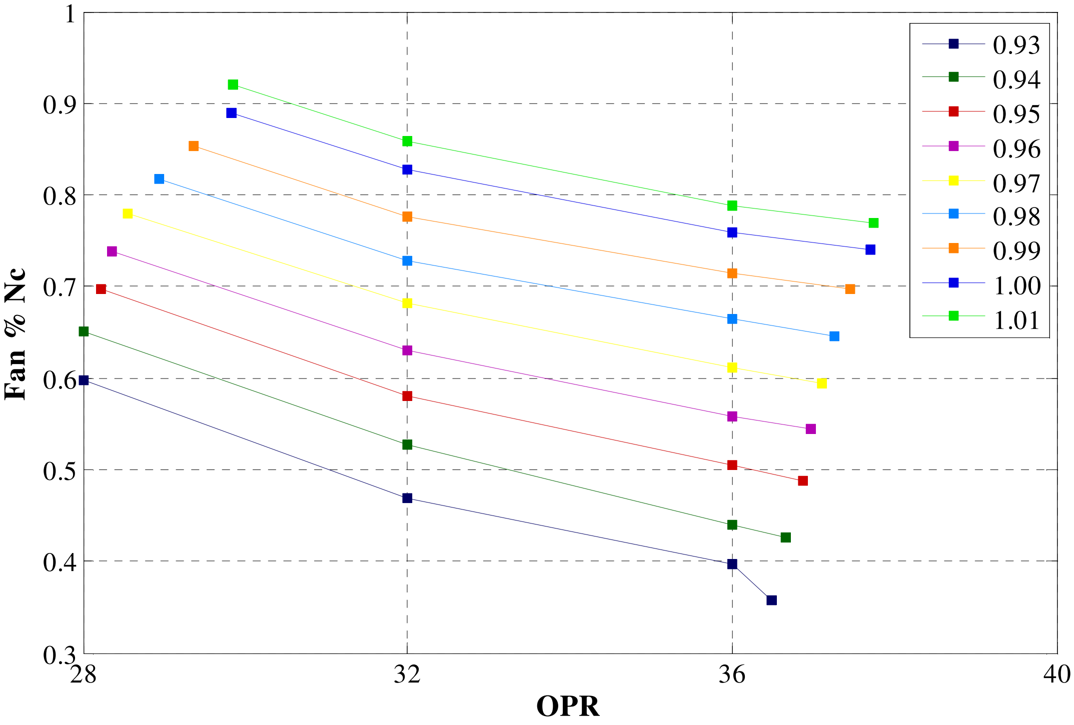


Figure 41. HPCDP Fan Nc vs. LPCDP OPR for Different HPCDP HPC Nc

A table of the HPC corrected speed and the associated adiabatic efficiency, mass flow and pressure ratio for different levels of OPR is listed in Table 12. The HPC adjusts from a corrected speed 95% to 102% for an increase in OPR from 28 to 44.

Table 12. HPCDP Design Point Performance Table

LPCDP OPR	HPC NC	HPC Eff.	HPC Wc	HPCPR
OPR ≤ 30, FPR ≤1.6	0.95	0.8823	186.3	11.60
OPR ≤ 30, FPR >1.6	0.93	0.8751	168.3	9.996
30 < OPR ≤ 34	0.98	0.8747	212.6	13.95
34 < OPR ≤ 38	0.99	0.8723	221.3	14.85
38 < OPR ≤ 42	1.00	0.8700	230.0	15.80
42 < OPR ≤ 50	1.01	0.8577	233.5	16.24
OPR > 50	1.02	0.8457	236.9	16.69

The nest five design points are all derived from the requirements and constraints. Each design point is specified to meet a minimum thrust level at a particular operating condition The operating conditions for TOC 1 are defined as Mach 0.85 at 39,000 ft and ISA. The engine cycle model is controlled to generate a minimum thrust of 16025 lbf at TOC1. The operating conditions for TOC 2 are defined as Mach 0.85 at 35,000 ft and ISA with a minimum thrust generation of 19600 lbf. The two TOC thrust design points will be used to specify maximum corrected speed and corrected flow for the engine. The operating conditions for TKO are defined as Mach 0.25, sea level and ISA +27 °F. The minimum thrust generated by the engine model at TKO is 75,000 lbf. The operating conditions for High Hot Day Take Off (HHDTKO) are defined as Mach 0.25 at an altitude of 5000 ft and ISA +27 °F. The thrust requirement is lower than the thrust requirement for TKO at 67,000 lbf. The cutback design point is defined as Mach 0.25 at an altitude of 2000 ft at standard acoustic day (ISA +18 °F). As this design point is a part power condition, the thrust generated for this condition is always 60,000 lbf.

The final two design points are Sea Level Static Installed (SLSI) and Sea Level Static Uninstalled (SLSU). The operating conditions for SLSI are Mach 0 , altitude 0, at ISA +27°F for flat rated temperature. The operating conditions for SLSU are Mach 0.0 , altitude 0, at ISA, horsepower and customer bleed are zero and inlet pressure recovery is set to one. The speed of the fan for SLSU is specified as 100% corrected speed. SLSU represents the test conditions for emissions testing. However, SLSI must match the same thrust as produced by the engine at SLSU without exceeding maximum T4 to maintain a flat rated thrust requirement.

The design points are created in the design run file found in Appendix C as assembly elements [58].The first element is the design point designated as the map scaling point. Within the element, the model file is included as well as the information necessary to output the cycle analysis results of the converged solution for the design point. Subsequent design points added to the model also include a preexecute command to include the file LCT_des_scl.int which contains the entire map scaling values from the first design point. This file can be found in Appendix C.

4.6 Design Point Mapping Matrix

The next step of the process is to create the DPMM. Since the DPMM links the design points with the design variables, performance requirements, component performance estimates and technology limits, it is unique to each experiment. Comparison of the different DPMMs best summarizes the differences between the five experiments. Each DPMM is listed in the experimental setup sections in chapter 5.

4.7 System of Nonlinear Equations

It has been stated that the design rules link the design points, performance requirements, technology rules and design variables to form a system of nonlinear equations. The design rules are dependent upon the specific design problem under consideration, the physics of the aerothermodynamic cycle, the selected architecture of the engine, the available technology level of the different engine components and the design experience of the cycle analyst. The simultaneous MDP method in effect records the design rules by the manner in which the independents, dependents and constraints are defined and linked together in the model to meet the performance requirements. These links will change depending on the application, engine architecture, level of technology or the design philosophy of the cycle analyst based prior engine cycle design experience. The simultaneous MDP method can handle a variety of different design rules as long as the rules produce feasible candidate engine cycles. The design rules can be adjusted for different applications, but the general simultaneous MDP method remains fixed to create a CDS.

4.7.1 Design Rules Summary for LCT Engine

The design rules incorporated into the simultaneous MDP method are influenced by many factors. These factors include the design problem under consideration, the physics of the aerothermodynamic cycle, the selected architecture of the engine, the available technology level of the different engine components and the design experience of the cycle analyst. Each of these factors are considered jointly when developing the design rules. Using his experience, the cycle analyst evaluates the design problem, physics of the cycle, engine architecture and component technology levels and devises a design philosophy as to how to use the various available cycle parameters to best meet the requirements of the engine. There may be several valid design philosophies for the cycle analyst to choose from that can be utilized for a single design problem. The design philosophy will definitely vary for different design problems and engine architectures. The simultaneous MDP method can be utilized with any valid philosophy that produces feasible candidate engine cycles. As the design problem changes, i.e. from a large commercial transport to a regional jet, supersonic business jet or military fighter, the design philosophy and design rules change based on the requirements and experience of the cycle analyst, but the simultaneous MDP method to create a feasible CDS remains the same.

The following is a summary of the design philosophy for the LCT engine used in the experiments. The LPCDP is the reference point for all components on the LP shaft. All of the component performance estimates and technology parameters for the components on the LP shaft are specified for this condition. Similarly, the HPCDP is the reference point for the components attached to the HP shaft. This design point ensures that the core components remain fixed for performance and geometry by adjusting the core component performance at LPCDP to match the specified performance at HPCDP. The T4 schedule is controlled at the most stringent take off design point, either TKO or HDDTKO. The T4 schedule is set by the FAR at LPCDP and the FAR at the most stringent take off point. The two FAR variables are adjusted to match the maximum T4 and thrust requirements at the most stringent take off condition. The mass flow schedule (engine sizing) is controlled at the most stringent top of climb design point, either TOC1 or TOC2. The mass flow schedule is set by the mass flow at LPCADP and the FAR at the most stringent top of climb design point which are adjusted to match the maximum overspeed of the engine and thrust requirement. The cooling flows for the HPT and LPT are calculated at TKO and HHDTKO and the highest value for each flow used for the rest of the design points. The rated thrust of the engine is determined by the minimum thrust calculated at SLSU, 100% fan Nc or SLSI with a combustor exit temperature equal to T4 max.

4.7.2 Independent-Dependent Combinations

The seventh step of the simultaneous MDP procedure is the most critical of the entire simultaneous MDP process, the construction of the system of nonlinear equations from the cycle design relations, engine matching relations and constraint relations. To form the system of equations the different relations must be divided into sets of independent and dependent parameters such that one independent parameter is manipulated by the modified Newton-Raphson solver to achieve the required value of one of the dependent parameters. The independent parameters must be unique, and there must be the same number of independent and dependent parameters to create a solvable system of nonlinear equations. All of the solver independent and dependent parameters for the different experiments are defined in Table 55 through Table 63 in Appendix B for each of the nine design points. The independent and dependent parameter definitions are added to NPSS in the NPSS function file found in Appendix C.

The independent-dependent combinations for the map scaling point will primarily derive from the cycle design relations. The independent-dependent combinations, shown for each design point in Table 13 though Table 21, should be considered as the solver adjusting the independent to obtain the dependent parameter value.

Table 13. LPCDP Independent-Dependent Combinations for Experiment 2

No.	Dependent	Independent	Class
1	LPCDP_FPR_Dep	LPCDP_FPR	User
2	LPCDP_OPR	LPCDP_LPCPR	User
3	LPCDP_Extr_Ratio	LPCDP_BPR	User
4	LPCDP_LPCEff_Error	LPCDP_LPCEff	User
5	LPCDP_HPNmech_Error	LPCDP_HPNmech	User
6	LPCDP_LPNmech_Error	LPCDP_LPNmech	User
7	LPCDP_Fan_Spec_Flow	LPCDP_Fan_InletMN	User
8	LPCDP_LPC_Spec_Flow	LPCDP_LPC_InletMN	User
9	SPL_Core_Area	LPCDP_SPL_Core_MN	User
10	SPL_Byp_Area	LPCDP_SPL_Byp_MN	User
11	Fan_Area_Out	LPCDP_Fan_Out_MN	User
12	LPC_Area_Out	LPCDP_LPC_Out_MN	User
13	LPT_Flow_Coef	LPCDP_Duct11_Out_MN	User
14	Duct13_Area_Out	LPCDP_Duct13_Out_MN	User
15	Duct15_Area_Out	LPCDP_Duct15_Out_MN	User
16	BypBld_Area_Out	LPCDP_BypBld_Out_MN	User
17	LPCDP.HP_Shaft.int_Nmech	LPCDP.HPT.S_map.parmMap	Engine Match
18	LPCDP.LP_Shaft.int_Nmech	LPCDP.LPT.S_map.parmMap	Engine Match

The first section of independent-dependent parameter combinations to be addressed are design variable related for LPCDP. These are the first 4 combinations listed in Table 13 and are user defined from the cycle design relations. The second group, combinations 5 and 6, are also user defined from the cycle design relations and are included to calculate the HP and LP shaft speeds. The final group of user defined combinations for LPCDP, combinations 7 through 16 set the flow area of the engine at key locations along the flowpath. The final two combinations, derived from the engine match relations, adjust the work produced by the turbines at LPCDP to match the work of the compressors through a torque balance on each shaft.

Each of the remaining eight design points will have a common set of independent-dependent combinations obtained from the engine match relations to match each design point to the component performance maps based on the map scaling information passed from LPCDP. These include the previously mentioned LP and HP shaft torque balances, a single combination for each of the fan, LPC, HPC, HPT, and LPT maps to match the incoming mass flow with the value in the performance map, and a combination for each of the nozzles to match the nozzle flow area. Each design point will have at least one additional user defined independent-dependent combination.

For the HPCDP, several user defined independent-dependent combinations were added to the solver to ensure that the core remained fixed. The first group of independent-dependent combinations, numbers 19 to 23 in Table 14, keep the performance map of the HPC fixed by ensuring the scale factors remain constant. The next group, combinations 24 to 26 keep the performance map of the HPT fixed while the last 4 user defined independent-dependent combinations, 27 through 30, fix the inlet and exit areas of both the HPC and HPT. Notice that not all of the independent parameters controlling the HPCDP dependent parameter functions were defined at HPCDP. It is possible to control a dependent at one design point with and independent from another design point. All of the user defined dependents for HPCDP, with the exception of HPCDP_HPC_Nc, demonstrate this principle.

Table 14. HPCDP Independent-Dependent Combinations for Experiment 2

No.	Dependent	Independent	Class
19	HPCDP_HPC_Nc	HPCDP_FAR	User
20	HPCDP_HPCPR	LPCDP_HPCPR	User
21	HPCDP_HPC_Wc	LPCDP_HPC_NcDes	User
22	HPCDP_HPCEff	LPCDP_HPCEff	User
23	HPCDP_HPC_Rline	LPCDP_HPC_Rline	User
24	HPCDP_HPT_s_effDes	LPCDP_HPT_effDes	User
25	HPCDP_HPT_s_WcDes	LPCDP_HPT_NcDes	User
26	HPCDP_HPT_s_PRdes	LPCDP_HPT_PRDes	User
27	HPCDP_HPC_InletArea	LPCDP_HPC_InletMN	User
28	HPCDP_HPC_ExitArea	LPCDP_HPC_ExitMN	User
29	HPCDP_HPT_InletArea	LPCDP_HPT_InletMN	User
30	HPCDP_HPT_ExitArea	LPCDP_HPT_ExitMN	User
31	HPCDP.Core_Nozz.Area	HPCDP.Ambient.W	Engine Match
32	HPCDP.Byp_Nozz.Area	HPCDP.Splitter.BPR	Engine Match
33	HPCDP.Fan.S_map.errWc	HPCDP.Fan.S_map.RlineMap	Engine Match
34	HPCDP.LPC.S_map.errWc	HPCDP.LPC.S_map.RlineMap	Engine Match
35	HPCDP.HPC.S_map.errWc	HPCDP.HPC.S_map.RlineMap	Engine Match
36	HPCDP.HPT.S_map.errWp	HPCDP.HPT.S_map.parmMap	Engine Match
37	HPCDP.LPT.S_map.errWp	HPCDP.LPT.S_map.parmMap	Engine Match
38	HPCDP.HP_Shaft.int_Nmech	HPCDP.HP_Shaft.Nmech	Engine Match
39	HPCDP.LP_Shaft.int_Nmech	HPCDP.LP_Shaft.Nmech	Engine Match

A total of three user defined independent-dependent combinations were included. for the design points TOC 1 and TOC 2, found in Table 15 and Table 16. Both adjust their respective independent parameter FAR to achieve the thrust defined in the requirements. In addition, the design variable TOC_NcMap was included as dependent number 40 for TOC1 to specify the operating condition at which the thrust must be met.. The fan speed for TOC 2 is not added to the independent-dependent combinations as it shares the same independent as TOC1_Fan_Nc. Therefore, the fan speed at TOC 2 will be added in a later section as a constraint for TOC1_Fan_Nc.

Table 15. TOC 1 Independent-Dependent Combinations for Experiment 2

No.	Dependent	Independent	Class
40	TOC1_Fan_Nc	LPCDP_W	User
41	TOC1_Thrust	TOC1_FAR	User
42	TOC1.Core_Nozz.Area	TOC1.Ambient.W	Engine Match
43	TOC1.Byp_Nozz.Area	TOC1.Splitter.BPR	Engine Match
44	TOC1.Fan.S_map.errWc	TOC1.Fan.S_map.RlineMap	Engine Match
45	TOC1.LPC.S_map.errWc	TOC1.LPC.S_map.RlineMap	Engine Match
46	TOC1.HPC.S_map.errWc	TOC1.HPC.S_map.RlineMap	Engine Match
47	TOC1.HPT.S_map.errWp	TOC1.HPT.S_map.parmMap	Engine Match
48	TOC1.LPT.S_map.errWp	TOC1.LPT.S_map.parmMap	Engine Match
49	TOC1.HP_Shaft.int_Nmech	TOC1.HP_Shaft.Nmech	Engine Match
50	TOC1.LP_Shaft.int_Nmech	TOC1.LP_Shaft.Nmech	Engine Match

Table 16. TOC 2 Independent-Dependent Combinations for Experiment 2

No.	Dependent	Independent	Class
51	TOC2_Thrust	TOC2_FAR	User
52	TOC2.Core_Nozz.Area	TOC2.Ambient.W	Engine Match
53	TOC2.Byp_Nozz.Area	TOC2.Splitter.BPR	Engine Match
54	TOC2.Fan.S_map.errWc	TOC2.Fan.S_map.RlineMap	Engine Match
55	TOC2.LPC.S_map.errWc	TOC2.LPC.S_map.RlineMap	Engine Match
56	TOC2.HPC.S_map.errWc	TOC2.HPC.S_map.RlineMap	Engine Match
57	TOC2.HPT.S_map.errWp	TOC2.HPT.S_map.parmMap	Engine Match
58	TOC2.LPT.S_map.errWp	TOC2.LPT.S_map.parmMap	Engine Match
59	TOC2.HP_Shaft.int_Nmech	TOC2.HP_Shaft.Nmech	Engine Match
60	TOC2.LP_Shaft.int_Nmech	TOC2.LP_Shaft.Nmech	Engine Match

As with TOC 1 and TOC 2, the independent-dependent combinations for design points TKO and HHDTKO, found in Table 17 and Table 18, both adjust the independent parameter FAR to meet their respective thrust requirements. TKO also includes the design variable T4_Max with dependent 62. The final four user defines independent-dependent combinations determine the necessary cooling flow for both the HPT and LPT. As before with the TOC 2 fan speed, T4 and the cooling flow calculations for HHDTKO share the same independents as the TKO dependents. Therefore, these dependents will also be included as constraints.

Table 17. TKO Independent-Dependent Combinations for Experiment 2

No.	Dependent	Independent	Class
61	TKO_Thrust	TKO_FAR	User
62	TKO_T4max	LPCDP_FAR	User
63	TKO.HPT.Cool.BleedFlow1	LPCDP_HPT_Cool1	User
64	TKO.HPT.Cool.BleedFlow2	LPCDP_HPT_Cool2	User
65	TKO.LPT.Cool.BleedFlow1	LPCDP_LPT_Cool1	User
66	TKO.LPT.Cool.BleedFlow2	LPCDP_LPT_Cool2	User
67	TKO.Core_Nozz.Area	TKO.Ambient.W	Engine Match
68	TKO.Byp_Nozz.Area	TKO.Splitter.BPR	Engine Match
69	TKO.Fan.S_map.errWc	TKO.Fan.S_map.RlineMap	Engine Match
70	TKO.LPC.S_map.errWc	TKO.LPC.S_map.RlineMap	Engine Match
71	TKO.HPC.S_map.errWc	TKO.HPC.S_map.RlineMap	Engine Match
72	TKO.HPT.S_map.errWp	TKO.HPT.S_map.parmMap	Engine Match
73	TKO.LPT.S_map.errWp	TKO.LPT.S_map.parmMap	Engine Match
74	TKO.HP_Shaft.int_Nmech	TKO.HP_Shaft.Nmech	Engine Match
75	TKO.LP_Shaft.int_Nmech	TKO.LP_Shaft.Nmech	Engine Match

Table 18. HHDTKO Independent-Dependent Combinations for Experiment 2

No.	Dependent	Independent	Class
76	HHDTKO_Thrust	HHDTKO_FAR	User
77	HHDTKO.Core_Nozz.Area	HHDTKO.Ambient.W	Engine Match
78	HHDTKO.Byp_Nozz.Area	HHDTKO.Splitter.BPR	Engine Match
79	HHDTKO.Fan.S_map.errWc	HHDTKO.Fan.S_map.RlineMap	Engine Match
80	HHDTKO.LPC.S_map.errWc	HHDTKO.LPC.S_map.RlineMap	Engine Match
81	HHDTKO.HPC.S_map.errWc	HHDTKO.HPC.S_map.RlineMap	Engine Match
82	HHDTKO.HPT.S_map.errWp	HHDTKO.HPT.S_map.parmMap	Engine Match
83	HHDTKO.LPT.S_map.errWp	HHDTKO.LPT.S_map.parmMap	Engine Match
84	HHDTKO.HP_Shaft.int_Nmech	HHDTKO.HP_Shaft.Nmech	Engine Match
85	HHDTKO.LP_Shaft.int_Nmech	HHDTKO.LP_Shaft.Nmech	Engine Match

The only user defined independent-dependent combination for the cutback design point found in Table 19 is combination 86, the cutback thrust requirement and independent parameter FAR.

Table 19. Cutback Independent-Dependent Combinations for Experiment 2

No.	Dependent	Independent	Class
86	Cutback_Thrust	Cutback_FAR	User
87	Cutback.Core_Nozz.Area	Cutback.Ambient.W	Engine Match
88	Cutback.Byp_Nozz.Area	Cutback.Splitter.BPR	Engine Match
89	Cutback.Fan.S_map.errWc	Cutback.Fan.S_map.RlineMap	Engine Match
90	Cutback.LPC.S_map.errWc	Cutback.LPC.S_map.RlineMap	Engine Match
91	Cutback.HPC.S_map.errWc	Cutback.HPC.S_map.RlineMap	Engine Match
92	Cutback.HPT.S_map.errWp	Cutback.HPT.S_map.parmMap	Engine Match
93	Cutback.LPT.S_map.errWp	Cutback.LPT.S_map.parmMap	Engine Match
94	Cutback.HP_Shaft.int_Nmech	Cutback.HP_Shaft.Nmech	Engine Match
95	Cutback.LP_Shaft.int_Nmech	Cutback.LP_Shaft.Nmech	Engine Match

The final two design points, SLSU and SLSI, each have a single user defined independent-dependent combination. as shown in Table 20 and Table 21. The SLSI FAR is adjusted to meet the SLSI thrust target. The SLSU FAR is adjusted to match the SLSL fan speed target.

Table 20. SLSI Independent-Dependent Combinations for Experiment 5

No.	Dependent	Independent	Class
96	SLSI_Thrust	SLSI_FAR	User
97	SLSI.Core_Nozz.Area	SLSI.Ambient.W	Engine Match
98	SLSI.Byp_Nozz.Area	SLSI.Splitter.BPR	Engine Match
99	SLSI.Fan.S_map.errWc	SLSI.Fan.S_map.RlineMap	Engine Match
100	SLSI.LPC.S_map.errWc	SLSI.LPC.S_map.RlineMap	Engine Match
101	SLSI.HPC.S_map.errWc	SLSI.HPC.S_map.RlineMap	Engine Match
102	SLSI.HPT.S_map.errWp	SLSI.HPT.S_map.parmMap	Engine Match
103	SLSI.LPT.S_map.errWp	SLSI.LPT.S_map.parmMap	Engine Match
104	SLSI.HP_Shaft.int_Nmech	SLSI.HP_Shaft.Nmech	Engine Match
105	SLSI.LP_Shaft.int_Nmech	SLSI.LP_Shaft.Nmech	Engine Match

Table 21. SLSU Independent-Dependent Combinations for Experiment 5

No.	Dependent	Independent	Class
106	SLSU_FanNcMap	SLSU_FAR	User
107	SLSU.Core_Nozz.Area	SLSU.Ambient.W	Engine Match
108	SLSU.Byp_Nozz.Area	SLSU.Splitter.BPR	Engine Match
109	SLSU.Fan.S_map.errWc	SLSU.Fan.S_map.RlineMap	Engine Match
110	SLSU.LPC.S_map.errWc	SLSU.LPC.S_map.RlineMap	Engine Match
111	SLSU.HPC.S_map.errWc	SLSU.HPC.S_map.RlineMap	Engine Match
112	SLSU.HPT.S_map.errWp	SLSU.HPT.S_map.parmMap	Engine Match
113	SLSU.LPT.S_map.errWp	SLSU.LPT.S_map.parmMap	Engine Match
114	SLSU.HP_Shaft.int_Nmech	SLSU.HP_Shaft.Nmech	Engine Match
115	SLSU.LP_Shaft.int_Nmech	SLSU.LP_Shaft.Nmech	Engine Match

The independent-dependent combinations are added to the design run file found in Appendix C. The autosolver command [58] adds the combinations from the engine match relations and the user defined combinations are added, usually as independent-dependent pairs by the cycle analyst. It should be noted that the solver does not consider the order that the user inputs the combinations nor the grouping as described in Table 13 through Table 19. The solver will simultaneously adjust all of the independents to match all of the dependent parameters. However, listing each combination separately ensures that the Jacobian matrix will have a nonzero determinant value which results in a solvable system of equations.

4.7.3 Constraint-Dependent Combinations

Each constraint included in the NPSS model is attached to a dependent parameter. The pairing of the constraints and the dependent parameters is shown in Table 22. The first four constraints are for the chargeable and nonchargeable cooling flows to the HPT and LPT. When active the cooling flow calculations are obtained at both the TKO and HHDTKO design points. For each bleed flow, the highest required value is matched by the modified Newton-Raphson solver. The next two constraints, maximum fan diameter and maximum cutback Vmix are both paired to FPR with the maximum fan diameter providing a minimum constraint on FPR. The relationship between fan diameter and FPR for the LCT engine is shown in Figure 42. For different values of OPR, the fan diameter constraint is of 128 inches is met ate between a FPR of 1.52 and 1.54. Cutback Vmix on the other hand, increases as FPR increases as shown in Figure 43. This makes cutback_Vmix a maximum constraint of FPR. The FPR design space is therefore limited to values that fall within these two constraints. The next four constraints are attached to OPR, a minimum LPCPR, and maximum T3 at either TKO, HHDTKO or SLSI design points. The minimum LPCPR is the minimum beneficial value that would be utilized on the engine. If values below the minimum were desired, then the LPC would be removed from the architecture of the engine. Similar to FPR, the OPR design space is limited between the minimum LPCPR constraint and the most severe T3, either at TKO of HHDTKO.

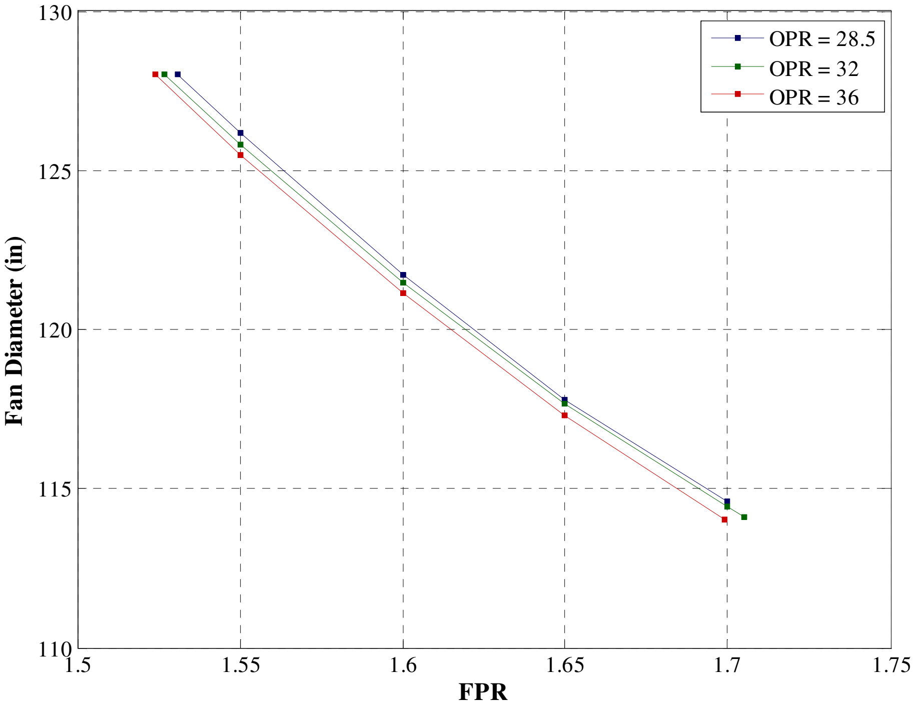


Figure 42. Constraint Fan Diameter Relation to FPR

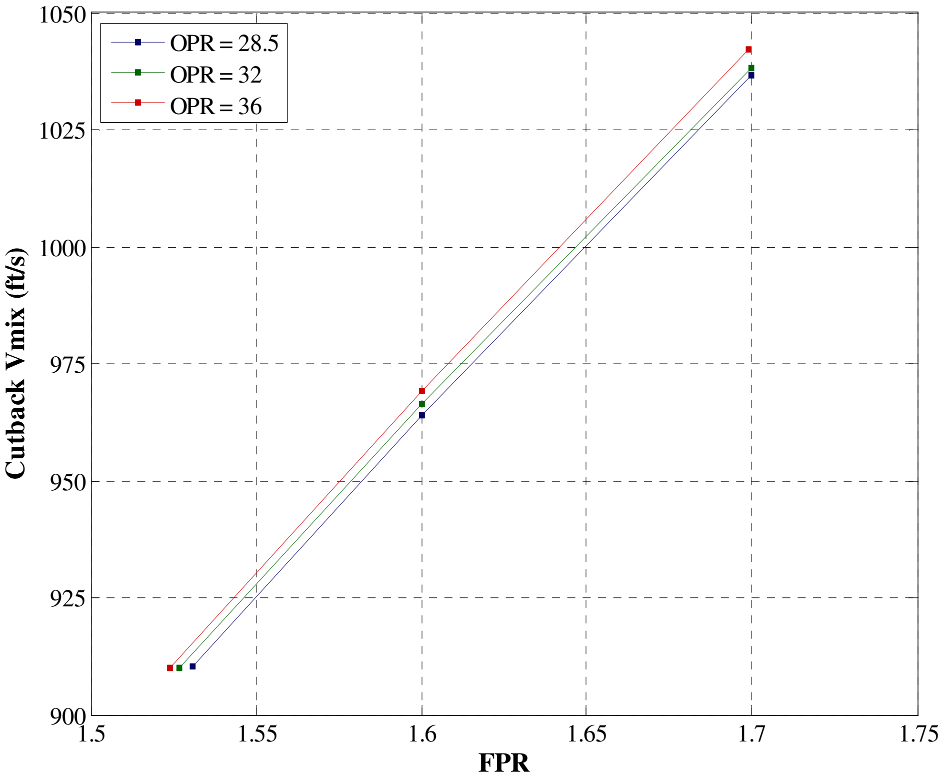


Figure 43. Constraint Cutback_Vmix Relation to FPR

Table 22. Constraint-Dependent Pairings for Experiment 2 through 5

Constraint	Dependent	Max/Min
HHDTKO_HPT_BleedFlow1	TKO.HPT.Cool.BleedFlow1	Max
HHDTKO_HPT_BleedFlow2	TKO.HPT.Cool.BleedFlow2	Max
HHDTKO_LPT_BleedFlow1	TKO.LPT.Cool.BleedFlow1	Max
HHDTKO_LPT_BleedFlow2	TKO.LPT.Cool.BleedFlow2	Max
Max_Fan_Diameter	LPCDP_FPR_Dep	Min
Cutback_Vmix	LPCDP_FPR_Dep	Max
LPCDP_LPCPRmin	LPCDP_OPR	Min
TKO_T3	LPCDP_OPR	Max
HHDTKO_T3	LPCDP_OPR	Max
SLSI_T3	LPCDP_OPR	Max
TOC2_Fan_Nc	TOC1_Fan_Nc	Max
HHDTKO_T4max	TKO_T4max	Max
SLSI_T4max	SLSI_Thrust	Max
SLSU_T4max	SLSU_FanNcMap	Max
TKO_Core_Bypass_dP	LPCDP_OPR	Max
HHDTKO_Core_Bypass_dP	LPCDP_OPR	Max

The TOC2_Fan_Nc constraint is attached to the same dependent parameter for TOC1. Whichever condition requires the most overspeed to obtain the required thrust is set to match the value from the DoE. While the next three constraints are for T4 at different conditions, HHDTKO, SLSI and SLSU, each constraint is attached to a different dependent. Therefore it is possible for all three constraints to be active at the same time. HHDTKO T4max is attached to TKO T4max. Whichever operating condition requires the higher T4 to obtain the required thrust is set to match the value from the DoE. SLSI T4max is attached to SLSI thrust and SLSU T4 max is attached to SLSU fan NcMap. If either of these is active than the respective thrust or fan speed is lowered until the T4 for that condition matched the value in the DoE. The last two constraints measure the pressure difference between the core and bypass streams at the exit of the HPC for TKO and HHDTKO design points. These constraints also limit the maximum OPR. However, these constraints are dependent on OPR and FPR, and as such could have been attached to FPR and provide a minimum limit. The choice is left to the discretion of the cycle analyst.

It should be noted that, unlike the link between independents and dependents, the solver does account for the link between constraints and dependents [58]. When a constraint is exceeded, it replaces the dependent parameter in the Jacobian. From an examination of the list of constraints, it is apparent that a dependent can be attached to multiple constraints. LPCDP_FPR_Dep is constrained by Max_Fan_Diameter and Cutback_Vmix while LPCDP_OPR is constrained by LPCDP_LPCPRmin, TKO_T3, and HHDTKO_T3. If more than one constraint is violated, then the most severe constraint is incorporated into the Jacobian. However, a constraint may not be attached to more than one dependent parameters as the solver would not be properly informed as to which dependent to replace should the constraint be violated.

Certain constraints are automatically enabled by the inclusion of related design points. Whenever, HHDTKO and TKO are included in an experiment the HHDTKO_HPT_BleedFlow1 constraint, HHDTKO_HPT_BleedFlow2 constraint, HHDTKO_LPT_BleedFlow1 constraint and HHDTKO_LPT_BleedFlow2 constraint are enabled. If the T3 constraint is enabled then it is applied to any of the following design points that are included in the experiment: TKO, HHDTKO and SLSI. The design points SLSI and SLSU are always included together in an experiment, and their T4 constraints are always enabled.

4.8 Design Variable Value Assignment

The eighth step of the simultaneous MDP method is the determination of the candidate engine cycles by assigning values to the design variables. The ranges of the design variables and the number of candidate engine cycles that populate the CDS are unique for each experiment and are provided in the experimental setup sections in chapter 5.

4.9 Solver Independent Parameters Initial Value

The ninth step is to determine the starting values of the independent parameters refer to as the initial iterate. The importance of this step and the process for determining the values is discussed as part of experiment 3 in chapter 5.

4.10 Solver Setup

The next step is to determine the appropriate settings for the modified Newton-Raphson solver. There are many consideration that go into determining the proper values of the different solve settings including: size of the design problem as reflected in the number of terms in the Jacobian, number of candidate engine cycles that populate the design space, importance to the cycle analyst in finding all of the feasible candidate engine cycles, and computational resources available to the cycle analyst. Some of the tradeoffs in the more important solver settings are discussed in chapter 5 as part of the third experiment.

One solver parameter that was held constant for all experiments was the solver convergence criteria. The solver convergence criteria is set as a fractional error term [58]. The value of the tolerance is set to 5e-8. This value was determined to reduce the error sufficiently that the differences between the same operating conditions from on-design and off-design analysis are acceptable. This tolerance applies to all but three of the dependent parameters in the LCT engine model; HPCDP_HPT_s_effDes, HPCDP_HPT_s_WcDes and HPCDP_HPT_s_PRdes – the scale factors for the HPT. The values of these three parameters are sufficiently close to zero to necessitate an absolute rather than a fractional error term for convergence[58]. The tolerance for the three absolute error terms was set to 0.001.

4.11 Solver Execution

The final step of the simultaneous MDP method is the execution of the solver. The step is repeated for every candidate engine cycle that populates the CDS. There are two parts to the execution; the initial attempt to find the solution for a candidate engine cycle and a repair algorithm to recover feasible candidate engines that initially fail to converge.

4.11.1 Computational Resources

One of the key metrics for the solver is its efficiency measured in terms of the number model passes required to reach convergence. This metric is important for reducing the time necessary to populate a large cycle design space. The comments will be with regards to executing the simultaneous MDP method on a single PC. However, many of the experiments will take advantage of distributive computing capabilities available to the author allowing for a shift in focus from efficiency to maximum convergence success rate.

4.11.2 Solver Convergence Failure Repair Algorithm

From Chapter 3 it was shown that there are three possible reasons for a the solver to fail to converge:1) there may not be a feasible solution for the solver to obtain, 2) the initial iterate is not close enough to the final solution for the solver to converge to the solution and 3) due to discontinuities, such as those created by the discrete points define in the performance maps, it is possible for the solver to get into an infinite loop and never achieve the tolerance established for the convergence criteria. The MDP repair algorithm addresses the last two reasons for convergence failure.

The MDP repair algorithm is depicted in Figure 44. The algorithm begins with the following information; the design variables and technology parameter values from for the failed case of the DoE (called the DoE Original values), the initial iterate, and the values of the design variables and technology parameters whose solution is the initial iterate (called the DoE Base values). The first step of the process is to calculate a step that is halfway between the DoE Original and DoE base values of the design variables and technology parameters. The solver then attempts to find a solution to the new values of the design variables and technology parameters (called DoE New). If the solver is successful, the repair algorithm updates the initial iterate to the solution for DoE New and then increments the design variables and technology parameters by the same step size. If on the other hand, the solver fails to converge, the step size is divided by two and the solver is executed for an altered Doe New that is closer to DoE Base. This process is repeated until either a solution is found for the original DoE case values or a prescribed number of iterations is met. This part of the process will find solutions for cases that fail for the second failure mode, that the initial iterate is not close enough to the final solution.

However, if the case fails for the third failure mode regarding an infinite loop, no matter how close the initial iterate approached the "solution", the solver will still fail to converge. In this instance, it can be desirable to alter the original values of the design variables and technology parameters by an extremely small step, called DoE Increment, and attempt to solve. This small increment will avoid the condition that causes the infinite loop and allow a solution to be found. Such a method would be useful in maintaining the structure of a DoE to provide better fits by not allowing failed cases to provide gaps in the regression data. The NPSS file containing the MDP repair algorithm can be found in Appendix C.

DoE Step = Step Size * (DoE Original – DoE Base)

DoE New = DoE Current + DoE Step

DoE Difference = DoE New – DoE Original

DoE Increment = DoE Original +/- DoE Delta

Check 1 = # of consecutive solver convergence failure

Check 2 = # of total solver runs

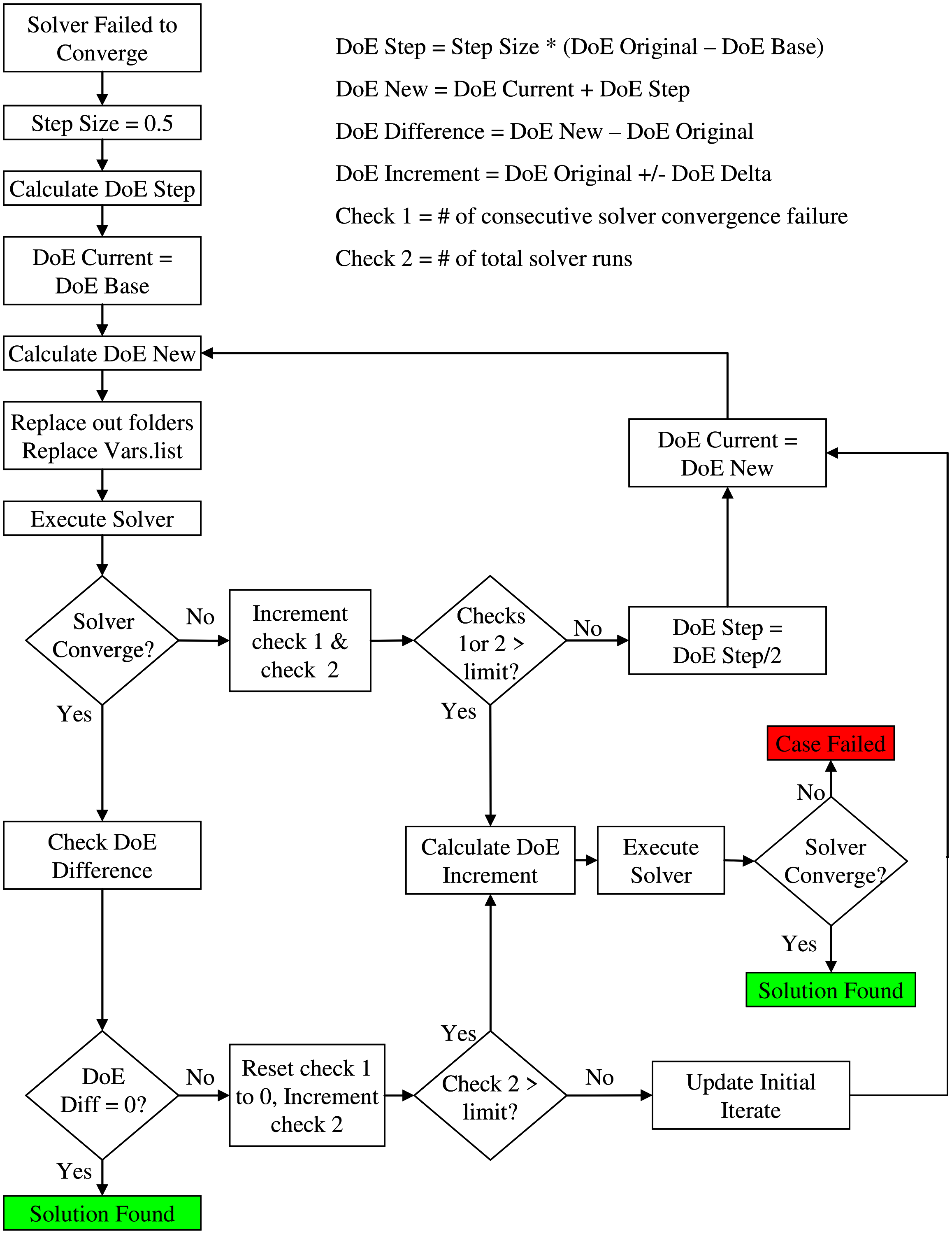


Figure 44. Flowchart for MDP Repair Algorithm for Cases that Fail to Converge

CHAPTER 5

RESULTS

This chapter examines the results from a series of experiments intended to provide insight to the following subjects in order to test the hypotheses described in Chapter 2 with the overall goal of producing a method capable of solving a cycle design problem with an extensive number of requirements and design points:

- Demonstrate the differences in CDSs created using SPD methods and MDP methods, particularly with regards to the feasibility of the CDS and the relocation of the optimum engine within the CDS.
- Illustrate the transformation of the design points, performance requirements technology rules and design variables into a solvable system of nonlinear equations.
- Investigate procedures and techniques for improving the robustness and efficiency of solver to allow for the inclusion of more design points.
- Determine the effects of altering technology level on the ability of the method to find feasible solutions for the CDS.

5.1 Experiment 1: Comparison of MDP and SPD On Design Cycle Analysis Method CDS

The objective of the first experiment is to demonstrate how the CDS is altered when switching from the traditional SPD cycle analysis method to an MDP cycle analysis method. The objective is accomplished by comparison of the CDS formed by both methods for the same engine cycle design problem. The experiment is designed to test the first hypothesis: The topography and feasible boundaries of the CDS will be altered between the MDP and SPD methods because many of the assumptions required by the

SPD method are unknown functions of the design variables and therefore cannot be correctly estimated prior to analysis.

5.1.1 Experiment 1 Setup

The design problem chosen for this experiment is the LCT engine with two thrust performance requirements, one at a TOC operating condition and the other at the TKO operating condition. The TOC operating condition is a key sizing condition of the engine that sets maximum corrected airflow through the fan and maximum corrected engine speed [91]. For a high bypass ratio separate flow turbofan intended for subsonic flight, the maximum T4 for the engine is achieved at the TKO operating condition [91]. The SPD cycle analysis method designates TOC as the design point while the MDP method includes both TOC and TKO design points. The metric of interest utilized for comparison of the two methods CDSs is uninstalled cruise TSFC. The cruise operating condition is analyzed in off-design for both methods. Other metrics that determine the feasibility of a design for the experiment are the TKO thrust, which is a constant from the MDP cycle analysis method but varies with the SPD cycle analysis method depending on the SPD assumptions and design variables, and the cooling flow requirements, specifically for the HPT, at TKO. The MDP method determines the exact amount of cooling flow required at TKO and incorporates that information during the analysis while the SPD method must assume a cooling flow requirement before the analysis and determine the feasibility of the design based upon the cooling flow assumption after the analysis. The specifications for the three operating conditions for the first experiment, TOC, TKO and cruise, are listed in Table 23.

Table 23. Operating Conditions for Experiment 1

Operating Condition	Mach	Altitude	Delta Temp.	Other Specifications
TOC	0.85	39,000 ft	0	Fn = 15,800 lbf
TKO	0.25	0 ft	+27 °F	T4 = 3300.0 °R
Cruise	0.84	35,000 ft	0	Fn = 12,000 lbf

The CDSs for both methods are defined by variations in the design variables FPR, OPR and EXTR, all established at the TOC operating condition. The established range of FPR varies from values of 1.4 to 1.76 while the range on OPR varies from values of 35 to 65. Both sets of ranges were chosen to ensure the minimum cruise TSFC would exist within the CDS. Other constraints, such as a maximum T3, which would normally limit the maximum OPR are not included in this experiment but will be included in subsequent experiments. For each combination of FPR and OPR, the extraction ratio is chosen which minimizes uninstalled cruise TSFC. The maximum T4 at TKO is fixed for both methods at 3300°R. The experiment is divided into three sub experiments. The first sub experiment compares the MDP and SPD CDSs for a common set of assumptions and requirements. The second sub experiment examines how the two methods change the CDS in response to a change in performance requirements. The third sub experiment examines the effects on the SPD CDS by a change in the design point assumptions. More than one CDS is required from each method in order to make the comparisons for the second and third sub experiment. The next two sections provide the information for the generation of the different MDP and SPD CDSs.

5.1.1.1. Single Point Design Setup for CDSs SPD 1 and SPD 2

For the SPD method, the TOC operating condition was designated as the design point. Additional assumptions, listed in Table 24, were required about the combustor exit temperature and the turbine cooling flow requirements at TOC. Two CDSs, designated

SPD 1 and SPD 2 are created using different TOC T4 assumptions of 2900°R and 2875°R respectively. In addition, the cooling flow assumptions for both the chargeable and nonchargeable cooling flows to both the HPT and LPT are modified for the two CDSs. The remaining operating conditions, TKO and cruise, are evaluated in off-design analysis mode at the conditions specified in Table 23.

Table 24. Single Point Design Analysis Assumptions for SPD Method

Designation	TKO T4 Requirement	T4 at TOC	HPT Cool (NC)	HPT Cool (Charge)	LPT Cool (NC)	LPT Cool (Charge)
SPD 1	3300 °R	2900 °R	0.115	0.075	0.035	0.020
SPD 2	3300 °R	2875 °R	0.125	0.090	0.030	0.010

5.1.1.2. Multi-Design Point Setup for CDSs MDP 1 and MDP 2

The MDP method includes two design points for the first experiment: TOC and TKO. A total of three performance requirements are defined for the MDP method at the two operating conditions; TOC thrust, TKO thrust and TKO T4. The values for the different performance requirements are listed in Table 25.In order to satisfy both thrust requirements and the T4 requirement, the modified Newton-Raphson solver varies the following three independent parameters; FAR at TKO, FAR at TOC and the mass flow at TOC. Two CDSs, designated MDP 1 and MDP 2, are created from the MDP method based on different TKO thrust requirement of 75,000 lbf and 77,000 lbf, respectively.

Table 25. Performance Requirements for MDP Method

Designation	TOC Thrust Requirement	TKO Thrust Requirement	TKO T4 Requirement
MDP 1	15,800 lbf	75,000 lbf	3300 °R
MDP 2	15,800 lbf	77,000 lbf	3300 °R

The DPMM for the MDP section of the first experiment, designating to which design point the design variables, performance requirements, component performance estimates and technology limits are specified is shown in Table 26.

Table 26. Design Point Mapping Matrix for Experiment 1

Design Points		**Top of Climb (TOC)** Mn 0.85 | 39 kft | dT 0	**Takeoff (TKO)** Mn 0.25 | 0 ft | dT +27
Comp. Map Scaling Point		TOC	
Design Variables	FPR	1.40 to 1.76	
	LPCPR	see OPR	
	HPCPR	16	
	OPR	35 to 65	
	T4		3300 °R
	Fan Nc Map	107%	
	LPC Nc Map	100%	
	HPC Nc Map	100%	
	Fan R line	2.05	
	LPC R line	2.05	
	HPC R line	2.05	
	Extraction Ratio	Optimized for TSFC	
Performance Requirements	TOC Thrust	15,800 lbf	
	TKO Thrust		75,000 lbf
Component Performance Estimation	Fan Adiabatic η	0.91	
	LPC Adiabatic η	0.895	
	HPC Adiabatic η	0.89	
	Fan Spec. Flow	45.0 lbm/ft^2-s	
	LPC Spec. Flow	26.5 lbm/ft^2-s	
	HPC Spec. Flow	31.5 lbm/ft^2-s	
	HPT Adiabatic η	0.90	
	LPT Adiabatic η	0.92	
	Burner η	0.995	
	HPT NC Cooling Effect.		1.5
	HPT Ch Cooling Effect.		0.8
	LPT NC Cooling Effect.		1.5
	HPT NC Cooling Effect.		1.2
	Bypass Nozzle Cv	Function of PR	
	Bypass Nozzle CdTh	Function of PR	
	Core Nozzle Cv	Function of PR	
	Core Nozzle CdTh	Function of PR	
Technology Limits	T4 Max		3300 °R
	HPT 1st Vane Temp.		2400 °R
	HPT Blade Temp.		2300 °R
	HPT Vane Temp.		2300 °R
	LPT Blade Temp.		2000 °R
	LPT Vane Temp.		2000 °R

All of the specifications established at TOC were set to the same values for both the MDP and SPD on-design cycle analysis methods. The required turbine cooling flows are determined at the TKO operating condition for the MDP method.

A successful test will demonstrate that the cycle designs space does transform such that the optimum – lowest cruise TSFC - combination of FPR and OPR will shift between the two cycle on-design methods. In addition, the test will demonstrate the feasibility of the CDS created by the MDP method while illustrating how the SPD method can create infeasible, feasible, and performance penalized candidate cycle engines.

5.1.2 Experiment 1 Results

The results of the first experiment are divided into three sections. The first section examines the differences between CDSs created by both methods with regards to topography and feasible boundaries to a single set of performance requirements and SPD assumptions. The second section evaluates how the two methods respond differently to a change in the performance requirements at conditions other than the SPD method design point. The third section assesses the affect that altering the assumptions for the SPD method has towards the CDS relative to the MDP method CDS. Evaluation of both methods in this manner will demonstrate the advantages of MDP over SPD on design cycle analysis methods. The first experiment focuses on the CDSs, demonstration of the details of the MDP method will be discussed in experiment 2.

5.1.2.1. Comparison of Cycle Design Spaces from MDP and SPD Methods

The evaluation of the two analysis methods begins with a surface and contour plot of uninstalled cruise TSFC for the MDP method, shown in Figure 45 as a function of FRP and OPR, as well as the same plots shown for the SPD method in Figure 46. These plots show the curvature of the surface and the location and value of the minimum cruise

TSFC for each CDS. For the MDP method, the minimum cruise TSFC occurs at a FPR of 1.47 and an OPR of 50.27. The value of cruise TSFC for this design is 0.5462 lbm/lbf-hr. This value differs from the minimum cruise TSFC determined from the SPD method of 0.5476 lbm/lbf-hr, which occurs at a FPR of 1.49 and an OPR of 59.47. The results for both methods are summarized in Table 27.

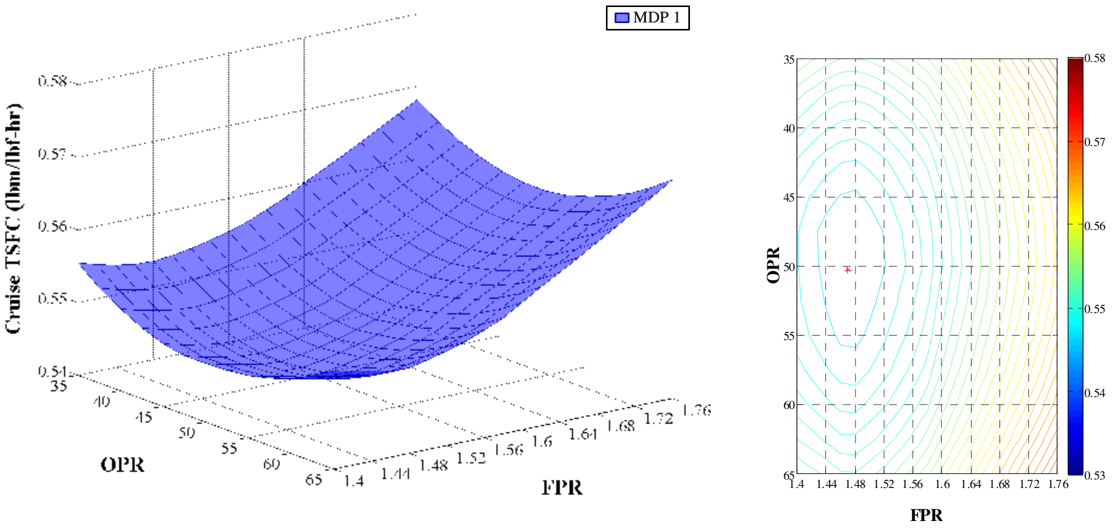


Figure 45. Cruise TSFC CDS Surface and Contour Plots for MDP 1

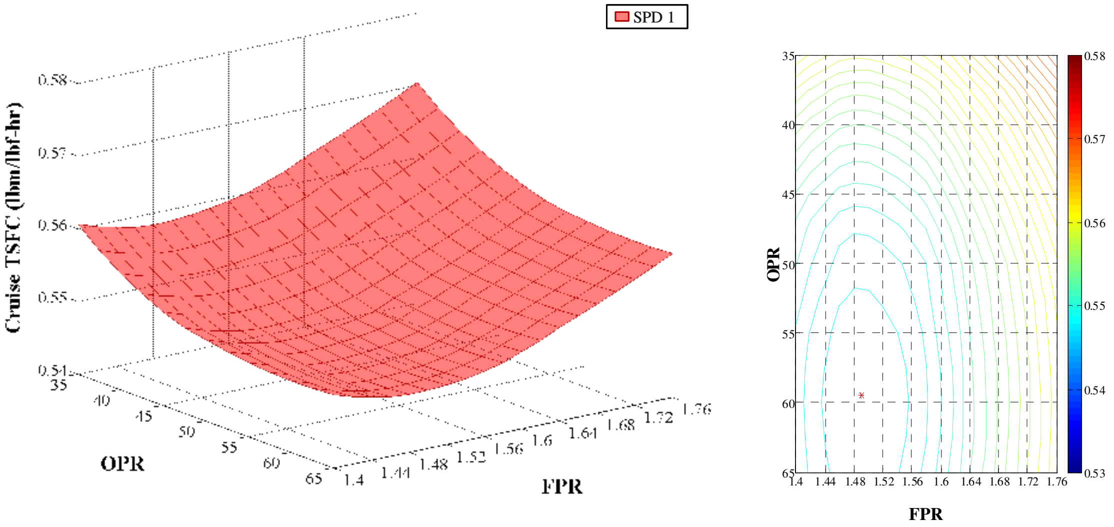


Figure 46. Cruise TSFC CDS Surface and Contour Plots for SPD 1

Table 27. Minimum Cruise TSFC Cycle Parameters for MDP1 and SPD 1

CDS	FPR	OPR	Cruise TSFC (lbm/lbf-hr)	TKO Thrust (lbf)
MDP 1	1.470	50.27	0.5462	75,000
SPD 1	1.490	59.47	0.5476	76,596

A better comparison of the two CDSs can be viewed by overlaying the two surface plots as shown in Figure 47. The two surfaces intersect each other with the SPD surface having lower cruise TSFC at higher values of OPR. The line of intersection increases with OPR as FPR is decreased. The location of the minimum TSFC values is also plotted in Figure 47. The two surfaces show that the topography of the CDS is different for each of the cycle analysis methods. This leads to the question, which of the two surfaces is "correct" and should be used to determine the cycle design for this problem?

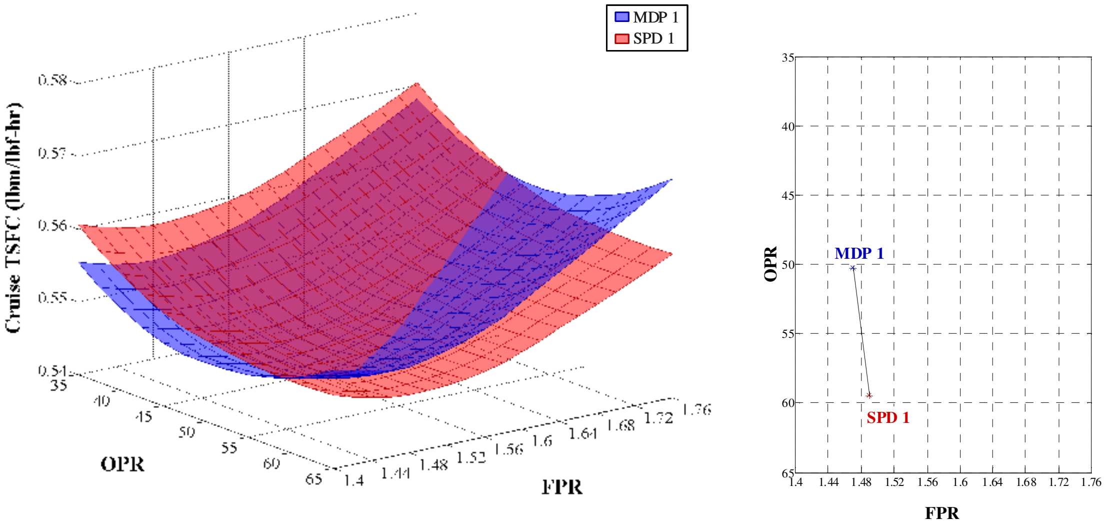


Figure 47. Cruise TSFC Comparison of MDP 1 and SPD 1

First, since both surfaces were created using the same engine model, the level of accuracy of both surfaces is the same. It is not the case that one surface is more accurate than the other. The difference between the two surfaces is not their accuracy, but their feasibility. To examine the feasibility of the CDS, the surface plot of TKO for both methods is shown in Figure 48. By design, the MDP method always generates an engine that produces the required 75,000 lbf of thrust at TKO. However, the TKO thrust produced by the SPD method varies from 72,500 to 78,000 lbf of thrust. Only those designs which produce thrust greater than 75,000, as indicated by the red SPD 1 surface that lies above the blue MDP 1 surface in Figure 48, are feasible. The intersection of the two surfaces forms a constraint line which becomes a feasible boundary for the SPD 1 CDS.

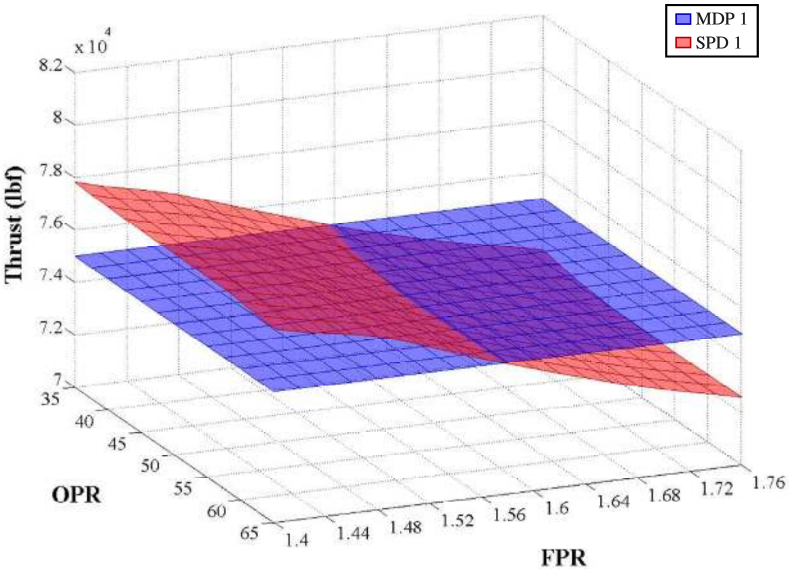


Figure 48. Comparison of MDP 1 and SPD 1 TKO Thrust

A similar argument can also be made for the required HPT cooling. The total cooling flow, both chargeable and nonchargeable cooling, for the HPT is shown for MDP and SPD methods in Figure 49. For the MDP method, the required cooling was determined at the TKO condition. While the combustor exit temperature was a constant 3300°R for all designs, the amount of cooling flow required increases with OPR. More flow is required to cool the turbine vanes and blades to the same material temperature because of the higher temperature of the incoming cooling flow from the compressor. The SPD method assumed the same value of turbine cooling for all designs. Therefore, for the higher OPR designs, insufficient cooling flow was extracted to cool the turbines making the designs infeasible. This assumption accounts for the SPD method's minimum TSFC occurring at a higher OPR than the MDP method, the negative impacts of increased cooling flow were not included to offset the positive impact of higher OPR on TSFC. As with the TKO thrust, the intersection of the two spaces forms a second feasible boundary for the SPD CDS. Designs with a higher OPR are infeasible for the SPD method.

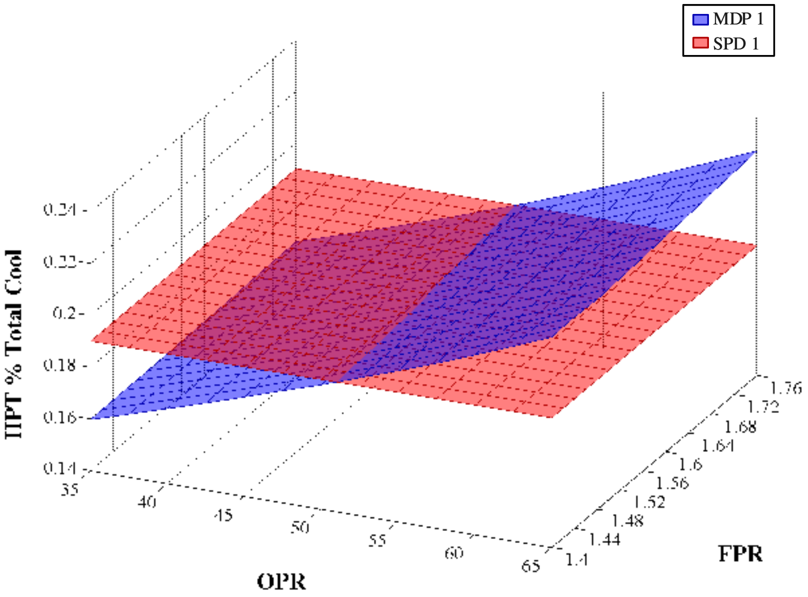


Figure 49. Comparison of MDP 1 and SPD 1 HPT Cooling Flow

A comparison of the feasible SPD 1 CDS with the MDP 1 CDS is shown in Figure 50. The feasible region of the SPD 1 CDS has been reduced to nearly a quarter of the size of the original surface. Furthermore, the minimum TSFC found from the original SDP CDS is not part of the feasible space. The result is a new feasible minimum TSFC for SPD 1 of 0.5486 lbm/lbf-hr, which occurs at a FPR of 1.488 and an OPR of 51.5 as summarized in Examination of the cruise TSFC surface plots in Figure 50 shows that the feasible SPD 1 cruise TSFC is always greater than the MDP cruise TSFC. This is a result of the engine being oversized to produce too much thrust and use too much cooling flow in the SPD method. Only at point A, where the feasible boundaries for thrust and HPT cooling flow converge, do the two surfaces meet. Therefore it is more appropriate to select a candidate engine cycle from the MDP CDS as all are properly sized to the requirements and use the correct amount of turbine cooling.

Table 28.

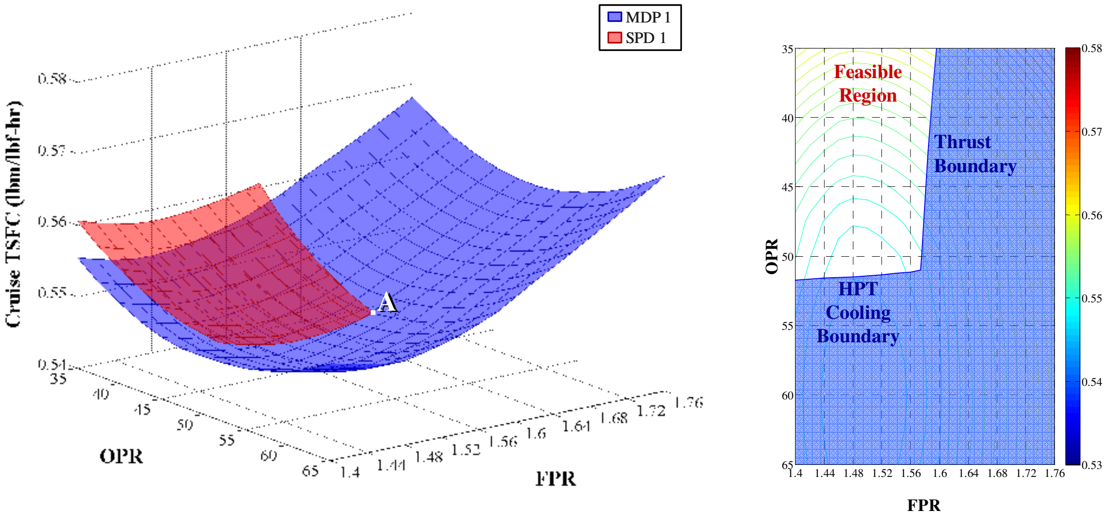


Figure 50. SPD 1 Feasible Cruise TSFC CDS for 75,000 lbf TKO Thrust

Examination of the cruise TSFC surface plots in Figure 50 shows that the feasible SPD 1 cruise TSFC is always greater than the MDP cruise TSFC. This is a result of the engine being oversized to produce too much thrust and use too much cooling flow in the SPD method. Only at point A, where the feasible boundaries for thrust and HPT cooling flow converge, do the two surfaces meet. Therefore it is more appropriate to select a candidate engine cycle from the MDP CDS as all are properly sized to the requirements and use the correct amount of turbine cooling.

Table 28. Minimum Cruise TSFC Comparison between MDP 1 and Feasible SPD 1

CDS	FPR	OPR	Cruise TSFC (lbm/lbf-hr)	TKO Thrust (lbf)
MDP 1	1.470	50.27	0.5462	75,000
SPD 1	1.490	59.47	0.5476	76,596
Feasible SPD1	1.488	51.50	0.5486	76,553

5.1.2.2. Response of MDP and SPD Methods to Changes in Performance Requirements

Section two of the first experiment evaluates how each method responds to a change in the performance requirements that occur at conditions other than the SPD design point. For this part of the experiment, the TKO thrust was increased from 75,000 lbf to 77,000 lbf. The CDS created from the MDP method with the adjusted TKO thrust, shown in Figure 51, is designated as design space MDP 2.

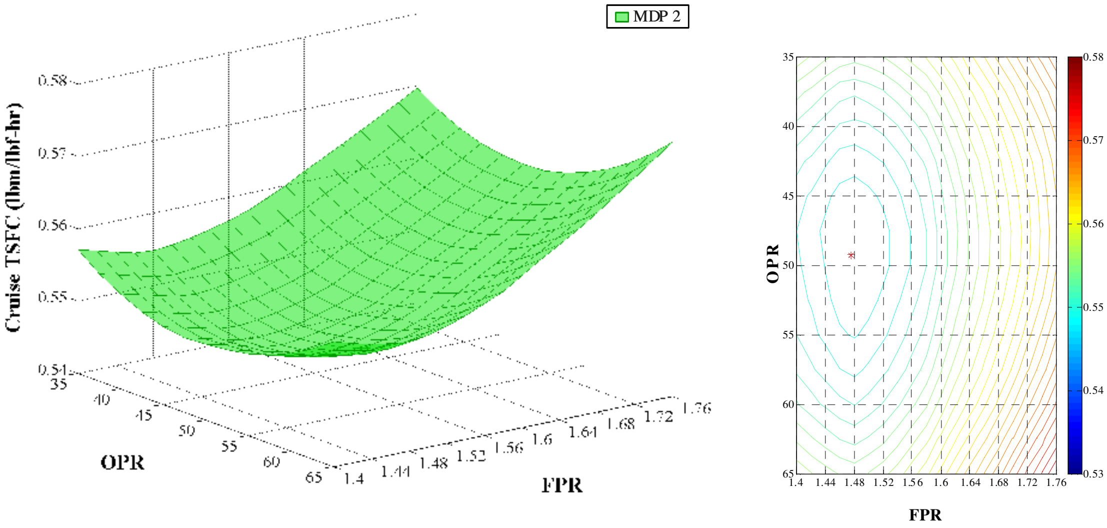


Figure 51. Cruise TSFC CDS Surface and Contour Plots for MDP 2

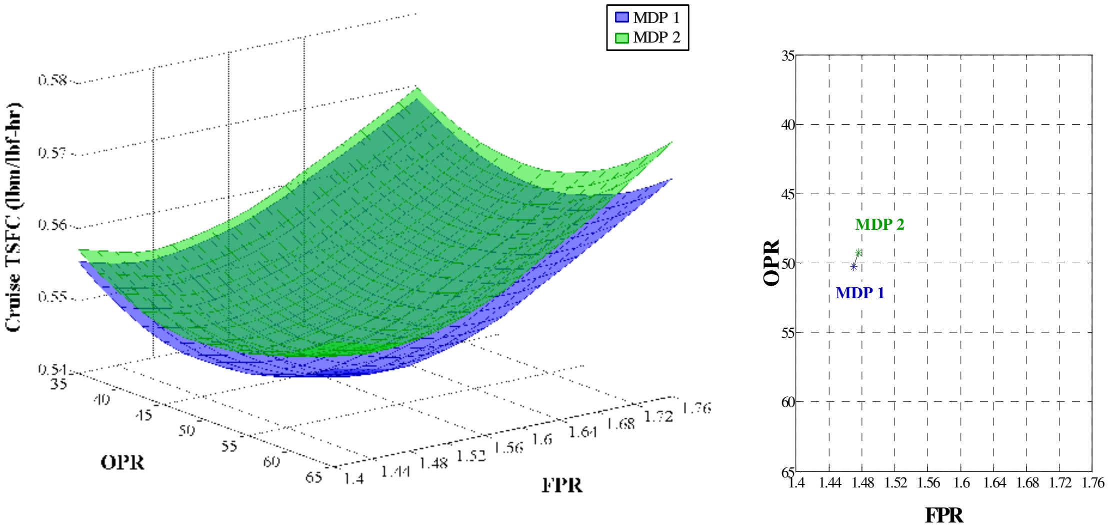


Figure 52. Cruise TSFC Comparison of MDP 1 and MDP 2

Comparison of the MDP 1 and MDP 2 CDSs, plotted together in Figure 52, shows that because of the larger engine needed to produce the extra 2000 lbf of thrust at TKO, the MDP 2 cruise TSFC surface has shifted above the MDP 1 cruise TSFC surface. Furthermore, the minimum TSFC for MDP 2 occurs at a different FPR and OPR. The minimum TSFC values are listed in Table 29. Therefore, the MDP method shifts the topography of the CDS to accommodate a change to the performance requirements.

Table 29. Minimum Cruise TSFC Cycle Parameters for MDP1 and MDP 2

CDS	FPR	OPR	Cruise TSFC (lbm/lbf-hr)	TKO Thrust (lbf)
MDP 1	1.470	50.27	0.5462	75,000
MDP 2	1.476	49.25	0.5482	77,000

For the SPD method, the on design cycle analysis is unaltered, as the change to the TKO thrust has no impact on the SPD analysis at TOC. Therefore the exact same engine designs are generated regardless of the value of the TKO thrust requirement. A comparison of the MDP 2 and SPD 1 cruise TSFC surface plots is shown in Figure 53. The location of the minimum cruise TSFC is plotted as well.

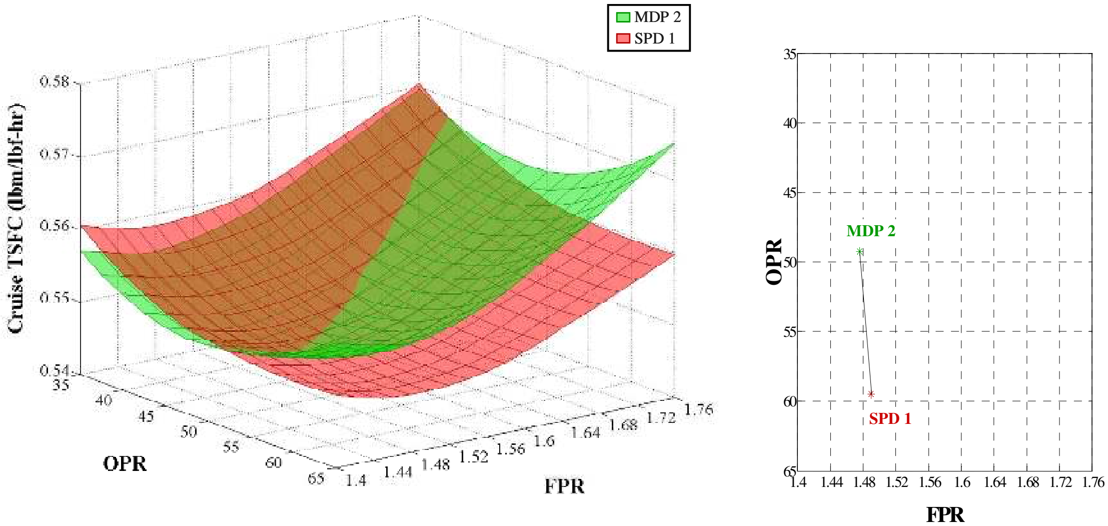


Figure 53. Cruise TSFC Comparison of MDP 2 and SPD 1

As with the first section of Experiment 1, an evaluation of the feasible CDS for SPD 1 must be made to properly compare the two CDSs. The increase in the TKO thrust requirement also resulted in an increase in the required HPT cooling flow. A comparison of the TKO thrust and HPT cooling flow for MDP 2 and SPD1 are shown in Figure 54 and Figure 55.

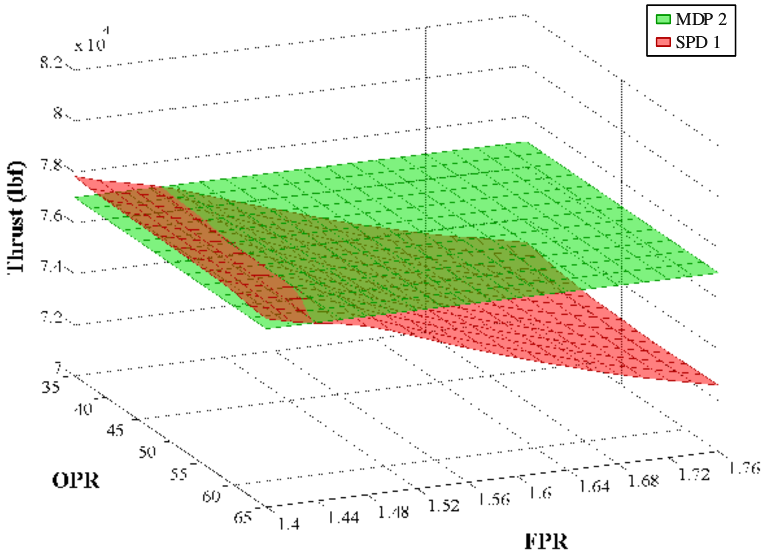


Figure 54. Comparison of MDP 2 and SPD 1 TKO Thrust

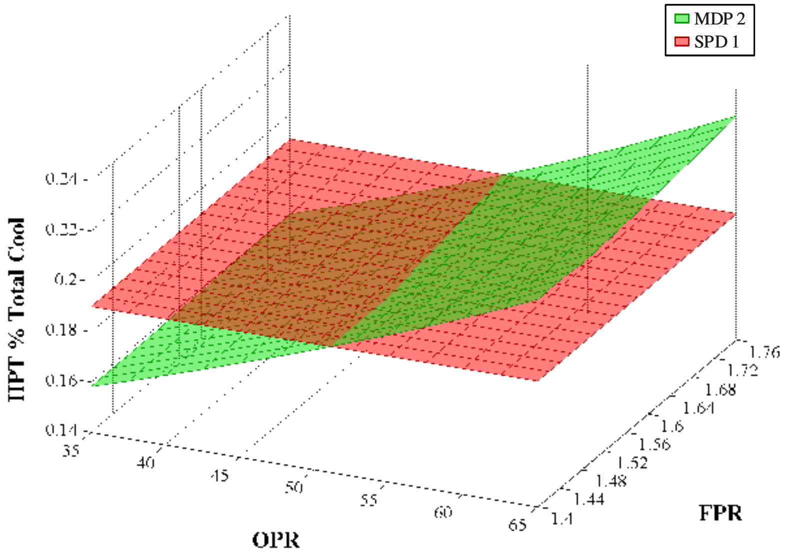


Figure 55. Comparison of MDP 2 and SPD 1 HPT Cooling Flow

The increase in the TKO thrust requirement has significantly reduced the feasible region of the SPD 1 CDS, with only the lowest values of FPR able to generate the required TKO thrust. The increased cooling flow requirement has also reduce the feasible region, but by only a small margin. The resulting feasible region of the CDS is plotted in

Figure 56. For the SPD 1 feasible region, the minimum cruise TSFC has increased to 0.5489 lbm/lbf-hr and has shifted to a FPR of 1.46 and an OPR of 51.06 as summarized in As before, the SPD 1 feasible surface intersects with the MDP 2 surface only at point A, which, for this case coincides with the minimum cruise TSFC for the feasible SPD 1 CDS. All other points within the feasible SPD 1 CDS exceed the MDP 2 values for cruise TSFC as the designs are oversized to provide too much thrust and cooling flow.

Table 30.

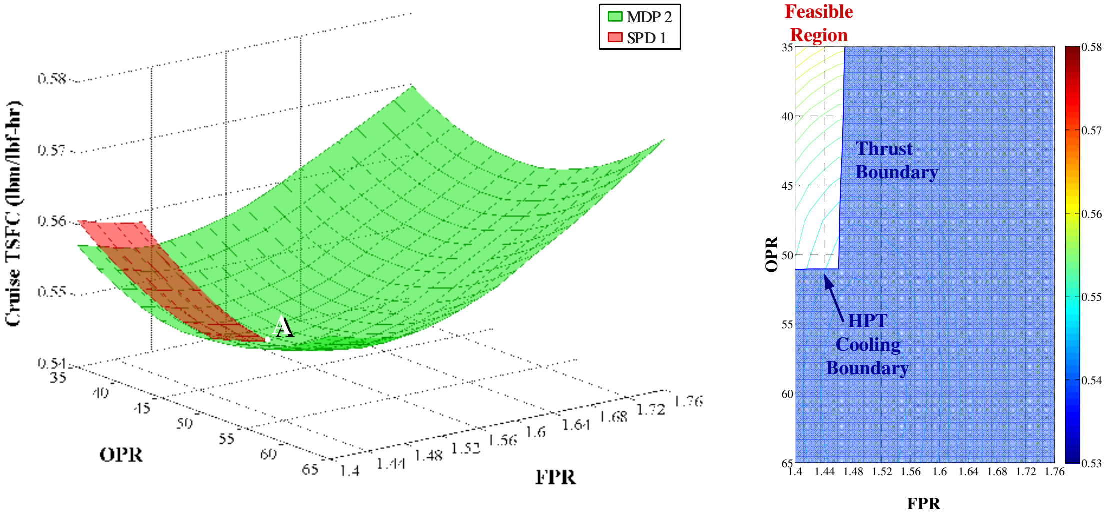


Figure 56. SPD 1 Feasible Cruise TSFC CDS for 77,000 lbf TKO Thrust

As before, the SPD 1 feasible surface intersects with the MDP 2 surface only at point A, which, for this case coincides with the minimum cruise TSFC for the feasible SPD 1 CDS. All other points within the feasible SPD 1 CDS exceed the MDP 2 values for cruise TSFC as the designs are oversized to provide too much thrust and cooling flow.

Table 30. Minimum TSFC Comparison between MDP 2 and Feasible SPD 1

CDS	FPR	OPR	Cruise TSFC (lbm/lbf-hr)	TKO Thrust (lbf)
MDP 2	1.476	49.25	0.5462	77,000
SPD 1	1.490	59.47	0.5476	76,596
Feasible SPD1	1.460	51.06	0.5489	77,000

It should be again noted that the underlying space for the SPD method has not changed with the change to the TKO thrust requirement. Rather, because the TKO thrust requirement has become more stringent, the feasible boundaries have shifted decreasing the available design space. This highlights the difference between the SPD and MDP methods. MDP methods change the topography of the CDS to meet a change in performance requirements and keep the candidate engine cycles feasible while the SPD method can only change the feasible boundaries of the CDS.

5.1.2.3. Impact of Assumptions for SPD Methods

The feasible boundaries of the SPD CDS are affected by the assumptions required to define the cycle by the SPD method at the design point, but with potentially undesirable consequences. To demonstrate this point, the assumptions for the SPD method were changed to a TOC T4 of 2875°R and increased cooling flow assumptions as summarized previously in Table 27. The resulting cruise TSFC CDS, designated SPD 2 is shown in Figure 57.

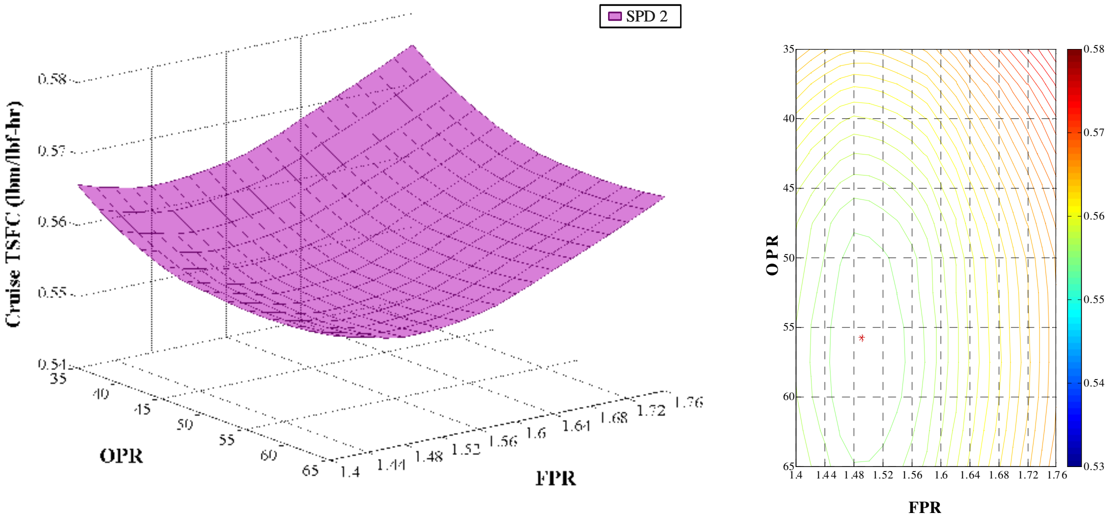


Figure 57. Cruise TSFC CDS Surface and Contour Plots for SPD 2

A comparison of the two CDSs, SPD 1 and SPD 2 in Figure 58, illustrates the impact of the TOC T4 assumption change to 2875°R from 2900 °R. The entire space has shifted to higher cruise TSFC values. Furthermore, the optimum TSFC value has shifted to a lower OPR.

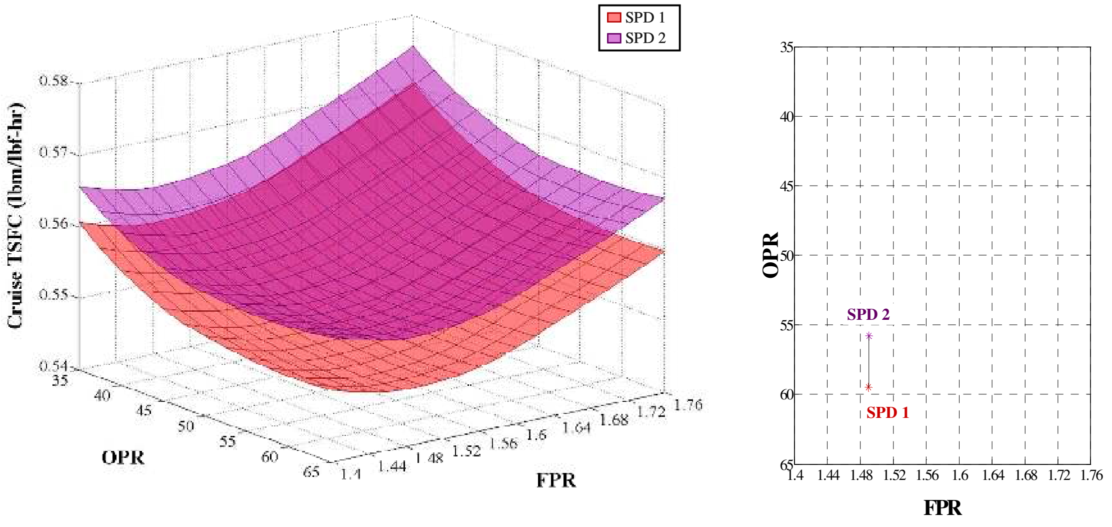


Figure 58. Cruise TSFC Comparison of SPD 1 and SPD 2

As with the first section of Experiment 1, an evaluation of the feasible CDS for SPD 2 must be made to properly compare the SPD 1 and SPD 2 CDSs. Compared to the

TKO thrust requirement of 75,000, most of the SPD 2 space is feasible as shown in Figure 59, with only the highest FPR value candidate engine cycles not meeting the thrust requirement.

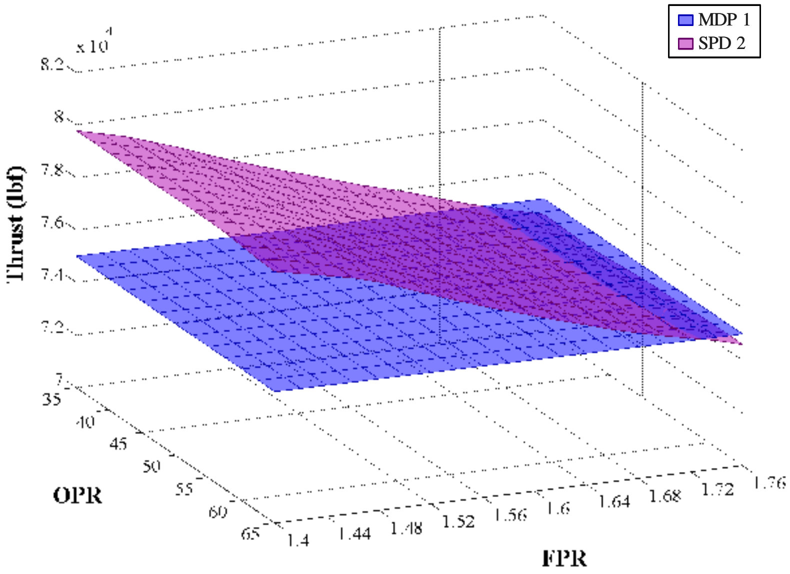


Figure 59. Comparison of MDP 1 and SPD 2 TKO Thrust

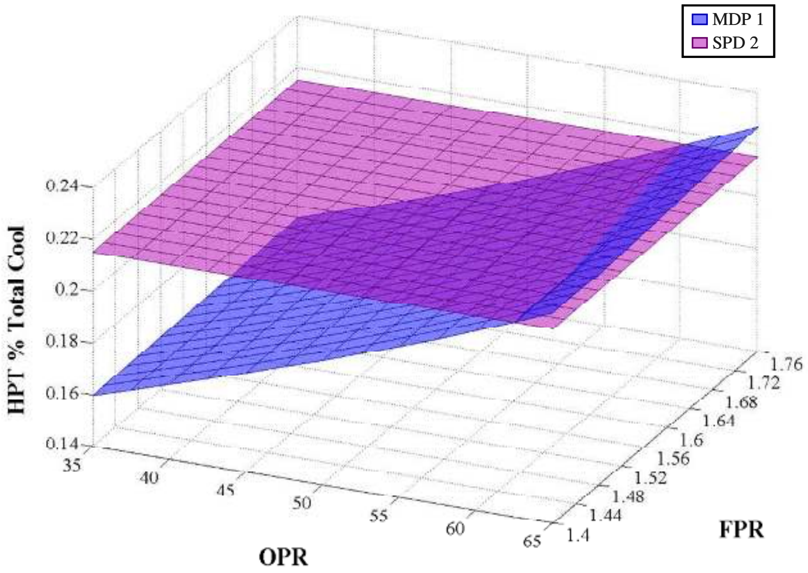


Figure 60. Comparison of MDP 1 and SPD 2 HPT Cooling Flow

As a result of the increase in the HPT cooling flows assumptions for SPD 2 shown in Figure 60, the CDS meets or exceed the required cooling flows established by MDP 1 at all design points except the highest OPRs. In combination with the decrease in the TOC T4 assumption, the feasible region of SPD 2 is more extensive than the feasible region for SPD1.

However, this increase in the feasible region of the SPD CDS is not without cost. The feasible region of the SPD 2 CDS and the MDP 1 CDS are compared in Figure 61 for cruise TSFC. While the feasible region is larger, the SPD 2 space has shifted further from the MDP 1 space. The penalty for increasing the feasible region is that the engines produced are even more oversized and do not perform as efficiently, in this case resulting in a performance penalty of increased cruise TSFC.

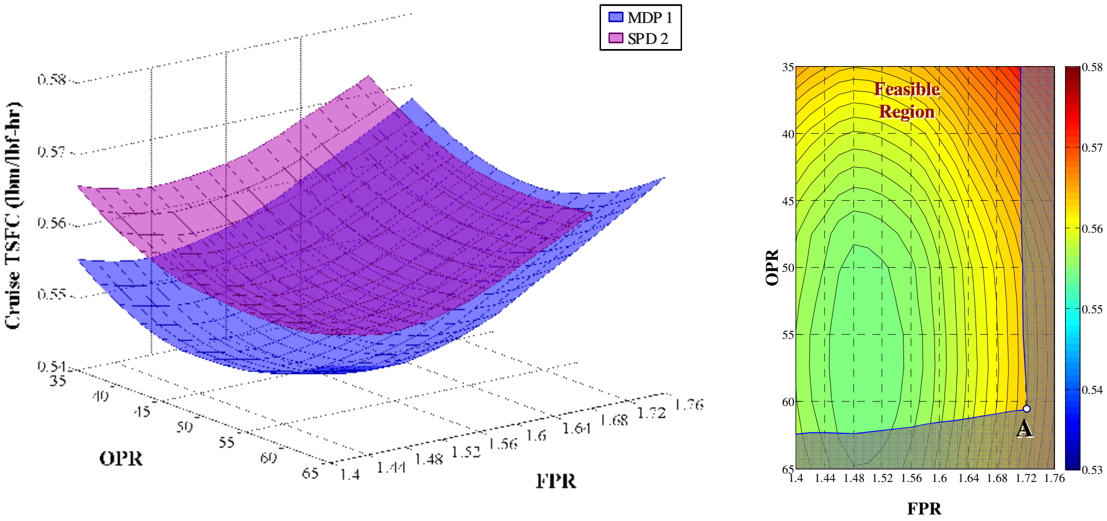


Figure 61. SPD 2 Feasible Cruise TSFC CDS for 75,000 lbf TKO Thrust

The performance data from Experiment 1 is summarized in Table 31. The MDP 1 CDS exactly meets the 75,000 lbf TKO thrust requirement and has the lowest cruise TSFC of 0.5462 lbm/lbf-hr at a FPR of 1.47 and OPR of 50.27. For the optimum cruise condition from the feasible region of SPD 1, a higher thrust of 76,553 lbf is generated at TKO. The optimum cruise TSFC increased to 0.5486 at FPR of 1.488 and OPR of 51.5.

Finally, by changing the assumptions for SPD 2, the optimum design for cruise TSFC produced 78,454 lbf of thrust at TKO and had an even higher cruise TSFC of 0.5542 at a FPR of 1.49 and OPR of 55.78.

Table 31. Minimum TSFC Comparison between MDP 1 and Feasible SPD 2

CDS	FPR	OPR	Cruise TSFC (lbm/lbf-hr)	TKO Thrust (lbf)
MDP 1	1.470	50.27	0.5462	75,000
Feasible SPD 1	1.488	51.50	0.5486	76,553
Feasible SPD 2	1.490	55.78	0.5542	78,454

The results of this experiment show the deficiencies of single point design to determine an optimum gas turbine engine for a given set of requirements. Due to the assumptions required by the SPD method, the engine is invariably oversized and suffers a performance penalty. Simply obtaining the optimum candidate engine cycle from an SPD CDS and then adjusting the assumption to meet the requirements will not work, because the optimum design shifts with the requirements. The MDP method always obtains the properly sized engine for a set of given requirements increasing the feasible region of the design space from which the optimum performance engine can be obtained.

5.2 Experiment 2: Design Rules Transformation

The second experiment focuses on the transformation of the design rules and engine matching relations into a system of nonlinear equations that can be solved employing a modified Newton-Raphson solver. Along with the fifth experiment, the second experiment is designed to test the second hypothesis that an approach will simultaneously meet multiple design points by formulating a system of nonlinear equations at on-design mode from design rules that couple the design points, performance requirements, technology rules and design variables.

The intent behind the second experiment is not to identify a “golden set” of design rules that can be utilized for any gas turbine application. Design rules will vary for different applications and architectures. Even within the same design problem, the cycle analyst’s design philosophy will influence the design rules so that multiple sets of valid design rules exist. The objective of this experiment is to test simultaneous MDP method’s flexibility to incorporate different types of design rules, demonstrate its ability to document the design rules and their interconnectedness and determine the impact of adding constraints to the modified Newton-Raphson solver’s ability to obtain solutions for a design problem incorporating multiple design points.

5.2.1 Experiment 2 Setup

The setup for experiment 2 is summarized by the DPMM in Table 32. Seven design points were included for the experiment; LPCDP, HPCDP, TOC1, TOC2, TKO, HHDTKO and cutback. The requirements and constraints enabled in the model for the experiment include thrust for TOC1, TOC2, TKO and HHDTKO, cutback maximum Vmix, maximum fan diameter, and maximum T3 and T4. The requirements and design points resulted in a system of 95 nonlinear equation for the solver. A DoE was formulated for 6 design variables and technology parameters; FPR, OPR, EXTR, T4_Max, T3_Max and TOC_NcMap. The ranges of the different variables within the DoE were: FPR from 1.5 to 1.75, OPR from 28 to 44, EXTR from 1 to 1.4 , T4_Max from 3275 to 3450 °R, T3_max from 1700 to 1850 °R and TOC_NcMap from 103% to 108% overspeed. The DoE consisted of 129 Central Composite Design (CCD) cases and 1000 Latin HypeCube (LHC) cases. The experiment is performed twice, once with constraints enabled and the other with the constraints disabled. The constraints included in the experiment are; maximum fan diameter of 128 inches, a minimum LPCPR of 1.15, and maximum cutback Vmix of 1050 ft/s. The MDP repair algorithm is not enabled for this experiment.

Table 32. Design Point Mapping Matrix for Experiment 2

Design Points		Low Pressure Aero Design Point (LPADP) Mn 0.80 | 35 kft | dT 0	High Pressure Aero Design Point (HPADP) Mn 0 | 0 kft | dT 0	Top of Climb 1 (TOC1) Mn 0.85 | 39 kft | dT 0	Top of Climb 2 (TOC2) Mn 0.85 | 35 kft | dT 0	Takeoff (TKO) Mn 0.25 | 0 ft | dT +27	High Hot-Day Takeoff (HHDTKO) Mn 0.25 | 5 kft | dT +27	Cutback Mn 0.25 | 2 kft | dT +18
Comp. Map Scaling Point		LPADP						
Design Variables	FPR	1.5 to 1.75						
	LPCPR	Function of OPR						
	OPR	28 to 44						
	Fan TOC Nc			1.03 to 1.08	1.03 to 1.08			
	Extraction Ratio	1.0 to 1.4						
Performance Requirements	TOC1 Thrust			16,025 lbf				
	TOC2 Thrust				19,600 lbf			
	TKO Thrust					75,000 lbf		
	HHDTKO Thrust						67,000 lbf	
	Cutback Thrust							60,000 lbf
	Fan Nc Map	100%						
	HPC Nc Map		Function of OPR					
Component Performance Estimation	Fan Adiabatic η	Function of FPR						
	LPC Adiabatic η	Function of LPCPR						
	HPC Adiabatic η		Fixed Map					
	Fan Spec. Flow	45 lbm/ft^2-s						
	LPC Spec. Flow	26.5 lbm/ft^2-s						
	HPC Spec. Flow		Fixed Area					
	HPT Adiabatic η		Fixed Map					
	LPT Adiabatic η	0.92						
	Burner η	0.999						
	HPT NC Cooling Effect.					1.5	1.5	
	HPT Ch Cooling Effect.					0.8	0.8	
	LPT NC Cooling Effect.					1.5	1.5	
	HPT NC Cooling Effect.					1.2	1.2	
	Bypass Nozzle Cv			Function of PR				
	Bypass Nozzle CdTh			Function of PR				
	Core Nozzle Cv			Function of PR				
	Core Nozzle CdTh			Function of PR				
Technology Limits	T4 Max.					3275 to 3450°R	3275 to 3450°R	
	HPT 1st Vane Temp.					2400 °R	2400 °R	
	HPT Blade Temp.					2300 °R	2300 °R	
	HPT Vane Temp.					2300 °R	2300 °R	
	LPT Blade Temp.					2000 °R	2000 °R	
	LPT Vane Temp.					2000 °R	2000 °R	
	T3 Max.					1700 to 1850 °R	1700 to 1850 °R	
	Min. LPCPR	1.15						
	Max. Fan Diameter	128 inches						
	Max Vmix							1050 ft/s

A successful test will demonstrate that the simultaneous MDP method can integrate multiple design points into the engine cycle model and fine solutions to candidate engine cycles within the CDS in an efficient manner. Furthermore the experiment will show the simultaneous MDP method can efficiently incorporate constraints into the engine cycle model.

5.2.2 Experiment 2 Results

The results section for experiment 2 is divided into three sections. The first section examines the effect of constraints on the key solver performance metrics. The second section explore the characteristic of the Jacobian matrix to better understand the system of nonlinear equations and how each is incorporated into the simultaneous MDP method. The final section explores the impact the constraints have on the CDS.

5.2.2.1. Summary of Solver Statistics

There are four possible outcomes from an attempt by the solver to find a solution to a candidate cycle engine within the CDS as illustrated in Table 33. The four possible outcomes can be categorized as: 1) that the candidate engine resides within the feasible space and the solver successfully converges on the solution, 2) that the candidate cycle engine resides within the feasible space and the solver fails to converge on the solution, 3) that no feasible solution exists yet the solver converges to a solution and 4) that no feasible solution exists and the solver fails to converge on a solution. The objective of the simultaneous MDP method is to maximize category I while minimizing if not eliminating altogether category II. Categories III and IV require a little more discussion. How is it possible for the solve to converge to a solution when no feasible solution exists? It must be remembered that the solver is only satisfying the conditions set forth in the nonlinear equations by adjusting the values of the independent parameters. The solver does not consider a good independent value from a bad independent value. If a solution is found with a bad independent value then the solver considers that outcome a successful convergence. This outcome most often happens when the solver extrapolates off of compressor maps by choosing Rline values less than 1 or greater than 2.8. The solution mathematically satisfies the system of nonlinear equations but is not a practical, physically obtainable solution. It is possible within NPSS to set limits on independent parameters. The solver is not prevented from exceeding the limits. Instead an error flag is generated to notify the cycle analyst the design is outside the feasible space. Categories III and IV are dependent upon the ranges of the design variables and technology parameters. The objective of the simultaneous MDP method is not to limit these categories but merely to identify when a candidate engine resides in the infeasible space.

Table 33. Categories of Possible Outcomes from Solver

I Feasible Space Convergence Success	**II** Feasible Space Convergence Failure
III Infeasible Space Convergence Success	**IV** Infeasible Space Convergence Failure

For each of the remaining experiments key solver performance metrics will be tracked to determine the robustness and efficiency with which solutions to candidate engine cycles are found by the modified Newton-Raphson solver. The key metrics include the number and percentage of cases converged the mean and standard deviation for the number of iterations for converged cases, the mean and standard deviation for the number of full Jacobian matrices generated for converged cases, and the mean and standard deviation for the number of model passes required for converged cases. Additional metrics for the solver will be added for subsequent experiments. The key solver metrics for the second experiment are listed in Table 34. Each of these metrics are compiled for two different design spaces, both with constraints enabled and constraints disabled. The range of the second design space, designated as space 2, is 50% larger than the first design space, designated as space 1, for all of the design variables.

Table 34. Solver Summary Results for Experiment 2

Solver Metric	Constrained Space 1	Constrained Space 2	Unconstrained Space 1	Unconstrained Space 2
Total Converged	1102	987	1094	981
Convergence Success Rate	97.61%	87.42%	96.99%	86.89%
Iterations: Mean	26.22	30.83	27.98	32.78
Iterations: StDev	12.66	12.80	12.55	13.83
Full Jacobians: Mean	1.60	1.75	1.62	1.89
Full Jacobians: StDev	0.64	0.70	1.51	0.94
Model Passes: Mean	177.85	197.44	181.80	212.13
Model Passes: StDev	66.93	71.75	149.08	95.88

The constrained design space has a convergence success rate of 97.6% and 87.4% for space 1 and 2. This result is without the MDP repair algorithm enabled. The modified Newton-Raphson solver requires a mean of 26.22 iterations and 177.85 model passes to obtain the constrained design space solution for space 1 and 30.83 iterations and 197.4 model passes for the constrained space 2. The modified Newton-Raphson solver for the unconstrained design space, on first appearance, performed only slightly worse than for the constrained space. The convergence success rate dropped by -0.62 % to 96.99% for space 1 and by -0.53% to 86.89 for space 2. The mean number of iterations increased 1.76 to 27.98 and 1.95 to 32.78 for space 1 and 2, while the mean number of model passes increased 3.95 to 181.8 and 14.69 to 212.13.

The results from Table 34 could lead one to conclude that the constraints have little impact on the modified Newton-Raphson solver's performance. However, these results do not provide the entire picture. The top model pass counts for both the constrained and unconstrained design spaces are shown in Figure 62 and Figure 63. Most of the difference in the performance metrics can be attributed to the top model pass counts for each design space. A large divergence occurs for the unconstrained design space 1 model pass counts for the highest 15 model pass counts. This effect is increased with the larger design variables of design space 2. The divergence between the constrained and unconstrained model occurs sooner for design space 2 covering the highest 28 model passes. Constraints therefore have a positive effect on the performance of the modified Newton-Raphson solver. The solver can adjust the design variables to make infeasible designs feasible. Furthermore, the constraints dampen the computational expense of the most difficult candidate engine cycles found by the modified Newton-Raphson solver. The degree of positive impact of the constraints on the solver performance is dependent on the width of the design variable range that defines the cycle design space.

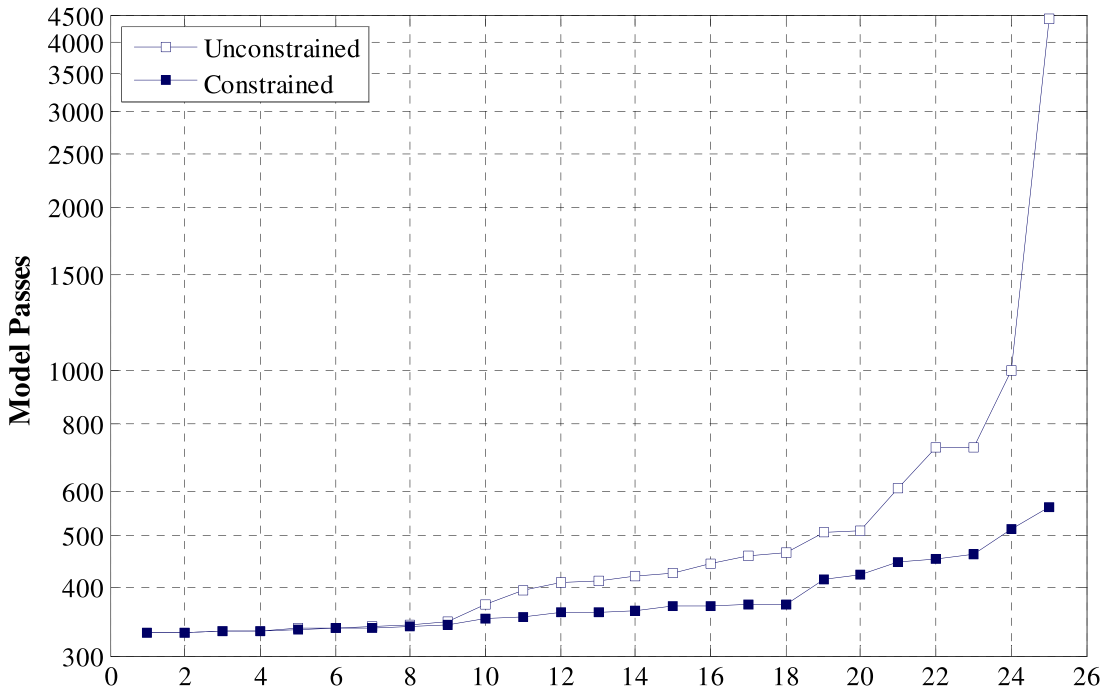


Figure 62. Highest Model Pass Count Comparison of Design Space 1

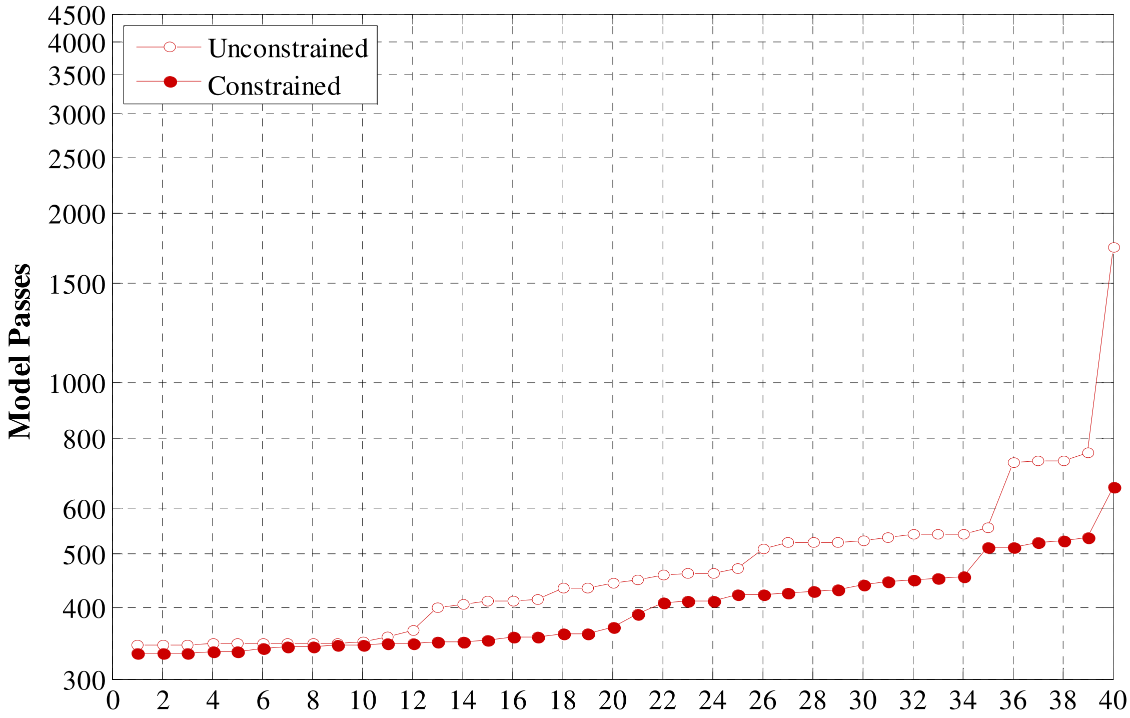


Figure 63. Highest Model Pass Count Comparison of Design Space 2

5.2.2.2. Examination of Jacobian Matrix

The system of nonlinear equations is represented within the solver by the Jacobian matrix. An understanding of the characteristics of the Jacobian will provide valuable insight into the simultaneous MDP method. The settings of the design variables for the sample case that is examined is listed in Table 35.

Table 35. Design Variable Settings for Jacobian Matrix Examination

Design Variable	Value	Units
FPR	1.64750	-
OPR	35.9619	-
EXTR	1.20476	-
T4_Max	3393.75	°R
TOC_NcMap	1.05321	Nc/100%Nc
T3_Max	1785.79	°R

As was discussed in Chapter 3, the Jacobian can be divided into several submatrices by design point for better organization as displayed in equation 20. The Jacobian for the sample case is displayed in Figure 64 segmented by design point. The degree of correlation between the independent parameters and dependent functions is represented within the chart. A strong correlation limit was established at +/- 10e-3, while a weak correlation was established as +/- 10e-6. Any partial derivative between -10e-6 and 10e-6 was considered to have no correlation.

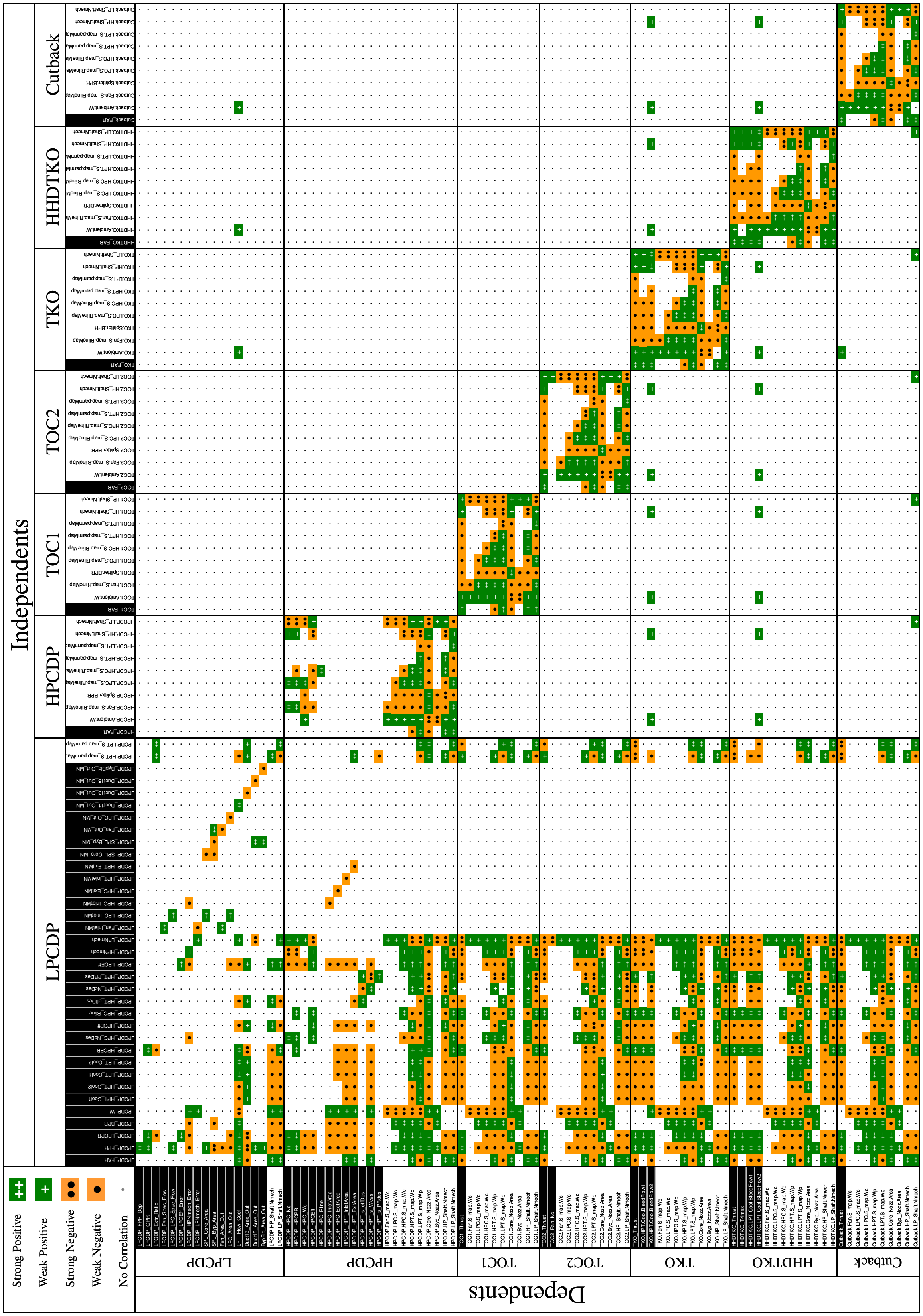


Figure 64. Example Jacobian Matrix for Experiment 2

Inspection of the Jacobian allows for some interesting observations. First, the entire matrix is not populated but rather populated along each diagonal submatrix and the submatrices for the map scaling independents of the LPCDP. The matrix can be considered for the non-map scaling off-diagonal submatrices as in Figure 65. The significance of this observation is that the additional design points are connected to each other through the maps scaling point, even those that share common constraints. While this knowledge of the Jacobian was not incorporated into the modified Newton-Raphson solver, possible advantages to this structure are discussed in the future work section in Chapter 6.

$\frac{\partial F^{LPCDP}}{\partial X^{LPCDP}}$	0	0	0	0	0	0
$\frac{\partial F^{HPCDP}}{\partial X^{LPCDP}}$	$\frac{\partial F^{HPCDP}}{\partial X^{HPCDP}}$	0	0	0	0	0
$\frac{\partial F^{TOC1}}{\partial X^{LPCDP}}$	0	$\frac{\partial F^{TOC1}}{\partial X^{TOC1}}$	0	0	0	0
$\frac{\partial F^{TOC2}}{\partial X^{LPCDP}}$	0	0	$\frac{\partial F^{TOC2}}{\partial X^{TOC2}}$	0	0	0
$\frac{\partial F^{TKO}}{\partial X^{LPCDP}}$	0	0	0	$\frac{\partial F^{TKO}}{\partial X^{TKO}}$	0	0
$\frac{\partial F^{HHDTKO}}{\partial X^{LPCDP}}$	0	0	0	0	$\frac{\partial F^{HHDTKO}}{\partial X^{HHDTKO}}$	0
$\frac{\partial F^{Cutback}}{\partial X^{LPCDP}}$	0	0	0	0	0	$\frac{\partial F^{Cutback}}{\partial X^{Cutback}}$

Figure 65. Jacobian Matrix Showing Populated and Unpopulated Submatrices

5.2.2.3. Active Constraints Summary for Experiment 2

Ten constraints are included in the model for the second experiment. The constraints influence on the modified Newton-Raphson solver only occurs when the constraint becomes active. The probability of each constraint being active is shown in Table 36. The first two constraints, the HPT cooling flows at HHDTKO were active approximately 40 % of the time. This means that the TKO condition set the cooling flow requirements approximately 60 % of the time and was usually the more stringent of the two takeoff conditions for HPT cooling flow. The LPT cooling flows constrained by the next two constraints, HHDTKO_LPT_BleedFlow1 and HHDTKO_LPT_BleedFlow2 were active 50% of the time. Therefore the TKO and HHDTKO design points were equally likely to be the more stringent design point for LPT cooling flows.

Table 36. Constraint Summary Results for Experiment 2

Constraint	Percent of Cases Constraint is Active
HHDTKO_HPT_BleedFlow1	36.2%
HHDTKO_HPT_BleedFlow2	40.9%
HHDTKO_LPT_BleedFlow1	49.7%
HHDTKO_LPT_BleedFlow2	50.5%
Max_Fan_Diam	9.8%
Cutback_Vmix	24.4%
TKO_T3	34.9%
HHDTKO_T3	1.7%
LPCDP_LPCPRmin	4.8%
TOC2_Fan_Nc	44.2%
HHDTKO_T4max	57.5%

The maximum fan diameter constraint was active 10% of the time and the cutback Vmix was active 25% of the time. This caused the design space along FPR to be constrained from achieving the value specified in the DoE for 35% of the cases. For 10% of the cases, FPR was constrained to a higher value than in the DoE and for 25% of the cases FPR was constrained to a lower value. A multivariate plot showing the FPR constrained cases is shown in Figure 66. The blue data points in Figure 66 show how far the constrained FPR was raised due to the fan diameter constraint. In most cases, only a small adjustment was required to make the candidate engine design feasible, but a shift of up to 0.1 in FPR did occur within the design space. Similarly, the green data points show

how far FPR was adjusted lower to make a feasible candidate engine due to the cutback Vmix constraint. The largest change in FPR from cutback Vmix was -0.2.

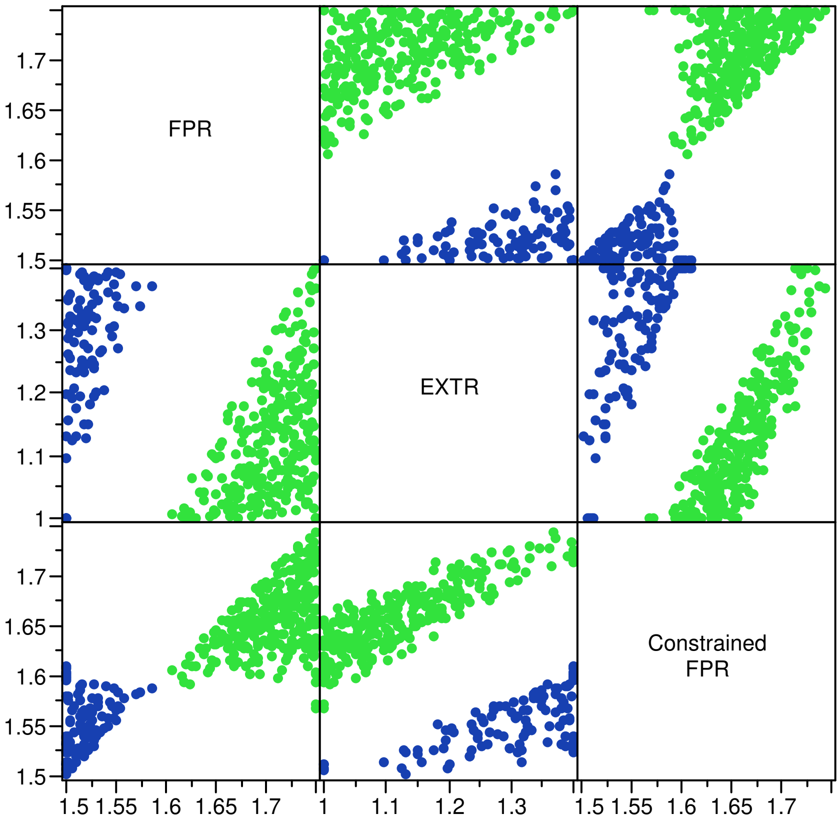


Figure 66. Multivariate Plot of Constrained FPR for Experiment 2

The TKO_T3 constraint was active for 35 % of the cases, HHTKO_T3 was active for only 2% of the cases and LPCDP_LPCPRmin was active for 5% of the cases. Thus the design space along OPR was constrained from achieving the specified DoE value for 42 % of the cases. LPCDP_LPCPRmin constrained the solver to use an OPR higher than the DoE value for 5% of the cases, while TKO_T3 and HHDTKO_T3 constrained OPR to a lower value than in the DoE. A multivariate plot showing the OPR constrained cases is shown in Figure 67. The blue data points in Figure 67 show the cases where OPR was constrained higher by the LPCDP_LPCPRmin constraint while the green data points show how far the OPR was constrained by the T3 constraints at TKO and HHDTKO. The maximum shift in OPR due to the LPCDP_LPCPRmin constraint was 2.5 while the maximum shift for OPR due to a T3 constraint was -12 in order to create a feasible candidate engine.

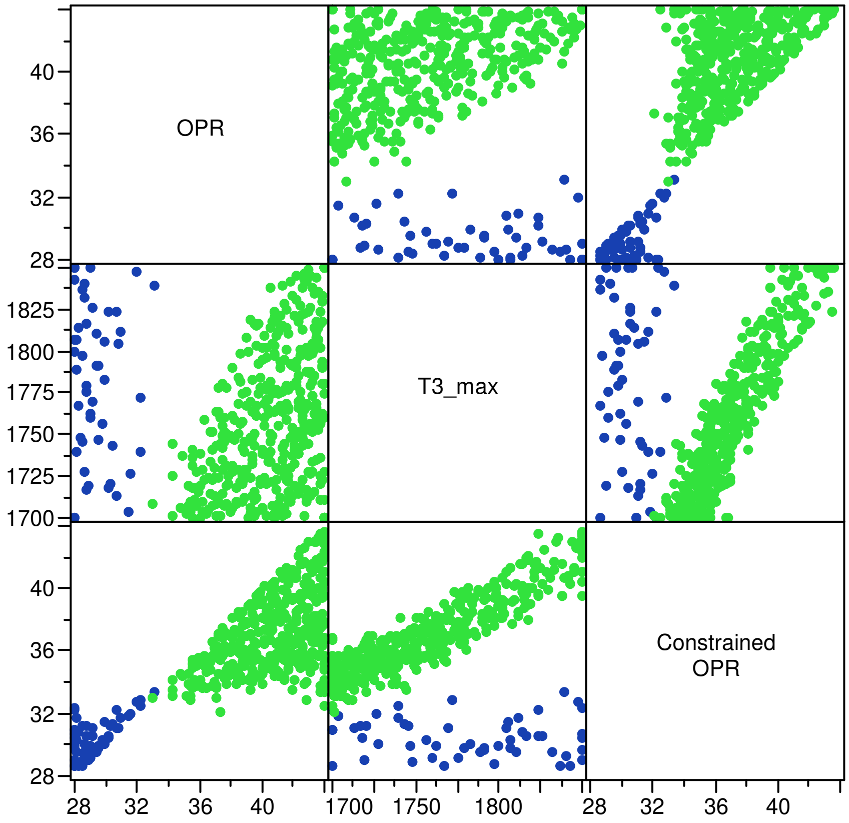


Figure 67. Multivariate Plot of Constrained OPR for Experiment 2

The final two constraints, TOC2_Fan_Nc and HHDTKO_T4max were active for 44% and 58% of the cases. The design variable TOC_NcMap from the DoE was set at TOC1 for 56% of the cases and TOC2 for 44% of the cases. Likewise, T4_max from the DoE was set at TKO for 42% of the cases and HHDTKO for 58% of the cases.

The number of constraints active for each case is summarized in a histogram in Figure 68. The maximum number of constraints active for any one case was 8 which was achieved for 8 % of the cases. Meanwhile, almost 16% of the cases had none of the constraints active.

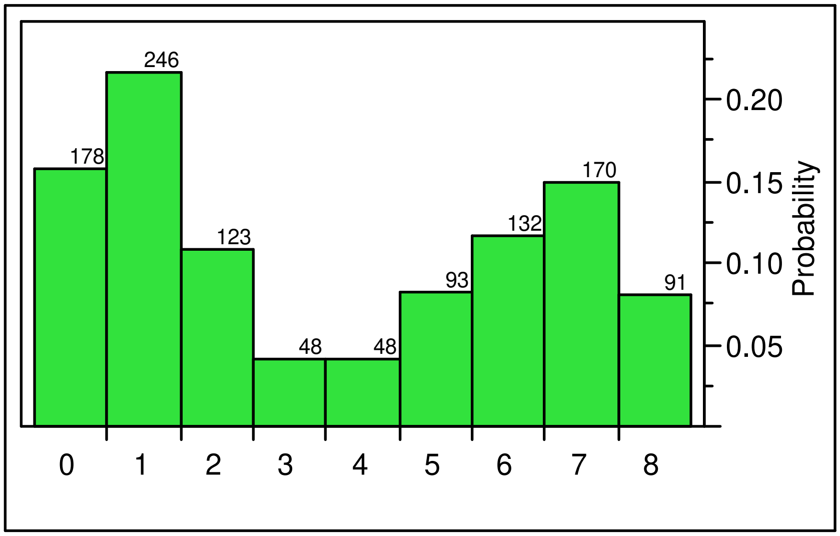


Figure 68. Histogram of Number of Active Constraints per Case for Experiment 2

In order to understand the relationship between the design space and active constraints, the correlation of the active constraints to the design variables is tabulated in Table 37. An understanding of the correlation can allow the cycle analyst the flexibility to move candidate engine cycles away from a constraint. The data is also provided in multivariate plots in Figure 102 through Figure 112 in Appendix I. For the two HPT bleed flow constraints the most important design variable is TOC_NcMap followed by the EXTR and FPR. The correlations show a candidate engine with a high TOC_NcMap, high FPR and low extraction ratio is most likely to activate the HPT bleed flows at HHDTKO. The same three design variables are also the most correlated to the LPT bleed flow constraints. There is not as much gradation between the three design variables in terms of importance to activating the LPT bleed floe constraints at HHDTKO. Again, a candidate engine with a high TOC_NcMap, high FPR and low extraction ratio is most likely to activate the LPT bleed flows at HHDTKO. The same three design variables are also the most influential design variables for T4 max set at HHDTKO constraint. The T3 constraint for HHDTKO had no high correlation to any of the design variables due to the fact that it is active for less than 2% of the cases. Examination of HHDTKO_T3 in

Appendix I shows that it is most likely to be active for high FPR, OPR, TOC_NcMap and low extraction ratio. The maximum fan diameter is correlated to FPR and TKO_T3 to OPR as expected. Cutback Vmix is correlated to FPR, but also has some negative correlation to EXTR. The maximum fan over speed at TOC 2 is strongly correlated to FPR.

Table 37. Active Constraint Correlation to Design Variables for Experiment 2

	FPR	OPR	EXTR	T4_Max	TOC_NcMap	T3_Max
Cutback_Vmix	**0.345**	0.015	**-0.406**	0.041	**0.566**	-0.009
HHDTKO_HPT__BleedFlow2	**0.364**	0.000	**-0.439**	0.046	**0.548**	-0.017
HHDTKO_LPT_BleedFlow1	**0.426**	0.003	**-0.473**	0.014	**0.492**	0.000
HHDTKO_LPT_BleedFlow2	**0.425**	0.005	**-0.499**	0.011	**0.472**	0.014
Max_Fan_Diam	**-0.468**	-0.047	0.259	-0.059	-0.188	-0.012
HHDTKO_T4max	**0.412**	-0.031	**-0.468**	0.025	**0.498**	-0.009
TKO_T3	-0.036	**0.704**	0.044	0.073	0.019	-0.323
HHDTKO_T3	0.097	0.157	-0.140	0.059	0.163	0.090
Cutback_Vmix	**0.646**	0.049	**-0.342**	0.036	0.150	0.035
TOC2_Fan_Nc	**0.805**	0.010	-0.155	0.001	0.104	-0.065
LPCDP_LPCPRmin	0.102	**-0.319**	0.198	0.254	-0.073	0.028

Experiment 2 showed that it is possible to simultaneously satisfy multiple gas turbine engine performance requirements and constraints at several design points. Adding constraints to the design problem was found to improve the performance of the modified Newton-Raphson solver by reducing the mean number of model passes for convergence. This trend was especially important as the ranges on the design variables expanded. The effect the constraints have on the value of certain design variables in order to create feasible candidate engines was captured. Finally, certain characteristics of the Jacobian matrix were analyzed leading to the conclusion that when creating the system of nonlinear equations, the design points are all linked to each other through the map scaling point.

5.3 Experiment 3: Simultaneous MDP Sensitivity Analysis

The objective of third experiment is to examine the sensitivity of convergence for the simultaneous MDP method by identifying the key parameters within the modified Newton-Raphson solver as well as examining the initial iterate for the independent parameters and determining their impact on the convergence rate through a systematic exploration of the design variables. The experiment, in conjunction with experiment 4 is designed to test the third hypothesis: Simultaneous MDP can find a solution for candidate engine cycles having a wide range of design variables and technology levels while maintaining a high level of efficiency.

There are several key characteristics of simultaneous MDP that are evaluated in the third experiment. The first is the sensitivity of convergence to the initial iterate for the modified Newton-Raphson solver. With this information, a method for systematically constructing an initial iterate for any cycle design problem is developed and tested. The second key feature is the trade off between the ability to find all of the feasible candidate engines in the CDS and the computational efficiency based on the use of step limits placed on the independent parameters for every iteration of the modified Newton-Raphson solver. Additionally, the comparison is made between the Newton-Raphson method with full calculation of the Jacobian for every iteration and the use of the Jacobian estimation technique, Broyden's method. Next the characteristics of the independent parameter solution space, the space defined by the solution values of the independent parameters for each case, are discussed. Finally, the third experiment evaluates the effectiveness of the MDP repair algorithm and determines the penalty in efficiency for recovering the feasible cases that initially fail.

5.3.1 Experiment 3 Setup

The third experiment is divided into 3 sub experiments. The first sub experiment maintains the same values for the design variables and technology parameters but varies the values of the initial iterate. The design variables are; FPR equal to 1.6, OPR equal to 35, EXTR equal to 1.2, T4_Max equal to 3350 and TOC_NcMap equal to 1.05. To reduce the dimensionality of the Jacobian and independent parameter solution space to be examined in experiment 3, only four design points are included in the experiment; LPCDP, HPCDP, TOC1 and TKO. This limited the number of independent-dependent combinations to 65. The ranges for the independent variable initial values are listed in Table 65 through Table 68 in Appendix D. The ranges can be summarized however as the minimum of +/- 20% from the actual solution or a physical limit that could not be exceeded such as turbine and compressor efficiencies greater than 1.0. No constraints are enabled for this test as the candidate engine does not change. To examine the initial iterate sensitivity, a 50,000 case LHC is assembled with the different initial iterate settings.

The second sub experiment examines the sensitivities of the different alternatives and settings for the modified Newton-Raphson solver. The same four design points, LPCDP, HPCDP, TOC1 and TKO are used in the sub experiment. To push the solver capabilities, the design variable ranges are substantially increased to FPR from 1.2 to 2, OPR from 25 to 65, T4_Max from 3100 to 3500 °R and TOC_NcMap which remains the same as experiment 2 at 1.03 to 1.08. The DoE is constructed of 43 CCD cases coupled with 4000 LHC cases. The goal of the second sub experiment is to choose the best alternatives and settings for the solver to optimize robustness and efficiency.

The objective of the third sub experiment is to test the MDP repair algorithm. The setup is the same as the second sub experiment. The only difference is the inclusion of the MDP repair algorithm. The results of the best modified Newton-Raphson solver for the second sub experiment are compared to the results of the third sub experiment with the repair algorithm. The DPMM for experiment 3 is shown in Table 38.

Table 38. Design Point Mapping Matrix for Experiment 3

Design Points		**Low Pressure Aero Design Point (LPADP)** Mn 0.80 | 35 kft | dT 0	**High Pressure Aero Design Point (HPADP)** Mn 0 | 0 kft | dT 0	**Top of Climb 1 (TOC1)** Mn 0.85 | 39 kft | dT 0	**Takeoff (TKO)** Mn 0.25 | 0 ft | dT +27
Comp. Map Scaling Point		LPADP			
Design Variables	FPR	1.2 to 2.0			
	LPCPR	Function of OPR			
	OPR	25 to 65			
	Fan TOC Nc			1.03 to 1.08	
	Extraction Ratio	0.8 to 1.5			
Performance Requirements	TOC1 Thrust			15,800 lbf	
	TKO Thrust				75,000 lbf
	Fan Nc Map	100%			
	HPC Nc Map		Function of OPR		
Component Performance Estimation	Fan Adiabatic η	Function of FPR			
	LPC Adiabatic η	Function of LPCPR			
	HPC Adiabatic η		Fixed Map		
	Fan Spec. Flow	45 lbm/ft^2-s			
	LPC Spec. Flow	26.5 lbm/ft^2-s			
	HPC Spec. Flow		Fixed Area		
	HPT Adiabatic η		Fixed Map		
	LPT Adiabatic η	0.92			
	Burner η	0.999			
	HPT NC Cooling Effect.				1.5
	HPT Ch Cooling Effect.				0.8
	LPT NC Cooling Effect.				1.5
	HPT NC Cooling Effect.				1.2
	Bypass Nozzle Cv			Function of PR	
	Bypass Nozzle CdTh			Function of PR	
	Core Nozzle Cv			Function of PR	
	Core Nozzle CdTh			Function of PR	
Technology Limits	T4 Max.				3100 to 3500 °R
	HPT 1st Vane Temp.				2400 °R
	HPT Blade Temp.				2300 °R
	HPT Vane Temp.				2300 °R
	LPT Blade Temp.				2000 °R
	LPT Vane Temp.				2000 °R
	T3 Max.				1750 °R
	Min. LPCPR	1.15			

5.3.2 Experiment 3 Results

In experiment 2, the variation in the key solver metrics demonstrates that the performance of the modified Newton-Raphson method must be understood and optimized for both robustness and efficiency. The results of experiment 3 are divided into five sections. The first section examines the modified Newton-Raphson solver’s sensitivity to the initial iterate and makes claims regarding the initial iterate that should be utilized to create a CDS. The second section discusses the means of systematically obtaining the initial iterate described in the first section. The third section examines the keys options within the modified Newton-Raphson solver, namely the use of full Jacobian matrix calculation versus a Jacobian matrix update with Broyden's method, and the setting of the step limits for the individual independent parameters. The fourth section analyses the independent parameter solution space to determine important characteristics of the independent parameters with regards to the solver. The final section examines the effectiveness of the MDP repair algorithm and the cost in terms of computational expense to improve the convergence success rate.

5.3.2.1. Initial Iterate Sensitivity

As was discussed in the experimental setup in Chapter 4, the initial iterate sensitivity study consists of a 50,000 case LHC DoE with the values of the initial iterate varied +/- 20% from the independent parameter solution to the fixed values of the design variables which were chosen to be near the middle of the CDS. The results of the experiment with regards to solver performance are summarized in Table 39.

Table 39. Summarized Results for Initial Iterate Sensitivity Study

Solver Metric	Value
Total Number of Cases	50,000
Cases Converged	4219
Convergence Success Rate	8.44%
Iterations: Mean	156.45
Iterations: StDev	69.59
Jacobians: Mean	4.44
Jacobians: StDev	1.80
Model Passes: Mean	445.36
Model Passes: StDev	180.45

Only 4219 of the 50,000 cases successfully converged to the solution for a for a paltry convergence success rate of 8.44%. This poor performance indicates that the initial iterate must carefully be selected in order to secure a robust solver for the design space. The correlation between the successfully converged cases and the settings of the initial iterate can be found in Table 70 through Table 73 in Appendix E. The correlations show that 10 to 12 independents have some measurable influence on the convergence, but even accounting for these independents would not improve the convergence success rate to a desirable level greater than 50%.

Not only is the convergence success rate poor, but the solver performance of the cases that did successfully converge were appalling. The mean number of iterations required to successfully converge with an arbitrary initial iterate is 156.5. Compare this to the results from experiment 2 where for a wide range of design variables the average number of iterations was between 26 and 32 per case. Similarly, the number of Jacobian increased from approximately 1.6 in experiment 2 to almost 4.5 in for the arbitrary initial iterate. Worst of all, the true measure of computation expense for the solver, the number of model passes increased 250% to 445. Clearly the method for selecting the initial iterate in experiment 2 was profoundly better that randomly choosing the initial iterate, even when the solution was known beforehand and the initial iterate was not allowed to deviate more than +/- 20% from the know solution.

Regarding the initial iterate, the simultaneous MDP method postulates that:

A single initial iterate can be utilized by the modified Newton-Raphson solver to efficiently and robustly find solutions to the candidate engine cycles within the CDS provided that the initial iterate ***is itself a solution to one of the candidate engines*** within the cycle design space.

The initial iterate in experiment 2, although not stated at the time, did follow this postulate and provides evidence to the postulates merit. This postulate has some powerful considerations. The first is the development of a single initial iterate that is not a function of the design variables but can be used for all candidate engine cycles within the feasible CDS. Such a relationship would be difficult to obtain and provide a significant barrier to the attractiveness of the simultaneous MDP method. The second consideration is the known location of the initial iterate within the CDS. This information will facilitate the MDP repair algorithm discussed in section 5.3.2.5. Further evidence to bolster this postulate is provided in the discussion of the independent parameter solution space in section 5.3.2.4.

5.3.2.2. Development of Initial Iterate for Cycle Design Space

The postulate regarding the initial iterate provides a perplexing conundrum, namely, if the initial iterate must be a solution to a candidate engine in the CDS, but a good initial iterate is required to find a solution to the candidate engine, how is the initial iterate originally obtained? One could use the “shotgun” approach of randomly choosing initial iterates and hope that one works, but as was seen in the initial iterate study the percentage of random initial iterates that will converge is low, even when the range was constrained around the actual solution which would not normally be possible. Without this knowledge the range of the possible initial iterates would increase further reducing the likelihood of convergence. Furthermore, the selection of the initial iterate is most often carried out during the development of the MDP model. The cycle analyst may not be able to distinguish convergence failure due to a poor initial iterate from a problem with the setup of the MDP model itself. A systematic method is required that can assure the development of a good initial iterate that meets the conditions of the initial iterate postulate.

The method for determining the initial iterate is simple, systematic, robust, and requires no special knowledge about MDP modeling other than the operating conditions of the design points from which the independent parameters are derived. The method is to utilize a SPD cycle model coupled with off-design analysis. First designate one of the design points, usually but not necessarily the map scaling point, as the SPD design point. Design the engine cycle using the SPD model by specifying the design variables that pertain to the design point to fall within the desired MDP CDS. Next, recreate the other design point operating conditions in off-design analysis as best as possible. From both the design and off-design points, the values of the independent parameters can be obtained for the initial iterate. When implementing the simultaneous MDP model, it may be necessary to initially change some of the design point operating conditions to match the off-design conditions from the SPD analysis, but this is usually not necessary.

This method for finding the initial iterate was implemented using the LPCDP as the SPD design point and with the other four design points analyzed in off-design. Assumptions regarding the HPT and LPT cooling flows, LPCDP T4 and LPCDP thrust were required for the SPD analysis. Further the exact operating conditions of the HPCDP were not precisely matched as the off-design points were controlled to achieve a thrust target not a specified HPC corrected shaft speed. The thrust for HPCDP was set to 90% part power. The cycle results from the SPD analysis for the five design points, with the values of the independent parameters highlighted, can be found in Figure 98 through Figure 101 located in Appendix H. The values of the independent parameters from the SPD analysis are listed in Table 40.

Table 40. Initial Iterate Values Obtained from Off-Design Analysis

Independent	Value	Independent	Value
LPCDP.Ambient.W	1373	LPCDP.HP_Shaft.Nmech	8390
LPCDP.Burner.FAR	0.02430	LPCDP.LP_Shaft.Nmech	2052
LPCDP.Fan.S_map.PRdes	1.547	HPCDP.Ambient.W	2930
LPCDP.LPC.S_map.PRdes	1.444	HPCDP.Burner.FAR	0.02530
LPCDP.LPC.S_map.effDes	0.9047	HPCDP.Fan.S_map.RlineMap	1.522
LPCDP.HPC.S_map.NcDes	0.9930	HPCDP.HP_Shaft.Nmech	8812
LPCDP.HPC.S_map.PRdes	14.51	HPCDP.HPC.S_map.RlineMap	1.813
LPCDP.HPC.S_map.effDes	0.8926	HPCDP.HPT.S_map.parmMap	5.137
LPCDP.HPC.S_map.RlineMap	1.985	HPCDP.LP_Shaft.Nmech	1979
LPCDP.HPT.S_map.parmMap	4.888	HPCDP.LPC.S_map.RlineMap	1.960
LPCDP.HPT.S_map.parmMapDes	4.888	HPCDP.LPT.S_map.parmMap	4.157
LPCDP.HPT.S_map.effDes	0.9001	HPCDP.Splitter.BPR	8.529
LPCDP.HPT.S_map.parmNcDes	100.6	TOC1.Ambient.W	1225
LPCDP.LPT.S_map.parmMap	5.415	TOC1.Burner.FAR	0.02670
LPCDP.Splitter.BPR	8.763	TOC1.Fan.S_map.RlineMap	2.050
LPCDP.Bld3.Cool1.fracW	0.1150	TOC1.HP_Shaft.Nmech	8600
LPCDP.Bld3.Cool2.fracW	0.07500	TOC1.HPC.S_map.RlineMap	2.050
LPCDP.HPC.Cool1.fracBldW	0.03500	TOC1.HPT.S_map.parmMap	4.800
LPCDP.HPC.Cool2.fracBldW	0.02000	TOC1.LP_Shaft.Nmech	2200
LPCDP.Inlet.Fl_O.MN	0.6508	TOC1.LPC.S_map.RlineMap	2.050
LPCDP.Fan.Fl_O.MN	0.4449	TOC1.LPT.S_map.parmMap	5.500
LPCDP.Splitter.Fl_02.MN	0.4372	TOC1.Splitter.BPR	8.269
LPCDP.Splitter.Fl_01.MN	0.3033	TKO.Ambient.W	3422
LPCDP.Duct4.Fl_O.MN	0.3093	TKO.Burner.FAR	0.03070
LPCDP.LPC.Fl_O.MN	0.3869	TKO.Fan.S_map.RlineMap	1.674
LPCDP.Duct6.Fl_O.MN	0.3927	TKO.HP_Shaft.Nmech	9369
LPCDP.Burner.Fl_O.MN	0.09004	TKO.HPC.S_map.RlineMap	1.833
LPCDP.HPC.Fl_O.MN	0.3003	TKO.HPT.S_map.parmMap	5.044
LPCDP.HPT.Fl_O.MN	0.3090	TKO.LP_Shaft.Nmech	2246
LPCDP.Duct11.Fl_O.MN	0.1965	TKO.LPC.S_map.RlineMap	2.040
LPCDP.BypBld.Fl_O.MN	0.4343	TKO.LPT.S_map.parmMap	4.759
LPCDP.Duct13.Fl_O.MN	0.3138	TKO.Splitter.BPR	8.422
LPCDP.Duct15.Fl_O.MN	0.4343		

The values from Table 40 are then added to the initial iterate for simultaneous MDP and the model run for the same design variable settings as in the previous initial iterate study. The result was successful convergence of the model in 54 iterations, with only 2 full Jacobians generated and 184 model passes. The solver required the 54 iterations to account for differences between the SPD assumptions and actual MDP values and the differences in HPCDP operating conditions.

A comparison of the SPD generated initial iterate versus the successful convergence results from the initial iterate sensitivity study is contained in Figure 69 through Figure 71 for iterations, full Jacobian generation, and the number of model passes. The histograms show that not only did the SPD method for initial iterate generation successfully determine a good initial iterate, but that the initial iterate, designated with an arrow in Figure 69 through Figure 71, performed as good or better than almost all of the randomly chosen initial iterate sensitivity cases.

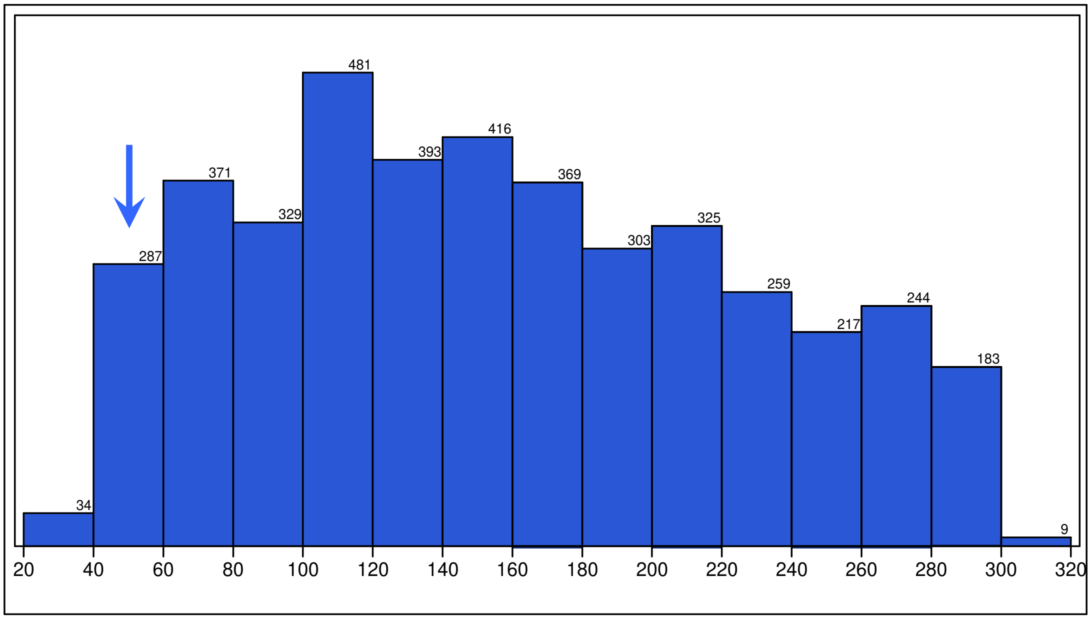


Figure 69. Iterations for Initial Iterate Sensitivity Analysis

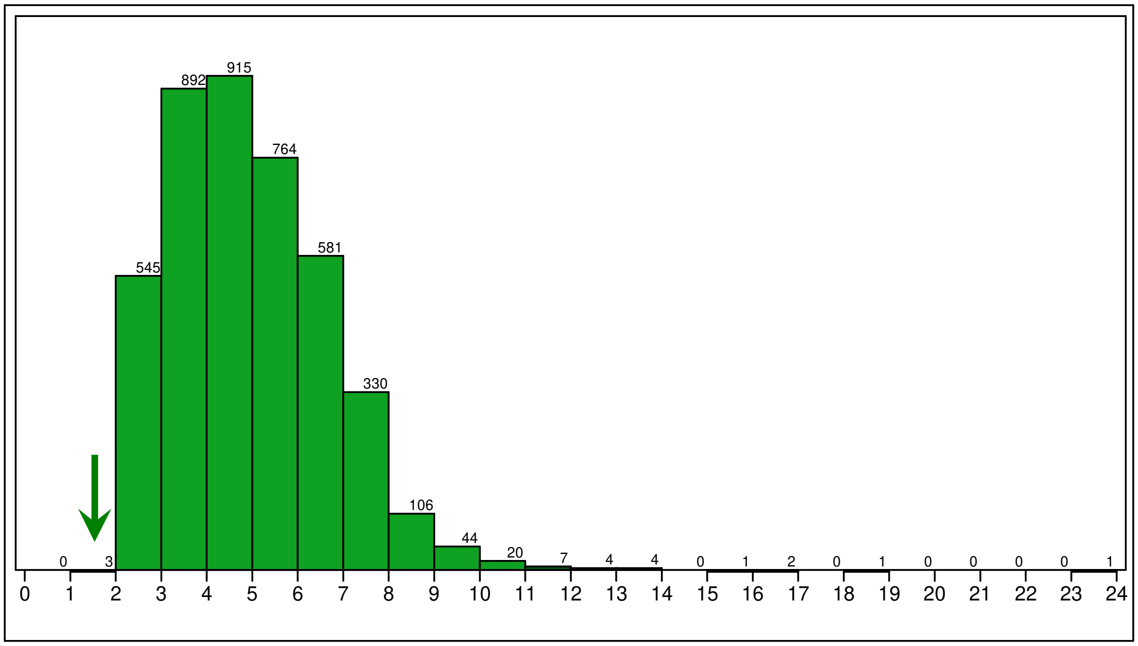


Figure 70. Full Jacobian Matrices for Initial Iterate Sensitivity Analysis

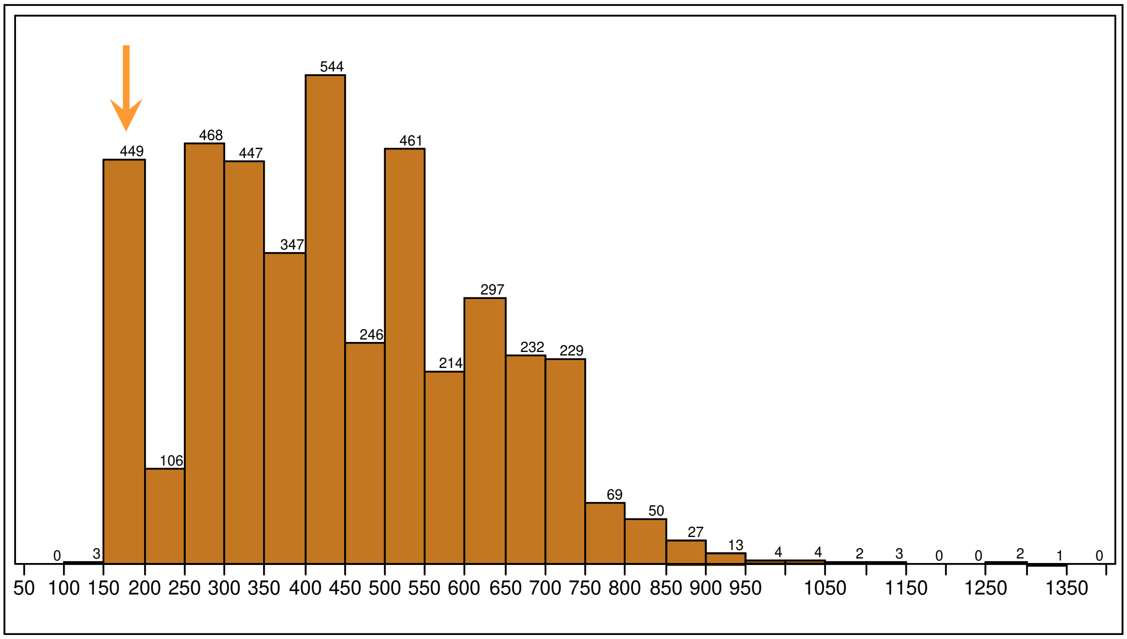


Figure 71. Model Passes for Initial Iterate Sensitivity Analysis

This method can be modified as well for different MDP modeling situations. Perhaps an MDP model already exists, but additional design points are desired to account for additional requirements or constraints. Instead of reverting to an SPD model, the cycle analyst can run the current MDP model with the existing design points and then analyze the additional design points in off-design mode to determine their independent parameters' values. Another possibility is that an MDP model exists with all of the desired design points, but a different thrust class engine is desired such as going from a 90,000 lbf class engine to a 20,000 lbf class engine. The independent parameters for these two design spaces, even if the engine architectures are the same, will have significantly different values. It may be necessary to step down in thrust class from 90k to 70k to 50k … all the way to 20k. At each step, the values of the independent parameters are updated to the solution of the previous thrust class. In this manner, the cycle analyst can quickly adjust the initial iterate from one thrust class CDS to another.

5.3.2.3. Modified Newton-Raphson Solver Settings

In the previous experiments, the convergence success rate, number of iterations full Jacobians and model passes were monitored as key solver performance metrics. In this section, the different settings and options for the modified Newton-Raphson solver are examined to determine their influence on the key performance metrics. The results of the first part of the experiment, summarized in Table 41, compare the solver performance of the solver using full Jacobian updates for almost every iteration to the use of Jacobians updated with Broyden's method for almost every iteration

Table 41. Summarized Results for All Jacobian vs. Broyden Update

Solver Metric	Full Jacobian	Broyden Update
Total Number of Cases	4043	4043
Cases Converged	2906	3137
Convergence Success Rate	71.88	77.6%
Iterations: Mean	33.88	38.65
Iterations: StDev	19.88	20.48
Jacobians: Mean	32.57	2.64
Jacobians: StDev	20.04	0.98
Model Passes: Mean	2150.69	210.36
Model Passes: StDev	1322.34	74.83

The results clearly show that full Jacobian calculations should be reduced to as few iterations as possible. First, the convergence success rate surprisingly decreased slightly for the full Jacobian calculations compared to the Broyden updates. There was no solver robustness benefit in terms of convergence success rate for the full Jacobian calculations. Second, while the number of iterations was less for the full Jacobian at 33.9 compared to the Broyden update at 38.7 iterations as predicted in Chapter 3, the average number of model passes increased by an order of magnitude for the full Jacobian calculation from 210.4 to 2150.7. Clearly there is a significant benefit to the efficiency of the solver by incorporating Broyden’s method. Furthermore, this benefit increases as the dimensionality of the Jacobian is increased though additional design points. The number of model passes for the full Jacobian calculations is proportional to the dimensionality of the Jacobian times the number of iterations, where as the number of model passes for the Broyden update is proportional to the dimensionality of the Jacobian times the number of full Jacobians + number of iterations –number of full Jacobians. Therefore it is important for solver efficiency to reduce the number of full Jacobian updates to as small a number as possible.

It was mentioned that it was surprising that the convergence success rate for full Jacobian calculations was less than for the Broyden update. A summary of the converged cases for both methods is found in Table 42 ands shows 252 cases which converged using Broyden's method updates but failed for the full Jacobian calculation.

Table 42. Summarized Results for All Jacobian vs. Broyden Update

Solver Metric	Value
Cases Converged for Both Full Jacobian and Broyden	3770
Cases Converged for Broyden, Failed Full Jacobian	252
Cases Converged for Full Jacobian, Failed Broyden	21

Examination of these failures revealed a pattern for a majority of the cases. When calculating full Jacobians for each iteration, the solver was more likely, as illustrated by an example case in Figure 72, of entering an infinite loop. The metric in Figure 72 is the error norm of the solver which is defined in equation 33:

$$error\_norm = \left(\sum_{i=1}^{u+m} error_i^2 \right)^{1/2} \tag{33}$$

The Broyden update case converged in 19 iterations, and, with 2 exceptions continuously reduced the error norm for each successive iteration. The Full Jacobian method, while initially outperforming the Broyden update method, after the ninth iteration found itself in an infinite loop from which it could not recover.

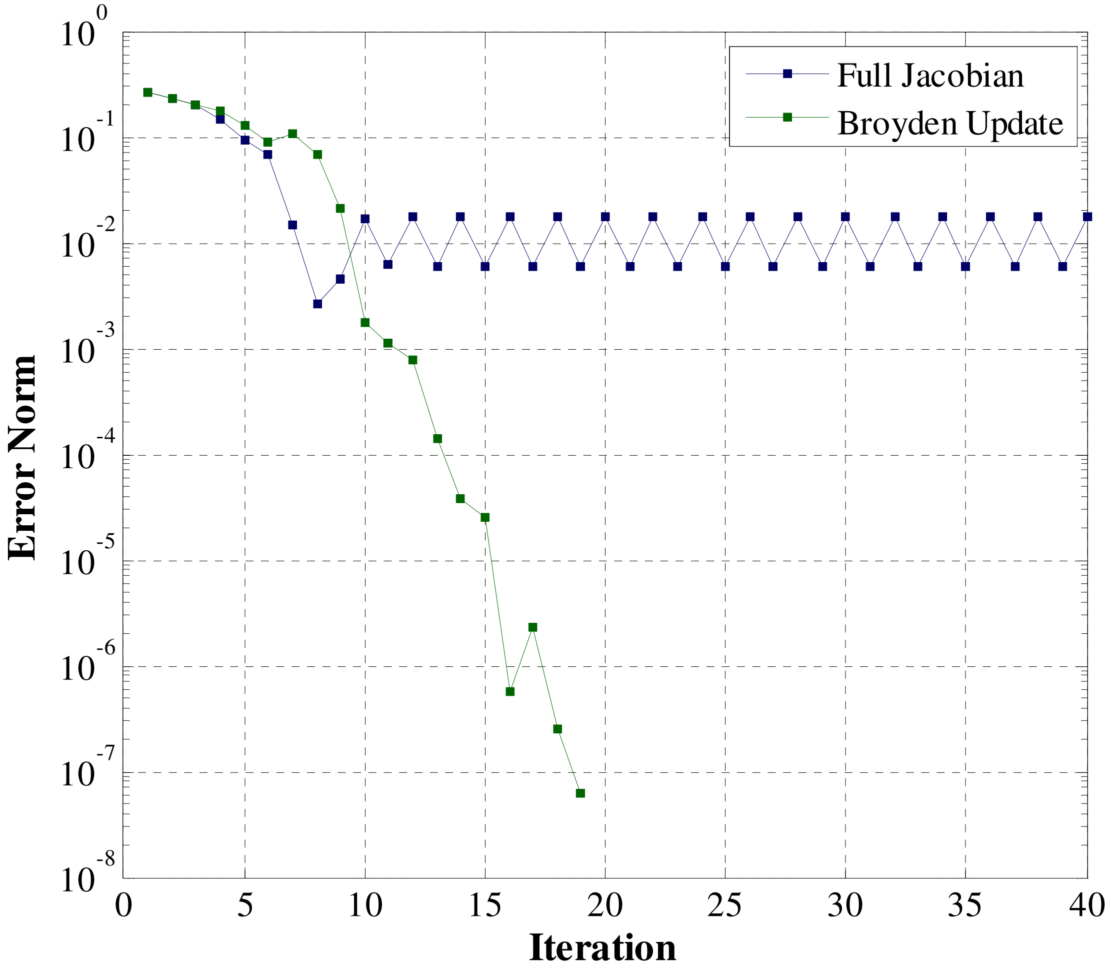


Figure 72. Error Norm for Failed Full Jacobian Case

The next solver setting of interest was the step limits placed on the independent parameters. Most of the solver independent step limits can be defined as a percentage of the independent parameter value. However, it is necessary for some of the independents to have absolute limits placed on their step size, either because the independent value can approach zero or a maximum amount of step limit is known. Within NPSS, the most limited independent is used to limit the movement of all independents[58] so it is important to ensure that the step limits are not too small. The absolute limits are specified for each independent separately and do not change for this experiment. The list of independents with fixed step limits is shown in Table 43. The list includes the LPT nonchargeable cooling flow, FPR and LPCPR, and all of the inlet and exit Mach numbers. All remaining independent parameters are set to the same percentage step limit tested in this sub experiment..

Table 43. Independent Parameters with Fixed Step Sizes

	Independent Parameter	Step Limit
1	LPCDP_LPT_Cool2	0.01
2	LPCDP_FPR	0.05
3	LPCDP_LPCPR	0.1
4	LPCDP_Fan_InletMN	0.02
5	LPCDP_LPC_InletMN	0.02
6	LPCDP_HPC_Rline	0.01
7	LPCDP_HPC_InletMN	0.02
8	LPCDP_HPC_ExitMN	0.02
9	LPCDP_HPT_InletMN	0.02
10	LPCDP_HPT_ExitMN	0.02
11	LPCDP_SPL_Core_MN	0.05
12	LPCDP_SPL_Byp_MN	0.05
13	LPCDP_Fan_Out_MN	0.05
14	LPCDP_LPC_Out_MN	0.05
15	LPCDP_Duct11_Out_MN	0.05
16	LPCDP_Duct13_Out_MN	0.05
17	LPCDP_Duct15_Out_MN	0.05
18	LPCDP_BypBld_Out_MN	0.05

The step limits for the remaining independent parameters were varied from 0.025 to 0.15. The results of the key solver performance metrics for each level of step limit are shown in Table 44.

Table 44. Summarized Results for Step Limit Sensitivity

Solver Metric	Step 0.025	Step 0.035	Step 0.050	Step 0.100	Step 0.150
Converged Cases	3149	3144	3137	3129	3127
Convergence Success Rate	77.89%	77.76%	77.595	77.395	77.345
Iterations: Mean	45.77	40.54	38.65	37.64	37.91
Iterations: StDev	21.90	20.57	20.48	20.89	21.48
Full Jacobians: Mean	2.66	2.66	2.64	2.68	2.70
Full Jacobians: StDev	1.28	1.27	0.98	1.09	1.21
Model Passes: Mean	218.89	213.69	210.36	211.94	213.31
Model Passes: StDev	94.1	92.89	74.83	82.01	89.26

Before selecting the optimal step limit for simultaneous MDP, it is instructive to examine the step limits effect for a particular sample case. The effects of the different step limit values on the error norm for a sample case is shown in Figure 73. The number of iterations required for convergence of the sample case ranged from 40 to 55. The performance of the solver should be considered as follows: while the solver is moving the values of the initial iterate to get close to the solution, the step limits are active and very little reduction in error norm is made by the solver. Once the solver reaches an area in the solution space close enough to the final solution, the step limits are no longer active and the modified Newton-Raphson solver convergence rate behaves quadratically, rapidly reducing the error to the final tolerance value. This process can be seen more clearly in Figure 74 and Figure 75. Examining the 0.1 step limit line in Figure 74, the first 40 iterations are constrained by the step size. During the first 40 iterations the error norm is reduced from 1.2 to 0.012 a factor of 10^{-2}. From iteration 41 to 46, the step limit is inactive and the error norm was reduced in just 6 iterations from 10^{-2} to 10^{-8}. A similar effect can be seen in the data for the 0.15, 0.0.5 and 0.05 step limits. The step limits size determines how many iterations are required to reach a place in the solution space where the step limit is no longer active at which point the solver rapidly converges to a solution in between 6 to 10 iterations. Too small of a step limit, such as the 0.025 data shown in Figure 73 results in the solver not having the flexibility to get to the correct area of the solution space in as few iterations as possible. However, too large of a step limit can cause the solver to overshoot the desired area of the solution space and thus take more iterations as well as evident by Figure 74.

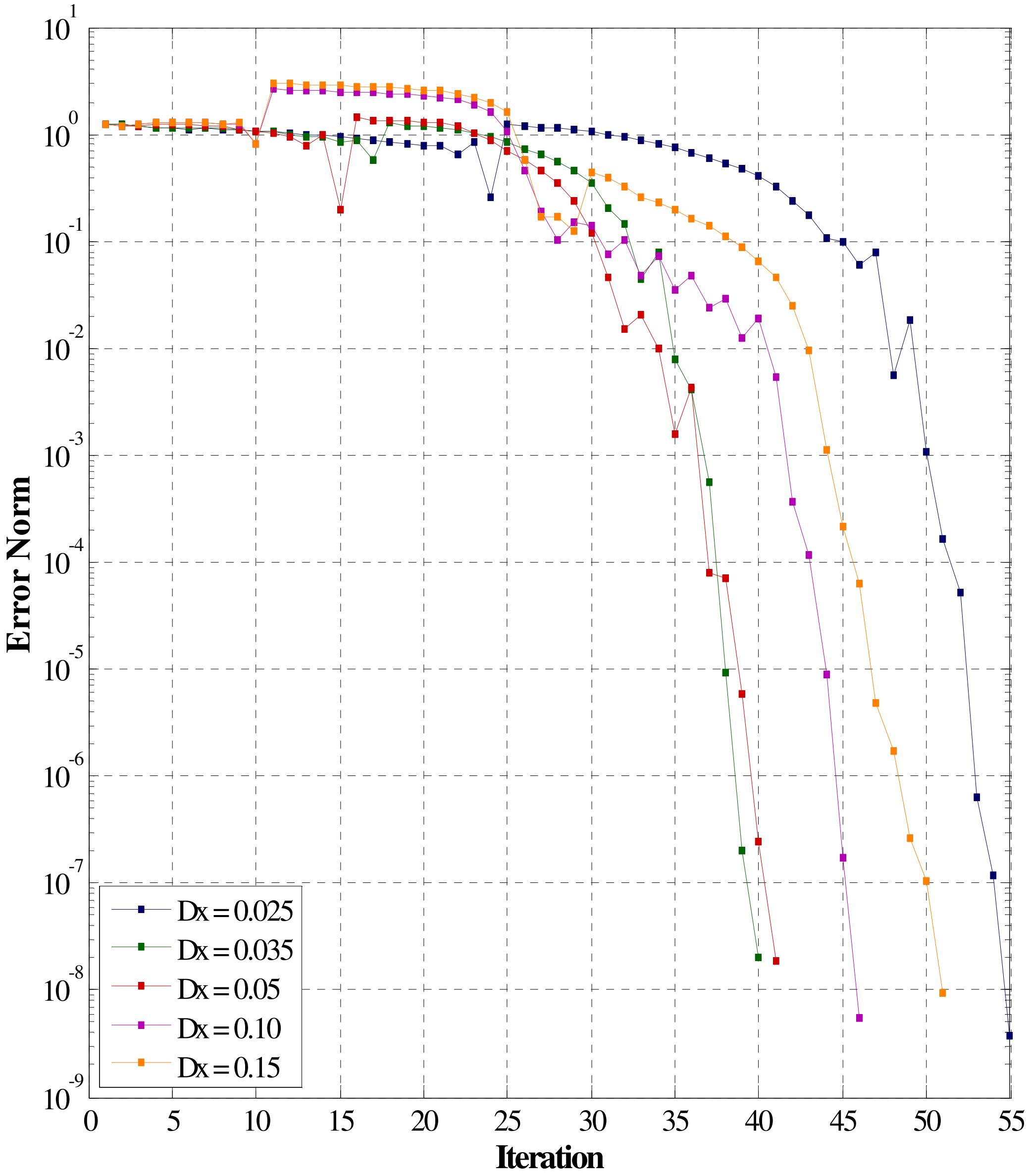


Figure 73. Effects of Step limits on Error Norm for Sample Case

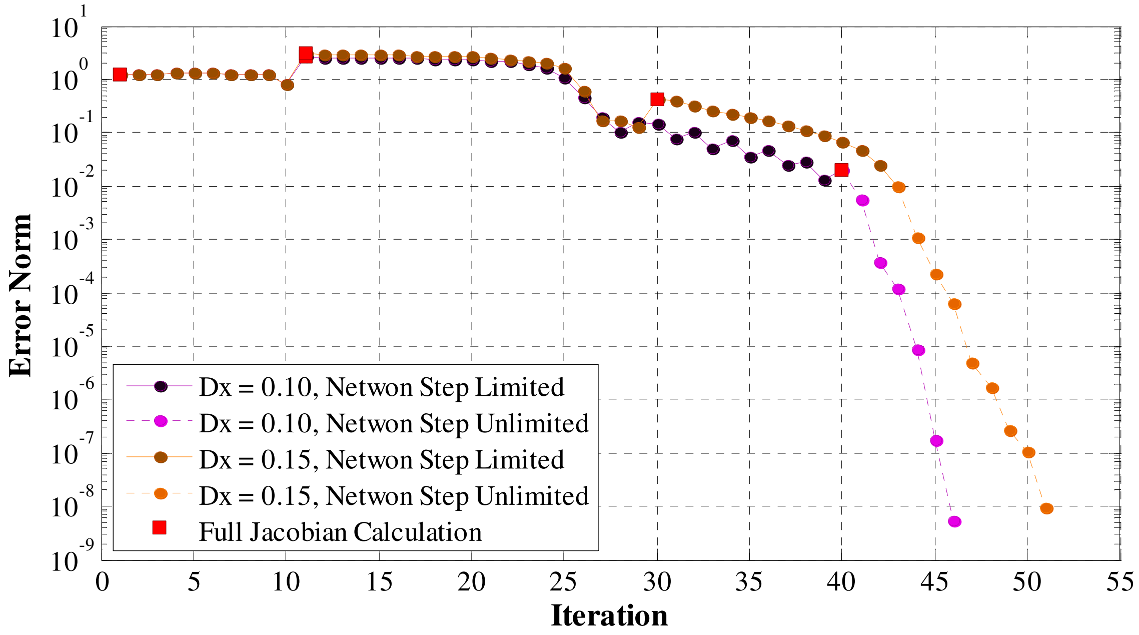


Figure 74. Effects of 0.1 and 0.15 Step limits on Error Norm for Sample Case

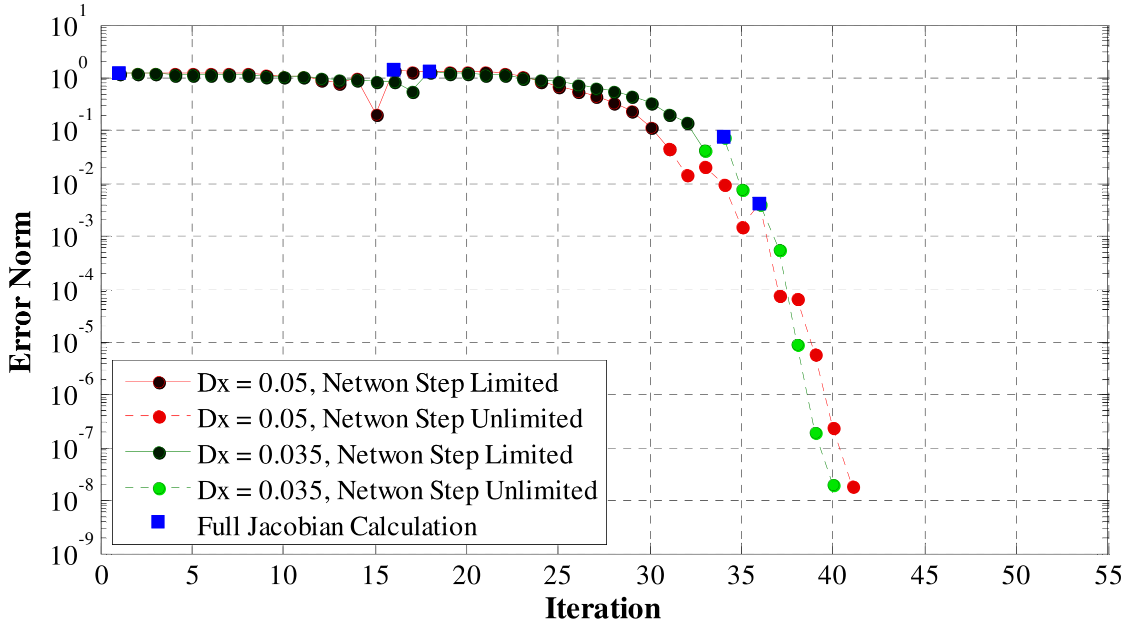


Figure 75. Effects of 0.035 and 0.05 Step limits on Error Norm for Sample Case

The convergence success rate for the different step limits consistently, but only marginally, improved as the step limit was reduced. The minimum mean number of model passes as well as the minimum variation in model passes occurred for the step limit size equal to 0.05. Based on this information, plotted in Figure 76, the step limit for independent parameters is set to 0.05 for the rest of the experiments.

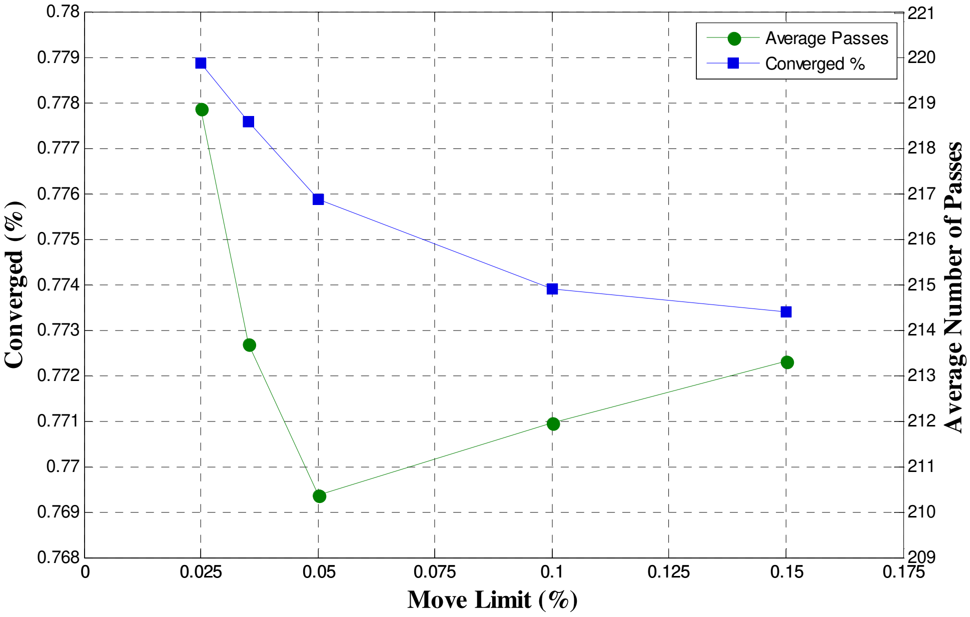


Figure 76. Effects of Step limits on Solver Statistics

The next solver setting to consider determines how often to recalculate the full Jacobian. Within the NPSS solver are different limits which can trigger the full Jacobian calculation [58]. The limits are defined by the convergence ratio which is calculated by dividing the current error norm by the previous iteration error norm. A convergence ratio less than 1 signifies the solver is improving the solution while a value grater than 1 means the solver has moved to a worse solution. There are two limits for the full Jacobian calculation. The first limit, called the divergence limit, causes an immediate full recalculation of the Jacobian when it is attained. The divergence limit is set to 1.5, that is if the solution is 50% worse than the previous solution, the Broyden update is assumed to be faulty and a full Jacobian calculation is required. The second limit, called the convergence limit, causes a full recalculation of the Jacobian after a certain number of occurrences. This limit is set to 1. An experiment is run to determine the optimal number of occurrence the convergence limit should be exceeded before recalculating the full Jacobian. The results of this experiment are shown in Table 45.

Table 45. Summarized Results for Convergence Limit Counter Sensitivity

Solver Metric	Converge Limit = 2	Converge Limit = 4	Converge Limit = 6	Converge Limit = 8
Converged Cases	3131	3137	3140	3140
Convergence Success Rate	77.44%	77.59%	77.67%	77.68%
Iterations: Mean	37.52	38.65	38.91	38.96
Iterations: StDev	19.21	20.48	20.78	20.87
Full Jacobians: Mean	3.00	2.64	2.59	2.58
Full Jacobians: StDev	1.43	0.98	0.95	0.95
Model Passes: Mean	232.44	210.36	207.04	206.68
Model Passes: StDev	103.01	74.83	72.97	72.72

The effect of the convergence limit counter on the convergence success rate was negligible. For a counter limit of 2 the convergence success rate was 77.4% while a counter limit of 8 only increased the success rate to 77.68%. The convergence limit counter did have an effect on the efficiency of the solver as shown in Figure 77. An average of 232.4 model passes were required for a convergence limit counter of 2 while a convergence limit counter of 8 only required an average of 206.7 model passes. This is a positive effect, but one that plateaus above a convergence counter limit of 6. Therefore, a convergence counter limit of 6 is chosen for the simultaneous MDP

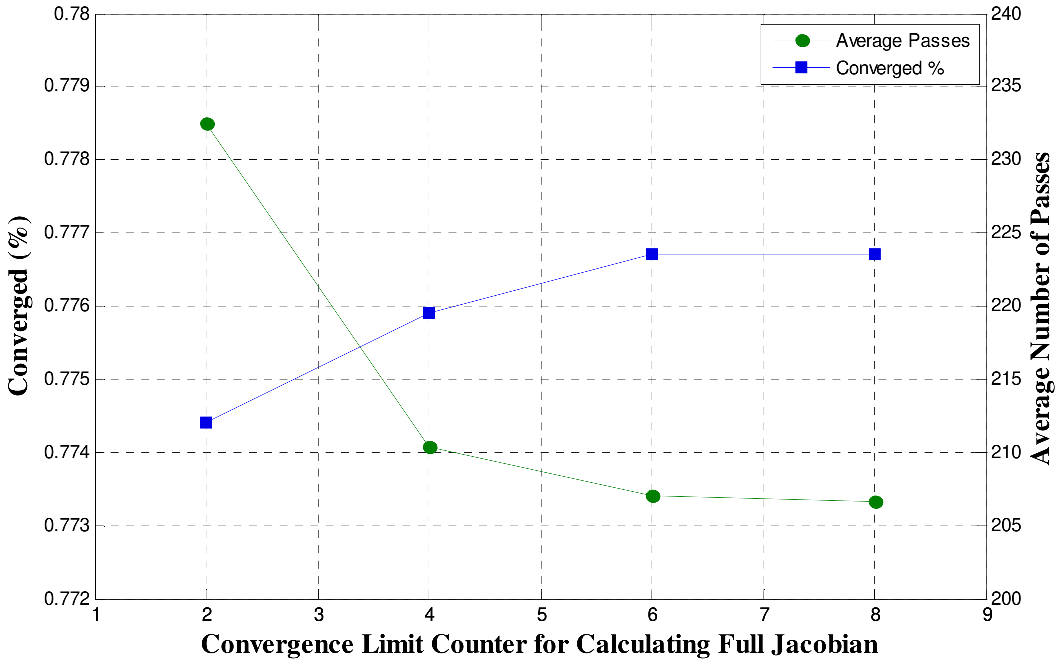
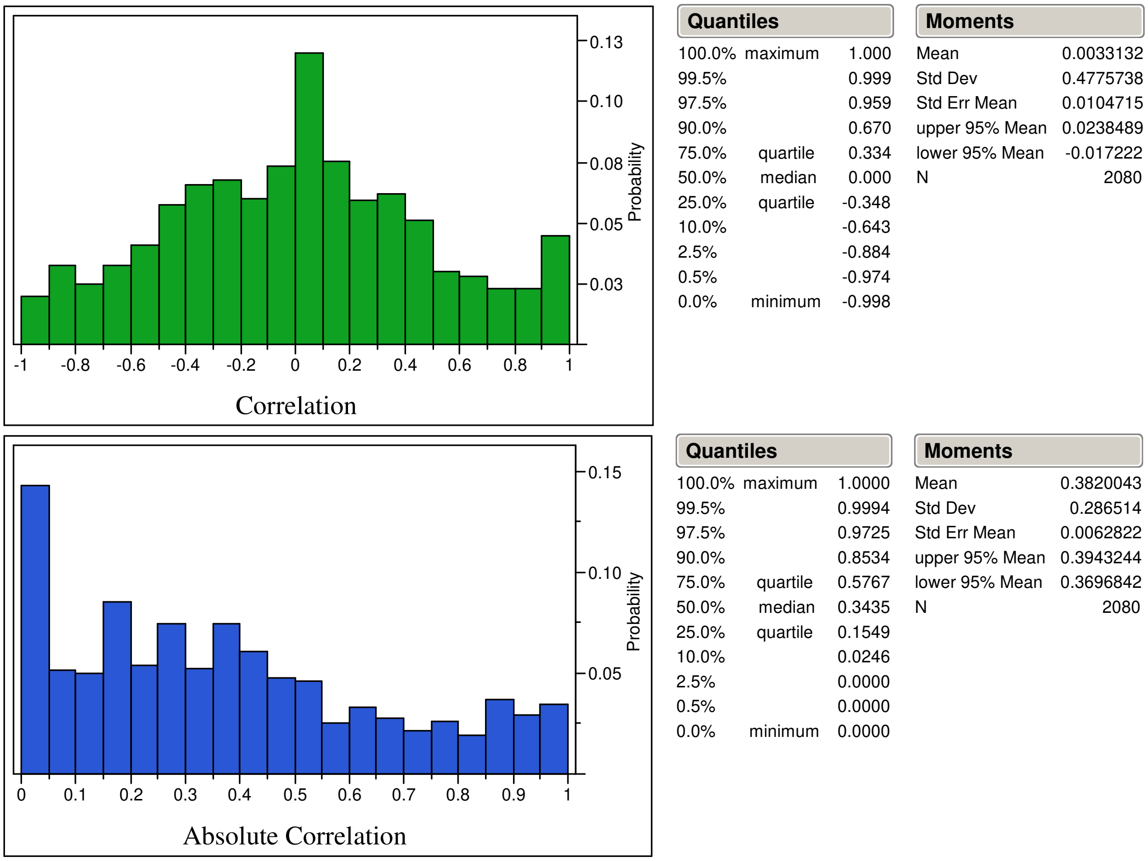


Figure 77. Effects of Converge Limit Counter for Full Jacobian Calculation

5.3.2.4. Independent Parameter Solution Space for Experiment 3

The independent parameter solutions space, the final value of each of the 65 independent parameters for all 4035 cases exhibits interesting behavior. Multivariate plots of the independent parameters showing the correlation existing between them can be found in Figure 90 through Figure 97 in Appendix F. The solution space should not be thought of as the minimum and maximum solution value for each independent, but rather the area formed by the collection of solutions in the multivariate plots. The multivariate plots show one important feature: the independent parameters within the solution space are highly correlated. The independent parameters are independent in the sense that the modified Newton-Raphson solver has control to set them, but they do not have the property of statistical independence from each other. The correlation data can be summarized by the histograms in Figure 78, which shows the distribution of the correlation and absolute correlation. Ten percent of the solution space has a correlation of 0.85 or higher, 25% of the solution space has an absolute correlation of at least 0.57 or higher and half the space has an absolute correlation of at least 0.34. This result provides further evidence for the postulate that the initial iterate must lie in the solution space. There are so many combinations of independent parameters that are correlated that randomly choosing an initial iterate, even knowing the ranges of the solution space, may leave you well outside the actual solution space and doomed to solver convergence failure



Quantiles

100.0%	maximum	1.000
99.5%		0.999
97.5%		0.959
90.0%		0.670
75.0%	quartile	0.334
50.0%	median	0.000
25.0%	quartile	-0.348
10.0%		-0.643
2.5%		-0.884
0.5%		-0.974
0.0%	minimum	-0.998

Moments

Mean	0.0033132
Std Dev	0.4775738
Std Err Mean	0.0104715
upper 95% Mean	0.0238489
lower 95% Mean	-0.017222
N	2080

Quantiles

100.0%	maximum	1.0000
99.5%		0.9994
97.5%		0.9725
90.0%		0.8534
75.0%	quartile	0.5767
50.0%	median	0.3435
25.0%	quartile	0.1549
10.0%		0.0246
2.5%		0.0000
0.5%		0.0000
0.0%	minimum	0.0000

Moments

Mean	0.3820043
Std Dev	0.286514
Std Err Mean	0.0062822
upper 95% Mean	0.3943244
lower 95% Mean	0.3696842
N	2080

Figure 78. Experiment 3 Independent Parameter Solution Space Correlation

The ranges of the solution space can be found in Table 75 through Table 78 in Appendix G. The initial iterate for the experiment was chosen as the solution of the midpoint of the design space. However, this did not result in the initial iterate being positioned in the middle of the independent parameter solution space. On average, the upper limit of the solution space for each independent parameter was 85% above the initial iterate value, while the lower limit was on average only 35% below the initial iterate value. The maximum upper limit was 412 % higher than the initial iterate for TOC1 BPR and the lower limit was -100% for the LPT chargeable cooling. It is not necessary to consider where in the solution space the initial iterate is located, as it is not possible to know the size of the solution space before the analysis of the CDS. It is only important that the initial iterate be located somewhere within the independent parameter solution space,

5.3.2.5. Simultaneous MDP Repair Algorithm Results

The final section of experiment 3 examines the effectiveness of the MDP repair algorithm for failed cases. A solver determines a failed case to be one that reaches 200 iterations or generates 50 full Jacobians without converging. The key solver metrics with data for the repair algorithm are shown in Table 46. Two different DoEs, one with an extended range for the design variables and one with a narrower range, were utilized for this section of experiment 3. For the extended range DoE, 902 cases initially failed. Of the 902 failed cases, 92 or 10.1% were converged utilizing the MDP repair algorithm. For the narrower range DoE, only 83 cases initially failed. A total of 44 or 53% of the initial failed cases were recovered with the MDP repair algorithm for the narrower DoE. It should be noted that for the vast majority of the failed cases that could not be repaired, a feasible solution does not exist. This is why there are many more failed cases for the extended range DoE and why the MDP repair algorithm could not recover more of the failed cases.

To better understand the logic for the MDP repair algorithm, two different cases are illustrated in Figure 79 and Figure 80. The first case illustrates an example where the initial iterate was moved close enough to the final solution that the solver could converge to the initial DoE values of the design variables. In the second case, the solver could not converge on the initial DoE values no matter how close to the final solution the initial iterate started. However, by incrementing the design variable values by a small amount, a solution could be found. The increment for each of the design variables is the DoE base value/1000.

Table 46. Solver Summary Results for Repair Algorithm

Solver Metric	Extended DoE	Narrow DoE
Converged Cases	3233	4004
Convergence Success Rate	79.97%	99.04%
Converged Cases [1st Attempt]	3141	3960
Convergence Success Rate [1st Attempt]	97.15%	98.90%
Iterations [1st Attempt]: Mean	38.99	31.89
Iterations [1st Attempt]: StDev	20.97	17.43
Iterations [Repaired]: Mean	881.7	750.6
Iterations [Repaired]: StDev	456.2	140.8
Full Jacobians [1st Attempt]: Mean	2.604	2.004
Full Jacobians [1st Attempt]: StDev	1.134	1.183
Full Jacobians [Repaired]: Mean	154.1	184.8
Full Jacobians [Repaired]: StDev	134.2	50.85
Model Passes [1st Attempt]: Mean	208.2	162.2
Model Passes [1st Attempt]: StDev	84.46	84.78
Model Passes [Repaired]: Mean	10898	12761
Model Passes [Repaired]: StDev	9006	3313

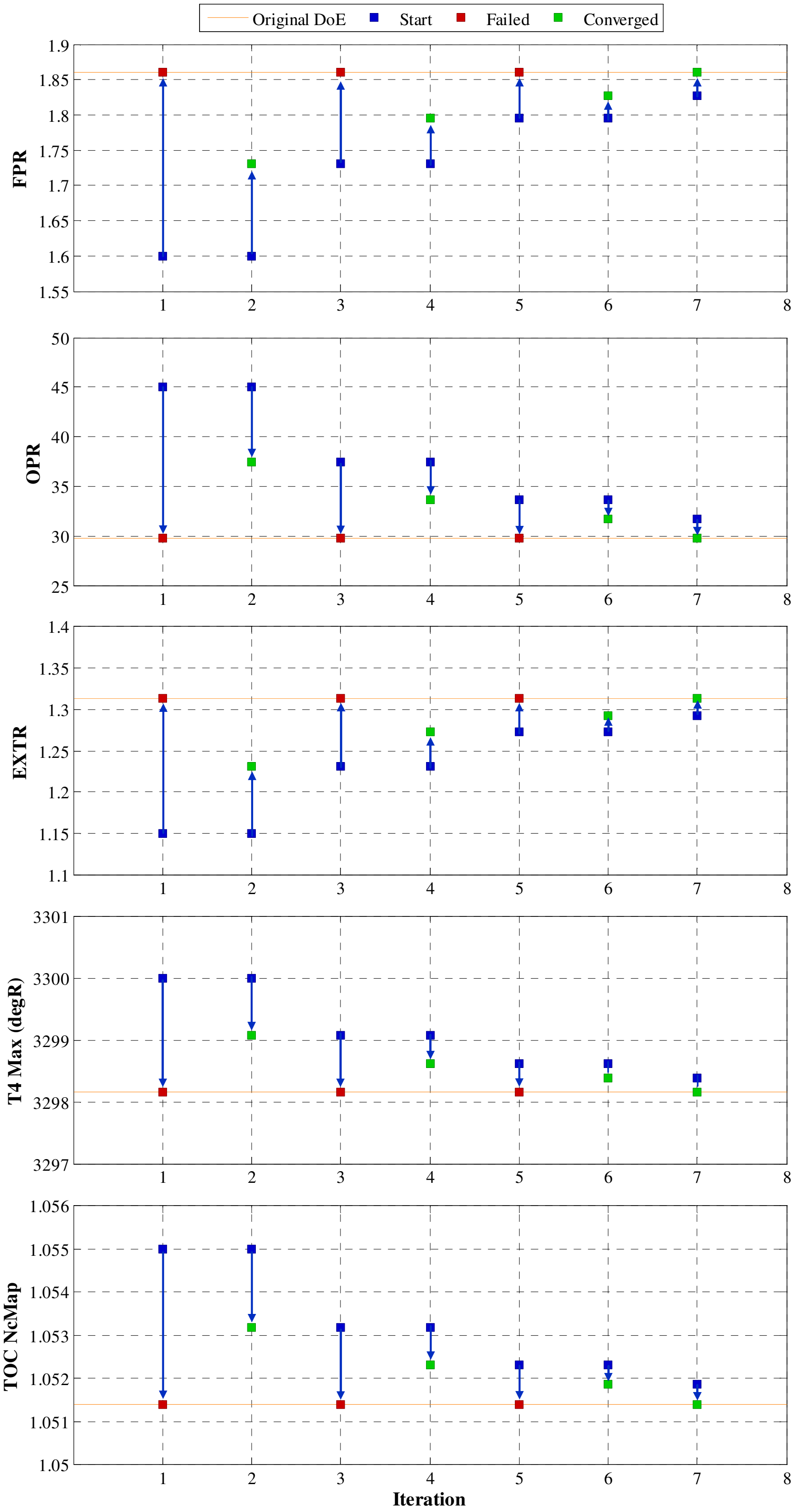


Figure 79. MDP Repair Algorithm Convergence to DoE Values of Design Variables

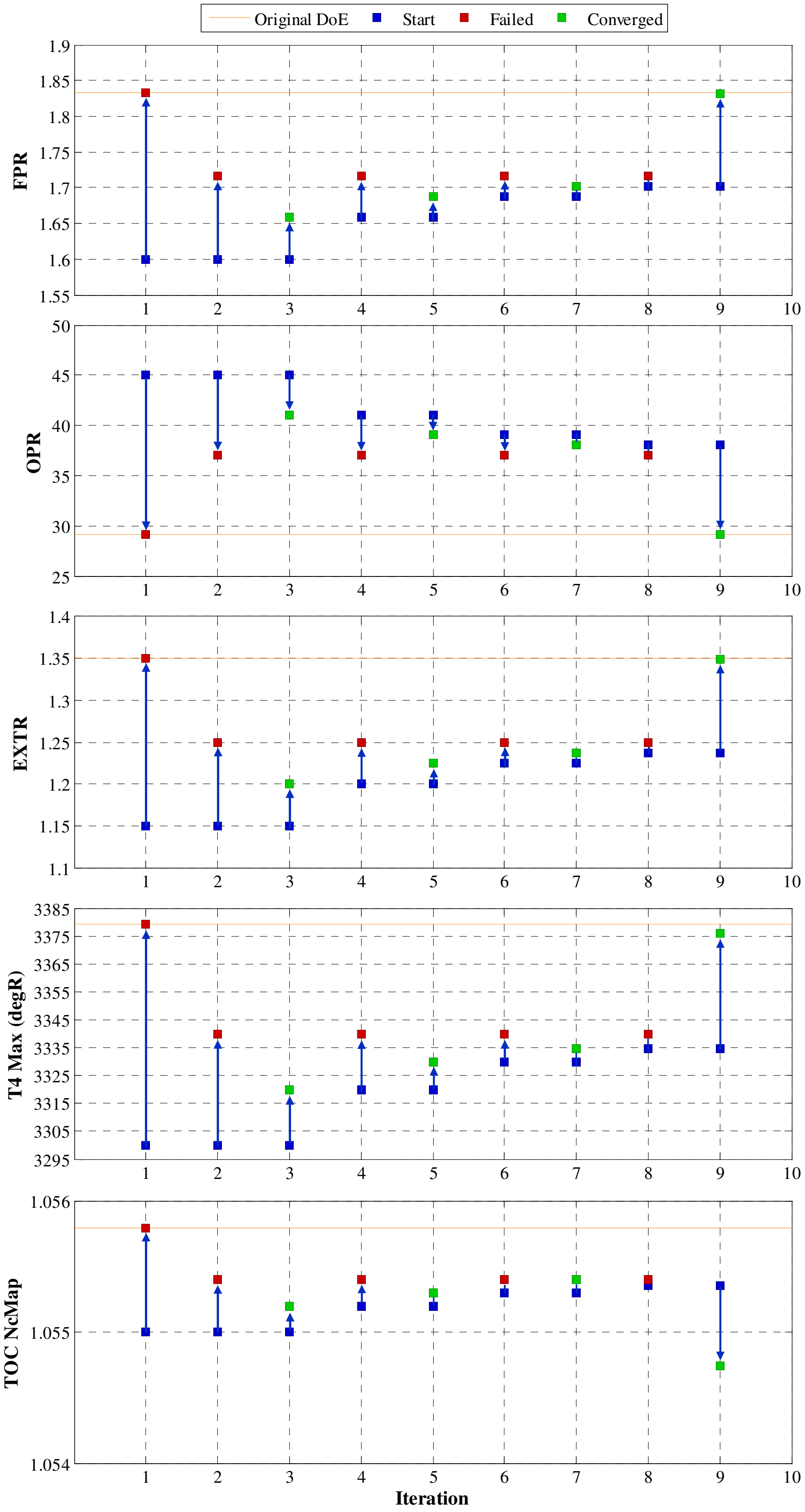


Figure 80. MDP Repair Algorithm Convergence with Increment of Design Variables

The pattern of the solver shown in Figure 79 is the most probable path taken by the MDP repair algorithm for cases that eventually converge to the original DoE values. The pattern consists of the solver failing initially and enabling the MDP repair algorithm. The initial step is calculated halfway between the DoE base and initial values. The solver finds the solution for the new design, updates the initial iterate and then tries to again converge to the original DoE values. The solver fails to converge, the step size is cut in half and a new attempt is made to converge on another solution. This process repeats with the initial iterate moving closer to the final solution until the solver converges or reach a repair algorithm iteration limit.

For the case displayed in Figure 79, seven iterations of the algorithm were required to reach convergence, with the initial iterate moving 87.5% of the distance between the initial iterate and the final solution in the CDS. The sample case in Figure 80 could not move 50 % of the way through the design space before the algorithm exceeded the number of allowable iterations. However, the initial iterate was close enough, that a solution could be found when the design variables were incremented from their original values. The design variables are always incremented towards the base DoE values. To ascertain the effect on the aerothermodynamic cycle, a few cycle metrics are listed in Table 47 for the same failed case that has been incremented towards and away from the base DoE.

Table 47. Comparison of DoE Incremented for Failed Case

Design Point Metrics	Increment Towards Base	Increment Away From Base	% Difference From Mean
LPCDP Thrust	17272	17206	+/- 0.38 %
LPCDP TSFC	0.6252	0.6258	+/- 0.10%
HPCDP Thrust	27038.9	28083	+/- 3.79%
HPCDP TSFC	0.3322	0.3329	+/- 0.21%
TOC1 TSFC	0.6658	0.6672	+/- 0.21%
TOC1 Wc	2729.15	2714.94	+/- 0.52%
TKO TSFC	0.4859	0.4869	+/- 0.21%
TKO Wc	2624.21	2612.83	+/- 0.44%
TKO HPT Cool 1	0.0925	0.0933	+/- 0.86%
TKO HPT Cool 2	0.0754	0.0765	+/- 1.45%
TKO LPT Cool 1	0.0421	0.0428	+/- 1.65%
TKO LPT Cool 2	0.0341	0.0351	+/- 2.89%

The performance of the original DoE values for the design variables falls somewhere in between and, to allow for a comparison, is assumed to be the mean of the two incremented values. The largest difference is the HPCDP thrust. Since HPCDP is a reference point, its difference is not as important to consider for the design. A more apt comparison would be for TKO and TOC1 TSFC and mass flow because both design points have to match their required thrust. For TSFC, both TKO and TOC1 have a difference of +/- 0.21% from their mean. The mass flow for TKO has a difference of +/- 0.44 % and the mass flow of TOC1 has a difference of +/- 0.52%. These results show that while not capturing the original candidate engine cycle from the DoE, the performance of a suitable replacement often can be found with the MDP repair algorithm.

One final discussion regarding the MDP repair algorithm is the computational expense associated with recovering the failed cases. As shown in Table 46, the average number of model passes increased almost two orders of magnitude from 150 – 200 model passes for initial convergence to 10,000 - 12,000 model passes for the repair algorithm, not to mention all of the model passes for the cases that failed to converge. Several trades can be made between the robustness and efficiency of the repair algorithm. The efficiency can be improved by either reducing the iteration or Jacobian limit definition for a failed case from the original values of 200 and 50 or reduce the number of attempts to find a solution in the repair algorithm before incrementing the design variables form the DoE values. For the remaining experiments, the repair algorithm will be set to maximize robustness to determine the maximum convergence success rate and the associated computational expense.

To recap the results for the third experiment, the ability for the solver to converge is extremely dependent upon the choice of the initial iterate values for the independent parameters. Even when the ranges of the independent parameters are narrowed around a known solution, the probability of randomly choosing a good initial iterate are small because of the high degree of correlation between the independent parameters. Therefore, the initial iterate utilized by the modified Newton-Raphson solver must be a solution to one of the candidate engines within the cycle design space so that the solver starts within the solution space. An initial iterate can be found that enables efficient performance for the modified Newton-Raphson solver through a SPD model and the use of off-design analysis to determine the values of the independent parameters. Additional options for the modified Newton-Raphson solver were chosen to optimize performance including the use of Broyden updates as often as possible, setting the independent parameter step limit to 5% of the independent's values and setting the convergence limit to 6 before recalculation of the full Jacobian. Finally, the MDP repair algorithm was tested and successfully found solutions to initially failed cases when a feasible candidate engine existed.

5.4 Experiment 4: Technology Effects on Simultaneous MDP and the CTS

The fourth experiment tests the simultaneous MDP method's ability to address changing technology parameters as well as design variables. The design problem is formulated equivalently to experiment 3 with the same design points. However the technology parameters such as component efficiencies and material temperature limits are allowed to vary within prescribed ranges. Therefore, this experiment evaluates simultaneous MDP capability to generate a CTS.

5.4.1 Experiment 4 Setup

The experiments are also divided into three sub experiments. The first sub experiment allows only the design variables to vary and incorporates the best modified Newton-Raphson solver settings determined from experiment 3. The second sub experiment adds variable technology parameters to the design variables, while the third sub experiment replaces the base fan and LPT maps . The different fan and LPT base maps are shown in Figure 81 and Figure 82. As with experiment 3, no constraints were enabled for this experiment.

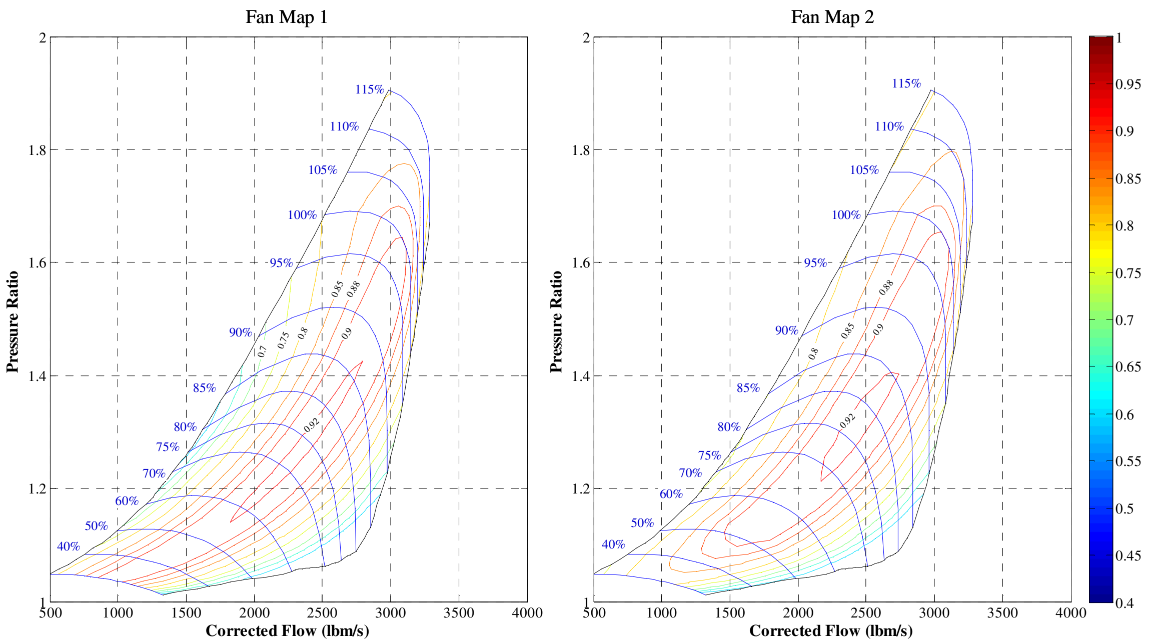


Figure 81. Fan Map Comparison for Experiment 4

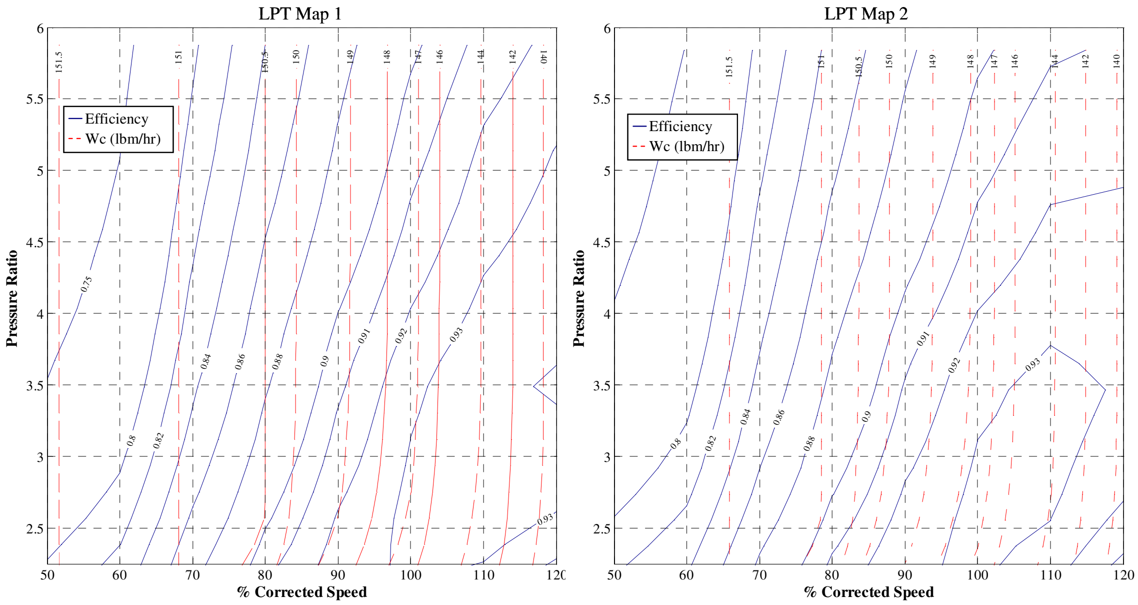


Figure 82. LPT Map Comparison for Experiment 4

The ranges for the design variables are: FPR from 1.5 to 1.75, OPR from 28 to 44, EXTR from 1 to 1.4, T4_Max from 3275 to 3450 °R and TOC_NcMap from 103% to 108%. The technology parameters are Fan, LPC and LPT delta adiabatic efficiencies from -.01 to 0.01, HPT first vane material temperature from 2400 to 2500 °R, HPT blade and vane material temperature from 2300 to 2400 °R and LPT blade and vane temperature from 2000 to 2100 °R.

Table 48. Design Point Mapping Matrix for Experiment 4

Design Points		Low Pressure Aero Design Point (LPADP) Mn 0.80 | 35 kft | dT 0	High Pressure Aero Design Point (HPADP) Mn 0 | 0 kft | dT 0	Top of Climb 1 (TOC1) Mn 0.85 | 39 kft | dT 0	Takeoff (TKO) Mn 0.25 | 0 ft | dT +27
Comp. Map Scaling Point		LPADP			
Design Variables	FPR	1.5 to 1.75			
	LPCPR	Function of OPR			
	OPR	28 to 44			
	Fan TOC Nc			1.03 to 1.08	
	Extraction Ratio	1.0 to 1.4			
Performance Requirements	TOC1 Thrust			16,025 lbf	
	TKO Thrust				75,000 lbf
	Fan Nc Map	100%			
	HPC Nc Map		Function of OPR		
Component Performance Estimation	Fan Adiabatic η	Function of FPR			
	Fan Adiabatic η – Delta	-0.01 to 0.01			
	LPC Adiabatic η	Function of LPCPR			
	LPC Adiabatic η - Delta	-0.01 to 0.01			
	HPC Adiabatic η		Fixed Map		
	Fan Spec. Flow	45 lbm/ft^2-s			
	LPC Spec. Flow	26.5 lbm/ft^2-s			
	HPC Spec. Flow		Fixed Area		
	HPT Adiabatic η		Fixed Map		
	LPT Adiabatic η	0.91 to 0.93			
	Burner η	0.999			
	HPT NC Cooling Effect.				1.5
	HPT Ch Cooling Effect.				0.8
	LPT NC Cooling Effect.				1.5
	HPT NC Cooling Effect.				1.2
	Bypass Nozzle Cv			Function of PR	
	Bypass Nozzle CdTh			Function of PR	
	Core Nozzle Cv			Function of PR	
	Core Nozzle CdTh			Function of PR	
Technology Limits	T4 Max.				3275 to 3450 °R
	HPT 1st Vane Temp.				2400 to 2500 °R
	HPT Blade Temp.				2300 to 2400 °R
	HPT Vane Temp.				2300 to 2400 °R
	LPT Blade Temp.				2000 to 2100 °R
	LPT Vane Temp.				2000 to 2100 °R
	T3 Max.				1750 °R
	Min. LPCPR	1.15			

The first sub experiment with only the design variables uses a 4043 case DoE, the same normalized DoE as in experiment 3 with different ranges applied to the design variables. The second and third sub experiments incorporate a 9000 case LHC DoE. The experiment compares the robustness and efficiency of each of the three sub experiments to identify any changes in the solver's performance due to the addition of technology parameters. A summary of the design points, design variables, and technology parameters is listed in the DPMM for experiment 3 in Table 48.

5.4.2 Experiment 4 Results

The fourth experiment tests the simultaneous MDP method's ability to address changing technology parameters as well as design variables capability to generate a CTS in response to the second half of the third hypothesis. Three separate sub experiments were conducted for experiment 4, one with a DoE consisting only of design variables, one consisting of a DoE with design variables and technology parameters, and the final sub experiment consisting of a DoE with design variables and technology parameters and altered base maps for the fan and LPT. For all three tests constraints were not enabled but the MDP repair algorithm was included. The results of the three tests are listed in Table 49. Because of the additional technology parameters, the DoE for the second and third sub experiment were increased from 4043 to 9000 cases. The same initial iterate was utilized for all three cases. Since the DoE base values of the delta efficiencies for the fan, LPC and LPT are zero, the repair algorithm increment for those technology parameters is set to 1e-5.

It was initially thought that the inclusion of technology parameters would have an impact on the performance of the modified Newton-Raphson solver. While these parameters have a significant impact on the candidate engine's predicted aerothermodynamic cycle performance, there was no impact on the solver's robustness or efficiency to find the candidate engine cycle. All three tests had a high convergence success rate above 99% with the tests including technology parameters increasing above 99.5%. More telling, the initial convergence success rate was nearly the same for all three experiments at 98.9% to 99.0%. The number of model passes for initial convergence remained the same between the design variable only DoE and the design variable and technology parameter DoE. Inclusion of different fan and LPT performance maps actually improved the efficiency of the solver lowering the average number of model passes from 162 to 139.

Table 49. Solver Summary Results for Experiment 4

Solver Metric	Only Design Variables	Design Variables and Tech. Parameters	Design Variables, Tech. Parameters and Maps
Total Cases	4043	9000	9000
Converged Cases	4004	8974	8968
Convergence Success Rate	99.04%	99.71%	99.64%
Converged Cases [1st Attempt]	3960	8883	8879
Convergence Success Rate [1st Attempt]	98.90%	98.99%	99.01%
Iterations [1st Attempt]: Mean	31.89	36.65	34.90
Iterations [1st Attempt]: StDev	17.43	17.38	17.33
Iterations [Repaired]: Mean	750.6	887.9	719.3
Iterations [Repaired]: StDev	140.8	299.9	200.1
Full Jacobians [1st Attempt]: Mean	2.004	1.916	1.604
Full Jacobians [1st Attempt]: StDev	1.183	1.045	1.000
Full Jacobians [Repaired]: Mean	184.8	223.4	165.7
Full Jacobians [Repaired]: StDev	50.85	89.1	68.1
Model Passes [1st Attempt]: Mean	162.2	161.2	139.2
Model Passes [1st Attempt]: StDev	84.78	74.99	73.63
Model Passes [Repaired]: Mean	12761	15412	11490
Model Passes [Repaired]: StDev	3313	5980	4482

The addition of technology parameters to create a CTS does not negatively effect the performance of the modified Newton-Raphson method. Therefore the simultaneous

MDP method is equally adept at generating handling design variables and technology parameters to generate CDSs and CTSs.

5.5 Experiment 5: Dimensionality Testing for Simultaneous MDP

The final experiment is designed to test the simultaneous MDP method by increasing the dimensionality of the design problem to give the cycle analyst increased control over the cycle design. The experiment is designed to test the overall hypothesis that a method that satisfies the performance requirements and constraints for all design points simultaneously would allow the cycle analyst to include as many design points as necessary to control all desired aspects of the aerothermodynamic cycle and would be robust and efficient enough to enable a comprehensive exploration of the cycle design space. All features and options are enabled for this experiment which combines the information obtained from the previous three experiments into one sever experiment.

5.5.1 Experiment 5 Setup

A nine design point problem is created to accomplish the goal of the experiment with LPCDP, HPCDP, TOC1, TOC2, TKO, HHDTKO, Cutback, SLSI and SLSU. A total of five thrust requirements are included in the experiment. TOC1 thrust of 16025 lbf, TOC2 thrust of 19600 lbf, TKO thrust of 75,000 lbf, HHDTKO thrust of 67,000 lbf and SLSI thrust which must match SLSU thrust. A total of 115 nonlinear equations are composed by the simultaneous MDP method for modified Newton-Raphson solver. The same design variables and technology parameters from the second sub experiment from experiment 4 with the same ranges are included in the experiment. However, unlike experiments 3 and 4, constraints are enabled for this experiment. The constraints include T3_Max, Cutback Vmix, maximum fan diameter, the pressure difference between the bypass and core stream at the exit of the HPC and the minimum allowable LPCPR.

Table 50. Design Point Mapping Matrix for Experiment 5

Design Points		Low Pressure Aero Design Point (LPADP) Mn 0.80 | 35 kft | dT 0	High Pressure Aero Design Point (HPADP) Mn 0 | 0 kft | dT 0	Top of Climb 1 (TOC1) Mn 0.85 | 39 kft | dT 0	Top of Climb 2 (TOC2) Mn 0.85 | 35 kft | dT 0	Takeoff (TKO) Mn 0.25 | 0 ft | dT +27	High Hot-Day Takeoff (HHDTKO) Mn 0.25 | 5 kft | dT +27	Cutback Mn 0.25 | 2 kft | dT +18	Sea Level Static Installed (SLSI) Mn 0 | 0 kft | dT +27	Sea Level Static Uninstalled (SLSU) Mn 0 | 0 kft | dT +27
Comp. Map Scaling Point		LPADP								
Design Variables	FPR	1.5 to 1.75								
	LPCPR	Function of OPR								
	OPR	28 to 44								
	Fan TOC Nc			1.03 to 1.08						
	Extraction Ratio	1.0 to 1.4								
Performance Requirements	TOC1 Thrust			16,025 lbf						
	TOC2 Thrust				19,600 lbf					
	TKO Thrust					75,000 lbf				
	HHDTKO Thrust						67,000 lbf			
	Cutback Thrust							60,000 lbf		
	SLSI Thrust								SLSU Thrust	
	Fan Nc Map	100%								100%
	HPC Nc Map		Function of OPR							
Component Performance Estimation	Fan Adiabatic η	Function of FPR								
	Fan Adiabatic η – Delta	-0.01 to 0.01								
	LPC Adiabatic η	Function of LPCPR								
	LPC Adiabatic η - Delta	-0.01 to 0.01								
	HPC Adiabatic η		Fixed Map							
	Fan Spec. Flow	43.6 lbm/ft^2-s								
	LPC Spec. Flow	26.5 lbm/ft^2-s								
	HPC Spec. Flow		Fixed Area							
	HPT Adiabatic η		Fixed Map							
	LPT Adiabatic η	0.91 to 0.93								
	Burner η	0.999								
	HPT NC Cooling Effect.					1.5	1.5			
	HPT Ch Cooling Effect.					0.8	0.8			
	LPT NC Cooling Effect.					1.5	1.5			
	HPT NC Cooling Effect.					1.2	1.2			
	Bypass Nozzle Cv	Function of PR								
	Bypass Nozzle CdTh	Function of PR								
	Core Nozzle Cv	Function of PR								
	Core Nozzle CdTh	Function of PR								
Technology Limits	T4 Max.					3275 to 3450 °R	3275 to 3450 °R		3275 to 3450 °R	3275 to 3450 °R
	HPT 1st Vane Temp.					2400 to 2500 °R	2400 to 2500 °R		2400 to 2500 °R	2400 to 2500 °R
	HPT Blade Temp.					2300 to 2400 °R	2300 to 2400 °R		2300 to 2400 °R	2300 to 2400 °R
	HPT Vane Temp.					2300 to 2400 °R	2300 to 2400 °R		2300 to 2400 °R	2300 to 2400 °R
	LPT Blade Temp.					2000 to 2100 °R	2000 to 2100 °R		2000 to 2100 °R	2000 to 2100 °R
	LPT Vane Temp.					2000 to 2100 °R	2000 to 2100 °R		2000 to 2100 °R	2000 to 2100 °R
	T3 Max.					1700 to 1850 °R	1700 to 1850 °R		1700 to 1850 °R	1700 to 1850 °R
	Min. LPCPR	1.15								
	Max. Fan Diameter	128 inches								
	Core-Bypass DP					550 psi	550 psi			
	Max Vmix							1050 ft/s		

The values of all of the design variables, performance requirements, component performance estimates and technology limits are summarized in Table 50. The last experiment contains a10,000 case LHC DoE of the design variables and technology parameters to create a well populated CTS. A successful test will show the practicality of densely populating the CTS with candidate cycle engines that meet the performance requirements and constraints specified at multiple design points. The experiment will show the ability to find solutions for all feasible candidate engine cycles for a CTS with a large number of design variables and technology parameters.

5.5.2 Experiment 5 Results

The final experiment is the ultimate test for the simultaneous MDP method that combines all of the knowledge obtained form the previous experiments and stresses the method by combining as many design points, requirements and constraints that could be devised for the LCT engine. The objective of the experiment is to test the simultaneous MDP method's ability to handle the increased dimensionality of the problem and still create a CTS saturated with candidate cycle engine in a robust and efficient manner. The DoE for this experiment consists of 10,000 cases to meet the objective of a saturated CTS. The results of the modified Newton Raphson solver performance are summarized in Table 51.

The outcomes of the 10,00 cases were as follows: 9908 category I outcomes, 0 category II outcomes, 84 category III outcomes and 8 category IV outcomes. Thus the solver successfully converged on 9992 of the candidate engine cycles with 84 being removed post analysis due to the infeasibility of the designs. The MDP repair algorithm was required for 245 of the 9992 converged cases. Of the 245 MDP repair algorithm cases, 89 converged after incrementing the design variables from the DoE values. For the cases that the modified Newton-Raphson converged on a solution in the first attempt, an average of 28 iterations, 1.64 full Jacobians and 216.9 model passes. The average iterations and full Jacobian did not appreciably change from the results for the constrained space 1 in experiment 2. Since there are more design points in experiment 5, the number of model passes increased due to the extra 30 equations in the Jacobian.

Table 51. Solver Summary Results for Experiment 5

Solver Metric	Value
Total Cases	10,000
Converged Cases	9992
Convergence Success Rate	99.92 %
Converged Cases [1st Attempt]	9747
Convergence Success Rate [1st Attempt]	97.55 %
Converged Cases [Repair]	245
Converged Cases [Repair]: Exact	156
Converged Cases [Repair]: Increment	89
Iterations [1st Attempt]: Mean	27.99
Iterations [1st Attempt]: StDev	13.51
Iterations [Repaired]: Mean	329.6
Iterations [Repaired]: StDev	135.2
Full Jacobians [1st Attempt]: Mean	1.643
Full Jacobians [1st Attempt]: StDev	1.722
Full Jacobians [Repaired]: Mean	68.6
Full Jacobians [Repaired]: StDev	30.8
Model Passes [1st Attempt]: Mean	216.9
Model Passes [1st Attempt]: StDev	206.12
Model Passes [Repaired]: Mean	8216
Model Passes [Repaired]: StDev	3644

For the fifth experiment the iteration and full Jacobian limits for convergence failure were set to 125 iterations and 20 full Jacobians. To understand the impact of the limits, the cumulative distribution of cases converged on the first attempt by the solver are shown in Figure 83 and Figure 84 for total iterations and total full Jacobians.

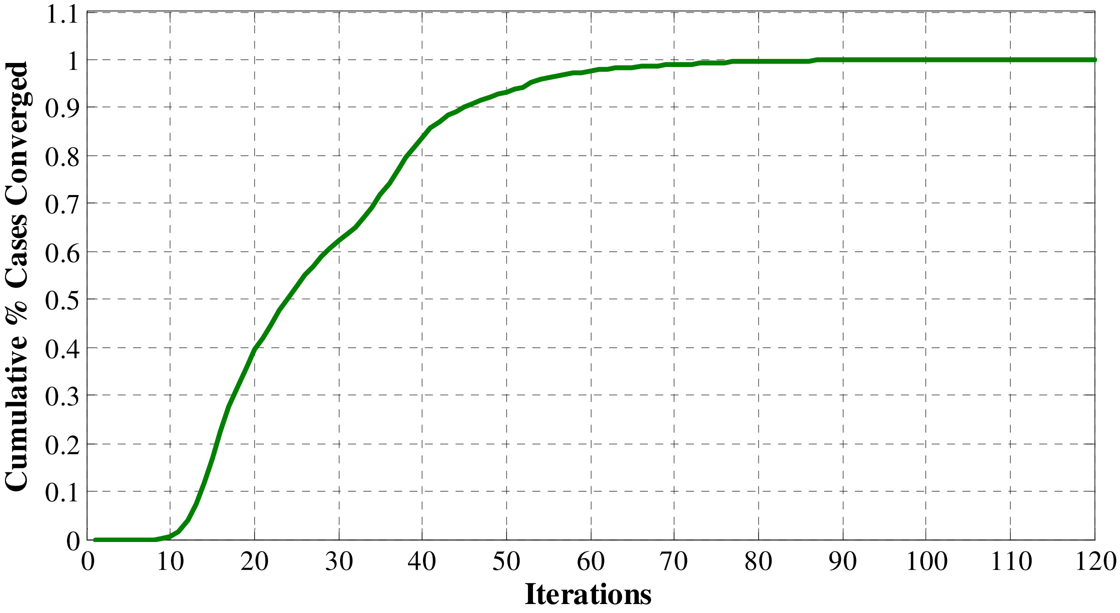


Figure 83. Cumulative Percentage of Converged Cases by Iteration

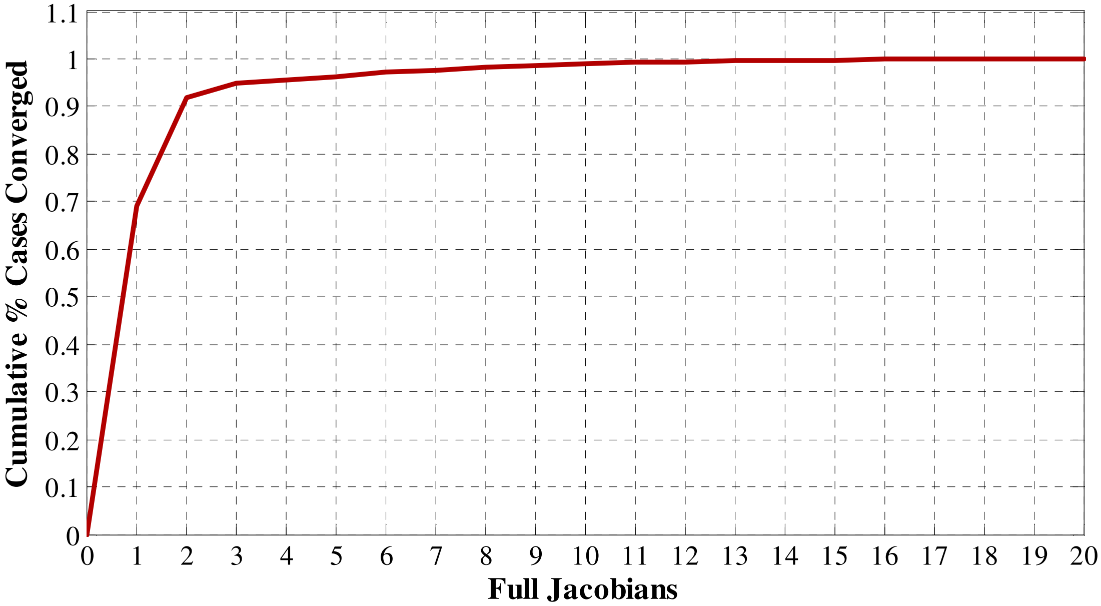


Figure 84. Cumulative Percentage of Converged Cases by Full Jacobian

Almost all of the cases converged within 50% of both limits. 98.1% of the cases converged in less than 63 total iterations while 99% of the cases converged with 10 or fewer full Jacobian calculations. An improvement in the solver efficiency could be made by reducing the limits, especially the full Jacobians. For this experiment, if the full Jacobian limit was reduced to 10, an additional 95 cases would initially fail to converge and be sent to the MDP repair algorithm for a total of 348 cases. The solver efficiency gains would occur in the MDP repair algorithm where steps that are going to fail would be ended sooner an not waste additional model passes.

The influence of the full Jacobian calculations on the total number of model passes can be seen in Figure 85. A clear separation is seen when the model passes are grouped by the number of Jacobians. A total of 6738 of the 10,000 cases required one full Jacobian calculation.

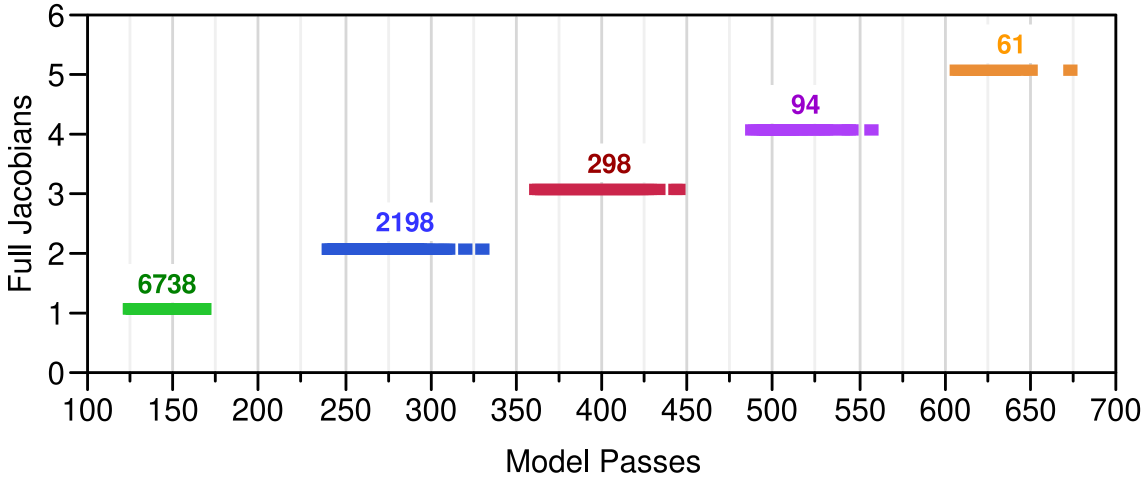
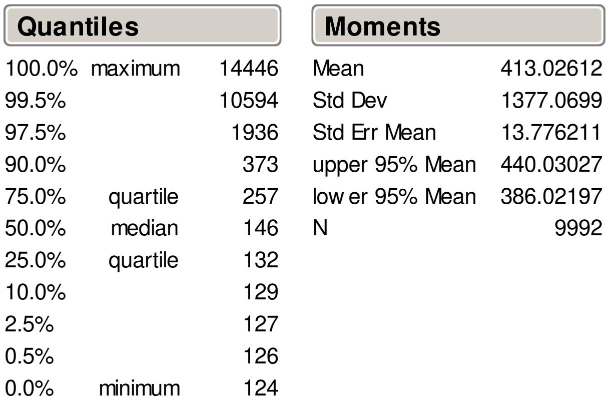


Figure 85. Variation of Model Passes for Experiment 5

The number of model passes for one Jacobian ranged from 124 to 173. Two full Jacobian calculations were required for 2198 of the remaining 3262 cases. The extra full Jacobian, which requires an additional 116 model passes to generate, caused the number of model passes to increase to a range of 230 to 330. The step in model passes for cases requiring 3, 4 and 5 full Jacobian calculations is shown in Figure 85 as well. A total of 93.8% of the cases required 5 or less full Jacobians. Four cases matched the maximum number of full Jacobians that converged on the first attempt with 19, ranging in the number of model passes from 2259 to 2296. The maximum number of full Jacobians used with the repair algorithm was 121. This case require 531 iterations and 14,446 model passes to converge, equivalent to the computational expense of 67 cases that converge in the first attempt.

The mean model pass count for all cases, shown in Table 52, was 413 or 3.59 model passes per independent-dependent combination. But this value is skewed upward by the 245 repair algorithm cases. A better statistic is the median model pass count which was 146, or 1.27 model passes per independent-dependent combination for experiment 5. In fact 90% of the cases required 373 or fewer model passes and 92% of the cases required the mean number of model passes or less to converge to the solution. These metrics again highlight the trade off between robustness and efficiency. Ninety percent of the space can be captured by the simultaneous MDP method very efficiently. Depending on the desired saturation of the design space and the available computation capability, modifications can be made to reduce the MDP repair algorithms robustness but improve its efficiency.

Table 52. Summary of Model Pass Results for Experiment 5



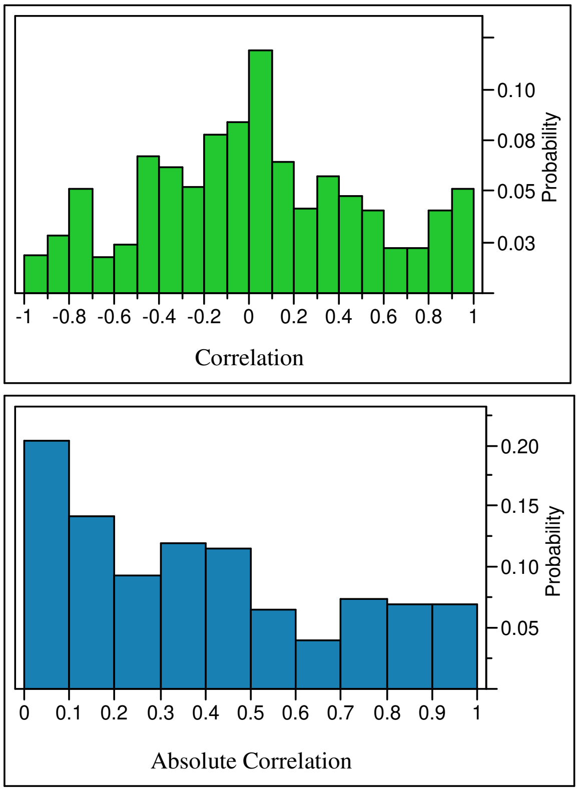
Quantiles		
100.0%	maximum	14446
99.5%		10594
97.5%		1936
90.0%		373
75.0%	quartile	257
50.0%	median	146
25.0%	quartile	132
10.0%		129
2.5%		127
0.5%		126
0.0%	minimum	124

Moments	
Mean	413.02612
Std Dev	1377.0699
Std Err Mean	13.776211
upper 95% Mean	440.03027
low er 95% Mean	386.02197
N	9992

5.5.2.1. Independent Parameter Solution Space for Experiment 5

The ranges for each of the 115 independent parameters within the solution space can be found in Table 79 through Table 87 in Appendix G. In experiment 3, where the design space was unconstrained and the design variables had a wider range than in experiment 5, the upper limit of the solution space was 85% above the initial iterate value and the lower limit on average was 35% below the initial iterate. With the constraints enabled resulting in a smaller cycle design space, the average limits of the solution space were 23% above the initial iterate value and -19% below. The change in size of the solution space did not change the correlation of the independent parameters.

As in experiment 3, the independent parameter solution space for experiment 5 is highly correlated. The distribution of independent parameter correlation and absolute correlation is shown in Figure 86. Fifty percent of the independent parameter solution space has an absolute correlation of 0.345 or greater, 25% of the independent parameter solution space has an absolute correlation of 0.62 or greater, while 10% of the solution space has an absolute correlation of 0.84.



Quantiles

100.0%	maximum	1.000
99.5%		1.000
97.5%		0.992
90.0%		0.777
75.0%	quartile	0.354
50.0%	median	0.000
25.0%	quartile	-0.334
10.0%		-0.707
2.5%		-0.863
0.5%		-0.962
0.0%	minimum	-0.999

Moments

Mean	0.0189603
Std Dev	0.494404
Std Err Mean	0.0061065
upper 95% Mean	0.0309311
lower 95% Mean	0.0069895
N	6555

Quantiles

100.0%	maximum	1.0000
99.5%		0.9997
97.5%		0.9920
90.0%		0.8430
75.0%	quartile	0.6203
50.0%	median	0.3451
25.0%	quartile	0.1204
10.0%		0.0423
2.5%		0.0000
0.5%		0.0000
0.0%	minimum	0.0000

Moments

Mean	0.3938886
Std Dev	0.299371
Std Err Mean	0.0036976
upper 95% Mean	0.4011372
lower 95% Mean	0.3866401
N	6555

Figure 86. Experiment 5 Independent Parameter Solution Space Correlation

5.5.2.2. Active Constraint Summary for Experiment 2

Sixteen constraints are included in the model for the fifth experiment. The probability of each of the constraints being active is shown in Table 53. Two of the constraints, SLSU_T4max and HHDTKO_Core_Bypass_dP were never activated in experiment 5. This is not an unusual result for constraints. Often it is not known at exactly which design point a constraint might be violated.

Table 53. Constraint Summary Results for Experiment 5

Constraint	Percent of Cases Constraint is Active
HHDTKO_HPT_BleedFlow1	43.6%
HHDTKO_HPT__BleedFlow2	48.3%
HHDTKO_LPT_BleedFlow1	56.0%
HHDTKO_LPT_BleedFlow2	56.8%
Max_Fan_Diam	16.9%
Cutback_Vmix	24.7%
TKO_T3	27.6%
HHDTKO_T3	2.2%
SLSI_T3	9.9%
LPCDP_LPCPRmin	3.8%
TKO_Core_Bypass_dP	4.3%
HHDTKO_Core_Bypass_dP	0.0%
HHDTKO_T4max	63.3%
SLSI_T4max	74.7%
SLSU_T4max	0.0%
TOC2_Fan_Nc	44.6%

It is better to add constraints at conditions that are never activated than to not constrain a design point and have it inadvertently exceed the constraint. Also, depending on the constraint values, certain constraints are more severe then others and will always constrain the design before the other is activated. The computational cost of unused constraints is minimal. A single model pass per full Jacobian calculation per extra constraint is all that is required. For the cases that only required one full Jacobian, an additional 2 model passes out of the 124-173 total model passes, ~1.3 %, were required to keep the two unused constraints

The first four constraints, the HPT and LPT bleed flows were active for nearly 50% of the cases. TKO was slightly more likely to set the HPT cooling while HHDTKO was slightly more likely to set the LPT cooling. The next two constraints, max fan diameter and maximum cutback Vmix constrained FPR as in experiment 2 with the max fan diameter setting the lower limit and the cutback Vmix establishing the upper limit. FPR was constrained for 41.6% of the cases, 16.9% for maximum fan diameter and 24.7% for cutback Vmix. The constrained FPR design space is shown in Figure 87.

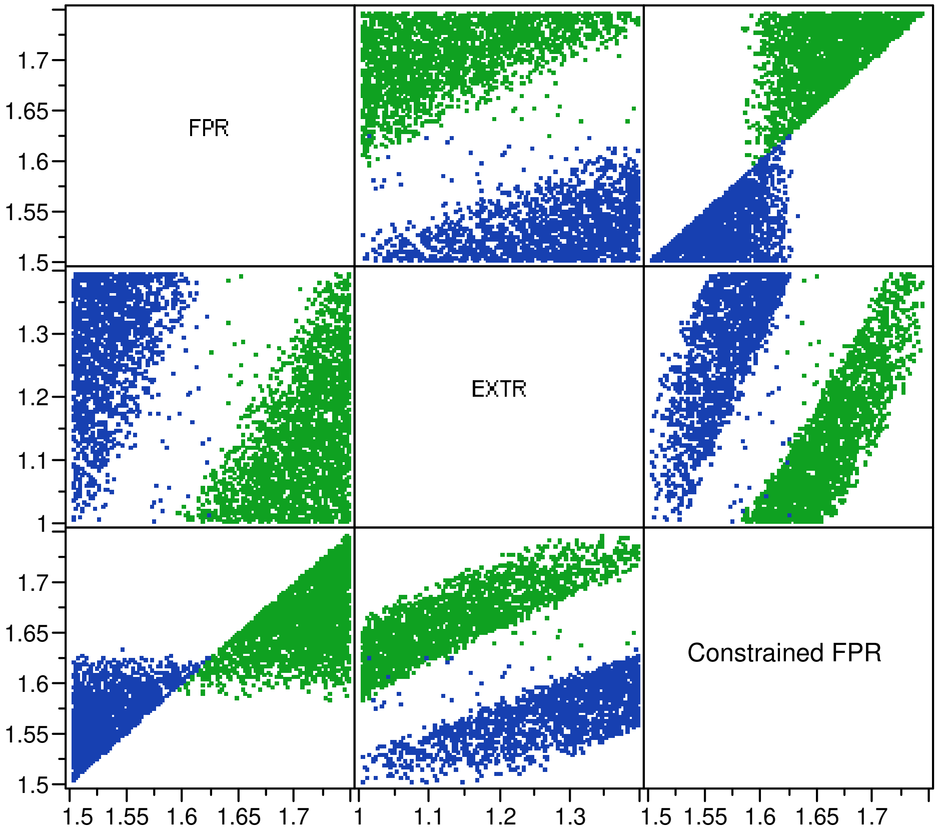


Figure 87. Multivariate Plot of Constrained FPR for Experiment 5

The blue data points represent the cases where FPR was raised by the solver to achieve a feasible fan diameter, while the green data points show the cases the solver lowered FPR to achieve a feasible cutback Vmix. The largest increase in FPR due to the fan diameter constraint is0.125 and the largest decrease due to cutback Vmix was -0.17.

Eight different constraints were included in the model which constrained OPR. Of the eight, only LPCDP_LPCPRmin, which was active for only 3.8 % of the cases, established a lower bound for OPR. The other seven constraints were active for 44% of the cases. The constrained OPR space, with the lower bound data points in blue and the upper bound data points in green, is shown in Figure 88. The maximum OPR rise from the solver was 4 and the maximum OPR drop was -14.

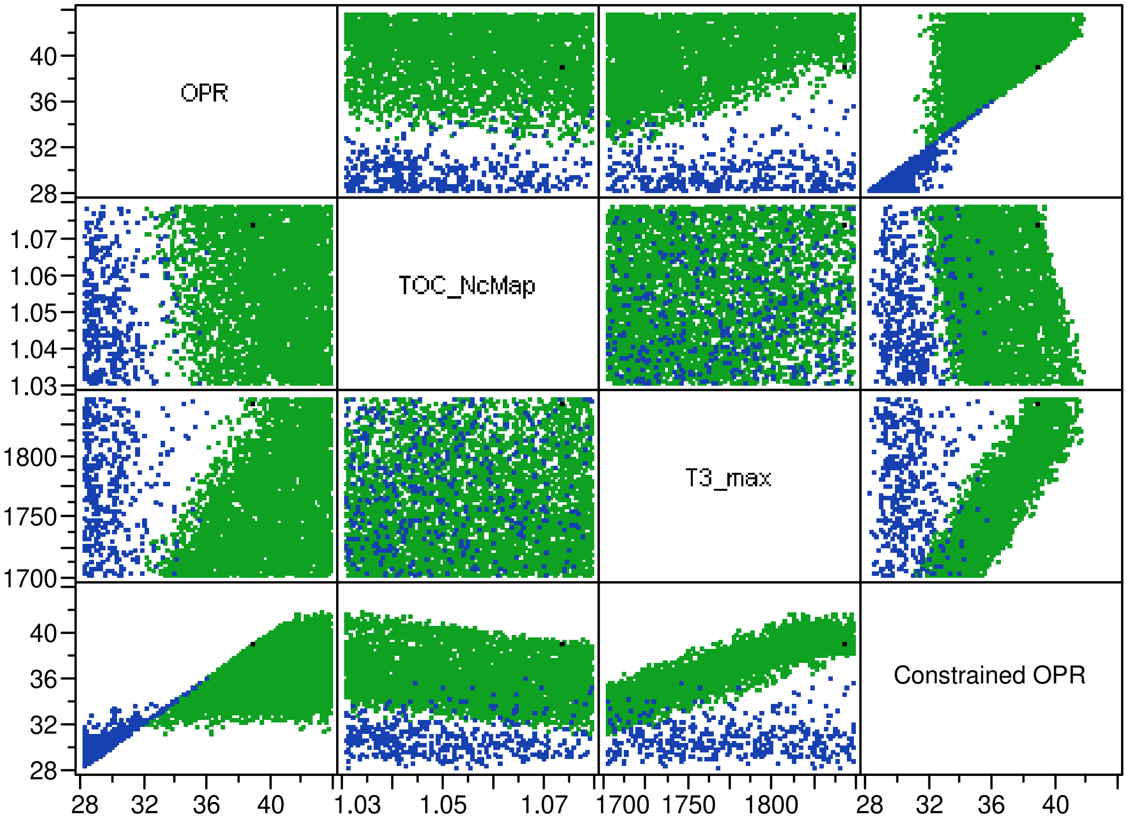


Figure 88. Multivariate Plot of Constrained OPR for Experiment 5

The next two constraints are related to the maximum T4, HHDTKO_T4max and SLSI_T4max. Both can be simultaneously active as each is paired to a different dependent. HHDTKO_T4max is paired to TKO_T4max while SLSI_T4max is paired with thrust at SLSI. The two constraints are active for 63% and 74% of the cases, respectively. For this design space, the design point HHDTKO was more likely to match peak combustor temperature than TKO. Also, the SLSI thrust could not match SLSU thrust for 74 % of the cases. The result is that the engine rated thrust for those cases is less than the thrust produced at SLSU 100% fan Nc. The final constraint, set the TOC_NcMap design variable to TOC2 for 44% of the cases and left it set at TOC1 for 56% of the cases.

The distribution of the number active constraints per case is shown in Figure 89. The maximum number of 9 active constraints was achieved for 260 of the cases. The minimum number of active constraints was achieved on 778 cases. Whereas in experiment 2, 16% of the cases were unconstrained, none of the cases for experiment 5 were unconstrained.

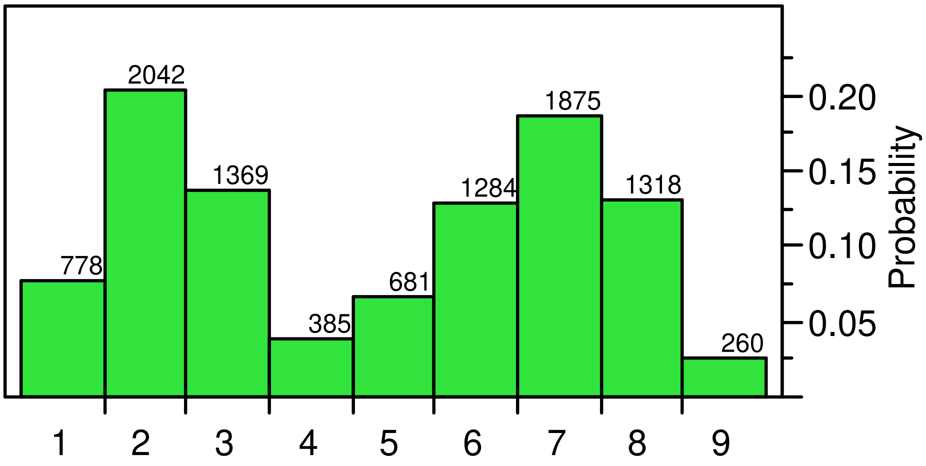


Figure 89. Histogram of Number of Active Constraints per Case for Experiment 5

Finally, to understand the relationship between active constraints and the design variables and technology parameters, the correlation matrix is summarized in Table 54. T4_max and the technology parameters, Fan_Deff, LPC_Deff, LPT_Deff, HPT_1VT, HPT_Temp, and LPT_Temp have no discernable correlation to the activation of the different constraints. The remaining design variables have similar influence as discussed in Chapter 2. Three additional constraints are included in Table 54 that were not included in experiment 2; SLSI_T3, SLSI_T4max, and TKO_Core_Bypass_dP. The HPC exit temperature at SLSI has little correlation to any of the design variables, but is most correlated with OPR. SLSI_T4 max, has a strong correlation to TOC_NcMap and weaker correlations to FPR and EXTR. The most probable scenario for activating SLSI_T4max is a low TOC_NcMap, low FPR and high EXTR. The core-bypass delta pressure at TKO constraint is weakly correlated to OPR and T3_Max. High values of OPR and T3_max provide the most likely scenario for TKO_Core_Bypass_dP activation.

Table 54. Active Constraint Correlation to Design Variables and Technology Parameters for Experiment 5

	FPR	OPR	EXTR	T4_Max	TOC_NcMap	T3_Max	Fan_Deff	LPC_Deff	LPT_Deff	HPT_1VT	HPT_Temp	LPT_Temp
HHDTKO_HPT_BleedFlow1	**0.38**	-0.02	**-0.43**	0.08	**0.54**	-0.03	0.01	-0.01	0.01	0.04	0.02	0.02
HHDTKO_HPT__BleedFlow2	**0.40**	-0.03	**-0.44**	0.07	**0.53**	-0.03	0.00	-0.01	0.00	0.01	0.05	0.02
HHDTKO_LPT_BleedFlow1	**0.43**	-0.03	**-0.46**	0.04	**0.49**	-0.03	-0.01	0.00	-0.01	0.00	0.01	0.03
HHDTKO_LPT_BleedFlow2	**0.42**	-0.03	**-0.48**	0.03	**0.48**	-0.03	-0.01	0.00	-0.01	0.00	0.01	0.03
Max_Fan_Diam	**-0.57**	-0.01	0.28	-0.01	-0.20	-0.01	0.02	0.00	0.01	0.00	0.00	0.00
HHDTKO_T4max	**0.40**	-0.06	**-0.48**	0.06	**0.48**	-0.03	0.00	0.00	-0.01	0.01	0.01	0.01
TKO_T3	-0.17	**0.53**	0.17	-0.01	-0.09	**-0.29**	-0.01	0.01	0.00	0.01	0.01	0.01
HHDTKO_T3	0.10	0.12	-0.16	0.09	0.18	-0.07	0.02	-0.02	0.02	0.02	0.04	0.01
Cutback_Vmix	**0.65**	0.03	**-0.32**	0.02	0.17	0.03	0.00	0.00	0.00	0.00	0.00	0.00
TOC2_Fan_Nc	**0.82**	-0.01	-0.17	0.01	0.10	-0.01	0.00	0.01	0.01	0.00	0.01	-0.01
LPCDP_LPCPRmin	0.07	**-0.28**	0.15	0.17	-0.08	0.00	0.08	0.00	0.06	0.04	0.07	0.05
SLSI_T3	0.09	**0.27**	-0.11	0.07	0.10	-0.19	0.01	0.00	0.01	0.02	0.03	0.03
SLSI_T4max	**-0.33**	-0.01	**0.36**	-0.03	**-0.54**	0.01	-0.01	0.01	0.01	-0.01	-0.01	0.00
TKO_Core_Bypass_dP	0.09	**0.27**	-0.08	-0.03	0.08	**0.28**	0.00	-0.01	-0.02	-0.02	-0.03	-0.03

The fifth experiment demonstrates the ultimate test of the LCT engine for the simultaneous MDP method. A total of nine design points, 15 constraints, and a system of

115 nonlinear equations are modeled. A DoE consisting of 10,000 candidate engines is developed with five design variables and seven technology parameters to create a saturated CTS. The simultaneous MDP method found solutions for 99.92 % of the candidate engine designs with 92% of the cases requiring less than 413 model passes or 3.59 model passes per independent-dependent combination . The MDP repair algorithm was utilized for 245 of the successfully converged cases. Between one and nine constraints are activated for each case. There is no correlation between the technology parameters and the performance of the modified Newton-Raphson method in terms of constraint activation, robustness or efficiency.

CHAPTER 6

CONCLUSIONS

This work develops a method for designing aerothermodynamic gas turbine engine cycles that simultaneously meet requirements and constraints at multiple design points. The resulting cycle design space would only contain feasible candidate engines from which the “best” engine could be identified during the cycle selection process depending on the cycle analysts selection criteria. The final chapter of this thesis is divided into four sections. The first section resolves the research questions and hypothesis presented in Chapter 3 including the overall hypothesis for this work. The next section summarizes the contributions of this work to the aerospace engineering community with regards to improvements in ability to generate feasible aerothermodynamic cycle. The third section discusses recommendations for future improvements to the simultaneous MDP method. The fourth section contains some final remarks from the author regarding considerations for other gas turbine applications other than a LCT engine

6.1.1 Research Questions and Hypotheses

The primary research question of this thesis was how can an on-design cycle analysis method be developed that incorporates all desired cycle performance requirements and constraints at different design point and generates aerothermodynamic cycles in a robust and efficient manner to populate the cycle design space? The overall hypothesis stated that a method that satisfies the performance requirements and constraints for all design points simultaneously would allow the cycle analyst to include as many design points as necessary to control all desired aspects of the aerothermodynamic cycle and would be robust and efficient enough to enable a comprehensive exploration of the cycle design space. The overall hypothesis was confirmed by the fifth experiment which created a CDS around the LCT engine with over 25 requirements and constraints at nine different design points. A population of 9992 feasible candidate cycle engines were successfully designed out of 10,000 attempted engines providing a comprehensive exploration of the CDS.

The first research question asked what the impacts were to the CDS as a result of using an MDP method versus the traditional SPD cycle analysis approach? The hypothesis is that the topography and feasible boundaries of the CDS are altered between the MDP and SPD methods. This hypothesis was shown to be true in the first experiment by direct comparison of CDSs created by the two methods. The results show that due to the assumptions required by the SPD method, the SPD candidate cycle engines are invariably oversized and suffer a performance penalty compared to their MDP counterparts. Furthermore, the feasible area of the SPD created cycle design space will always be less than the MDP CDS and may not contain the "optimum" candidate cycle engine.

The second research question asked how a cycle analysis approach can account for multiple design points simultaneously. The hypothesis stated that the solution to a system of nonlinear equations formulated at on-design mode from design rules that couple the design points, performance requirements, technology rules and design variables will simultaneously meet multiple design points. This hypothesis was proven to be true in the second experiment which showed how a modified Newton-Raphson solver could be utilized to find the solution to a set of nonlinear equation. These equations were formulated in the setup for experiment 2 discussed in Chapter 4. The inclusion of constraints provided a benefit to the solver by limiting the size of the cycle design space to a feasible region which in turn reduced the size of the independent parameter solution space preventing the need for the solver to find extraneous solutions.

The final research question asked about the robustness of the simultaneous MDP method against a wide range of design variables and altering technology levels and the impact on the solver's efficiency to ensure a high level of robustness? The hypothesis stated that the simultaneous MDP can find a solution for candidate engine cycles having a wide range of design variables and technology levels while maintaining a high level of efficiency. The proof for this hypothesis was supplied by experiments 3 and 4. In experiment three it was found that with the proper settings, the modified Newton-Raphson solver could find a high percentage of the candidate cycle engines efficiently. The solution to feasible candidate cycle engines which initially elude the solver could be found using the MDP repair algorithm. Experiment 4 added various technology parameters to modify the level of technology with no impact to the solvers ability to find the feasible solution.

6.1.2 Summary of Contributions to Aerospace Engineering

The principal contributions from the thesis are:

- The development of the simultaneous MDP cycle analysis method to convert design rules into a system of nonlinear equations by identifying a single independent parameter for every user defined requirement and constraint and coupling them with the engine match independent-dependent combinations that ensure the engine obeys the laws of continuity and energy conservation at all design points. The method identifies the most stringent design point operating conditions and applies the appropriate performance requirements and constraints resulting in a feasible cycle design space.
- The development of a postulate regarding the initial iterate that states it must be a solution to a candidate cycle engine within the CDS. A process was devised to systematically obtain an initial iterate that adheres to the postulate.
- The development of a repair algorithm to resolve candidate engine cycles that initially fail to converge in the solver. The repair algorithm accounts for cases that

fail due to the initial iterate not being close enough to the final solution and cases that culminate in an infinite loop.

The simultaneous MDP methods robustness and efficiency allow for a large number of candidate engines to comprehensively populate the cycle design space

6.1.3 Recommendations for Future Development

The most computationally expensive step of the modified Newton-Raphson solver in the simultaneous MDP method is the calculation of the full Jacobian. The full Jacobian requires n+1 model passes to calculate, where n is the number of nonlinear equations to be solved. All possible steps were taken to reduce this effect by limiting the number of full Jacobians generated. However, for experiment 5, 70% to 90% of all model passes were still devoted to the full generation of Jacobian matrices. There would be great value in reducing the computational expense of the full Jacobian. Fortunately, the structure of the Jacobian, reproduced here from Figure 65, provides a possibility for further improvement in efficiency.

$\frac{\partial F^{LPCDP}}{\partial X^{LPCDP}}$	0	0	0	0	0	0
$\frac{\partial F^{HPCDP}}{\partial X^{LPCDP}}$	$\frac{\partial F^{HPCDP}}{\partial X^{HPCDP}}$	0	0	0	0	0
$\frac{\partial F^{TOC1}}{\partial X^{LPCDP}}$	0	$\frac{\partial F^{TOC1}}{\partial X^{TOC1}}$	0	0	0	0
$\frac{\partial F^{TOC2}}{\partial X^{LPCDP}}$	0	0	$\frac{\partial F^{TOC2}}{\partial X^{TOC2}}$	0	0	0
$\frac{\partial F^{TKO}}{\partial X^{LPCDP}}$	0	0	0	$\frac{\partial F^{TKO}}{\partial X^{TKO}}$	0	0
$\frac{\partial F^{HHDTKO}}{\partial X^{LPCDP}}$	0	0	0	0	$\frac{\partial F^{HHDTKO}}{\partial X^{HHDTKO}}$	0
$\frac{\partial F^{Cutback}}{\partial X^{LPCDP}}$	0	0	0	0	0	$\frac{\partial F^{Cutback}}{\partial X^{Cutback}}$

Figure 65. Jacobian Matrix Showing Populated and Unpopulated Submatrices

A model pass consists of many sets of calculations. These calculations are repeated for each design point or NPSS assembly. If the number of design points in the model is d, then the number of design point calculations required to calculate the full Jacobian is d*(n+1), that is one set of d calculations at the base value of the independents plus a perturbation of each of the independents for every design point. For experiment 5 with 9 design points and 115 nonlinear equations, 1044 design point calculations are required for every full Jacobian calculation. Looking at the structure of the Jacobian, many of the design point calculations are wasted computations as the partial derivatives for the dependents are already known to be zero for certain independent parameters. A scheme that takes advantage of this knowledge could significantly reduce the computational expense of a full Jacobian calculation.

Let z be the number of independents based at the map scaling point. Then using the known structure of the Jacobian, the calculations required to calibrate the full Jacobian are one set of d calculations at the base value of the independents, plus the number of map scaling independent parameters at each of the design points (d*z) plus a single calculation for each of the remaining independents at its corresponding design point (n-z). Thus the total number of design point calculations required to generate the full Jacobian becomes d+(d*z)+(n-z). For experiment 5 there were a total of 35 independents for LPCDP, the map scaling point. The number of design point calculations following the new scheme would be 9 + (9 * 35) +(115-35) = 404, a reduction for calculating the full Jacobian of 61.3%. As a disproportionate number of the model passes are for the full Jacobian, such a reduction could reduce the overall computational expense by 50%.

6.1.4 Final Remarks Regarding Other Gas Turbine Applications

This thesis used a large commercial transport engine to demonstrate the capabilities of the simultaneous MDP method for generating feasible design spaces. This method would be ideally suited for engines with even more requirements and design points to consider. Two applications that would generate addition requirements and design points include military fighter engines and high speed civil transport engines. Both of theses applications would add additional design points at supersonic speeds. Often such engines employ variable cycles that would have to be integrated into the modified Newton-Raphson solver. These applications would represent the next challenge for the simultaneous MDP method.

APPENDIX A

TERMS AND DEFINITIONS

Candidate Engine Cycle: A single engine cycle in the cycle design space characterized by a specific combination of cycle design variables that meet the cycle design space criteria.

Component Performance estimates: Functions, tables or maps which estimate component technology parameters for a given technology level as a function of design variables.

Component Technology Parameters: The set of parameters that define the performance of an individual engine component

Cycle Analysis: Process of determining thermodynamic performance for a particular engine as defined by a set of cycle design variables and technology level while meeting all of the cycle performance requirements.

Cycle Analyst: Person or person responsible for performing cycle analysis.

Cycle Constraints: Limits placed on the cycle which may not be exceeded. The constraints are composed of the technology limits and performance limits. Constraints may be specified for a specific operating condition or apply to all operating conditions.

Cycle Design Space: The space created by executing cycle analysis while parametrically varying the cycle design variables to create candidate engine cycles.

Cycle Design Space Criteria: The criteria necessary to construct a consistent and feasible design space, namely consistent technology level and consistent performance capability.

Cycle Model: A mathematical representation of the aerothermodynamics of the internal flow within the engine

Cycle Model Pass: A single execution through the cycle model by the cycle analysis program. Cycle model pass is the cycle model equivalent to a function call.

Cycle On-Design Analysis: A process to generate an engine sized to meet the cycle performance requirements at the design point based on technology rules and values of design variables.

Cycle Off-Design Analysis: A process to evaluate aerothermodynamic performance of a fixed engine cycle as conditions change within the operational envelope.

Cycle Performance Requirement: Specific requisites defining the expected performance characteristics of the cycle throughout the operational envelope that, along with the technology rules, shape and constrain the cycle design space. Cycle performance requirements can be divided into two categories, performance targets (equality constraints) and performance limits (inequality constraints).

Cycle Selection: A process of choosing from a cycle design space the “best” engine or engines to maximize mission figures of merit.

Design Variables: Independent cycle parameters which the cycle analyst has complete authority to set. Design variables are parametrically varied in cycle on-design analysis and chosen in the cycle selection process.

Design Rules: Rules that link the design points, performance requirements, technology rules and design variables to form a system of nonlinear equations.

Design Point: An engine operating condition which establishes some portion of the engine cycle performance.

Figure of Merit (FOM): Overall engine/aircraft performance characteristics used to evaluate competing engine cycle designs which satisfy the performance requirements in order to selects the best combination of design variables for a particular mission.

MDP Robustness: The ability to converge to the solution for all possible candidate engine cycles within the cycle design space.

MDP Efficiency: The ratio of the number of cycle model passes required to find a solution to the number of equations solved.

Operating Condition: The external environment in which the engine is operating. The operating condition is specified by the atmospheric properties such as pressure, temperature, density, and humidity and the flight Mach number.

Operational Envelope: The combination of all operating conditions and throttle settings within which the engine would be expected to operate.

Parametric Engine Cycle Design: Overall process which evaluates multiple candidate engine cycles for a particular aircraft/mission. Process can be divided into two sub processes, cycle analysis and cycle selection.

Performance Capability: The set of cycle performance requirements.

Sequential Single Point Design (Sequential SPD): An iterative method that designs the cycle and sizes the engine at a single design point like SPD, but iterates on certain cycle design variables to meet performance requirements at other operating conditions.

Simultaneous Multi-Design Point (MDP): An iterative method that simultaneously designs the cycle and sizes the engine at multiple design points.

Single Point Design (SPD): Method that designs the cycle and sizes the engine at a single design point in a single pass without regard to performance requirements at other operating conditions.

Technology Level: A set of capabilities defined for each engine component. Assigns values to the component technology parameters.

Technology Limits: Constraints established by the technology level which can not be exceeded. Constraining values for differing technology limits can be found at a variety of different operating conditions.

Technology Rules: Rules to describe how the technology parameters change as a function of the design variables to maintain a consistent technology level that, along with the performance requirements, shape and constrain the cycle design space. They can be divided into two categories, component performance estimates and technology limits.

APPENDIX B

SOLVER INDEPENDENT, DEPENDNT AND CONSTRAINT DEFINITION

The following tables define the solver dependents, user defined independents and constraints used in Experiments 2-5. The dependents, listed in each table by their design point, are defined by the dependent name and the left hand side (LHS) and right hand side (RHS) terms. The LHS and RHS terms must match within the specified tolerance for convergence. The LHS and RHS terms can be variables or equations. The constraints are defined in the same manner as the dependents parameters with an name, LHS and RHS terms. The user defined independent parameters are defined by the independent name and its corresponding NPSS variable name.

Table 55. Dependent Parameter Definition for LPCDP

Dependent Name	Dependent LHS	Dependent RHS
LPCDP_FPR_Dep	LPCDP.Fan.Fl_O.Pt/LPCDP.Fan.Fl_I.Pt	FPR
LPCDP_OPR	LPCDP.PERF.myOPR	OPR
LPCDP_Extr_Ratio	LPCDP.Byp_Nozz.Fl_I.Pt/LPCDP.Core_Nozz.Fl_I.Pt	EXTR
LPCDP_Fan_Spec_Flow	144.0*LPCDP.Fan.Wc/LPCDP.Inlet.Fl_O.Aphy	45.0
LPCDP_LPC_Spec_Flow	144.0*LPCDP.LPC.Wc/LPCDP.Duct4.Fl_O.Aphy	26.5
LPCDP_LPCEff_Error	LPCDP.LPC.S_map.effDes/LPC_Eff_Eq	1.0
LPCDP_HPNmech_Error	LPCDP.HP_Shaft.Nmech	LPC_HP_Nmech_Eq
LPCDP_LPNmech_Error	LPCDP.LP_Shaft.Nmech	LPC_LP_Nmech_Eq
SPL_Core_Area	LPCDP.Splitter.Fl_01.Aphy/LPCDP.Duct4.Fl_O.Aphy	1.0
SPL_Byp_Area	LPCDP.Splitter.Fl_01.Aphy+LPCDP.Splitter.Fl_02.Aphy	LPCDP.Fan.Fl_O.Aphy
Fan_Area_Out	LPCDP.Fan.Fl_O.Aphy/LPCDP.Fan.Fl_I.Aphy	0.9
LPC_Area_Out	LPCDP.LPC.Fl_O.Aphy/LPCDP.LPC.Fl_I.Aphy	0.75
LPT_Flow_Coef	LPT_Flow_Coef_Eq1	LPT_Flow_Coef_Eq2
Duct13_Area_Out	LPCDP.Duct13.Fl_O.Aphy/LPCDP.LPT.Fl_O.Aphy	0.95
Duct15_Area_Out	LPCDP.Duct15.Fl_O.Aphy/LPCDP.Splitter.Fl_02.Aphy	1.0
BypBld_Area_Out	LPCDP.BypBld.Fl_O.Aphy/LPCDP.Splitter.Fl_02.Aphy	1.0
LPCDP.HP_Shaft.int_Nmech	LPCDP.HP_Shaft.trqNet	0.0
LPCDP.LP_Shaft.int_Nmech	LPCDP.LP_Shaft.trqNet	0.0

Table 56. Dependent Parameter Definition for HPCDP

Dependent Name	Dependent LHS	Dependent RHS
HPCDP_HPC_Nc	HPCDP.HPC.S_map.NcMap	From Table
HPCDP_HPCPR	HPCDP.HPC.PR	From Table
HPCDP_HPC_Wc	HPCDP.HPC.Wc	From Table
HPCDP_HPCEff	HPCDP.HPC.eff	From Table
HPCDP_HPC_Rline	HPCDP.HPC.S_map.RlineMap	2.0
HPCDP_HPC_InletArea	HPCDP.Duct6.Fl_O.Aphy	875
HPCDP_HPC_ExitArea	HPCDP.HPC.Fl_O.Aphy	100
HPCDP_HPT_InletArea	HPCDP.Burner.Fl_O.Aphy	425
HPCDP_HPT_ExitArea	HPCDP.HPT.Fl_O.Aphy	700
HPCDP_HPT_s_effDes	HPCDP.HPT.S_map.s_eff	1.0
HPCDP_HPT_s_WcDes	HPCDP.HPT.S_map.s_Wp	1.0
HPCDP_HPT_s_PRdes	HPCDP.HPT.S_map.s_dPqP	1.0
HPCDP.Core_Nozz.Area	HPCDP.Core_Nozz.WqAE	HPCDP.Core_Nozz.WqAEdem
HPCDP.Byp_Nozz.Area	HPCDP.Byp_Nozz.WqAE	HPCDP.Byp_Nozz.WqAEdem
HPCDP.Fan.S_map.errWc	HPCDP.Fan.S_map.Wc	HPCDP.Fan.S_map.WcCalc
HPCDP.LPC.S_map.errWc	HPCDP.LPC.S_map.Wc	HPCDP.LPC.S_map.WcCalc
HPCDP.HPC.S_map.errWc	HPCDP.LPC.S_map.Wc	HPCDP.LPC.S_map.WcCalc
HPCDP.HPT.S_map.errWp	HPCDP.HPT.S_map.WpIn	HPCDP.HPT.S_map.WpOut
HPCDP.LPT.S_map.errWp	HPCDP.LPT.S_map.WpIn	HPCDP.LPT.S_map.WpOut
HPCDP.HP_Shaft.int_Nmech	HPCDP.HP_Shaft.trqNet	0.0
HPCDP.LP_Shaft.int_Nmech	HPCDP.LP_Shaft.trqNet	0.0

Table 57. Dependent Parameter Definition for TOC 1 Design Point

Dependent Name	Dependent LHS	Dependent RHS
TOC1_Thrust	TOC1.PERF.myFn	16025
TOC1_Fan_Nc	TOC1.Fan.S_map.NcMap	TOC_NcMap
TOC1.Core_Nozz.Area	TOC1.Core_Nozz.WqAE	TOC1.Core_Nozz.WqAEdem
TOC1.Byp_Nozz.Area	TOC1.Byp_Nozz.WqAE	TOC1.Byp_Nozz.WqAEdem
TOC1.Fan.S_map.errWc	TOC1.Fan.S_map.Wc	TOC1.Fan.S_map.WcCalc
TOC1.LPC.S_map.errWc	TOC1.LPC.S_map.Wc	TOC1.LPC.S_map.WcCalc
TOC1.HPC.S_map.errWc	TOC1.LPC.S_map.Wc	TOC1.LPC.S_map.WcCalc
TOC1.HPT.S_map.errWp	TOC1.HPT.S_map.WpIn	TOC1.HPT.S_map.WpOut
TOC1.LPT.S_map.errWp	TOC1.LPT.S_map.WpIn	TOC1.LPT.S_map.WpOut
TOC1.HP_Shaft.int_Nmech	TOC1.HP_Shaft.trqNet	0.0
TOC1.LP_Shaft.int_Nmech	TOC1.LP_Shaft.trqNet	0.0

Table 58. Dependent Parameter Definition for TOC 2 Design Point

Dependent Name	Dependent LHS	Dependent RHS
TOC2_Thrust	TOC2.PERF.myFn	19600
TOC2.Core_Nozz.Area	TOC2.Core_Nozz.WqAE	TOC2.Core_Nozz.WqAEdem
TOC2.Byp_Nozz.Area	TOC2.Byp_Nozz.WqAE	TOC2.Byp_Nozz.WqAEdem
TOC2.Fan.S_map.errWc	TOC2.Fan.S_map.Wc	TOC2.Fan.S_map.WcCalc
TOC2.LPC.S_map.errWc	TOC2.LPC.S_map.Wc	TOC2.LPC.S_map.WcCalc
TOC2.HPC.S_map.errWc	TOC2.LPC.S_map.Wc	TOC2.LPC.S_map.WcCalc
TOC2.HPT.S_map.errWp	TOC2.HPT.S_map.WpIn	TOC2.HPT.S_map.WpOut
TOC2.LPT.S_map.errWp	TOC2.LPT.S_map.WpIn	TOC2.LPT.S_map.WpOut
TOC2.HP_Shaft.int_Nmech	TOC2.HP_Shaft.trqNet	0.0
TOC2.LP_Shaft.int_Nmech	TOC2.LP_Shaft.trqNet	0.0

Table 59. Dependent Parameter Definition for TKO Design Point

Dependent Name	Dependent LHS	Dependent RHS
TKO_Thrust	TKO.PERF.myFn	75,000
TKO_T4max	TKO.Burner.TtCombOut	T4_Max
TKO.HPT.Cool.BleedFlow1	TKO.HPT.Cool.WbleedC1	TKO.HPT.Cool.WbleedIn1
TKO.HPT.Cool.BleedFlow2	TKO.HPT.Cool.WbleedC2	TKO.HPT.Cool.WbleedIn2
TKO.LPT.Cool.BleedFlow1	TKO.LPT.Cool.WbleedC1	TKO.LPT.Cool.WbleedIn1
TKO.LPT.Cool.BleedFlow2	TKO.LPT.Cool.WbleedC2	TKO.LPT.Cool.WbleedIn2
TKO.Core_Nozz.Area	TKO.Core_Nozz.WqAE	TKO.Core_Nozz.WqAEdem
TKO.Byp_Nozz.Area	TKO.Byp_Nozz.WqAE	TKO.Byp_Nozz.WqAEdem
TKO.Fan.S_map.errWc	TKO.Fan.S_map.Wc	TKO.Fan.S_map.WcCalc
TKO.LPC.S_map.errWc	TKO.LPC.S_map.Wc	TKO.LPC.S_map.WcCalc
TKO.HPC.S_map.errWc	TKO.LPC.S_map.Wc	TKO.LPC.S_map.WcCalc
TKO.HPT.S_map.errWp	TKO.HPT.S_map.WpIn	TKO.HPT.S_map.WpOut
TKO.LPT.S_map.errWp	TKO.LPT.S_map.WpIn	TKO.LPT.S_map.WpOut
TKO.HP_Shaft.int_Nmech	TKO.HP_Shaft.trqNet	0.0
TKO.LP_Shaft.int_Nmech	TKO.LP_Shaft.trqNet	0.0

Table 60. Dependent Parameter Definition HHDTKO Design Point

Dependent Name	Dependent LHS	Dependent RHS
HHDTKO_Thrust	HHDTKO.PERF.myFn	60,000
HHDTKO.Core_Nozz.Area	HHDTKO.Core_Nozz.WqAE	HHDTKO.Core_Nozz.WqAEdem
HHDTKO.Byp_Nozz.Area	HHDTKO.Byp_Nozz.WqAE	HHDTKO.Byp_Nozz.WqAEdem
HHDTKO.Fan.S_map.errWc	HHDTKO.Fan.S_map.Wc	HHDTKO.Fan.S_map.WcCalc
HHDTKO.LPC.S_map.errWc	HHDTKO.LPC.S_map.Wc	HHDTKO.LPC.S_map.WcCalc
HHDTKO.HPC.S_map.errWc	HHDTKO.LPC.S_map.Wc	HHDTKO.LPC.S_map.WcCalc
HHDTKO.HPT.S_map.errWp	HHDTKO.HPT.S_map.WpIn	HHDTKO.HPT.S_map.WpOut
HHDTKO.LPT.S_map.errWp	HHDTKO.LPT.S_map.WpIn	HHDTKO.LPT.S_map.WpOut
HHDTKO.HP_Shaft.int_Nmech	HHDTKO.HP_Shaft.trqNet	0.0
HHDTKO.LP_Shaft.int_Nmech	HHDTKO.LP_Shaft.trqNet	0.0

Table 61. Dependent Parameter Definition Cutback Design Point

Dependent Name	Dependent LHS	Dependent RHS
Cutback _Thrust	Cutback.PERF.myFn	60,000
Cutback.Core_Nozz.Area	Cutback.Core_Nozz.WqAE	Cutback.Core_Nozz.WqAEdem
Cutback.Byp_Nozz.Area	Cutback.Byp_Nozz.WqAE	Cutback.Byp_Nozz.WqAEdem
Cutback.Fan.S_map.errWc	Cutback.Fan.S_map.Wc	Cutback.Fan.S_map.WcCalc
Cutback.LPC.S_map.errWc	Cutback.LPC.S_map.Wc	Cutback.LPC.S_map.WcCalc
Cutback.HPC.S_map.errWc	Cutback.LPC.S_map.Wc	Cutback.LPC.S_map.WcCalc
Cutback.HPT.S_map.errWp	Cutback.HPT.S_map.WpIn	Cutback.HPT.S_map.WpOut
Cutback.LPT.S_map.errWp	Cutback.LPT.S_map.WpIn	Cutback.LPT.S_map.WpOut
Cutback.HP_Shaft.int_Nmech	Cutback.HP_Shaft.trqNet	0.0
Cutback.LP_Shaft.int_Nmech	Cutback.LP_Shaft.trqNet	0.0

Table 62. Dependent Parameter Definition for SLSI Design Point

Dependent Name	Dependent LHS	Dependent RHS
SLSI_Thrust	SLSI.PERF.myFn	SLSU.PERF.myFn
SLSI.Core_Nozz.Area	SLSI.Core_Nozz.WqAE	SLSI.Core_Nozz.WqAEdem
SLSI.Byp_Nozz.Area	SLSI.Byp_Nozz.WqAE	SLSI.Byp_Nozz.WqAEdem
SLSI.Fan.S_map.errWc	SLSI.Fan.S_map.Wc	SLSI.Fan.S_map.WcCalc
SLSI.LPC.S_map.errWc	SLSI.LPC.S_map.Wc	SLSI.LPC.S_map.WcCalc
SLSI.HPC.S_map.errWc	SLSI.LPC.S_map.Wc	SLSI.LPC.S_map.WcCalc
SLSI.HPT.S_map.errWp	SLSI.HPT.S_map.WpIn	SLSI.HPT.S_map.WpOut
SLSI.LPT.S_map.errWp	SLSI.LPT.S_map.WpIn	SLSI.LPT.S_map.WpOut
SLSI.HP_Shaft.int_Nmech	SLSI.HP_Shaft.trqNet	0.0
SLSI.LP_Shaft.int_Nmech	SLSI.LP_Shaft.trqNet	0.0

Table 63. User Independent Parameter Definition

Name	NPSS Variable		Name	NPSS Variable
LPCDP_FPR	LPCDP.Fan.S_map.PRdes		LPCDP_HPT_InletMN	LPCDP.Burner.Fl_O.MN
LPCDP_LPCPR	LPCDP.LPC.S_map.PRdes		LPCDP_HPT_ExitMN	LPCDP.HPT.Fl_O.MN
LPCDP_HPCPR	LPCDP.HPC.S_map.PRdes		LPCDP_SPL_Core_MN	LPCDP.Splitter.Fl_01.MN
LPCDP_HPT_PRDes	LPCDP.HPT.S_map.parmMapDes		LPCDP_SPL_Byp_MN	LPCDP.Splitter.Fl_02.MN
LPCDP_BPR	LPCDP.Splitter.BPR		LPCDP_Duct11_Out_MN	LPCDP.Duct11.Fl_O.MN
LPCDP_W	LPCDP.Ambient.W		LPCDP_Duct13_Out_MN	LPCDP.Duct13.Fl_O.MN
LPCDP_FAR	LPCDP.Burner.FAR		LPCDP_Duct15_Out_MN	LPCDP.Duct15.Fl_O.MN
LPCDP_HPC_NcDes	LPCDP.HPC.S_map.NcDes		LPCDP_BypBld_Out_MN	LPCDP.BypBld.Fl_O.MN
LPCDP_HPT_NcDes	LPCDP.HPT.S_map.parmNcDes		LPCDP_HPT_Cool1	LPCDP.Bld3.Cool1.fracW
LPCDP_LPCEff	LPCDP.LPC.S_map.effDes		LPCDP_HPT_Cool2	LPCDP.Bld3.Cool2.fracW
LPCDP_HPCEff	LPCDP.HPC.S_map.effDes		LPCDP_LPT_Cool1	LPCDP.HPC.Cool1.fracBldW
LPCDP_HPT_effDes	LPCDP.HPT.S_map.effDes		LPCDP_LPT_Cool2	LPCDP.HPC.Cool2.fracBldW
LPCDP_HPC_Rline	LPCDP.HPC.S_map.RlineMap		HPCDP_FAR	HPCDP.Burner.FAR
LPCDP_HPNmech	LPCDP.HP_Shaft.Nmech		TOC1_FAR	TOC1.Burner.FAR
LPCDP_LPNmech	LPCDP.LP_Shaft.Nmech		TOC2_FAR	TOC2.Burner.FAR
LPCDP_Fan_InletMN	LPCDP.Inlet.Fl_O.MN		TKO_FAR	TKO.Burner.FAR
LPCDP_Fan_Out_MN	LPCDP.Fan.Fl_O.MN		HHDTKO_FAR	HHDTKO.Burner.FAR
LPCDP_LPC_InletMN	LPCDP.Duct4.Fl_O.MN		Cutback_FAR	Cutback.Burner.FAR
LPCDP_LPC_Out_MN	LPCDP.LPC.Fl_O.MN		SLSI_FAR	SLSI.Burner.FAR
LPCDP_HPC_InletMN	LPCDP.Duct6.Fl_O.MN		SLSU_FAR	SLSU.Burner.FAR
LPCDP_HPC_ExitMN	LPCDP.HPC.Fl_O.MN			

Table 64. Constraint Parameter Definition

Constraint Name	Dependent LHS	Dependent RHS
HHDTKO_HPT_BleedFlow1	HHDTKO.HPT.Cool.WbleedC1	HHDTKO.HPT.Cool. WbleedIn1
HHDTKO_HPT_BleedFlow2	HHDTKO.HPT.Cool.WbleedC2	HHDTKO.HPT.Cool. WbleedIn2
HHDTKO_LPT_BleedFlow1	HHDTKO.LPT.Cool.WbleedC1	HHDTKO.LPT.Cool. WbleedIn1
HHDTKO_LPT_BleedFlow2	HHDTKO.LPT.Cool.WbleedC2	HHDTKO.LPT.Cool. WbleedIn2
Max_Fan_Diameter	LPCDP.Fan.Fl_I.Aphy	Fan_Area_Eq
Cutback_Vmix	Cutback_Vmix_q1	1050
LPCDP_LPCPRmin	LPCDP.LPC.Fl_O.Pt/LPCDP.LPC.Fl_I.Pt	1.15
TKO_T3	TKO.HPC.Fl_O.Tt	T3_Max
HHDTKO_T3	HHDTKO.HPC.Fl_O.Tt	T3_Max
SLSI_T3	SLSI.HPC.Fl_O.Tt	T3_Max
TOC2_Fan_Nc	TOC2.Fan.S_map.NcMap	TOC_NcMap
HHDTKO_T4max	HHDTKO.Burner.TtCombOut	T4_Max
SLSI_T4max	SLSI.Burner.TtCombOut	T4_Max
SLSU_T4max	SLSU.Burner.TtCombOut	T4_Max
TKO_Core_Bypass_dP	TKO.HPC.Fl_O.Ps- TKO.Duct15.Fl_O.Ps	550
HHDTKO_Core_Bypass_dP	HHDTKO.HPC.Fl_O.Ps- HHDTKO.Duct15.Fl_O.Ps	550

APPENDIX C

NPSS FILES

The NPSS files listed were used in experiments 2-5:

- C1: NPSS Model File – base model file.
- C2: NPSS Variable List File – contains updates to model file and definitions of all design points.
- C3: NPSS Run File – NPSS execution file. Includes solver setup along with coupling and activation of independents, dependents and constraint parameters.
- C4: NPSS Map Scalar File – file which passes all component performance map scale values to the design points.
- C5: NPSS Function File – file which defines the independents, dependents and constraints.
- C6: NPSS Repair Algorithm – file which containing repair algorithm for failed solver cases

C1: NPSS Model File

```
//-----------------------------------------------------------------------------------------
//  File Name:     LCT_MDP.mdl                                                          |
//  Date(s):       August 1, 2008                                                       |
//  Author:        Jeff Schutte                                                         |
//                                                                                      |
//  Description:   Separate Flow Turbofan Engine for 300 PAX Twin                       |
//                 and 600 PAX Quad Intercontinental Aircraft multi-                    |
//                 design point model                                                   |
//-----------------------------------------------------------------------------------------

Table PercentPower( real PC ) {
  PC =

 PartPwr = committe }

Element FlightConditions Ambient aa

Element Inlet Inlet {
   Afs = 3500.0;
   Subelement ramRecovery S_rec {
     eRamDes = 0.9970;
     Table TB_rec( real Mach ) {
       Mach = {  0.000,  0.025,  0.100,  0.200,  0.250,  0.300,  0.400,  0.500,  0.600,  0.700,  0.900 }
       etar = {  0.993,  0.996,  0.995,  0.993,  0.9925, 0.992,  0.998,  0.998,  0.997,  0.997,  0.997 } } } }
```

```
Element Compressor Fan {
  #include <FAN.map>
  Fl_O.MN = 0.45;   real spec_flow_dsn = 40.0;   S_map.alpha = 0.00;   S_map.RlineMap = 1.6000;
  S_map.NcDes = 0.900;   S_map.PRdes = 1.500;   S_map.effDes = 0.8900;
  S_map.S_eff.TB_Wc.NcorrMap.extrap = "linear" ;     S_map.S_eff.TB_eff.NcorrMap.extrap = "linear" ;
  S_map.S_eff.TB_PR.NcorrMap.extrap = "linear" ;     S_map.S_eff.TB_Wc.RlineMap.extrap = "linear" ;
  S_map.S_eff.TB_eff.RlineMap.extrap = "linear" ;    S_map.S_eff.TB_PR.RlineMap.extrap = "linear" ;   }

Element Splitter Splitter {
  BPR = 8.50;   Fl_01.MN = 0.45;    Fl_02.MN = 0.45;   }

Element Duct Duct4 { dPqPbase = 0.010;    Fl_O.MN = 0.40;   }

Element Compressor LPC {
  #include <LPC.map>
  Fl_O.MN = 0.40;    real spec_flow_dsn = 38.5;   S_map.RlineMap = 1.8000;
  S_map.NcDes = 1.0;    S_map.PRdes = 1.300;    S_map.effDes = 0.9000;   }

Element Duct Duct6 {  dPqPbase = 0.015;     Fl_O.MN = 0.45;   }

Element Compressor HPC {
  #include <HPC.map>
  Fl_O.MN = 0.30;   real spec_flow_dsn = 38.5;   real BleedFlow = 3.930;
  S_map.RlineMap = 2.0;   S_map.NcDes = 0.985;   S_map.PRdes = 20.00;    S_map.effDes = 0.8650;

  InterStageBleedOutPort  Cool1 {    // LPT inlet cooling flow
    fracBldWork = 0.3500;     fracBldP = 0.1465;      fracBldW = 0.01075;     }

  InterStageBleedOutPort  Cool2 {    // LPT exit cooling flow
    fracBldWork = 0.3500;      fracBldP = 0.1465;      fracBldW = 0.00625;     }

  InterStageBleedOutPort  CustBld {
    fracBldWork = 0.3500;      fracBldP = 0.1465;      fracBldW = 0.0000;     }

  void preexecute() {
    CustBld.fracBldW = BleedFlow / Fl_I.W ;    } }

Element Bleed Bld3 {   Fl_O.MN = 0.30;
  BleedOutPort Cool1 {fracW = 0.12;     }            // HPT stator cooling flow
  BleedOutPort Cool2 { fracW = 0.070;   } }          // HPT rotor cooling flow


Element FuelStart FUEL36 {    LHV = 18400. }

Element Burner Burner {
  effBase = 0.999;     switchHotLoss = "input";     dPqPBase = 0.045;     Fl_O.MN = 0.10;
  switchBurn = "FAR";     FAR = 0.0286;     TtCombOut = 3500.0;
  Subelement NeppEI Emissions {  switchCalcType = "EDS";     altName = "Ambient.alt";     }
  void postexecute() {   Emissions.execute();    } }

Element Turbine HPT {
  #include <HPT.map>
  FlowStation FS_41;     Fl_O.MN = 0.30;     S_map.effDes = 0.9200;     S_map.parmMap = 5.280;
  S_map.parmGeomDes  = 1.0;     S_map.parmMapDes = 4.800;     S_map.parmNcDes = 100.0;
  InterStageBleedInPort Non_ChargeableBld {
    Pfract  = 1.0;      diaPump = 0.0;
  }
  InterStageBleedInPort ChargeableBld {
    Pfract  = 0.0;      diaPump = 0.0;
  }
  real delta_eff = 0.0;
    Subelement CoolIt Cool {
    cool1 = "Non_ChargeableBld";      cool2 = "ChargeableBld";
    desBladeTemp[1] = 2300;      desVaneTemp[1] = 2400;      desBladeTemp[2] = 2300;       desVaneTemp[2] = 2300;
    nStages = 2;       xFactor = 1.0;        xFactor1 =1.0;     }

  void postexecute() {
    FS_41.copyFlow("Fl_I");     FS_41.add("Non_ChargeableBld");      Cool.run();      delta_eff = 0.0;     } }

Element Duct Duct11 {  dPqPbase = 0.005;    Fl_O.MN = 0.30;  }
Element Turbine LPT {
  #include <LPT.map>
  FlowStation FS_49;     Fl_O.MN = 0.35;     S_map.effDes = 0.9300;     S_map.parmMap = 5.010;     S_map.parmGeomDes = 1.0;
  S_map.parmMapDes = 5.500;     S_map.parmNcDes = 100.0;
  InterStageBleedInPort Non_ChargeableBld {     Pfract  = 1.0;      diaPump = 0.0;     }
  InterStageBleedInPort ChargeableBld {     Pfract  = 0.0;      diaPump = 0.0;     }
  real delta_eff = 0.0;
  Subelement CoolIt Cool {
   cool1 = "Non_ChargeableBld";      cool2 = "ChargeableBld";
   desBladeTemp[1] = 2000;      desVaneTemp[1] = 2000;      desBladeTemp[2] = 2000;     desVaneTemp[2] = 2000;
   desBladeTemp[3] = 2000;      desVaneTemp[3] = 2000;      nStages = 3;     xFactor = 1.0;     xFactor1 = 1.0;     }

  void postexecute() {
   FS_49.copyFlow("Fl_I");      FS_49.add("Non_ChargeableBld");      Cool.run();      delta_eff = 0.0;     } }

Element Duct Duct13 {     dPqPbase = 0.010;     Fl_O.MN = 0.25;  }

Element Nozzle Core_Nozz {
  switchType = "CONIC";     switchCoef = "CV";     Cang = 0.997;
  Table DischargeCoeff( real PR ) {
     PR =   {1.0, 1.08, 1.25, 1.6, 2, 2.25,  2.5,   3.0,  4.0, 5.0, 7.0}
     CdTh = {0.89, 0.907, 0.922, 0.945, 0.96, 0.966, 0.968, 0.97, 0.9725, 0.974, 0.975}
     PR.interp = "linear";  }
  Subelement CDTH S_CdTh { }
  Table VelCoeff( real PR ) {
    PR = {1.0,  2.1, 2.6444, 3.1889, 3.7333, 4.2778, 4.8222, 5.3667, 5.9111, 6.4556, 7.0}
    Cv = {0.9962, 0.9952, 0.9936, 0.9886, 0.9825, 0.9757, 0.9691, 0.9631, 0.9574, 0.952, 0.947}
   PR.interp = "linear";   }
  Subelement CVELOCITY S_Cv { }
  PsExhName = "Ambient.Ps";   }

Element FlowEnd Core_NozzEnd { }

Element Bleed BypBld {    BleedOutPort Lkg;      Lkg.fracW = 0.0050;     Fl_O.MN = 0.45;  }

Element Duct Duct15 {     dPqPbase = 0.015;     Fl_O.MN = 0.45;  }

Element Nozzle Byp_Nozz {
  switchType = "CONIC";     switchCoef = "CV";     Cang = 0.997;
  Table DischargeCoeff( real PR ) {
    PR =   {1.0, 1.08, 1.25, 1.6, 2, 2.25,  2.5,   3.0,  4.0, 5.0, 7.0}
    CdTh = {0.89, 0.907, 0.922, 0.945, 0.96, 0.966, 0.968, 0.97, 0.9725, 0.974, 0.975}
    PR.interp = "linear";     }
  Subelement CDTH S_CdTh { }
  Table VelCoeff( real PR ) {
    PR = {1.0,  2.1, 2.6444, 3.1889, 3.7333, 4.2778, 4.8222, 5.3667, 5.9111, 6.4556, 7}
    Cv = {0.9962, 0.9952, 0.9936, 0.9886, 0.9825, 0.9757, 0.9691, 0.9631, 0.9574, 0.952, 0.947}
    PR.interp = "linear";     }
  Subelement CVELOCITY S_Cv { }
  PsExhName = "Ambient.Ps";  }

Element FlowEnd Byp_NozzEnd { }

Element Shaft HP_Shaft {     ShaftInputPort HPC, HPT;     HPX = 100.0;     Nmech=10000.0;  }

Element Shaft LP_Shaft {     ShaftInputPort FAN, LPC, LPT;     Nmech=2450.0;  }

Element FlowEnd OverBrd1 {  }

Element FlowEnd OverBrd2 {  }

Element EngPerf PERF {
  real Nozzle_Atot, VjetRatio, OPR_Jon, myOPR;
  real delta_KE, Thrml_Eff, Prop_Eff;
  FlowStation FS_A, FS_B;
  real myFn, myTSFC, myFg, myWfuelHr;
  real theta, theta_n;
  real exp_theta = 0.84;
```

```
   void postexecute() {
     Nozzle_Atot = Core_Nozz.Ath + Byp_Nozz.Ath;        VjetRatio = Core_Nozz.Vactual / Byp_Nozz.Vactual;
     OPR_Jon = Fan.PR * LPC.PR * HPC.PR;       myFg = Core_Nozz.Fg + Byp_Nozz.Fg ;
                                 myFn = myFg - Inlet.Fram;       myWfuelHr = Burner.Wfuel*3600.0;
     myTSFC = myWfuelHr/myFn;       myOPR = HPC.Fl_O.Pt/Fan.Fl_I.Pt;
     FS_A.copyFlow( "HPC.Fl_I" );        FS_B.copyFlow( "HPT.Fl_O" );
     FS_A.setStaticPs(Ambient.Ps);    // expand to Pamb to get ideal velocity
     FS_B.setStaticPs(Ambient.Ps);    // expand to Pamb to get ideal velocity
     delta_KE = (0.5*FS_B.W/C_GRAVITY)*FS_B.V**2.0 -
             (0.5*FS_A.W/C_GRAVITY)*FS_A.V**2.0 ;
     Thrml_Eff = delta_KE / (Burner.Wfuel*FUEL36.LHV*C_BTUtoFT_LBF);
     Prop_Eff = myFn*Ambient.Fl_O.V / delta_KE;
   theta = Fan.Fl_I.Tt/518.67;     theta_n = theta**exp_theta;   } }

//------------------------------------------------------------------------
//                         Component Linkages
//------------------------------------------------------------------------
linkPorts( "Ambient.Fl_O"    , "Inlet.Fl_I"       , "FS_1"  ) ;
linkPorts( "Inlet.Fl_O"      , "Fan.Fl_I"         , "FS_2"  ) ;
linkPorts( "Fan.Fl_O"        , "Splitter.Fl_I"    , "FS_21" ) ;
linkPorts( "Splitter.Fl_01"  , "Duct4.Fl_I"       , "FS_22" ) ;
linkPorts( "Duct4.Fl_O"      , "LPC.Fl_I"         , "FS_23" ) ;
linkPorts( "LPC.Fl_O"        , "Duct6.Fl_I"       , "FS_24" ) ;
linkPorts( "Duct6.Fl_O"      , "HPC.Fl_I"         , "FS_25" ) ;
linkPorts( "HPC.Fl_O"        , "Bld3.Fl_I"        , "FS_3"  ) ;
linkPorts( "Bld3.Fl_O"       , "Burner.Fl_I"      , "FS_36" ) ;
linkPorts( "FUEL36.Fu_O"     , "Burner.Fu_I"      , "FuelIn") ;
linkPorts( "Burner.Fl_O"     , "HPT.Fl_I"         , "FS_4"  ) ;
linkPorts( "HPT.Fl_O"        , "Duct11.Fl_I"      , "FS_45" ) ;
linkPorts( "Duct11.Fl_O"     , "LPT.Fl_I"         , "FS_48" ) ;
linkPorts( "LPT.Fl_O"        , "Duct13.Fl_I"      , "FS_5"  ) ;
linkPorts( "Duct13.Fl_O"     , "Core_Nozz.Fl_I"   , "FS_7"  ) ;
linkPorts( "Core_Nozz.Fl_O"  , "Core_NozzEnd.Fl_I", "FS_9"  ) ;

linkPorts( "Splitter.Fl_02"  , "BypBld.Fl_I"      , "FS_13" ) ;
linkPorts( "BypBld.Fl_O"     , "Duct15.Fl_I"      , "FS_15" ) ;
linkPorts( "Duct15.Fl_O"     , "Byp_Nozz.Fl_I"    , "FS_17" ) ;
linkPorts( "Byp_Nozz.Fl_O"   , "Byp_NozzEnd.Fl_I" , "FS_19" ) ;

//------------------------------------------------------------------------
//                         Bleed Connections
//------------------------------------------------------------------------
linkPorts( "Bld3.Cool1"      , "HPT.Non_ChargeableBld",  "C_HPTinlt" );
linkPorts( "Bld3.Cool2"      , "HPT.ChargeableBld"    ,  "C_HPTexit" );
linkPorts( "HPC.Cool1"       , "LPT.Non_ChargeableBld",  "C_LPTinlt" );
linkPorts( "HPC.Cool2"       , "LPT.ChargeableBld"    ,  "C_LPTexit" );
linkPorts( "HPC.CustBld"     , "OverBrd2.Fl_I"        ,  "CstmrBld"  ) ;
linkPorts( "BypBld.Lkg"      , "OverBrd1.Fl_I"        ,  "Duct15Lk"  ) ;

//------------------------------------------------------------------------
//                         Shaft Connections
//------------------------------------------------------------------------
linkPorts( "HPC.Sh_O"  , "HP_Shaft.HPC" ,  "HPCworkin"   ) ;
linkPorts( "HPT.Sh_O"  , "HP_Shaft.HPT" ,  "HPTworkout"  ) ;
linkPorts( "Fan.Sh_O"  , "LP_Shaft.FAN" ,  "FANworkin"   ) ;
linkPorts( "LPC.Sh_O"  , "LP_Shaft.LPC" ,  "LPCworkin"   ) ;
linkPorts( "LPT.Sh_O"  , "LP_Shaft.LPT" ,  "LPTworkout"  ) ;
```

C2: NPSS Variable List File

```
/* ****************************************************
*  File Name: LCTVARStemplate.list                          *
*  Date: January 24, 2007                                   *
*  For: Aerospace Systems Design Lab @ Georgia Tech         *
*  Description: Multiple Design Point NPSS Variable List    *
*****************************************************/
//----------------------------------------------------
// LPCDP (Low Pressure Component Design Point)
//----------------------------------------------------

//LPCDP Flight Conditions
LPCDP.Ambient.alt = 35000.0; LPCDP.Ambient.MN = 0.8; LPCDP.Ambient.dTs = 0.0;

// Fan
LPCDP_FPR_Dep.eq_rhs = " 1.5 "; LPCDP.Fan.S_map.NcDes = 1.00;          LPCDP.Fan.spec_flow_dsn = 43.3;
LPCDP.Fan.S_map.RlineMap = 2.0; real max_fan_diam = 128;real fan_hubtotip = 0.3;
real Fan_PR = 1.5;real Fan_Deff = 0;

Max_Fan_Diameter.eq_rhs = " 3.14159265/4*(1-fan_hubtotip**2)* max_fan_diam**2"; //.714712328*dia^2
real Fan_Utip = -508.1708 * Fan_PR**2 + 2968.65545 * Fan_PR - 2208;
real Fan_Poly = -0.0000000107547 * Fan_Utip**2 - 0.00000458201 * Fan_Utip + 0.93;
LPCDP.Fan.S_map.effDes = ((Fan_PR**0.28555)-1)/(Fan_PR**(0.28555/Fan_Poly)-1) + Fan_Deff;

//Splitter
LPCDP_Extr_Ratio.eq_rhs = " 1.10 ";

// LPC
LPCDP.LPC.S_map.NcDes = 1.0;    LPCDP.LPC.spec_flow_dsn = 26.5;   LPCDP.LPC.S_map.RlineMap = 2.0;
LPCDP_OPR.eq_rhs = " 30.0 ";        LPCDP_LPCPRmin.eq_rhs = " 1.10 ";real LPC_Deff = 0;
LPCDP_LPCEff_Error.eq_rhs = " (-0.00017183105 * (LPCDP.LPC.PR**3) + 0.00242746662*(LPCDP.LPC.PR**2)-
0.0157813509*LPCDP.LPC.PR + 0.918596973 + LPC_Deff) ";

// HPC
real HPC_BleedFlow = 4.00;  LPCDP.HPC.Fl_O.MN = 0.26;  LPCDP_T3.eq_rhs = " 1750 ";  real HPC_hubtotip = 0.475;

// Ducts
real Duct4_dPqPbase = 0.010;           real Duct6_dPqPbase = 0.008;real Duct11_dPqPbase = 0.010;
real Duct13_dPqPbase = 0.008;real Duct15_dPqPbase = 0.018;

// Burner
real Burner_eff = 0.995;real Burner_dPqPBase = 0.045;

// HPT
real HPT_Deff = 0;      real HPT_1VT = 2400;
real HPT_Temp = 2300;     LPCDP.HPT.S_map.effDes = 0.90 + HPT_Deff;

// LPT
real LPT_Deff = 0;real LPT_Temp = 2200;            LPT_hubtotip = 0.825;
LPCDP.LPT.S_map.effDes = 0.92 + LPT_Deff;
LPT_Flow_Coef.eq_rhs = " 7 * (1+ LPT_hubtotip) / sqrt ( 1 - LPT_hubtotip**2) ";

//Shafts
LPCDP_LPNmech_Error.eq_rhs = " Fan_Utip * 60 * 12 / (2 * PI) * sqrt(PI * (1-fan_hubtotip**2) / LPCDP.Fan.Fl_I.Aphy) ";
LPCDP_HPNmech_Error.eq_rhs = " LPCDP.HPC.S_map.NcDes * 1325 * 60 * 12 / (2 * PI) * sqrt (PI * (1- HPC_hubtotip**2) /
LPCDP.HPC.Fl_I.Aphy) ";

//Nozzles
real S_CoreNozz = 1.0;real S_BypNozz = 1.0;
real A_CoreNozz = 0.975-(S_CoreNozz*0.975); //0.975 = highest value in CdTh table in mdl
real A_BypNozz = 0.975-(S_BypNozz*0.975); //0.975 = highest value in CdTh table in mdl

//Turbine Cooling
LPCDP.HPT.Cool.xFactor = 0.8;//chargeable (EXIT)
LPCDP.HPT.Cool.xFactor1 = 1.5;//non-chargeable (INLET)
LPCDP.LPT.Cool.xFactor = 1.2;//chargeable (EXIT)
LPCDP.LPT.Cool.xFactor1 = 1.5;//non-chargeable (INLET)
```

```
//------------------------------------------------------
//    HPCDP (High Pressure Component Design Point)
//------------------------------------------------------
HPCDP.Ambient.alt = 0.0;                 HPCDP.Ambient.MN = 0.0; HPCDP.Ambient.dTs = 0.0;
real LPCDP_OPR_Target = 30;HPCDP_HPC_Rline.eq_rhs = " 2.0 ";
int HPCimap;

real NC_array[]  = {0.86,    0.87,    0.88,    0.89,    0.90,    0.91,    0.92,    0.93,    0.94,    0.95,    0.96,    0.97,    0.98,    0.99,    1.00,
1.01,    1.02,    1.03,    1.04,    1.05,    1.06,    1.07,    1.08,    1.09,    1.1};

real Eff_array[] = {0.8355,  0.8427,  0.8497,  0.8565,  0.8632,  0.8673,  0.8713,  0.8751,  0.8788,  0.8823,  0.8797,  0.8771,  0.8747,
0.8723,  0.87,    0.8577,  0.8457,  0.8338,  0.8221,  0.8107,  0.8033,  0.7958,  0.7883,  0.7807,  0.7731};

real Wc_array[]  = {113.962, 120.739, 127.596, 134.525, 141.523, 150.396, 159.325, 168.295, 177.294, 186.306, 195.1,   203.883,
212.637, 221.348, 230,     233.468, 236.93,  240.385, 243.834, 247.273, 248.543, 249.796, 251.033, 252.251, 253.454};

real PR_array[]  = {6.1829,  6.5922,  7.033,   7.5077,  8.0189,  8.6257,  9.2834,  9.996,   10.7681, 11.6046, 12.3362, 13.118,  13.9534,
14.8461, 15.8,    16.2421, 16.6922, 17.1503, 17.6164, 18.0903, 18.3174, 18.5355, 18.7441, 18.9427, 19.1308};

if ( LPCDP_OPR_Target <= 30) {
if   (Fan_PR <= 1.6){ HPCimap = 9;}
else {  HPCimap = 7;}
}
else if ( LPCDP_OPR_Target <= 34) { HPCimap = 12;}
else if ( LPCDP_OPR_Target <= 38) { HPCimap = 13;}
else if ( LPCDP_OPR_Target <= 42) { HPCimap = 14;}
else if ( LPCDP_OPR_Target <= 46) { HPCimap = 15;}
else if ( LPCDP_OPR_Target <= 50) { HPCimap = 15;}
else { HPCimap = 16;}

HPCDP_HPC_Nc.eq_rhs = " NC_array[HPCimap]"; HPCDP_HPCPR.eq_rhs  = " PR_array[HPCimap] ";
HPCDP_HPC_Wc.eq_rhs = " Wc_array[HPCimap] "; HPCDP_HPCEff.eq_rhs = " Eff_array[HPCimap] ";
HPCDP_HPC_InletArea.eq_rhs = " 875 "; HPCDP_HPC_ExitArea.eq_rhs = "100";
HPCDP_HPT_InletArea.eq_rhs = "425"; HPCDP_HPT_ExitArea.eq_rhs = "700";

//-----------------------------------------------------
//              TOC1 (Top of Climb)
//-----------------------------------------------------
TOC1.Ambient.alt = 39000.0;TOC1.Ambient.MN = 0.85;TOC1.Ambient.dTs = 0.0;
TOC1_Thrust.eq_rhs = " 16025 "; TOC1_T3.eq_rhs = " 1750 ";  TOC1_Fan_Nc.eq_rhs = " 1.05 ";

//-----------------------------------------------------
//              TOC2 (Top of Climb)
//-----------------------------------------------------
TOC2.Ambient.alt = 35000.0;TOC2.Ambient.MN = 0.85;TOC2.Ambient.dTs = 0.0;
TOC2_Thrust.eq_rhs = " 19600 ";TOC2_T3.eq_rhs = " 1750 ";TOC2_Fan_Nc.eq_rhs = " 1.05 ";

//-----------------------------------------------------
//               TKO (Takeoff)
//-----------------------------------------------------
TKO.Ambient.alt = 0.0;TKO.Ambient.MN = 0.25;TKO.Ambient.dTs = 27.0;
TKO_Thrust.eq_rhs = " 75000 ";TKO_T4max.eq_rhs = " 3300 "; TKO_T3.eq_rhs = " 1750 ";

//-----------------------------------------------------
//       HHDTKO (High Hot Day Takeoff)
//-----------------------------------------------------
HHDTKO.Ambient.alt = 5000.0;HHDTKO.Ambient.MN = 0.25;HHDTKO.Ambient.dTs = 27.0;
HHDTKO_Thrust.eq_rhs = " 67000 ";HHDTKO_T4max.eq_rhs = " 3300 ";HHDTKO_T3.eq_rhs = " 1750 ";

//-----------------------------------------------------
//       Cutback
//-----------------------------------------------------
Cutback.Ambient.alt = 2000.0;Cutback.Ambient.MN = 0.25;Cutback.Ambient.dTs = 18.0;
Cutback_Thrust.eq_rhs = " 60000 ";Cutback_T3.eq_rhs = " 1750 ";Cutback_Vmix.eq_rhs = " 1050 ";

//-----------------------------------------------------
//       SLSI
//-----------------------------------------------------
SLSI.Ambient.alt = 0.0;SLSI.Ambient.MN = 0.0;SLSI.Ambient.dTs = 27.0;
SLSI_Thrust.eq_rhs = " SLSU.PERF.myFn ";SLSI_T4max.eq_rhs = " 3300 ";SLSI_T3.eq_rhs = " 1750 ";
```

```
//-----------------------------------------------------
//      SLSU
//-----------------------------------------------------
SLSU.Ambient.alt = 0.0; SLSU.Ambient.MN = 0.0;SLSU.Ambient.dTs = 0.0;
SLSU_T3.eq_rhs = " 1750 ";SLSU_T4max.eq_rhs = " 3300 ";SLSU.HP_Shaft.HPX = 0.0;
SLSU.Inlet.S_rec.s_eRam = 0.0;SLSU.Inlet.S_rec.a_eRam = 1.0;

//-----------------------------------------------------
//              All Point Setup
//-----------------------------------------------------
int temp_i;
for(temp_i = 0; temp_i <= 8; temp_i++) {
string Design_point[] = {"LPCDP", "HPCDP", "TOC1", "TOC2", "TKO", "HHDTKO", "Cutback", "SLSI", "SLSU" };
Design_point[temp_i]->Duct4.dPqPbase              = Duct4_dPqPbase;
Design_point[temp_i]->Duct6.dPqPbase              = Duct6_dPqPbase;
Design_point[temp_i]->Burner.dPqPBase             = Burner_dPqPBase;
Design_point[temp_i]->Duct11.dPqPbase             = Duct11_dPqPbase;
Design_point[temp_i]->Duct13.dPqPbase             = Duct13_dPqPbase;
Design_point[temp_i]->Duct15.dPqPbase             = Duct15_dPqPbase;
Design_point[temp_i]->Burner.effBase              = Burner_eff;
Design_point[temp_i]->Core_Nozz.DischargeCoeff.s_rtn = S_CoreNozz;
Design_point[temp_i]->Byp_Nozz.DischargeCoeff.s_rtn  = S_BypNozz;
Design_point[temp_i]->Core_Nozz.DischargeCoeff.a_rtn = A_CoreNozz;
Design_point[temp_i]->Byp_Nozz.DischargeCoeff.a_rtn  = A_BypNozz;
Design_point[temp_i]->HPT.Cool.desVaneTemp[1] = HPT_1VT;
Design_point[temp_i]->HPT.Cool.desBladeTemp[1] = HPT_Temp;
   Design_point[temp_i]->HPT.Cool.desBladeTemp[2]  = HPT_Temp;
   Design_point[temp_i]->HPT.Cool.desVaneTemp[2] = HPT_Temp;
Design_point[temp_i]->LPT.Cool.desBladeTemp[1] = LPT_Temp;
   Design_point[temp_i]->LPT.Cool.desVaneTemp[1] = LPT_Temp;
   Design_point[temp_i]->LPT.Cool.desBladeTemp[2] = LPT_Temp;
   Design_point[temp_i]->LPT.Cool.desVaneTemp[2] = LPT_Temp;
   Design_point[temp_i]->LPT.Cool.desBladeTemp[3] = LPT_Temp;
   Design_point[temp_i]->LPT.Cool.desVaneTemp[3] = LPT_Temp;
  if (Design_point[temp_i] == "SLSU") {
 Design_point[temp_i]->HPC.BleedFlow = 0;
  } else {
Design_point[temp_i]->HPC.BleedFlow = HPC_BleedFlow;}}
```

C3: NPSS Run File

```
//------------------------------------------------------------------------
//   File Name:     LCT_MDP.run                                          |
//   Date(s):       June 15, 2006                                        |
//   Author:        Jeff Schutte                                         |
//   Description:   File to run the SFTF design model                    |
//------------------------------------------------------------------------
setThermoPackage("GasTbl");
//------------------------------------------------------------------------
//                       User-Defined Elements
//------------------------------------------------------------------------
#include <CoolIt.int>
#include <Emission.int>
#include <CompressorReynoldsEffects_mod.int>;
#include <CompressorMap_mod.int>;
#include <TurbineReynoldsEffects_mod.int>;
#include <TurbineNeppMap_mod.int>;
#include <print_macros.fnc>
#include <inlet_install.int>
#include <inlet_read_maps.int>
#include <nozzle_read_cv.int>
#include <nozzle_read_cdth.int>
#include <solver_macrosMAX.fnc>
#include <guesslogic.fnc>
//------------------------------------------------------------------------
//                         Model Definition
//------------------------------------------------------------------------
MODELNAME = "High BPR Turbofan Multiple-Point Design Model";
AUTHOR = "";
real PC;
Element Assembly LPCDP {
  #include <LCT_MDP.mdl>
  OutFileStream point1Stream { filename = "LPCDP.viewOut"; }
  DataViewer PageViewer point1 {
    #include <point.view>
    outStreamHandle = "point1Stream";}}

Element Assembly HPCDP {
  #include <LCT_MDP.mdl>
  OutFileStream point2Stream { filename = "HPCDP.viewOut"; }
  DataViewer PageViewer point2 {
    #include <point.view>
    outStreamHandle = "point2Stream";}
  void preexecute() {
    #include <LCT_des_scl.int>   } }

Element Assembly TOC1 {
  #include <LCT_MDP.mdl>
  OutFileStream point3Stream { filename = "TOC1.viewOut"; }
  DataViewer PageViewer point3 {
    #include <point.view>
    outStreamHandle = "point3Stream";}
  void preexecute() {
    #include <LCT_des_scl.int>   } }

Element Assembly TOC2 {
  #include <LCT_MDP.mdl>
  OutFileStream point4Stream { filename = "TOC2.viewOut"; }
  DataViewer PageViewer point4 {
    #include <point.view>
    outStreamHandle = "point4Stream";}
  void preexecute() {
    #include <LCT_des_scl.int>   } }

Element Assembly TKO{
  #include <LCT_MDP.mdl>
  Burner.Emissions.altName = "TKO.Ambient.alt";
  OutFileStream point5Stream { filename = "TKO.viewOut"; }
  DataViewer PageViewer point5 {
    #include <point.view>
    outStreamHandle = "point5Stream";}
  void preexecute() {
    #include <LCT_des_scl.int>   } }

Element Assembly HHDTKO{
  #include <LCT_MDP.mdl>
  Burner.Emissions.altName = "HHDTKO.Ambient.alt";
  OutFileStream point6Stream { filename = "HHDTKO.viewOut"; }
  DataViewer PageViewer point6 {
    #include <point.view>
    outStreamHandle = "point6Stream";}
  void preexecute() {
    #include <LCT_des_scl.int>   } }

Element Assembly Cutback{
  #include <LCT_MDP.mdl>
  Burner.Emissions.altName = "Cutback.Ambient.alt";
  OutFileStream point7Stream { filename = "Cutback.viewOut"; }
  DataViewer PageViewer point7 {
    #include <point.view>
    outStreamHandle = "point7Stream";}
  void preexecute() {
    #include <LCT_des_scl.int>   } }

Element Assembly SLSI{
  #include <LCT_MDP.mdl>
  Burner.Emissions.altName = "SLSI.Ambient.alt";
  OutFileStream point8Stream { filename = "SLSI.viewOut"; }
  DataViewer PageViewer point8 {
    #include <point.view>
    outStreamHandle = "point8Stream";}
  void preexecute() {
    #include <LCT_des_scl.int>   } }

Element Assembly SLSU{
  #include <LCT_MDP.mdl>
  Burner.Emissions.altName = "SLSU.Ambient.alt";
  OutFileStream point9Stream { filename = "SLSU.viewOut"; }
  DataViewer PageViewer point9 {
    #include <point.view>
    outStreamHandle = "point9Stream";}
  void preexecute() {
    #include <LCT_des_scl.int>   } }

#include <solver_macros.fnc>
#include <LCT_MDP.fnc>
//---------------------------------------------------------------------------
// Changes to Baseline Engine
//---------------------------------------------------------------------------
#include <vars.list>
#include <guesslogic.int>
//---------------------------------------------------------------------------
// RUN DESIGN POINT(s)
//---------------------------------------------------------------------------
LPCDP  { setOption( "switchDes", "DESIGN" );}
HPCDP  { setOption( "switchDes", "OFFDESIGN" );}   TOC1 { setOption( "switchDes", "OFFDESIGN" );}
TOC2 { setOption( "switchDes", "OFFDESIGN" );}     TKO     { setOption( "switchDes", "OFFDESIGN" );}
HHDTKO  { setOption( "switchDes", "OFFDESIGN" );}  Cutback { setOption( "switchDes", "OFFDESIGN" );}
SLSI    { setOption( "switchDes", "OFFDESIGN" );}  SLSU    { setOption( "switchDes", "OFFDESIGN" );}

solver.defaultTolerance = 0.00000005; solver.defaultToleranceType = "FRACTIONAL"; solver.maxJacobians = 20;
solver.maxIterations = 125; solver.defaultDxLimit = 0.05; solver.broydenLimit = 0;
solver.convergenceLimit = 1.0; solver.divergenceLimit = 1.5; solver.maxConvergeFailures = 6;

//Constraints ("name", "max or min", priority (1 to n higher is more priority, slope(1 or -1))
LPCDP_FPR_Dep.addConstraint("Max_Fan_Diameter", "MIN",1,-1);
LPCDP_OPR.addConstraint("LPCDP_LPCPRmin","MIN");
TKO_T4max.addConstraint("HHDTKO_T4max", "MAX");
```

```
LPCDP_OPR.addConstraintGroup("T3limits");
LPCDP_OPR.addConstraint("TKO_Core_Bypass_dP", "MAX");
LPCDP_OPR.addConstraint("HHDTKO_Core_Bypass_dP", "MAX");
TKO.HPT.Cool.dep_BleedFlow1.addConstraint("HHDTKO.HPT.Cool.dep_BleedFlow1", "MAX");
TKO.HPT.Cool.dep_BleedFlow2.addConstraint("HHDTKO.HPT.Cool.dep_BleedFlow2", "MAX");
TKO.LPT.Cool.dep_BleedFlow1.addConstraint("HHDTKO.LPT.Cool.dep_BleedFlow1", "MAX");
TKO.LPT.Cool.dep_BleedFlow2.addConstraint("HHDTKO.LPT.Cool.dep_BleedFlow2", "MAX");
LPCDP_FPR_Dep.addConstraint("Cutback_Vmix", "MAX");
TOC1_Fan_Nc.addConstraint("TOC2_Fan_Nc", "MAX");
SLSI_Thrust.addConstraint("SLSI_T4max", "MAX");
SLSU_FanNcMap.addConstraint("SLSU_T4max", "MAX");
autoSolverSetup();

// AutoSolver added 4 turbine cooling flow Dependents
solver.addIndependent( "LPCDP_HPT_Cool1" );solver.addIndependent( "LPCDP_HPT_Cool2" );
solver.addIndependent( "LPCDP_LPT_Cool1" );solver.addIndependent( "LPCDP_LPT_Cool2" );
solver.removeDependent( "LPCDP.HPT.Cool.dep_BleedFlow1" );solver.removeDependent( "LPCDP.HPT.Cool.dep_BleedFlow2" );
solver.removeDependent( "LPCDP.LPT.Cool.dep_BleedFlow1" );solver.removeDependent( "LPCDP.LPT.Cool.dep_BleedFlow2" );
solver.addDependent  ( "TKO.HPT.Cool.dep_BleedFlow1" );solver.addDependent  ( "TKO.HPT.Cool.dep_BleedFlow2" );
solver.addDependent  ( "TKO.LPT.Cool.dep_BleedFlow1" );solver.addDependent  ( "TKO.LPT.Cool.dep_BleedFlow2" );
solver.addIndependent( "LPCDP_FPR" );solver.addDependent  ( "LPCDP_FPR_Dep" );
solver.addIndependent( "LPCDP_LPCPR" );solver.addDependent  ( "LPCDP_OPR" );
solver.addIndependent( "LPCDP_BPR" );solver.addDependent  ( "LPCDP_Extr_Ratio" );
solver.addIndependent( "LPCDP_W" );solver.addDependent  ( "TOC1_Fan_Nc" );
solver.addIndependent ( "TOC1_FAR" );solver.addDependent( "TOC1_Thrust" );
solver.addIndependent ( "TOC2_FAR" );solver.addDependent( "TOC2_Thrust" );
solver.addIndependent ( "Cutback_FAR" );solver.addDependent( "Cutback_Thrust" );
solver.addIndependent( "TKO_FAR" );solver.addDependent  ( "TKO_Thrust" );
solver.addIndependent ( "HHDTKO_FAR" );solver.addDependent( "HHDTKO_Thrust" );
solver.addIndependent( "LPCDP_FAR" );solver.addDependent( "TKO_T4max" );
solver.addIndependent ( "SLSI_FAR" ); solver.addDependent( "SLSI_Thrust" );
solver.addIndependent ( "SLSU_FAR" ); solver.addDependent( "SLSU_FanNcMap" );
solver.addIndependent ( "LPCDP_Fan_InletMN" );solver.addDependent( "LPCDP_Fan_Spec_Flow" );
solver.addIndependent ( "LPCDP_LPC_InletMN" );solver.addDependent( "LPCDP_LPC_Spec_Flow" );
solver.addIndependent( "HPCDP_FAR" );solver.addDependent  ( "HPCDP_HPC_Nc" );
solver.addIndependent( "LPCDP_HPCPR" );solver.addDependent  ( "HPCDP_HPCPR" );
solver.addIndependent( "LPCDP_HPC_NcDes" );solver.addDependent  ( "HPCDP_HPC_Wc" );
solver.addIndependent( "LPCDP_HPCEff" );solver.addDependent  ( "HPCDP_HPCEff" );
solver.addIndependent( "LPCDP_HPC_Rline" );solver.addDependent  ( "HPCDP_HPC_Rline" );
solver.addIndependent( "LPCDP_HPC_InletMN" );solver.addDependent  ( "HPCDP_HPC_InletArea" );
solver.addIndependent( "LPCDP_HPC_ExitMN" );solver.addDependent  ( "HPCDP_HPC_ExitArea" );
solver.addIndependent( "LPCDP_HPT_InletMN" );solver.addDependent  ( "HPCDP_HPT_InletArea" );
solver.addIndependent( "LPCDP_HPT_ExitMN" );solver.addDependent  ( "HPCDP_HPT_ExitArea" );
solver.addIndependent( "LPCDP_HPT_effDes" );solver.addDependent  ( "HPCDP_HPT_s_effDes" );
solver.addIndependent( "LPCDP_HPT_NcDes" );solver.addDependent  ( "HPCDP_HPT_s_WcDes" );
solver.addIndependent( "LPCDP_HPT_PRDes" );solver.addDependent  ( "HPCDP_HPT_s_PRdes" );
solver.addIndependent( "LPCDP_LPCEff" );solver.addDependent  ( "LPCDP_LPCEff_Error" );
solver.addIndependent( "LPCDP_HPNmech" );solver.addDependent  ( "LPCDP_HPNmech_Error" );
solver.addIndependent( "LPCDP_LPNmech" );solver.addDependent  ( "LPCDP_LPNmech_Error" );
solver.addIndependent( "LPCDP_SPL_Core_MN" ); solver.addDependent  ( "SPL_Core_Area" );
solver.addIndependent( "LPCDP_SPL_Byp_MN" ); solver.addDependent  ( "SPL_Byp_Area" );
solver.addIndependent( "LPCDP_Fan_Out_MN" ); solver.addDependent  ( "Fan_Area_Out" );
solver.addIndependent( "LPCDP_LPC_Out_MN" ); solver.addDependent  ( "LPC_Area_Out" );
solver.addIndependent( "LPCDP_Duct11_Out_MN" ); solver.addDependent  ( "LPT_Flow_Coef" );
solver.addIndependent ( "LPCDP_Duct13_Out_MN" ); solver.addDependent   ( "Duct13_Area_Out" );
solver.addIndependent ( "LPCDP_Duct15_Out_MN" ); solver.addDependent   ( "Duct15_Area_Out" );
solver.addIndependent ( "LPCDP_BypBld_Out_MN" ); solver.addDependent   ( "BypBld_Area_Out" );

#include <solver_output.npss>
verify();
run();

save_engine_data();
update_guesslogic();
```

C4: NPSS Map Scalar File

```
//------------------------------------------------------------------------
//   File Name:     LCT_des_scl.int                                  |
//   Date(s):       June 15, 2007                                    |
//   Author:        Jeff Schutte                                     |
//   Description:   File to pass map scalars to design points        |
//------------------------------------------------------------------------

// INLET SCALARS
Inlet.eRam = LPCDP.Inlet.eRam;

// TURBOMACHINERY SCALARS
Fan.S_map.s_effDes = LPCDP.Fan.S_map.s_effDes;
Fan.S_map.s_NcDes  = LPCDP.Fan.S_map.s_NcDes;
Fan.S_map.s_PRdes  = LPCDP.Fan.S_map.s_PRdes;
Fan.S_map.s_WcDes  = LPCDP.Fan.S_map.s_WcDes;
LPC.S_map.s_effDes = LPCDP.LPC.S_map.s_effDes;
LPC.S_map.s_NcDes  = LPCDP.LPC.S_map.s_NcDes;
LPC.S_map.s_PRdes  = LPCDP.LPC.S_map.s_PRdes;
LPC.S_map.s_WcDes  = LPCDP.LPC.S_map.s_WcDes;
HPC.S_map.s_effDes = LPCDP.HPC.S_map.s_effDes;
HPC.S_map.s_NcDes  = LPCDP.HPC.S_map.s_NcDes;
HPC.S_map.s_PRdes  = LPCDP.HPC.S_map.s_PRdes;
HPC.S_map.s_WcDes  = LPCDP.HPC.S_map.s_WcDes;
HPT.S_map.s_eff    = LPCDP.HPT.S_map.s_eff;
HPT.S_map.s_Np     = LPCDP.HPT.S_map.s_Np;
HPT.S_map.s_dPqP   = LPCDP.HPT.S_map.s_dPqP;
HPT.S_map.s_Wp     = LPCDP.HPT.S_map.s_Wp;
LPT.S_map.s_eff    = LPCDP.LPT.S_map.s_eff;
LPT.S_map.s_Np     = LPCDP.LPT.S_map.s_Np;
LPT.S_map.s_dPqP   = LPCDP.LPT.S_map.s_dPqP;
LPT.S_map.s_Wp     = LPCDP.LPT.S_map.s_Wp;

// COMPONENT AREAS
Inlet.Fl_O.Aphy    = LPCDP.Inlet.Fl_O.Aphy;
Fan.Fl_O.Aphy      = LPCDP.Fan.Fl_O.Aphy;
Splitter.Fl_01.Aphy= LPCDP.Splitter.Fl_01.Aphy;
Splitter.Fl_02.Aphy= LPCDP.Splitter.Fl_02.Aphy;
Duct4.Fl_O.Aphy    = LPCDP.Duct4.Fl_O.Aphy;
LPC.Fl_O.Aphy      = LPCDP.LPC.Fl_O.Aphy;
Duct6.Fl_O.Aphy    = LPCDP.Duct6.Fl_O.Aphy;
HPC.Fl_O.Aphy      = LPCDP.HPC.Fl_O.Aphy;
Bld3.Fl_O.Aphy     = LPCDP.Bld3.Fl_O.Aphy;
Burner.Fl_O.Aphy   = LPCDP.Burner.Fl_O.Aphy;
HPT.Fl_O.Aphy      = LPCDP.HPT.Fl_O.Aphy;
Duct11.Fl_O.Aphy   = LPCDP.Duct11.Fl_O.Aphy;
LPT.Fl_O.Aphy      = LPCDP.LPT.Fl_O.Aphy;
Duct13.Fl_O.Aphy   = LPCDP.Duct13.Fl_O.Aphy;
BypBld.Fl_O.Aphy   = LPCDP.BypBld.Fl_O.Aphy;
Duct15.Fl_O.Aphy   = LPCDP.Duct15.Fl_O.Aphy;
Core_Nozz.AthCold  = LPCDP.Core_Nozz.AthCold;
Byp_Nozz.AthCold   = LPCDP.Byp_Nozz.AthCold;

// BLEED FLOW FRACTIONS       // Set cooling flows for takeoff condition
HPC.Cool1.fracBldW = LPCDP.HPC.Cool1.fracBldW;
HPC.Cool2.fracBldW = LPCDP.HPC.Cool2.fracBldW;
Bld3.Cool1.fracW   = LPCDP.Bld3.Cool1.fracW;
Bld3.Cool2.fracW   = LPCDP.Bld3.Cool2.fracW;

//DESIGN CORRECTED SPEEDS
Fan.NcDes = LPCDP.Fan.Nc;
LPC.NcDes = LPCDP.LPC.Nc;
HPC.NcDes = LPCDP.HPC.Nc;
Fan.S_map.NcDes = LPCDP.Fan.S_map.NcDes;
LPC.S_map.NcDes = LPCDP.LPC.S_map.NcDes;
HPC.S_map.NcDes = LPCDP.HPC.S_map.NcDes;

//Design Reynolds # and Corrected Flow for Reynolds effects
```

```
Fan.S_map.WcDes  = LPCDP.Fan.S_map.WcDes;
Fan.S_map.ReDes  = LPCDP.Fan.S_map.ReDes;
Fan.S_map.RNI_des = LPCDP.Fan.S_map.RNI_des;
LPC.S_map.WcDes  = LPCDP.LPC.S_map.WcDes;
LPC.S_map.ReDes  = LPCDP.LPC.S_map.ReDes;
LPC.S_map.RNI_des = LPCDP.LPC.S_map.RNI_des;
HPC.S_map.WcDes  = LPCDP.HPC.S_map.WcDes;
HPC.S_map.ReDes  = LPCDP.HPC.S_map.ReDes;
HPC.S_map.RNI_des = LPCDP.HPC.S_map.RNI_des;

LPT.S_map.TtMap  = LPCDP.LPT.S_map.TtMap;
LPT.S_map.PtMap  = LPCDP.LPT.S_map.PtMap;
LPT.S_map.RNImap  = LPCDP.LPT.S_map.RNImap;
HPT.S_map.TtMap  = LPCDP.HPT.S_map.TtMap;
HPT.S_map.PtMap  = LPCDP.HPT.S_map.PtMap;
HPT.S_map.RNImap  = LPCDP.HPT.S_map.RNImap;

HP_Shaft.HPX = LPCDP.HP_Shaft.HPX;

//Cooling
HPT.Cool.xFactor  = LPCDP.HPT.Cool.xFactor;                //chargeable
HPT.Cool.xFactor1 = LPCDP.HPT.Cool.xFactor1;               //non-chargeable
LPT.Cool.xFactor  = LPCDP.LPT.Cool.xFactor;                //chargeable
LPT.Cool.xFactor1 = LPCDP.LPT.Cool.xFactor1;               //non-chargeable
```

C5: NPSS Function File

```
//------------------------------------------------------------------------
//  File Name:    LCT_MDP.fnc                                              |
//  Date(s):      June 15, 2007                                            |
//  Author:       Jeff Schutte                                             |
//  Description:  Defines NPSS Independents and Dependents                 |
//------------------------------------------------------------------------


//==============================================================
//  NPSS Independents
//==============================================================
Independent LPCDP_W              { varName = "LPCDP.Ambient.W"; }
Independent LPCDP_BPR            { varName = "LPCDP.Splitter.BPR"; }
Independent LPCDP_FPR            { varName = "LPCDP.Fan.S_map.PRdes";  dxLimit = 0.05;    dxLimitType = "ABSOLUTE";}
Independent LPCDP_FAR            { varName = "LPCDP.Burner.FAR"; }
Independent HPCDP_FAR            { varName = "HPCDP.Burner.FAR"; }
Independent LPCDP_HPCPR          { varName = "LPCDP.HPC.S_map.PRdes"; }
Independent LPCDP_HPC_NcDes      { varName = "LPCDP.HPC.S_map.NcDes"; }
Independent TOC1_FAR             { varName = "TOC1.Burner.FAR"; }
Independent TOC2_FAR             { varName = "TOC2.Burner.FAR"; }
Independent TKO_FAR              { varName = "TKO.Burner.FAR"; }
Independent Cutback_FAR          { varName = "Cutback.Burner.FAR"; }
Independent HHDTKO_FAR           { varName = "HHDTKO.Burner.FAR"; }
Independent LPCDP_HPCEff         { varName = "LPCDP.HPC.S_map.effDes"; }
Independent LPCDP_HPC_Rline   { varName = "LPCDP.HPC.S_map.RlineMap"; dxLimit = 0.01;  dxLimitType = "ABSOLUTE";}
Independent LPCDP_LPCPR          { varName = "LPCDP.LPC.S_map.PRdes"; dxLimit = 0.1;    dxLimitType = "ABSOLUTE";}
Independent LPCDP_LPCEff         { varName = "LPCDP.LPC.S_map.effDes"; }
Independent LPCDP_HPT_effDes     { varName = "LPCDP.HPT.S_map.effDes"; }
Independent LPCDP_HPT_PRDes      { varName = "LPCDP.HPT.S_map.parmMapDes"; }
Independent LPCDP_HPT_NcDes      { varName = "LPCDP.HPT.S_map.parmNcDes"; }
Independent LPCDP_HPT_Cool1      { varName = "LPCDP.Bld3.Cool1.fracW"; }
Independent LPCDP_HPT_Cool2      { varName = "LPCDP.Bld3.Cool2.fracW"; }
Independent LPCDP_LPT_Cool1      { varName = "LPCDP.HPC.Cool1.fracBldW"; }
Independent LPCDP_HPNmech        { varName = "LPCDP.HP_Shaft.Nmech"; }
Independent LPCDP_LPNmech        { varName = "LPCDP.LP_Shaft.Nmech"; }
Independent LPCDP_LPT_Cool2      { varName = "LPCDP.HPC.Cool2.fracBldW";   dxLimit = 0.4; }
Independent LPCDP_Fan_InletMN    { varName = "LPCDP.Inlet.Fl_O.MN";        dxLimit = 0.02;    dxLimitType = "ABSOLUTE"; }
Independent LPCDP_LPC_InletMN    { varName = "LPCDP.Duct4.Fl_O.MN";      dxLimit = 0.02;  dxLimitType = "ABSOLUTE"; }
Independent LPCDP_HPC_InletM   { varName = "LPCDP.Duct6.Fl_O.MN";      dxLimit = 0.02;    dxLimitType = "ABSOLUTE"; }
Independent LPCDP_HPC_ExitMN     { varName = "LPCDP.HPC.Fl_O.MN";     dxLimit = 0.02;    dxLimitType = "ABSOLUTE"; }
Independent LPCDP_HPT_InletMN    { varName = "LPCDP.Burner.Fl_O.MN";    dxLimit = 0.02;   dxLimitType = "ABSOLUTE"; }
Independent LPCDP_HPT_ExitMN     { varName = "LPCDP.HPT.Fl_O.MN";   dxLimit = 0.02;    dxLimitType = "ABSOLUTE"; }
Independent LPCDP_SPL_Core_MN  { varName = "LPCDP.Splitter.Fl_01.MN";  dxLimit = 0.05;  dxLimitType = "ABSOLUTE"; }
Independent LPCDP_SPL_Byp_MN    { varName = "LPCDP.Splitter.Fl_02.MN"; dxLimit = 0.05;  dxLimitType = "ABSOLUTE"; }
Independent LPCDP_Fan_Out_MN     { varName = "LPCDP.Fan.Fl_O.MN";        dxLimit = 0.05;    dxLimitType = "ABSOLUTE"; }
Independent LPCDP_LPC_Out_MN     { varName = "LPCDP.LPC.Fl_O.MN";     dxLimit = 0.05;    dxLimitType = "ABSOLUTE"; }
Independent LPCDP_LPT_Out_MN     { varName = "LPCDP.LPT.Fl_O.MN";     dxLimit = 0.05;    dxLimitType = "ABSOLUTE"; }
Independent LPCDP_Duct11_Out_MN  { varName = "LPCDP.Duct11.Fl_O.MN"; dxLimit = 0.05;  dxLimitType = "ABSOLUTE"; }
Independent LPCDP_Duct13_Out_MN  { varName = "LPCDP.Duct13.Fl_O.MN";  dxLimit = 0.05; dxLimitType = "ABSOLUTE"; }
Independent LPCDP_Duct15_Out_MN  { varName = "LPCDP.Duct15.Fl_O.MN";  dxLimit = 0.05; dxLimitType = "ABSOLUTE"; }
Independent LPCDP_BypBld_Out_MN  { varName = "LPCDP.BypBld.Fl_O.MN"; dxLimit = 0.05; dxLimitType = "ABSOLUTE"; }
Independent FAR_IND          { varName = "Burner.FAR"; }
Independent SLSI_FAR           { varName = "SLSI.Burner.FAR"; }
Independent SLSU_FAR           { varName = "SLSU.Burner.FAR"; }

//==============================================================
//  NPSS Dependents and Constraints
//==============================================================
Dependent TKO_Thrust           { eq_lhs = "TKO.PERF.myFn";             eq_rhs = "79000.0"; }
Dependent TOC1_Fan_Nc          { eq_lhs = "TOC1.Fan.S_map.NcMap";      eq_rhs = "1.05"; }
Dependent TOC2_Fan_Nc          { eq_lhs = "TOC2.Fan.S_map.NcMap";      eq_rhs = "1.05"; }
Dependent HPCDP_HPC_Nc  { eq_lhs = "HPCDP.HPC.S_map.NcMap";    eq_rhs = "1.0"; }
Dependent HPCDP_HPCPR     { eq_lhs = "HPCDP.HPC.PR";             eq_rhs = "16.0"; }
Dependent LPCDP_OPR           { eq_lhs = "LPCDP.PERF.myOPR";         eq_rhs = "40"; }
Dependent HPCDP_HPC_Rline        { eq_lhs = "HPCDP.HPC.S_map.RlineMap"; eq_rhs = "2.0"; }
Dependent TOC1_Thrust          { eq_lhs = "TOC1.PERF.myFn";            eq_rhs = "15000.0"; }
Dependent TOC2_Thrust          { eq_lhs = "TOC2.PERF.myFn";            eq_rhs = "19000.0"; }
Dependent HHDTKO_Thrust        { eq_lhs = "HHDTKO.PERF.myFn";        eq_rhs = "75000.0"; }
Dependent Cutback_Thrust       { eq_lhs = "Cutback.PERF.myFn";       eq_rhs = "60000.0"; }
Dependent TKO_T4max            { eq_lhs = "TKO.Burner.TtCombOut";     eq_rhs = "3285.0"; }
Dependent HHDTKO_T4max         { eq_lhs = "HHDTKO.Burner.TtCombOut";   eq_rhs = "3285.0"; }
Dependent SLSI_T4max           { eq_lhs = "SLSI.Burner.TtCombOut";     eq_rhs = "3285.0"; }
Dependent SLSU_T4max           { eq_lhs = "SLSU.Burner.TtCombOut";     eq_rhs = "3285.0"; }
Dependent Max_Fan_Diameter { eq_lhs = "LPCDP.Fan.Fl_I.Aphy";  eq_rhs = "10000"; }
Dependent LPCDP_T3{ eq_lhs = "LPCDP.HPC.Fl_O.Tt";  eq_rhs = "1750"; }
Dependent TKO_T3    { eq_lhs = "TKO.HPC.Fl_O.Tt";         eq_rhs = "1750"; }
Dependent HHDTKO_T3            { eq_lhs = "HHDTKO.HPC.Fl_O.Tt";      eq_rhs = "1750"; }
Dependent Cutback_T3         { eq_lhs = "Cutback.HPC.Fl_O.Tt";     eq_rhs = "1750"; }
Dependent TOC1_T3            { eq_lhs = "TOC1.HPC.Fl_O.Tt";        eq_rhs = "1750"; }
Dependent TOC2_T3            { eq_lhs = "TOC2.HPC.Fl_O.Tt";        eq_rhs = "1750"; }
Dependent SLSI_T3           { eq_lhs = "SLSI.HPC.Fl_O.Tt";        eq_rhs = "1750"; }
Dependent SLSU_T3            { eq_lhs = "SLSU.HPC.Fl_O.Tt";        eq_rhs = "1750"; }
Dependent HPCDP_HPC_InletArea  { eq_lhs = "HPCDP.Duct6.Fl_O.Aphy";   eq_rhs = "875"; }
Dependent HPCDP_HPC_ExitArea   { eq_lhs = "HPCDP.HPC.Fl_O.Aphy";      eq_rhs = "100"; }
Dependent HPCDP_HPT_InletArea  { eq_lhs = "HPCDP.Burner.Fl_O.Aphy";   eq_rhs = "425"; }
Dependent HPCDP_HPT_ExitArea   { eq_lhs = "HPCDP.HPT.Fl_O.Aphy";       eq_rhs = "700"; }
Dependent LPCDP_HPNmech_Error   { eq_lhs = "LPCDP.HP_Shaft.Nmech";     eq_rhs = "8600"; }
Dependent LPCDP_LPNmech_Error    { eq_lhs = "LPCDP.LP_Shaft.Nmech";       eq_rhs = "2200"; }
Dependent HPCDP_HPT_s_WcDes  { eq_lhs = "HPCDP.HPT.S_map.s_Wp"; eq_rhs = "1.0"; tolerance = 0.001;  toleranceType =
"ABSOLUTE"; }
Dependent HPCDP_HPT_s_PRdes    { eq_lhs = "HPCDP.HPT.S_map.s_dPqP";   eq_rhs = "1.0";    tolerance = 0.001;   toleranceType
= "ABSOLUTE"; }
Dependent HPCDP_HPT_s_effDes   { eq_lhs = "HPCDP.HPT.S_map.s_eff";    eq_rhs = "1.0";    tolerance = 0.001;   toleranceType =
"ABSOLUTE"; }
Dependent HPCDP_HPCEff          { eq_lhs = "HPCDP.HPC.eff/HPCDP.HPC.S_map.s_effRe";              eq_rhs = "0.87"; }
Dependent HPCDP_HPC_Wc         { eq_lhs = "HPCDP.HPC.Wc/HPCDP.HPC.S_map.s_effRe";               eq_rhs = "230"; }
Dependent LPCDP_Extr_Ratio     { eq_lhs = "LPCDP.Byp_Nozz.Fl_I.Pt/LPCDP.Core_Nozz.Fl_I.Pt";      eq_rhs = "1.1"; }
Dependent LPCDP_Fan_Spec_Flow  { eq_lhs = "144.0*LPCDP.Fan.Wc/LPCDP.Inlet.Fl_O.Aphy";     eq_rhs =
"LPCDP.Fan.spec_flow_dsn"; }
Dependent LPCDP_LPC_Spec_Flow  { eq_lhs = "144.0*LPCDP.LPC.Wc/LPCDP.Duct4.Fl_O.Aphy";            eq_rhs =
"LPCDP.LPC.spec_flow_dsn"; }
Dependent LPCDP_FPR_Dep        { eq_lhs = "LPCDP.Fan.Fl_O.Pt/LPCDP.Fan.Fl_I.Pt";            eq_rhs = "1.5"; }
Dependent LPCDP_LPCPRmin         { eq_lhs = "LPCDP.LPC.Fl_O.Pt/LPCDP.LPC.Fl_I.Pt";               eq_rhs = "1.1"; }
Dependent SPL_Core_Area       { eq_lhs = "LPCDP.Splitter.Fl_01.Aphy/LPCDP.Duct4.Fl_O.Aphy";                 eq_rhs = "1.0"; }
Dependent Fan_Area_Out         { eq_lhs = "LPCDP.Fan.Fl_O.Aphy/LPCDP.Fan.Fl_I.Aphy";           eq_rhs = "0.90"; }
Dependent LPC_Area_Out          { eq_lhs = "LPCDP.LPC.Fl_O.Aphy/LPCDP.LPC.Fl_I.Aphy";           eq_rhs = "0.75"; }
Dependent SPL_Byp_Area          { eq_lhs = "LPCDP.Splitter.Fl_01.Aphy+LPCDP.Splitter.Fl_02.Aphy"; eq_rhs =
"LPCDP.Fan.Fl_O.Aphy"; }
Dependent Duct13_Area_Out       { eq_lhs = "LPCDP.Duct13.Fl_O.Aphy/LPCDP.LPT.Fl_O.Aphy";        eq_rhs = ".95"; }
Dependent Duct15_Area_Out       { eq_lhs = "LPCDP.Duct15.Fl_O.Aphy/LPCDP.Splitter.Fl_02.Aphy";     eq_rhs = "1.0"; }
Dependent BypBld_Area_Out       { eq_lhs = "LPCDP.BypBld.Fl_O.Aphy/LPCDP.Splitter.Fl_02.Aphy";     eq_rhs = "1.0"; }
Dependent LPT_Flow_Coef        { eq_lhs = "LPCDP.Duct11.Fl_O.Vflow*(LPCDP.LP_Shaft.Nmech/(60*12))/ sqrt(PI *
LPCDP.Duct11.Fl_O.Aphy)";   eq_rhs = "6"; }
Dependent LPCDP_LPCEff_Error    { eq_lhs = "LPCDP.LPC.S_map.effDes";            eq_rhs = "(-0.00017183105 *
(LPCDP.LPC.PR**3) + 0.00242746662*(LPCDP.LPC.PR**2)-0.0157813509*LPCDP.LPC.PR + 0.918596973)"; }
Dependent Cutback_Vmix         { eq_lhs = "(Cutback.Byp_Nozz.Fl_O.V*Cutback.Byp_Nozz.Fl_O.W +
Cutback.Core_Nozz.Fl_O.V*Cutback.Core_Nozz.Fl_O.W)/(Cutback.Byp_Nozz.Fl_O.W + Cutback.Core_Nozz.Fl_O.W)";    eq_rhs
= "1500";}
Dependent Max_NcFan          { eq_lhs = "Fan.NcqNcDes";  eq_rhs = "1.0"; }
Dependent Target_Fnet          { eq_lhs = "PERF.myFn";     eq_rhs = "TargetThrust"; }
Dependent TKO_Core_Bypass_dP    { eq_lhs = "TKO.HPC.Fl_O.Ps-TKO.Duct15.Fl_O.Ps";     eq_rhs = "550"; }
Dependent HHDTKO_Core_Bypass_dP { eq_lhs = "HHDTKO.HPC.Fl_O.Ps-HHDTKO.Duct15.Fl_O.Ps";  eq_rhs = "550"; }
Dependent SLSI_Thrust           { eq_lhs = "SLSI.PERF.myFn";            eq_rhs = "70000.0"; }
Dependent SLSU_FanNcMap         { eq_lhs = "SLSU.Fan.S_map.NcMap";      eq_rhs = "1.0"; }
ConstraintGroup T3limits;
T3limits.addConstraint ( "TKO_T3", "MAX");
T3limits.addConstraint ( "HHDTKO_T3", "MAX");
T3limits.addConstraint ( "SLSI_T3", "MAX");
```

C6: MDP Repair Algorithm

```
//------------------------------------------------------------------------
//   File Name:     LCT_Repair.fnc                                        |
//   Date(s):       June 15, 2007                                         |
//   Author:        Jeff Schutte                                          |
//   Description:   Defines NPSS Independents and Dependents              |
//------------------------------------------------------------------------


npssstatus  = 0;
varslistcheck = 1;
#ifdef DEBUG2
  cout << "Design Failed to Converge - Resetting Guess Logic\n";
#endif
real con_count = 0;  real total_count = 0;  real Max_Diff;  real con_count_limit = 3;  real total_count_limit = 6;
int solver_flag = 0;  int num_passes;  int num_iterations;  int num_jacobians;  int num_Broydens;
real DOE_Base[12]; real DOE_Orig[12]; real DOE_Current[12]; real DOE_Delta[12]; real DOE_Step[12]; real DOE_Diff[12];
DOE_Orig[0] = FPR;  DOE_Base[0] = 1.625;  DOE_Delta[0] = 1.625*.001;
DOE_Orig[1] = OPR;  DOE_Base[1] = 36;  DOE_Delta[1] = 36*.001;
DOE_Orig[2] = EXTR;  DOE_Base[2] = 1.2;  DOE_Delta[2] = 1.2*.001;
DOE_Orig[3] = T3_max;  DOE_Base[3] = 1775;  DOE_Delta[3] = 1775*.001;
DOE_Orig[4] = T4_max;  DOE_Base[4] = 3362.5;  DOE_Delta[4] = 3362.5*.001;
DOE_Orig[5] = TOC_NcMap;  DOE_Base[5] = 1.055;  DOE_Delta[5] = 1.055*.001;
DOE_Orig[6] = Fan_Deff;  DOE_Base[6] = 0;  DOE_Delta[6] = 0.01*0.001;
DOE_Orig[7] = LPC_Deff;  DOE_Base[7] = 0;  DOE_Delta[7] = 0.01*0.001;
DOE_Orig[8] = LPT_Deff;  DOE_Base[8] = 0;  DOE_Delta[8] = 0.01*0.001;
DOE_Orig[9] = HPT_1VT;  DOE_Base[9] = 2450;  DOE_Delta[9] = 2450*.001;
DOE_Orig[10] = HPT_Temp;  DOE_Base[10] = 2350;  DOE_Delta[10] = 2350*.001;
DOE_Orig[11] = LPT_Temp;  DOE_Base[11] = 2050;  DOE_Delta[11] = 2050*.001;

for (i = 0; i < 12; i++){
   DOE_Step[i]=0.5*(DOE_Orig[i]-DOE_Base[i]);
   DOE_Current[i]=DOE_Base[i];
  }

num_iterations = toInt(parse("solverTemp", 2, 1, 0, "solverTemp.out"));
num_passes = toInt(parse("solverTemp", 3, 1, 0, "solverTemp.out"));
num_jacobians = toInt(parse("solverTemp", 4, 1, 0, "solverTemp.out"));
num_Broydens = toInt(parse("solverTemp", 5, 1, 0, "solverTemp.out"));
remove_file("solverTemp.out");

#ifdef DEBUG2
   cout << "num_iterations = "  << num_iterations << "\n";
   cout << "num_passes = "  << num_passes << "\n";
   cout << "num_jacobians = "  << num_jacobians << "\n";
   cout << "num_Broydens = "  << num_Broydens << "\n";
  cout << "Start While Loop\n";
#endif

while(solver_flag == 0) {
   for (i = 0; i < 12; i++){
     DOE_New[i]=DOE_Current[i]+DOE_Step[i];
    }

   #ifdef DEBUG2
      cout << "\n\n\n DOE_New = " <<  DOE_New << "\n";
   #endif

   // Get vars.list ready for replacement
   remove_file("vars.list");   remove_dir("out\\temp");   make_dir("out\\temp");
   copy("inp\\LCTVARStemplate_MDP.list", "out\\temp\\temp.var0");
   i = 0;

   // ----- Replace NPSS Variables -----
   replace_NPSS_Variables(); // found in setup_functions.npss

   // vars.list is ready for use
   move("out\\temp\\temp.var" + toStr(i), "vars.list");
   system("run run\\" + model + "_MDP_design.run" + NPSScontrols);
   npsscheck2 = toReal(parse("converge =", 12, 1, 0, "LPCDP.viewOut"));

   num_iterations += toInt(parse("solverTemp", 2, 1, 0, "solverTemp.out"));
   num_passes += toInt(parse("solverTemp", 3, 1, 0, "solverTemp.out"));
   num_jacobians += toInt(parse("solverTemp", 4, 1, 0, "solverTemp.out"));
   num_Broydens += toInt(parse("solverTemp", 5, 1, 0, "solverTemp.out"));
   remove_file("solverTemp.out");

   #ifdef DEBUG2
    cout << "num_iterations = "  << num_iterations << "\n";
    cout << "num_passes = "  << num_passes << "\n";
    cout << "num_jacobians = "  << num_jacobians << "\n";
    cout << "num_Broydens = "  << num_Broydens << "\n";
   #endif

   if (npsscheck2 == 0){
    #ifdef DEBUG2
    cout << "Failed to converge\n";
    #endif
     con_count += 1;  total_count += 1;

     if ((total_count > total_count_limit)||(con_count > con_count_limit)){
         #ifdef DEBUG2
      cout << "Failed to converge - Exceeded allowable iterations\n";
      #endif
      solver_flag = 1;     npssstatus = 0;
     } else {
      #ifdef DEBUG2
      cout << "Failed to converge - Recalulating step size\n";
      #endif
      for (i = 0; i < 12; i++){
      DOE_Step[i]=DOE_Step[i]/2;
       }
      }
    } else {
    #ifdef DEBUG2
    cout<< "Solver Converged check if DOE back to original\n";
    #endif
     for (i = 0; i < 12; i++){
     DOE_Diff[i]=abs((DOE_New[i]-DOE_Orig[i])/DOE_Orig[i]);
    }
     #ifdef DEBUG2
     cout << "DOE_Diff = "<< DOE_Diff << "\n";
     #endif
     Max_Diff =
max(DOE_Diff[0],DOE_Diff[1],DOE_Diff[2],DOE_Diff[3],DOE_Diff[4],DOE_Diff[5],DOE_Diff[6],DOE_Diff[7],DOE_Diff[8],D
OE_Diff[9],DOE_Diff[10] );
     if (Max_Diff < 0.0001) {
     solver_flag = 1;     npssstatus = 1;
     #ifdef DEBUG2
     cout << "DOE back to original \n";
     #endif
    } else {
     con_count = 0;  total_count += 1;
     if (total_count > total_count_limit) {
      solver_flag = 1;      npssstatus = 0;
     } else {
      for (i = 0; i < 12; i++){
       DOE_Current[i]= DOE_New[i];
       }
      }
    }
   }
  }

   if  ((solver_flag == 1) && (npssstatus == 1)) {
  //Solution found
  } else {
```

```
#ifdef DEBUG2
cout << "\n\n\nIncrementing original DOE\n";
cout << " DOE_Orig = " <<  DOE_Orig << "\n";
cout << " DOE_Delta = " <<  DOE_Delta << "\n";
#endif

for (i = 0; i < 12; i++){
 if (DOE_Orig[i]> DOE_Base[i]){
  DOE_New[i] = DOE_Orig[i] - DOE_Delta[i];
 } else {
  DOE_New[i] = DOE_Orig[i] + DOE_Delta[i];
 }
}

#ifdef DEBUG2
cout << " DOE_New = " <<  DOE_New << "\n";
#endif
remove_file("vars.list");   remove_dir("out\\temp");   make_dir("out\\temp");
copy("inp\\LCTVARStemplate_MDP.list", "out\\temp\\temp.var0");
i = 0;

// ----- Replace NPSS Variables -----
replace_NPSS_Variables(); // found in setup_functions.npss

// vars.list is ready for use
move("out\\temp\\temp.var" + toStr(i), "vars.list");
system("run run\\" + model + "_MDP_design.run" + NPSScontrols);

num_iterations += toInt(parse("solverTemp", 2, 1, 0, "solverTemp.out"));
num_passes += toInt(parse("solverTemp", 3, 1, 0, "solverTemp.out"));
num_jacobians += toInt(parse("solverTemp", 4, 1, 0, "solverTemp.out"));
num_Broydens += toInt(parse("solverTemp", 5, 1, 0, "solverTemp.out"));
remove_file("solverTemp.out");

#ifdef DEBUG2
 cout << "num_iterations = "  << num_iterations << "\n";
 cout << "num_passes = "  << num_passes << "\n";
 cout << "num_jacobians = "  << num_jacobians << "\n";
 cout << "num_Broydens = "  << num_Broydens << "\n";
#endif
npsscheck2 = toReal(parse("converge =", 12, 1, 0, "LPCDP.viewOut"));

if (npsscheck2 == 1) {
 npssstatus = 1;
} else {
 npssstatus = 0;
}

}
```

APPENDIX D

INITIAL ITERATE RANGES FOR EXPERIMENT 3

The following tables list the maximum and minimum values used in the initial iterate sensitivity analysis for experiment 3. The solution value represents the final value of the independent parameters when the solver reaches convergence. For most independents the max/min values were +/- 20% of the final solution unless such values exceeded physical limits.

Table 65. Ranges on HPCDP Initial Iterate for Sensitivity Analysis 3

Name	Solution	Min	Max
HPCDP.Ambient.W	2029.79	1623.83	2435.74
HPCDP.Burner.FAR	0.0162090	0.0129672	0.0194508
HPCDP.Fan.S_map.RlineMap	1.52222	1.21777	1.82666
HPCDP.HP_Shaft.Nmech	8096.85	6477.48	9716.22
HPCDP.HPC.S_map.RlineMap	2.00000	1.60000	2.40000
HPCDP.HPT.S_map.parmMap	6.11730	4.89384	7.34076
HPCDP.LP_Shaft.Nmech	1849.52	1479.62	2219.43
HPCDP.LPC.S_map.RlineMap	2.18216	1.74573	2.61860
HPCDP.LPT.S_map.parmMap	2.70841	2.16673	3.25009
HPCDP.Splitter.BPR	6.10836	4.88669	7.33003

Table 66. Ranges on LPCDP Initial Iterate for Sensitivity Analysis 3

Name	Solution	Min	Max
LPCDP.Ambient.W	1241.47	993.174	1489.76
LPCDP.Burner.FAR	0.0228348	0.0182678	0.0274017
LPCDP.Fan.S_map.PRdes	1.60000	1.28000	1.92000
LPCDP.LPC.S_map.PRdes	1.26859	1.05000	1.52231
LPCDP.LPC.S_map.effDes	0.902133	0.721706	0.950000
LPCDP.HPC.S_map.NcDes	1.05778	0.846228	1.10000
LPCDP.HPC.S_map.PRdes	17.5582	14.0466	21.0698
LPCDP.HPC.S_map.effDes	0.800936	0.640749	0.950000
LPCDP.HPC.S_map.RlineMap	2.41672	1.93338	2.90006
LPCDP.HPT.S_map.parmMap	5.65936	4.52749	6.79123
LPCDP.HPT.S_map.parmMapDes	5.65936	4.52749	6.79123
LPCDP.HPT.S_map.effDes	0.911999	0.729599	0.950000
LPCDP.HPT.S_map.parmNcDes	93.8422	75.0738	110.000
LPCDP.LPT.S_map.parmMap	4.42997	3.54398	5.31596
LPCDP.Splitter.BPR	6.25523	5.00419	7.50628
LPCDP.Bld3.Cool1.fracW	0.111136	0.0889086	0.133363
LPCDP.Bld3.Cool2.fracW	0.0830797	0.0664638	0.0996956
LPCDP.HPC.Cool1.fracBldW	0.0311630	0.0249304	0.0373957
LPCDP.HPC.Cool2.fracBldW	0.0166899	0.0133519	0.0200279
LPCDP.Inlet.Fl_O.MN	0.703486	0.562789	0.844183
LPCDP.Fan.Fl_O.MN	0.442760	0.354208	0.531312
LPCDP.Splitter.Fl_02.MN	0.470236	0.376188	0.564283
LPCDP.Splitter.Fl_01.MN	0.327789	0.262231	0.393347
LPCDP.Duct4.Fl_O.MN	0.331594	0.265275	0.397912
LPCDP.LPC.Fl_O.MN	0.366795	0.293436	0.440154
LPCDP.Duct6.Fl_O.MN	0.585565	0.468452	0.702678
LPCDP.Burner.Fl_O.MN	0.105275	0.0842202	0.126330
LPCDP.HPC.Fl_O.MN	0.390377	0.312301	0.468452
LPCDP.HPT.Fl_O.MN	0.383089	0.306472	0.459707
LPCDP.Duct11.Fl_O.MN	0.210996	0.168797	0.253196
LPCDP.BypBld.Fl_O.MN	0.467139	0.373711	0.560567
LPCDP.Duct13.Fl_O.MN	0.375317	0.300253	0.450380
LPCDP.Duct15.Fl_O.MN	0.478522	0.382817	0.574226
LPCDP.HP_Shaft.Nmech	8468.61	6774.89	10162.34
LPCDP.LP_Shaft.Nmech	2370.90	1896.72	2845.07

Table 67. Ranges on TKO Initial Iterate for Sensitivity Analysis 3

Name	Solution	Min	Max
TKO.Ambient.W	3128.19	2502.55	3753.82
TKO.Burner.FAR	0.0292176	0.0233741	0.0350612
TKO.Fan.S_map.RlineMap	1.79451	1.43561	2.15341
TKO.HP_Shaft.Nmech	9554.61	7643.69	11465.53
TKO.HPC.S_map.RlineMap	2.11405	1.69124	2.53685
TKO.HPT.S_map.parmMap	5.83817	4.67054	7.00581
TKO.LP_Shaft.Nmech	2594.35	2075.48	3113.22
TKO.LPC.S_map.RlineMap	2.07362	1.65890	2.48835
TKO.LPT.S_map.parmMap	4.03228	3.22582	4.83873
TKO.Splitter.BPR	6.09459	4.87568	7.31351

Table 68. Ranges on TOC1 Initial Iterate for Sensitivity Analysis 3

Name	Solution	Min	Max
TOC1.Ambient.W	1096.09	876.874	1315.31
TOC1.Burner.FAR	0.0249897	0.0199918	0.0299877
TOC1.Fan.S_map.RlineMap	2.11437	1.69150	2.53724
TOC1.HP_Shaft.Nmech	8755.87	7004.70	10507.05
TOC1.HPC.S_map.RlineMap	2.43701	1.94961	2.92441
TOC1.HPT.S_map.parmMap	5.64770	4.51816	6.77724
TOC1.LP_Shaft.Nmech	2495.17	1996.13	2994.20
TOC1.LPC.S_map.RlineMap	2.01552	1.61242	2.41863
TOC1.LPT.S_map.parmMap	4.47328	3.57863	5.36794
TOC1.Splitter.BPR	6.06167	4.84934	7.27401

APPENDIX E

INITIAL ITERATE CORRELATION FOR EXPERIMENT 3

The following tables show the correlation of the independent parameters initial values to a successful convergence of the solver for the initial iterate sensitivity analysis for experiment 3. In addition, each independent is listed with a histogram of % of converged cases versus independent parameter value. The bin sizes for the histogram are 10% of the range shown in Appendix D. The data shows that none of the independent parameters has a strong correlation to successful convergence indicating the difficulty in finding an initial iterate that will converge and demonstrates the necessity of developing a rigorous method for determining the initial iterate.

Table 69. Histogram Bins for Initial Iterate

Histogram Bins (% Cases Converged)											
1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13
🟥	🟥	🟪	🟪	🟦	🟦	⬛	🟨	🟨	🟩	🟩	🟩

Table 70. Correlation of HPCDP Initial Iterate to Successful Solver Convergence

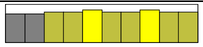
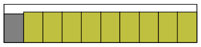
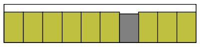
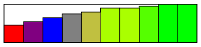
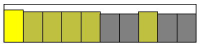
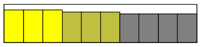
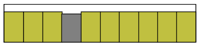
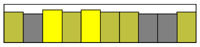
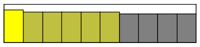
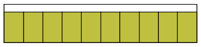
Independent	Correlation	Histogram
HPCDP.Ambient.W	0.0127	
HPCDP.Burner.FAR	0.0054	
HPCDP.Fan.S_map.RlineMap	-0.0046	
HPCDP.HP_Shaft.Nmech	0.1299	
HPCDP.HPC.S_map.RlineMap	-0.0177	
HPCDP.HPT.S_map.parmMap	-0.0331	
HPCDP.LP_Shaft.Nmech	0.0033	
HPCDP.LPC.S_map.RlineMap	-0.0085	
HPCDP.LPT.S_map.parmMap	-0.0258	
HPCDP.Splitter.BPR	-0.0027	

Table 71. Correlation of LPCDP Initial Iterate to Successful Solver Convergence

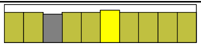
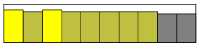
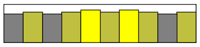
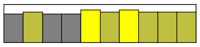
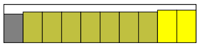
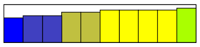
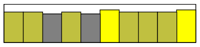
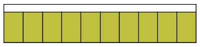
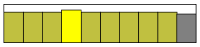
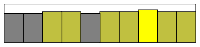
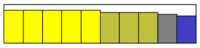
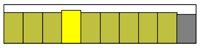
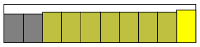
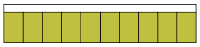
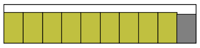
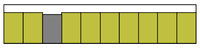
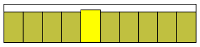
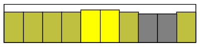
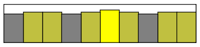
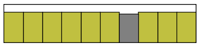
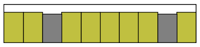
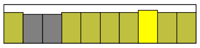
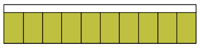
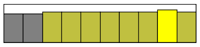
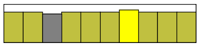
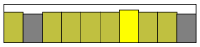
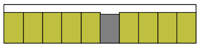
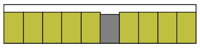
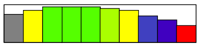
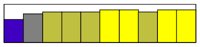
Independent	Correlation	Histogram
LPCDP.Ambient.W	-0.0026	
LPCDP.Burner.FAR	-0.0222	
LPCDP.Fan.S_map.PRdes	0.0034	
LPCDP.LPC.S_map.PRdes	0.0091	
LPCDP.LPC.S_map.effDes	0.0188	
LPCDP.HPC.S_map.NcDes	0.0568	
LPCDP.HPC.S_map.PRdes	0.0145	
LPCDP.HPC.S_map.effDes	0.0943	
LPCDP.HPC.S_map.RlineMap	0.009	
LPCDP.HPT.S_map.parmMap	-0.0008	
LPCDP.HPT.S_map.parmMapDes	0.0004	
LPCDP.HPT.S_map.effDes	0.0131	
LPCDP.HPT.S_map.parmNcDes	-0.0338	
LPCDP.LPT.S_map.parmMap	-0.0076	
LPCDP.Splitter.BPR	0.0135	
LPCDP.Bld3.Cool1.fracW	0.0041	
LPCDP.Bld3.Cool2.fracW	-0.0096	
LPCDP.HPC.Cool1.fracBldW	-0.0047	
LPCDP.HPC.Cool2.fracBldW	-0.0002	
LPCDP.Inlet.Fl_O.MN	-0.007	
LPCDP.Fan.Fl_O.MN	-0.0083	
LPCDP.Splitter.Fl_02.MN	0.0054	
LPCDP.Splitter.Fl_01.MN	0.0001	
LPCDP.Duct4.Fl_O.MN	-0.0019	
LPCDP.LPC.Fl_O.MN	0.0103	
LPCDP.Duct6.Fl_O.MN	-0.0024	
LPCDP.Burner.Fl_O.MN	0.0004	
LPCDP.HPC.Fl_O.MN	0.0112	
LPCDP.HPT.Fl_O.MN	-0.0003	
LPCDP.Duct11.Fl_O.MN	0.0052	
LPCDP.BypBld.Fl_O.MN	0.0002	
LPCDP.Duct13.Fl_O.MN	0.0021	
LPCDP.Duct15.Fl_O.MN	0.0061	
LPCDP.HP_Shaft.Nmech	-0.0796	
LPCDP.LP_Shaft.Nmech	0.0375	

Table 72. Correlation of TOC1 Initial Iterate to Successful Solver Convergence

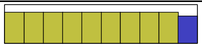
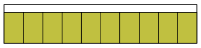
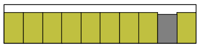
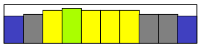
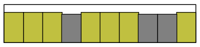
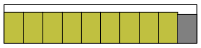
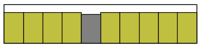
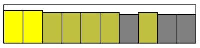
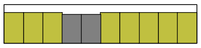
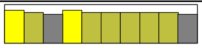
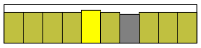
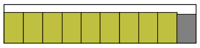
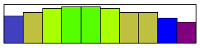
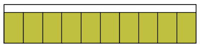
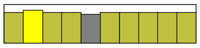
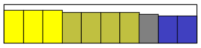
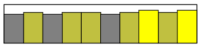
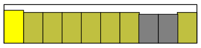
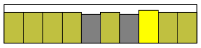
Independent	Correlation	Histogram
TOC1.Ambient.W	-0.0164	
TOC1.Burner.FAR	-0.0001	
TOC1.Fan.S_map.RlineMap	-0.0087	
TOC1.HP_Shaft.Nmech	-0.0129	
TOC1.HPC.S_map.RlineMap	-0.0096	
TOC1.HPT.S_map.parmMap	-0.0012	
TOC1.LP_Shaft.Nmech	0.0006	
TOC1.LPC.S_map.RlineMap	0.0046	
TOC1.LPT.S_map.parmMap	-0.0179	
TOC1.Splitter.BPR	-0.0011	

Table 73. Correlation of TKO Initial Iterate to Successful Solver Convergence

Independent	Correlation	Histogram
TKO.Ambient.W	-0.012	
TKO.Burner.FAR	-0.0026	
TKO.Fan.S_map.RlineMap	-0.0011	
TKO.HP_Shaft.Nmech	-0.0401	
TKO.HPC.S_map.RlineMap	-0.0005	
TKO.HPT.S_map.parmMap	-0.0058	
TKO.LP_Shaft.Nmech	-0.0328	
TKO.LPC.S_map.RlineMap	0.0136	
TKO.LPT.S_map.parmMap	-0.0137	
TKO.Splitter.BPR	-0.003	

APPENDIX F

EXPERIMENT 3 INDEPENDENT PARAMETER SOLUTION SPACE CORRELATION

The following figures graphically display the solution space for experiment 3. Each dot in the solution space represents the final converged value of the independent parameters for a unique combination of design variables. The solution space defines the area that the solver must work within in order to find the solution. If during any part of the iterative analysis the solver gets outside the solution space, the probability of the solver finding the solution is materially reduced. Therefore the initial iterate must start the solver within the solution space. The strong correlation between many of the independent parameters makes it very unlikely that randomly assigning values to the initial iterate will be successful. Therefore a means of finding an initial iterate that lies within the solution space must be devised.

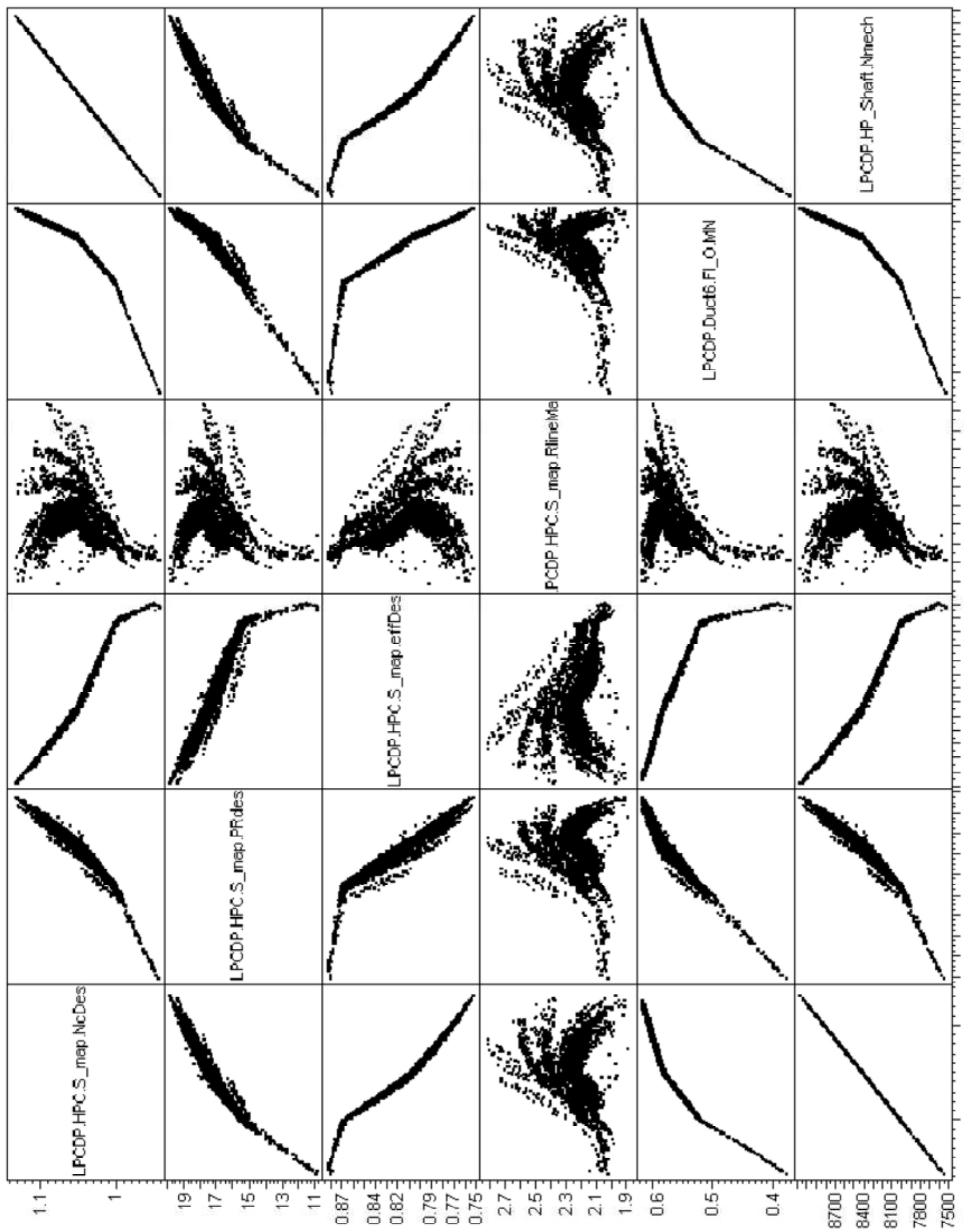


Figure 90. Multivariate Plot of LPCDP Independent Parameter Solution Space (1)

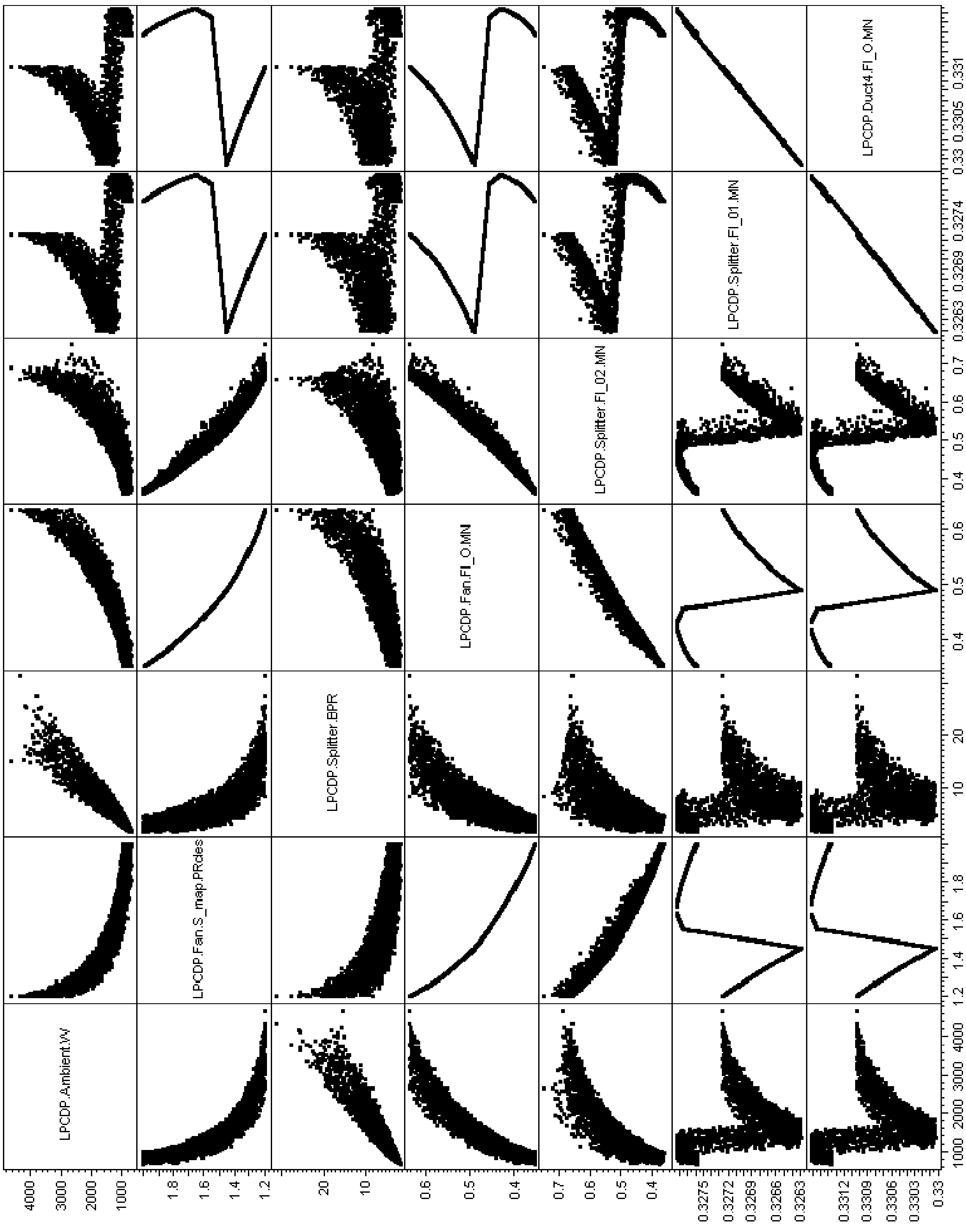


Figure 91. Multivariate Plot of LPCDP Independent Parameter Solution Space (2)

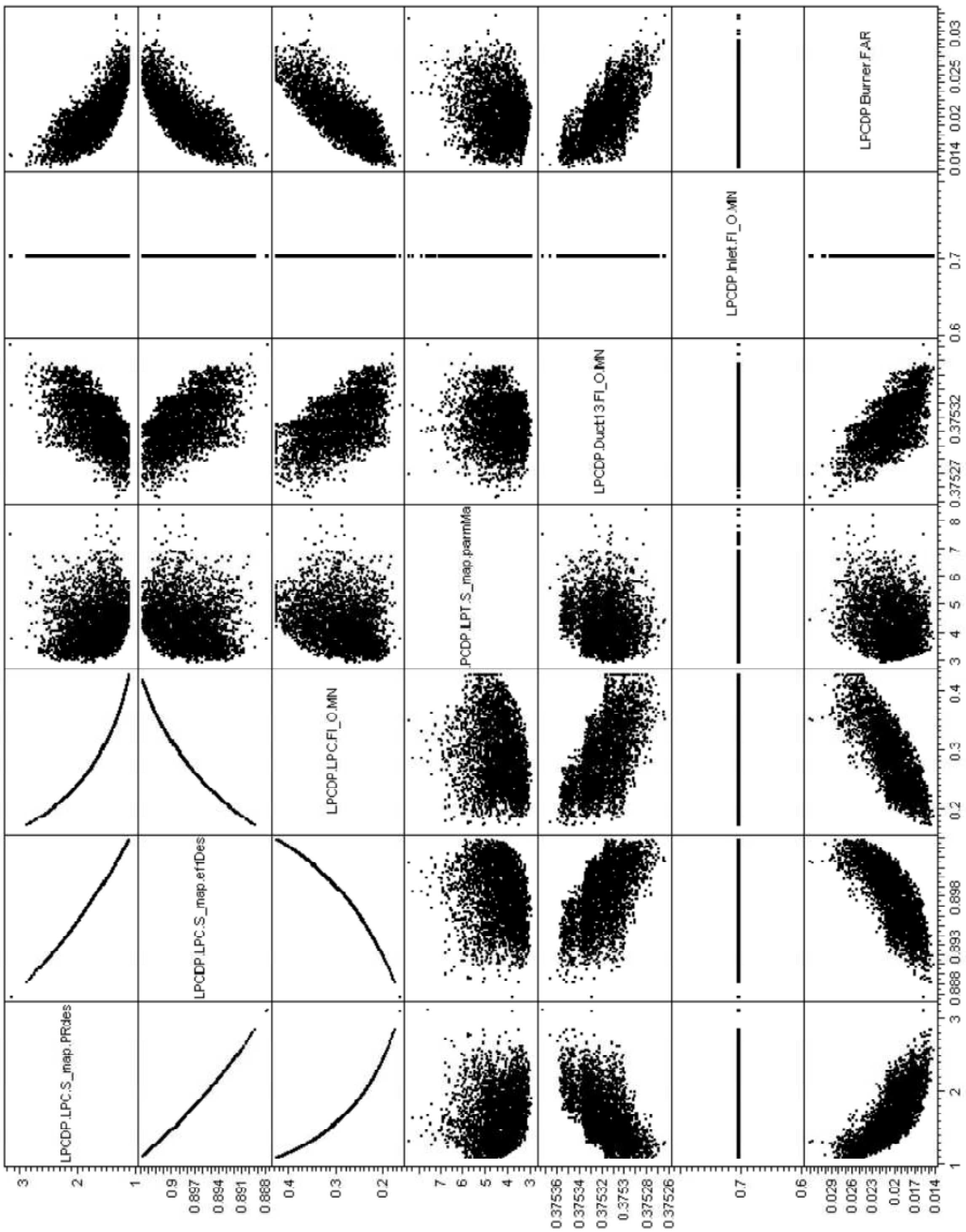


Figure 92. Multivariate Plot of LPCDP Independent Parameter Solution Space (2)

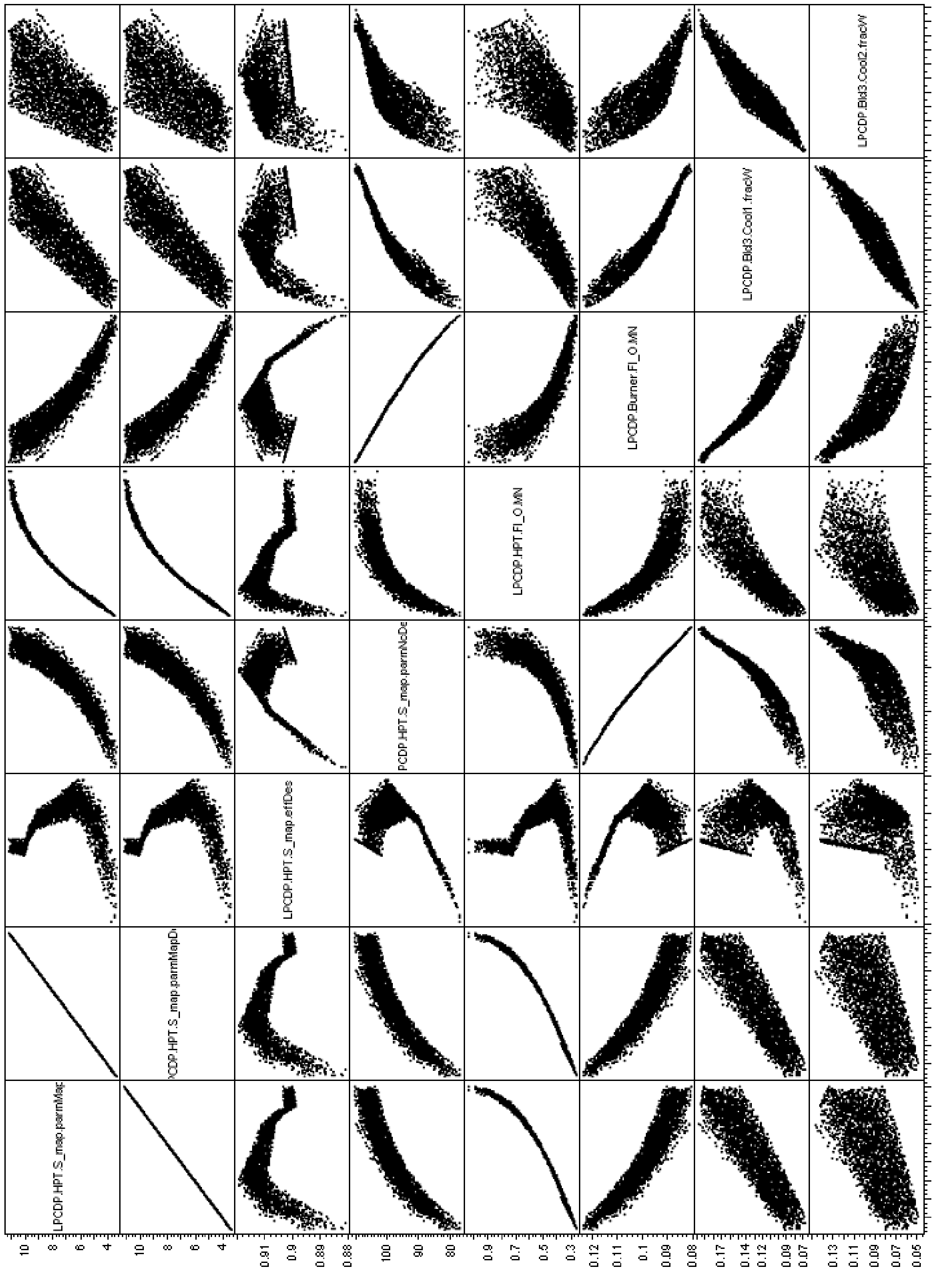


Figure 93. Multivariate Plot of LPCDP Independent Parameter Solution Space (4)

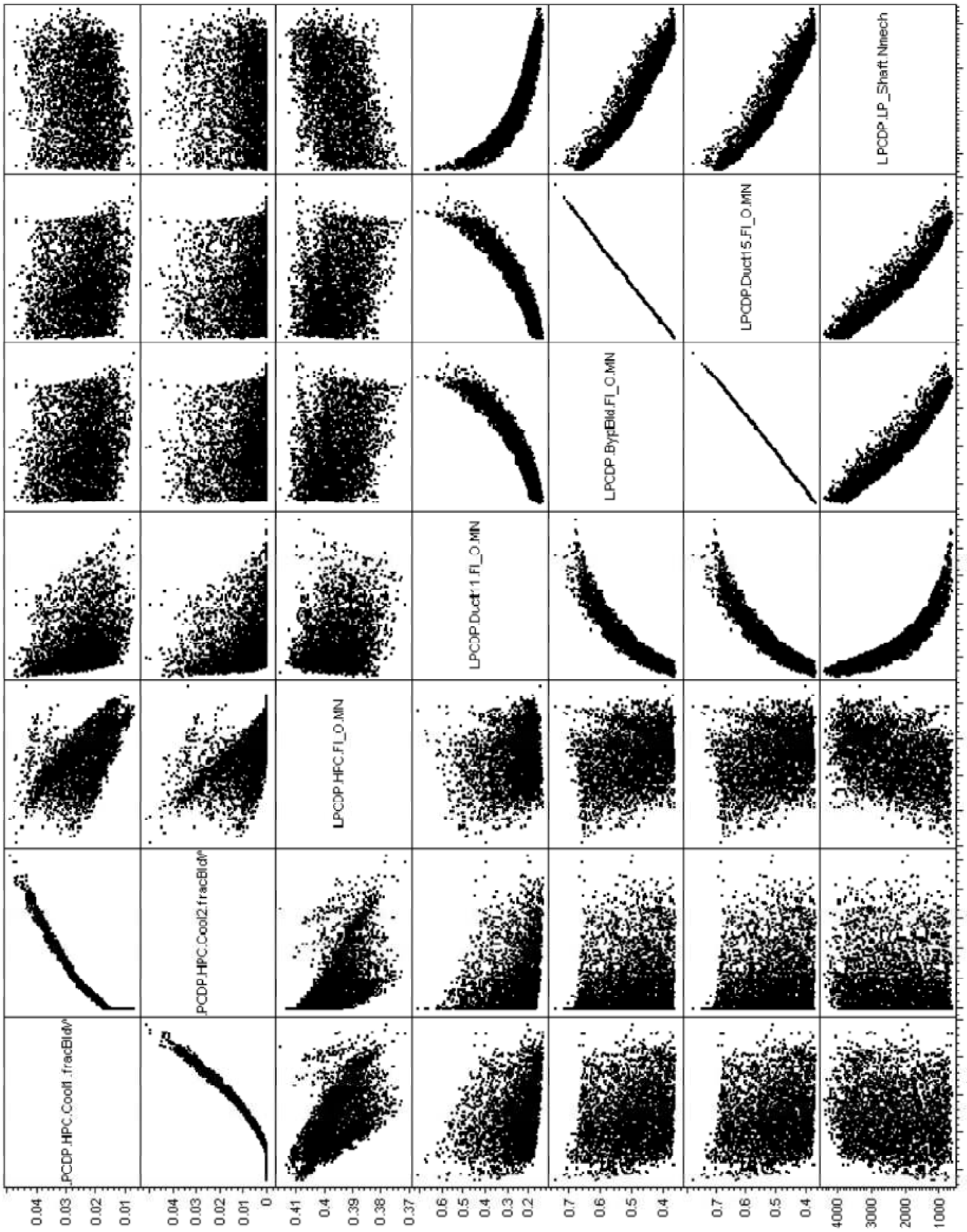


Figure 94. Multivariate Plot of LPCDP Independent Parameter Solution Space (5)

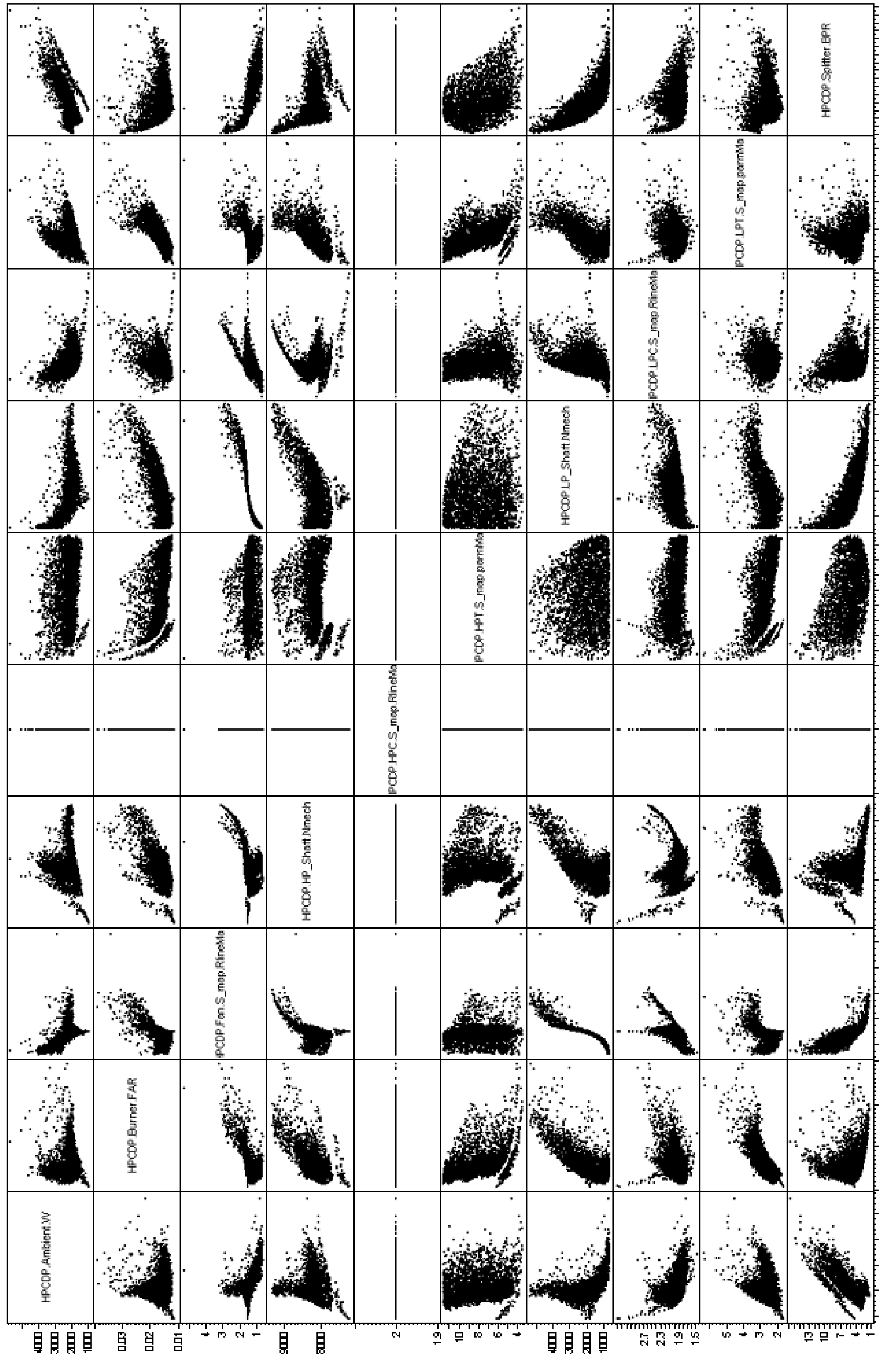


Figure 95. Multivariate Plot of HPCDP Independent Parameter Solution Space

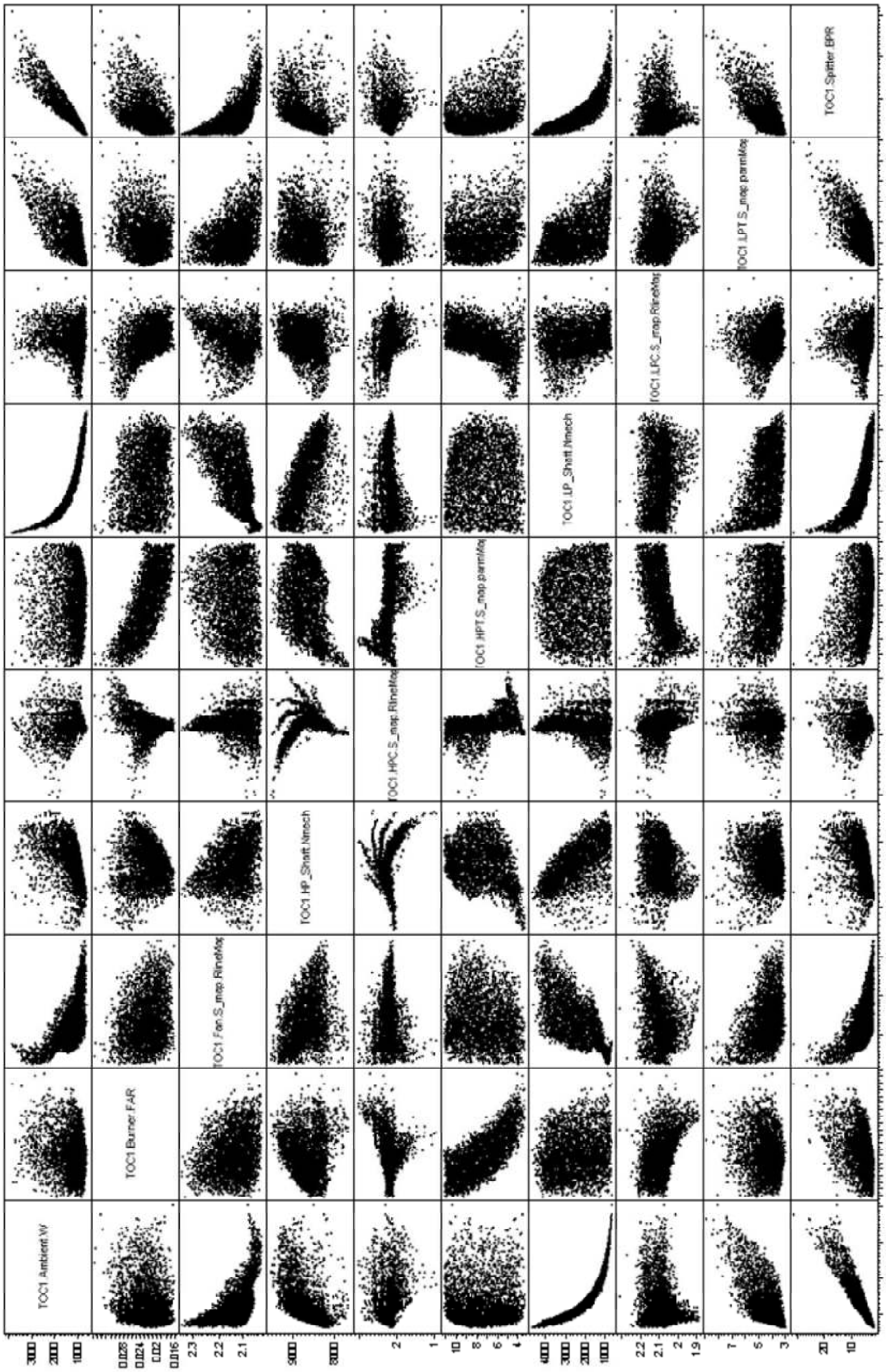


Figure 96. Multivariate Plot of TOC1 Independent Parameter Solution Space

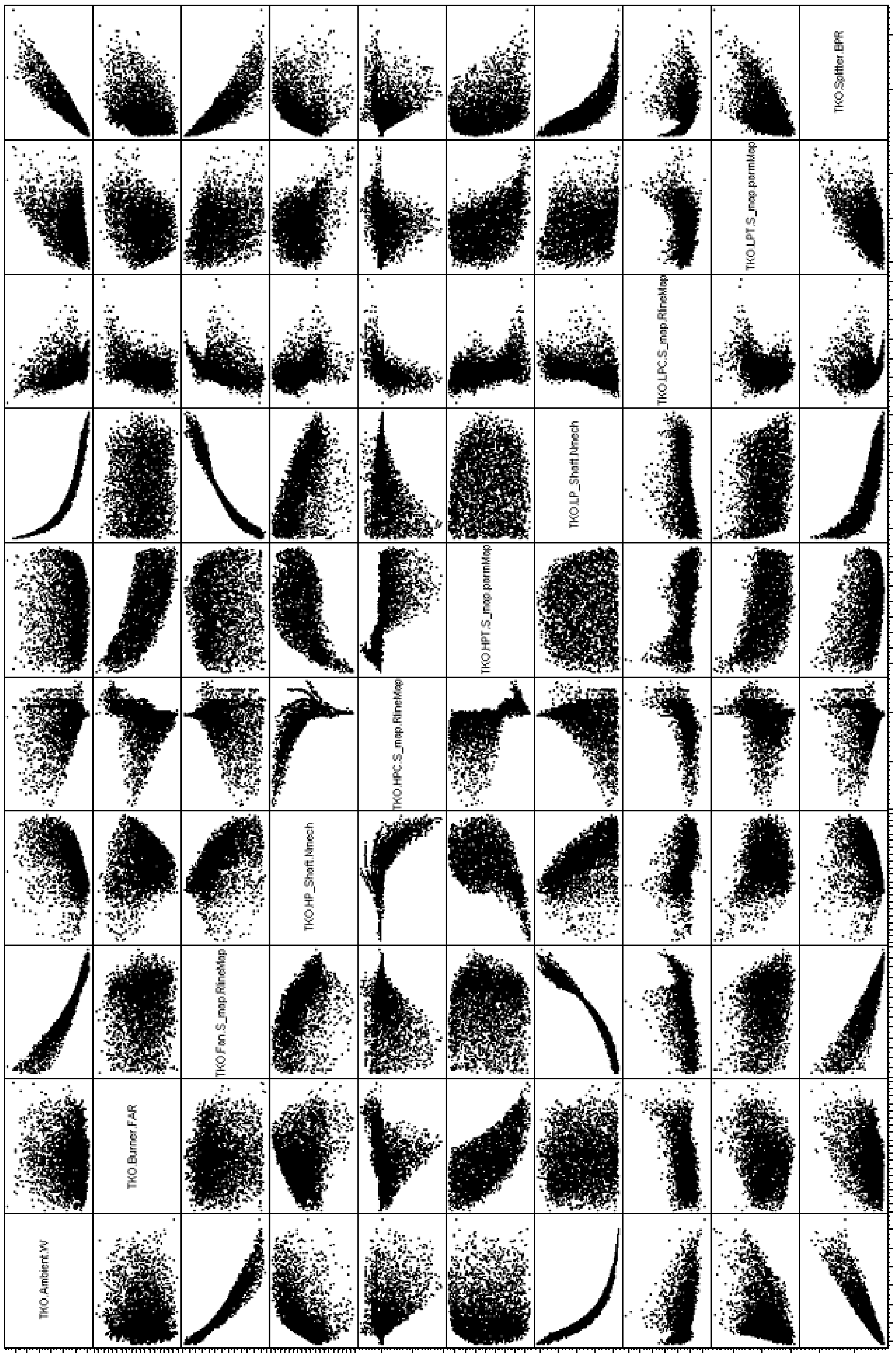


Figure 97. Multivariate Plot of TKO Independent Parameter Solution Space

Table 74. Experiment 3 Independent Parameter Solution Space Correlation

	LPCDP.Ambient.W	LPCDP.Burner.FAR	LPCDP.Fan.S_map.PRdes	LPCDP.LPC.S_map.PRdes	LPCDP.LPC.S_map.effDes	LPCDP.HPC.S_map.NcDes	LPCDP.HPC.S_map.PRdes	LPCDP.HPC.S_map.effDes	LPCDP.HPC.S_map.RlineMap	LPCDP.HPT.S_map.parmMap	LPCDP.HPT.S_map.parmMapDes	LPCDP.HPT.S_map.effDes	LPCDP.HPT.S_map.parmNcDes	LPCDP.LPT.S_map.parmMap	LPCDP.Splitter.BPR	LPCDP.Bld3.Cool1.fracW	LPCDP.Bld3.Cool2.fracW	LPCDP.HPC.Cool1.fracBldW	LPCDP.HPC.Cool2.fracBldW	LPCDP.Inlet.Fl_O.MN	LPCDP.Fan.Fl_O.MN	LPCDP.Splitter.Fl_02.MN	LPCDP.Splitter.Fl_01.MN	LPCDP.Duct4.Fl_O.MN	LPCDP.LPC.Fl_O.MN	LPCDP.Duct6.Fl_O.MN	LPCDP.Burner.Fl_O.MN	LPCDP.HPC.Fl_O.MN	LPCDP.HPT.Fl_O.MN	LPCDP.Duct11.Fl_O.MN	LPCDP.BypBld.Fl_O.MN	LPCDP.Duct13.Fl_O.MN	LPCDP.Duct15.Fl_O.MN	LPCDP.HP_Shaft.Nmech	LPCDP.LP_Shaft.Nmech	HPCDP.Ambient.W	HPCDP.Burner.FAR	HPCDP.Fan.S_map.RlineMap	HPCDP.HP_Shaft.Nmech	HPCDP.HPC.S_map.RlineMap	HPCDP.HPT.S_map.parmMap	HPCDP.LP_Shaft.Nmech	HPCDP.LPC.S_map.RlineMap	HPCDP.LPT.S_map.parmMap	HPCDP.Splitter.BPR	TOC1.Ambient.W	TOC1.Burner.FAR	TOC1.Fan.S_map.RlineMap	TOC1.HP_Shaft.Nmech	TOC1.HPC.S_map.RlineMap	TOC1.HPT.S_map.parmMap	TOC1.LP_Shaft.Nmech	TOC1.LPC.S_map.RlineMap	TOC1.LPT.S_map.parmMap	TOC1.Splitter.BPR	TKO.Ambient.W	TKO.Burner.FAR	TKO.Fan.S_map.RlineMap	TKO.HP_Shaft.Nmech	TKO.HPC.S_map.RlineMap	TKO.HPT.S_map.parmMap	TKO.LP_Shaft.Nmech	TKO.LPC.S_map.RlineMap	TKO.LPT.S_map.parmMap
LPCDP.Burner.FAR	0.2	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.Fan.S_map.PRdes	-0.9	-0.1	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.LPC.S_map.PRdes	0.2	-0.8	-0.2	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.LPC.S_map.effDes	-0.2	0.8	0.2	-1.0	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.HPC.S_map.NcDes	0.5	0.0	-0.6	0.2	-0.2	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.HPC.S_map.PRdes	0.4	-0.2	-0.5	0.3	-0.3	1.0	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.HPC.S_map.effDes	-0.5	0.0	0.6	-0.2	0.2	-1.0	-0.9	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.HPC.S_map.RlineMap	0.2	0.5	-0.3	-0.4	0.4	0.3	0.2	-0.4	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.HPT.S_map.parmMap	0.0	-0.8	0.0	0.9	-0.9	0.3	0.5	-0.3	-0.4	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.HPT.S_map.parmMapDes	0.0	-0.8	0.0	0.9	-0.9	0.3	0.5	-0.3	-0.4	1.0	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.HPT.S_map.effDes	-0.2	0.1	0.3	-0.4	0.3	0.3	0.3	-0.3	0.2	-0.2	-0.2	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.HPT.S_map.parmNcDes	-0.2	-0.6	0.1	0.7	-0.7	0.4	0.5	-0.4	-0.4	0.9	0.9	0.2	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.LPT.S_map.parmMap	0.6	0.2	-0.4	-0.2	0.2	0.1	0.0	-0.1	0.1	-0.3	-0.3	-0.3	-0.4	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.Splitter.BPR	0.9	0.4	-0.8	-0.1	0.1	0.4	0.2	-0.4	0.4	-0.3	-0.3	-0.2	-0.4	0.7	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.Bld3.Cool1.fracW	-0.2	-0.4	0.1	0.7	-0.7	0.3	0.4	-0.3	-0.3	0.9	0.9	0.0	0.9	-0.4	-0.4	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.Bld3.Cool2.fracW	-0.2	-0.1	0.2	0.4	-0.4	0.3	0.3	-0.3	-0.2	0.6	0.6	0.1	0.8	-0.4	-0.4	0.9	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.HPC.Cool1.fracBldW	0.0	0.9	0.1	-0.9	0.9	-0.3	-0.4	0.3	0.4	-0.9	-0.9	0.2	-0.7	0.2	0.3	-0.6	-0.2	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.HPC.Cool2.fracBldW	0.0	0.9	0.0	-0.8	0.8	-0.3	-0.5	0.3	0.4	-0.9	-0.9	0.0	-0.8	0.2	0.3	-0.6	-0.3	1.0	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.Inlet.Fl_O.MN	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.Fan.Fl_O.MN	0.9	0.1	-1.0	0.2	-0.2	0.6	0.5	-0.6	0.3	0.0	0.0	-0.3	-0.1	0.4	0.9	-0.1	-0.2	-0.1	0.0	0.0	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.Splitter.Fl_02.MN	0.9	0.0	-1.0	0.4	-0.4	0.6	0.5	-0.6	0.3	0.1	0.1	-0.3	0.0	0.3	0.8	0.0	-0.1	-0.2	-0.1	0.0	1.0	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.Splitter.Fl_01.MN	-0.5	-0.1	0.7	-0.2	0.2	-0.4	-0.4	0.4	-0.2	0.0	0.0	0.2	0.1	-0.2	-0.5	0.1	0.1	0.1	0.0	0.0	-0.7	-0.7	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.Duct4.Fl_O.MN	-0.5	-0.1	0.7	-0.2	0.2	-0.4	-0.4	0.4	-0.2	0.0	0.0	0.2	0.1	-0.2	-0.5	0.1	0.1	0.1	0.0	0.0	-0.7	-0.7	1.0	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.LPC.Fl_O.MN	-0.2	0.8	0.2	-1.0	1.0	-0.2	-0.3	0.2	0.5	-0.9	-0.9	0.3	-0.7	0.2	0.1	-0.7	-0.4	0.9	0.8	0.0	-0.2	-0.4	0.2	0.2	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.Duct6.Fl_O.MN	0.4	-0.1	-0.5	0.2	-0.2	0.9	1.0	-0.9	0.4	0.4	0.4	0.3	0.4	0.0	0.3	0.3	0.3	-0.3	-0.4	0.0	0.5	0.5	-0.3	-0.3	-0.2	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.Burner.Fl_O.MN	0.2	0.6	-0.1	-0.7	0.8	-0.4	-0.5	0.3	0.4	-0.9	-0.9	-0.1	-1.0	0.4	0.4	-1.0	-0.8	0.7	0.8	0.0	0.1	0.0	-0.1	-0.1	0.8	-0.4	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.HPC.Fl_O.MN	-0.4	-0.8	0.3	0.5	-0.5	-0.5	-0.4	0.5	-0.3	0.4	0.4	-0.4	0.2	-0.2	-0.5	0.2	0.0	-0.6	-0.5	0.0	-0.3	-0.3	0.3	0.3	-0.5	-0.4	-0.3	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.HPT.Fl_O.MN	0.0	-0.7	-0.1	0.9	-0.9	0.3	0.4	-0.3	-0.3	1.0	1.0	-0.3	0.9	-0.2	-0.3	0.9	0.6	-0.8	-0.8	0.0	0.1	0.2	-0.1	-0.1	-0.9	0.3	-0.9	0.4	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.Duct11.Fl_O.MN	0.9	-0.1	-0.9	0.4	-0.4	0.6	0.5	-0.6	0.1	0.3	0.3	-0.2	0.1	0.4	0.8	0.1	0.0	-0.3	-0.2	0.0	0.9	0.9	-0.6	-0.6	-0.4	0.5	-0.1	-0.2	0.3	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.BypBld.Fl_O.MN	0.9	0.0	-1.0	0.4	-0.4	0.6	0.5	-0.6	0.3	0.1	0.1	-0.3	0.0	0.3	0.8	0.0	-0.1	-0.2	-0.1	0.0	1.0	1.0	-0.7	-0.7	-0.4	0.5	0.0	-0.3	0.2	0.9	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.Duct13.Fl_O.MN	0.1	-0.7	0.0	0.6	-0.6	0.2	0.3	-0.1	-0.3	0.6	0.6	-0.1	0.5	0.1	-0.1	0.4	0.2	-0.7	-0.7	0.0	0.0	0.1	0.0	0.0	-0.6	0.2	-0.5	0.4	0.6	0.2	0.1	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.Duct15.Fl_O.MN	0.9	0.0	-1.0	0.4	-0.4	0.6	0.5	-0.6	0.2	0.1	0.1	-0.3	0.0	0.3	0.8	0.0	-0.1	-0.2	-0.1	0.0	1.0	1.0	-0.7	-0.7	-0.4	0.5	0.0	-0.3	0.2	0.9	1.0	0.1	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.HP_Shaft.Nmech	0.5	0.0	-0.6	0.2	-0.2	1.0	1.0	-1.0	0.3	0.3	0.3	0.3	0.4	0.1	0.4	0.3	0.3	-0.3	-0.3	0.0	0.6	0.6	-0.4	-0.4	-0.2	0.9	-0.4	-0.5	0.3	0.6	0.6	0.2	0.6	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
LPCDP.LP_Shaft.Nmech	-0.9	-0.1	1.0	-0.2	0.2	-0.6	-0.5	0.6	-0.3	0.0	0.0	0.3	0.2	-0.5	-0.8	0.2	0.2	0.1	0.0	0.0	-1.0	-1.0	0.7	0.7	0.2	-0.5	-0.2	0.4	0.0	-0.9	-1.0	0.0	-1.0	-0.6	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
HPCDP.Ambient.W	0.9	0.1	-0.8	0.2	-0.2	0.3	0.1	-0.2	0.0	-0.2	-0.2	-0.2	-0.3	0.5	0.8	-0.4	-0.4	0.0	0.0	0.0	0.8	0.8	-0.5	-0.5	-0.2	0.1	0.3	-0.2	-0.2	0.8	0.8	0.0	0.8	0.3	-0.8	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
HPCDP.Burner.FAR	-0.4	0.2	0.5	-0.2	0.2	-0.7	-0.7	0.7	-0.3	-0.4	-0.4	0.1	-0.2	-0.3	-0.3	-0.2	0.0	0.5	0.4	0.0	-0.4	-0.5	0.3	0.3	0.2	-0.8	0.2	0.0	-0.4	-0.5	-0.5	-0.3	-0.5	-0.7	0.5	0.0	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
HPCDP.Fan.S_map.RlineMap	-0.8	-0.2	0.8	-0.1	0.1	-0.6	-0.6	0.7	-0.4	0.0	0.0	0.2	0.1	-0.5	-0.8	0.1	0.1	0.1	0.0	0.0	-0.8	-0.8	0.5	0.5	0.1	-0.6	-0.1	0.4	-0.1	-0.8	-0.8	-0.1	-0.8	-0.6	0.9	-0.6	0.7	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
HPCDP.HP_Shaft.Nmech	-0.2	-0.4	0.3	0.3	-0.3	-0.3	-0.3	0.4	-0.4	0.1	0.1	0.3	0.2	-0.5	-0.3	0.2	0.1	-0.2	-0.3	0.0	-0.2	-0.2	0.2	0.2	-0.3	-0.3	-0.2	0.3	0.1	-0.2	-0.2	0.1	-0.2	-0.3	0.4	0.1	0.7	0.6	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
HPCDP.HPC.S_map.RlineMap	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
HPCDP.HPT.S_map.parmMap	0.0	-0.7	-0.1	0.9	-0.9	0.4	0.5	-0.4	-0.3	1.0	1.0	-0.2	0.9	-0.3	-0.3	0.9	0.6	-0.9	-0.8	0.0	0.0	0.2	0.0	0.0	-0.9	0.4	-0.9	0.4	1.0	0.3	0.2	0.6	0.2	0.4	0.0	-0.2	-0.4	-0.1	0.1	0.0	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
HPCDP.LP_Shaft.Nmech	-0.8	-0.2	0.9	-0.1	0.1	-0.7	-0.6	0.7	-0.4	0.0	0.0	0.3	0.1	-0.5	-0.7	0.1	0.2	0.1	0.0	0.0	-0.9	-0.8	0.6	0.6	0.1	-0.6	-0.1	0.4	-0.1	-0.8	-0.8	-0.1	-0.8	-0.7	0.9	-0.5	0.7	0.9	0.6	0.0	-0.1	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
HPCDP.LPC.S_map.RlineMap	-0.5	0.2	0.6	-0.4	0.5	-0.3	-0.3	0.3	0.1	-0.3	-0.3	0.3	-0.1	0.0	-0.4	-0.1	0.1	0.4	0.3	0.0	-0.6	-0.6	0.4	0.4	0.5	-0.3	0.1	0.0	-0.3	-0.6	-0.6	-0.1	-0.6	-0.3	0.6	-0.5	0.3	0.6	0.0	0.0	-0.3	0.5	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
HPCDP.LPT.S_map.parmMap	-0.3	0.0	0.4	-0.2	0.2	-0.7	-0.7	0.7	-0.4	-0.4	-0.4	0.0	-0.4	0.0	-0.1	-0.4	-0.4	0.3	0.2	0.0	-0.4	-0.4	0.3	0.3	0.2	-0.8	0.4	0.2	-0.5	-0.4	-0.4	-0.2	-0.4	-0.7	0.4	0.2	0.8	0.5	0.6	0.0	-0.5	0.6	0.2	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
HPCDP.Splitter.BPR	0.8	0.5	-0.8	-0.3	0.3	0.3	0.2	-0.4	0.4	-0.5	-0.5	-0.2	-0.6	0.7	1.0	-0.5	-0.5	0.4	0.4	0.0	0.8	0.7	-0.5	-0.5	0.3	0.2	0.6	-0.5	-0.4	0.7	0.7	-0.2	0.7	0.3	-0.8	0.7	-0.3	-0.7	-0.5	0.0	-0.4	-0.8	-0.3	-0.2	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
TOC1.Ambient.W	1.0	0.2	-0.9	0.2	-0.2	0.5	0.4	-0.5	0.2	0.0	0.0	-0.2	-0.2	0.6	0.9	-0.2	-0.2	0.0	0.0	0.0	0.9	0.9	-0.5	-0.5	-0.2	0.4	0.2	-0.4	0.0	0.9	0.9	0.1	0.9	0.5	-0.9	0.9	-0.4	-0.8	-0.2	0.0	0.0	-0.8	-0.5	-0.3	0.8	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
TOC1.Burner.FAR	0.2	1.0	-0.2	-0.8	0.8	0.0	-0.2	0.0	0.6	-0.8	-0.8	0.1	-0.6	0.2	0.4	-0.4	-0.1	0.9	0.9	0.0	0.2	0.1	-0.1	-0.1	0.8	-0.1	0.6	-0.7	-0.7	0.0	0.1	-0.7	0.1	0.0	-0.2	0.1	0.2	-0.2	-0.4	0.0	-0.7	-0.2	0.2	0.0	0.5	0.2	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
TOC1.Fan.S_map.RlineMap	-0.7	-0.3	0.6	0.0	-0.1	-0.5	-0.4	0.5	-0.3	0.2	0.2	0.2	0.3	-0.4	-0.6	0.3	0.3	-0.1	-0.1	0.0	-0.6	-0.6	0.4	0.4	-0.1	-0.4	-0.3	0.5	0.1	-0.6	-0.6	0.2	-0.6	-0.5	0.7	-0.5	0.3	0.6	0.4	0.0	0.1	0.7	0.4	0.2	-0.7	-0.6	-0.3	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
TOC1.HP_Shaft.Nmech	0.5	0.0	-0.6	0.2	-0.2	1.0	0.9	-1.0	0.4	0.3	0.3	0.3	0.4	0.1	0.4	0.3	0.3	-0.3	-0.3	0.0	0.6	0.6	-0.5	-0.5	-0.2	0.9	-0.4	-0.5	0.3	0.6	0.6	0.2	0.6	1.0	-0.6	0.3	-0.7	-0.6	-0.3	0.0	0.4	-0.7	-0.3	-0.7	0.3	0.5	0.0	-0.4	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
TOC1.HPC.S_map.RlineMap	-0.1	0.4	0.1	-0.5	0.5	-0.3	-0.3	0.2	0.7	-0.5	-0.5	0.0	-0.5	0.1	0.1	-0.4	-0.3	0.4	0.5	0.0	-0.1	-0.2	0.1	0.1	0.5	-0.1	0.5	0.1	-0.4	-0.3	-0.2	-0.3	-0.2	-0.3	0.1	-0.2	0.0	0.0	-0.3	0.0	-0.4	0.0	0.2	0.0	0.2	-0.1	0.4	0.0	-0.2	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·	·
TOC1.HPT.S_map.parmMap	0.0	-0.7	-0.1	0.9	-0.9	0.4	0.5	-0.4	-0.4	1.0	1.0	-0.2	0.9	-0.3	-0.3	0.9	0.7	-0.9	-0.9	0.0	0.0	0.2	0.0	0.0	-0.9	0.4	-0.9	0.4	1.0	0.3	0.2	0.6	0.2	0.4	0.0	-0.2	-0.4	-0.1	0.1	0.0	1.0	-0.1	-0.3	-0.5	-0.5	0.0	-0.7	0.1	0.4	-0.5	1.0	·	·	·	·	·	·	·	·	·	·	·	·	·
TOC1.LP_Shaft.Nmech	-0.9	-0.2	1.0	-0.2	0.2	-0.6	-0.5	0.6	-0.3	0.0	0.0	0.3	0.2	-0.5	-0.8	0.2	0.2	0.1	0.0	0.0	-1.0	-1.0	0.7	0.7	0.2	-0.5	-0.2	0.4	0.0	-0.9	-1.0	0.0	-1.0	-0.6	1.0	-0.8	0.5	0.9	0.4	0.0	0.0	0.9	0.6	0.4	-0.8	-0.9	-0.2	0.7	-0.6	0.1	0.0	1.0	·	·	·	·	·	·	·	·	·	·	·	·
TOC1.LPC.S_map.RlineMap	0.2	-0.6	-0.3	0.6	-0.7	0.0	0.1	0.0	-0.3	0.5	0.5	-0.2	0.4	-0.2	0.0	0.4	0.2	-0.6	-0.5	0.0	0.3	0.4	-0.2	-0.2	-0.7	0.0	-0.4	0.4	0.5	0.4	0.4	0.4	0.4	0.0	-0.2	0.3	0.0	-0.1	0.4	0.0	0.5	0.0	-0.5	0.0	-0.1	0.2	-0.5	0.3	0.2	-0.4	0.5	-0.2	1.0	·	·	·	·	·	·	·	·	·	·	·
TOC1.LPT.S_map.parmMap	0.7	0.2	-0.4	-0.2	0.2	0.1	0.0	-0.1	0.1	-0.3	-0.3	-0.3	-0.4	1.0	0.7	-0.4	-0.4	0.1	0.2	0.0	0.4	0.3	-0.3	-0.2	0.2	0.1	0.4	-0.2	-0.2	0.5	0.3	0.1	0.3	0.1	-0.5	0.5	-0.3	-0.5	-0.5	0.0	-0.2	-0.5	0.0	0.0	0.7	0.7	0.2	-0.4	0.1	0.1	-0.3	-0.5	-0.2	1.0	·	·	·	·	·	·	·	·	·	·
TOC1.Splitter.BPR	0.9	0.4	-0.8	-0.1	0.1	0.4	0.2	-0.4	0.4	-0.3	-0.3	-0.2	-0.4	0.7	1.0	-0.4	-0.4	0.3	0.3	0.0	0.9	0.8	-0.5	-0.5	0.1	0.3	0.4	-0.5	-0.3	0.8	0.8	-0.1	0.8	0.4	-0.8	0.8	-0.3	-0.8	-0.3	0.0	-0.3	-0.7	-0.4	-0.1	1.0	0.9	0.4	-0.7	0.4	0.1	-0.3	-0.8	0.0	0.7	1.0	·	·	·	·	·	·	·	·	·
TKO.Ambient.W	1.0	0.2	-0.9	0.2	-0.2	0.5	0.4	-0.5	0.3	0.0	0.0	-0.2	-0.2	0.7	0.9	-0.2	-0.2	0.0	0.0	0.0	0.9	0.9	-0.6	-0.6	-0.2	0.4	0.2	-0.4	0.0	0.9	0.9	0.1	0.9	0.5	-0.9	0.8	-0.4	-0.8	-0.3	0.0	0.0	-0.8	-0.5	-0.3	0.9	1.0	0.2	-0.7	0.5	-0.1	0.0	-0.9	0.2	0.7	0.9	1.0	·	·	·	·	·	·	·	·
TKO.Burner.FAR	0.1	1.0	-0.1	-0.8	0.8	0.0	-0.2	0.0	0.5	-0.7	-0.7	0.1	-0.5	0.2	0.3	-0.3	0.0	0.9	0.9	0.0	0.1	0.0	0.0	0.0	0.8	-0.1	0.5	-0.7	-0.6	-0.1	0.0	-0.6	0.0	0.0	-0.1	0.0	0.3	-0.1	-0.4	0.0	-0.7	-0.1	0.3	0.0	0.4	0.1	1.0	-0.2	0.0	0.4	-0.7	-0.1	-0.5	0.2	0.3	0.1	1.0	·	·	·	·	·	·	·
TKO.Fan.S_map.RlineMap	-0.9	-0.2	0.9	-0.2	0.2	-0.6	-0.5	0.6	-0.3	0.0	0.0	0.3	0.2	-0.5	-0.9	0.2	0.3	0.0	0.0	0.0	-1.0	-0.9	0.7	0.7	0.2	-0.5	-0.2	0.4	0.0	-0.9	-0.9	0.0	-0.9	-0.6	1.0	-0.8	0.5	0.9	0.3	0.0	0.0	0.9	0.6	0.4	-0.8	-0.9	-0.2	0.7	-0.6	0.1	0.0	1.0	-0.2	-0.5	-0.9	-1.0	-0.1	1.0	·	·	·	·	·	·
TKO.HP_Shaft.Nmech	0.4	-0.1	-0.5	0.3	-0.3	1.0	0.9	-0.9	0.2	0.5	0.5	0.3	0.5	0.1	0.3	0.4	0.4	-0.4	-0.4	0.0	0.5	0.5	-0.4	-0.4	-0.3	0.9	-0.5	-0.4	0.4	0.6	0.5	0.3	0.5	1.0	-0.5	0.2	-0.7	-0.6	-0.3	0.0	0.5	-0.6	-0.2	-0.7	0.2	0.4	-0.1	-0.3	1.0	-0.3	0.5	-0.5	0.1	0.1	0.3	0.5	-0.1	-0.5	1.0	·	·	·	·	·
TKO.HPC.S_map.RlineMap	-0.4	0.2	0.4	-0.4	0.4	-0.6	-0.6	0.6	0.4	-0.5	-0.5	-0.1	-0.5	-0.1	-0.2	-0.5	-0.4	0.4	0.4	0.0	-0.4	-0.4	0.3	0.3	0.5	-0.4	0.5	0.3	-0.5	-0.5	-0.4	-0.4	-0.4	-0.6	0.4	-0.3	0.3	0.3	0.0	0.0	-0.5	0.4	0.2	0.4	-0.1	-0.4	0.2	0.2	-0.6	0.7	-0.5	0.4	-0.3	-0.2	-0.2	-0.4	0.2	0.4	-0.7	1.0	·	·	·	·
TKO.HPT.S_map.parmMap	0.0	-0.7	-0.1	0.9	-0.9	0.4	0.5	-0.4	-0.4	1.0	1.0	-0.2	0.9	-0.3	-0.3	0.9	0.6	-0.9	-0.8	0.0	0.1	0.2	-0.1	-0.1	-0.9	0.4	-0.9	0.4	1.0	0.3	0.2	0.6	0.2	0.4	0.0	-0.1	-0.4	-0.1	0.1	0.0	1.0	-0.1	-0.3	-0.5	-0.4	0.0	-0.7	0.1	0.4	-0.5	1.0	0.0	0.5	-0.2	-0.3	0.0	-0.7	0.0	0.5	-0.5	1.0	·	·	·
TKO.LP_Shaft.Nmech	-0.9	-0.2	1.0	-0.2	0.2	-0.6	-0.5	0.6	-0.3	0.0	0.0	0.3	0.2	-0.5	-0.8	0.2	0.3	0.1	0.0	0.0	-1.0	-0.9	0.7	0.7	0.2	-0.5	-0.2	0.4	0.0	-0.9	-0.9	0.0	-0.9	-0.6	1.0	-0.7	0.6	0.9	0.4	0.0	0.0	0.9	0.5	0.4	-0.8	-0.9	-0.2	0.7	-0.6	0.1	0.0	1.0	-0.2	-0.5	-0.8	-0.9	-0.1	1.0	-0.5	0.4	0.0	1.0	·	·
TKO.LPC.S_map.RlineMap	-0.3	0.3	0.3	-0.4	0.4	-0.4	-0.4	0.4	0.3	-0.4	-0.4	0.0	-0.3	0.1	-0.1	-0.2	-0.1	0.5	0.4	0.0	-0.3	-0.3	0.2	0.2	0.5	-0.3	0.3	0.0	-0.4	-0.4	-0.3	-0.2	-0.3	-0.4	0.3	-0.2	0.3	0.3	0.0	0.0	-0.4	0.3	0.5	0.2	0.0	-0.3	0.4	0.3	-0.4	0.4	-0.4	0.3	-0.2	0.1	-0.1	-0.3	0.4	0.3	-0.4	0.5	-0.4	0.3	1.0	·
TKO.LPT.S_map.parmMap	0.4	0.3	-0.1	-0.4	0.4	-0.2	-0.3	0.1	0.1	-0.5	-0.5	-0.3	-0.6	0.9	0.5	-0.6	-0.6	0.3	0.3	0.0	0.1	0.0	0.0	0.0	0.4	-0.2	0.6	-0.1	-0.4	0.1	0.0	0.0	0.0	-0.2	-0.2	0.3	-0.1	-0.2	-0.5	0.0	-0.5	-0.3	0.2	0.2	0.6	0.4	0.2	-0.3	-0.2	0.2	-0.5	-0.2	-0.4	0.9	0.5	0.4	0.2	-0.2	-0.2	0.1	-0.5	-0.2	0.3	1.0
TKO.Splitter.BPR	0.9	0.5	-0.8	-0.2	0.2	0.4	0.2	-0.4	0.4	-0.4	-0.4	-0.2	-0.5	0.7	1.0	-0.5	-0.4	0.3	0.3	0.0	0.8	0.8	-0.5	-0.5	0.2	0.3	0.5	-0.5	-0.3	0.7	0.8	-0.1	0.8	0.4	-0.8	0.8	-0.3	-0.8	-0.4	0.0	-0.4	-0.8	-0.4	-0.2	1.0	0.9	0.5	-0.7	0.4	0.1	-0.4	-0.8	0.0	0.7	1.0	0.9	0.4	-0.9	0.3	-0.1	-0.3	-0.8	0.0	0.5

APPENDIX G

INDEPENDENT PARAMETER SOLUTION SPACE RANGES FOR EXPERIMENTS 3 AND 5

The following tables list the maximum and minimum values of the independent parameter final converged values for experiment 3 and experiment5. There are two important items to note from these tables. First, the center of the design space, i.e. the cycle design defined at the mean values of the design variables is not the center of the solution space. The base solution in the tables is the value of the independent parameters for the center of the design space. This value is not centered in the solution space as demonstrated by the variation in % below and % above for each individual independent. Setting the initial iterate to the center of the solution space would improve the modified Newton-Raphson performance, but this is impractical as the solution space and design space centers do not coincide. Therefore, it is important that the initial iterate resided within the solution space, but it is not necessary that it be centered within the solution space or the design space.

The second and far more important item to note is that some of the independent parameters have exceeded physical limits that make the converged solution infeasible. The solver is only finding the solution to a set of equations and does not consider that the values of the independent parameters are not physically achievable. This most often occurs when the solver extrapolates off of the performance maps and either is operating in stall of choked flow regions. Therefore it is important that the designs in which the solver converges to an infeasible design be identified by the cycle analyst and removed from consideration.

Table 75. Range of Solutions for LPCDP Independents for Experiment 3

Independent	Min Value	Max Value	Base Solution	% Below	% Above
LPCDP.Ambient.W	682.7	4652	1241	-45.0%	274.7%
LPCDP.Burner.FAR	0.01406	0.03198	0.02283	-38.4%	40.1%
LPCDP.Fan.S_map.PRdes	1.200	2.000	1.600	-25.0%	25.0%
LPCDP.LPC.S_map.PRdes	1.100	3.138	1.269	-13.3%	147.3%
LPCDP.LPC.S_map.effDes	0.8877	0.9040	0.9021	-1.6%	0.2%
LPCDP.HPC.S_map.NcDes	0.9415	1.132	1.058	-11.0%	7.0%
LPCDP.HPC.S_map.PRdes	10.83	19.793	17.56	-38.3%	12.7%
LPCDP.HPC.S_map.effDes	0.7529	0.8814	0.8009	-6.0%	10.0%
LPCDP.HPC.S_map.RlineMap	1.892	2.845	2.417	-21.7%	17.7%
LPCDP.HPT.S_map.parmMap	3.354	11.098	5.659	-40.7%	96.1%
LPCDP.HPT.S_map.parmMapDes	3.354	11.098	5.659	-40.7%	96.1%
LPCDP.HPT.S_map.effDes	0.8808	0.9192	0.9120	-3.4%	0.8%
LPCDP.HPT.S_map.parmNcDes	76.99	110.0	93.84	-18.0%	17.2%
LPCDP.LPT.S_map.parmMap	2.928	8.385	4.430	-33.9%	89.3%
LPCDP.Splitter.BPR	1.363	31.432	6.255	-78.2%	402.5%
LPCDP.Bld3.Cool1.fracW	0.06794	0.1970	0.1111	-38.9%	77.3%
LPCDP.Bld3.Cool2.fracW	0.04971	0.1484	0.08308	-40.2%	78.7%
LPCDP.HPC.Cool1.fracBldW	0.00726	0.04880	0.03116	-76.7%	56.6%
LPCDP.HPC.Cool2.fracBldW	4.9E-55	0.05100	0.01669	-100.0%	205.5%
LPCDP.Inlet.Fl_O.MN	0.7035	0.7035	0.7035	0.0%	0.0%
LPCDP.Fan.Fl_O.MN	0.3525	0.6330	0.4428	-20.4%	43.0%
LPCDP.Splitter.Fl_02.MN	0.3588	0.7511	0.4702	-23.7%	59.7%
LPCDP.Splitter.Fl_01.MN	0.3263	0.3278	0.3278	-0.5%	0.0%
LPCDP.Duct4.Fl_O.MN	0.3300	0.3316	0.3316	-0.5%	0.0%
LPCDP.LPC.Fl_O.MN	0.1607	0.4255	0.3668	-56.2%	16.0%
LPCDP.Duct6.Fl_O.MN	0.3733	0.6186	0.5856	-36.3%	5.6%
LPCDP.Burner.Fl_O.MN	0.08064	0.1238	0.1053	-23.4%	17.6%
LPCDP.HPC.Fl_O.MN	0.3713	0.4151	0.3904	-4.9%	6.3%
LPCDP.HPT.Fl_O.MN	0.2530	1.035	0.3831	-34.0%	170.2%
LPCDP.Duct11.Fl_O.MN	0.1341	0.7068	0.2110	-36.5%	235.0%
LPCDP.BypBld.Fl_O.MN	0.3567	0.7415	0.4671	-23.6%	58.7%
LPCDP.Duct13.Fl_O.MN	0.3753	0.3754	0.3753	0.0%	0.0%
LPCDP.Duct15.Fl_O.MN	0.3644	0.7793	0.4785	-23.8%	62.9%
LPCDP.HP_Shaft.Nmech	7537	9064	8469	-11.0%	7.0%
LPCDP.LP_Shaft.Nmech	614.5	4369	2371	-74.1%	84.3%

Table 76. Range of Solutions for HPCDP Independents for Experiment 3

Independent	Min Value	Max Value	Base Solution	% Below	% Above
HPCDP.Ambient.W	942.4	5655	2030	-53.6%	178.6%
HPCDP.Burner.FAR	0.01105	0.03979	0.01621	-31.9%	145.5%
HPCDP.Fan.S_map.RlineMap	0.6241	5.400	1.522	-59.0%	254.7%
HPCDP.HP_Shaft.Nmech	7261	9299	8097	-10.3%	14.8%
HPCDP.HPC.S_map.RlineMap	2.000	2.000	2.000	0.0%	0.0%
HPCDP.HPT.S_map.parmMap	3.461	11.730	6.117	-43.4%	91.7%
HPCDP.LP_Shaft.Nmech	512.5	5467	1850	-72.3%	195.6%
HPCDP.LPC.S_map.RlineMap	1.483	3.307	2.182	-32.0%	51.5%
HPCDP.LPT.S_map.parmMap	1.643	6.258	2.708	-39.3%	131.1%
HPCDP.Splitter.BPR	1.230	16.890	6.108	-79.9%	176.5%

Table 77. Range of Solutions for TOC1 Independents for Experiment 3

Independent	Min Value	Max Value	Base Solution	% Below	% Above
TOC1.Ambient.W	602.6	4077	1096	-45.0%	271.9%
TOC1.Burner.FAR	0.01596	0.03458	0.02499	-36.1%	38.4%
TOC1.Fan.S_map.RlineMap	2.028	2.342	2.114	-4.1%	10.8%
TOC1.HP_Shaft.Nmech	7670	9593	8756	-12.4%	9.6%
TOC1.HPC.S_map.RlineMap	0.9508	3.084	2.437	-61.0%	26.5%
TOC1.HPT.S_map.parmMap	3.317	11.045	5.648	-41.3%	95.6%
TOC1.LP_Shaft.Nmech	634.4	4652	2495	-74.6%	86.4%
TOC1.LPC.S_map.RlineMap	1.882	2.306	2.016	-6.6%	14.4%
TOC1.LPT.S_map.parmMap	2.940	8.915	4.473	-34.3%	99.3%
TOC1.Splitter.BPR	1.321	31.075	6.062	-78.2%	412.6%

Table 78. Range of Solutions for TKO Independents for Experiment 3

Independent	Min Value	Max Value	Base Solution	% Below	% Above
TKO.Ambient.W	1820	8566	3128	-41.8%	173.8%
TKO.Burner.FAR	0.01928	0.03679	0.02922	-34.0%	25.9%
TKO.Fan.S_map.RlineMap	1.010	2.285	1.795	-43.7%	27.3%
TKO.HP_Shaft.Nmech	8338	10179	9555	-12.7%	6.5%
TKO.HPC.S_map.RlineMap	0.01340	2.649	2.114	-99.4%	25.3%
TKO.HPT.S_map.parmMap	3.429	11.618	5.838	-41.3%	99.0%
TKO.LP_Shaft.Nmech	621.6	5178	2594	-76.0%	99.6%
TKO.LPC.S_map.RlineMap	1.872	2.507	2.074	-9.7%	20.9%
TKO.LPT.S_map.parmMap	2.884	5.611	4.032	-28.5%	39.1%
TKO.Splitter.BPR	1.302	24.705	6.095	-78.6%	305.4%

Table 79. Range of Solutions for LPCDP Independents for Experiment 5

Independent	Max Value	Min Value	Base Solution	% Above	% Below
LPCDP.Ambient.W	1360	1101	1204	12.9%	-8.6%
LPCDP.Burner.FAR	0.02752	0.01970	0.02249	22.3%	-12.4%
LPCDP.Fan.S_map.PRdes	0.1233	0.09571	0.1052	17.2%	-9.0%
LPCDP.LPC.S_map.PRdes	0.4756	0.3933	0.4325	10.0%	-9.1%
LPCDP.LPC.S_map.effDes	0.2520	0.1679	0.2089	20.6%	-19.6%
LPCDP.HPC.S_map.NcDes	0.3753	0.3753	0.3753	0.0%	0.0%
LPCDP.HPC.S_map.PRdes	0.4873	0.4022	0.4425	10.1%	-9.1%
LPCDP.HPC.S_map.effDes	0.3316	0.3307	0.3316	0.0%	-0.3%
LPCDP.HPC.S_map.RlineMap	0.6195	0.3818	0.5970	3.8%	-36.0%
LPCDP.HPT.S_map.parmMap	0.4495	0.3845	0.4153	8.2%	-7.4%
LPCDP.HPT.S_map.parmMapDes	1.750	1.501	1.625	7.7%	-7.6%
LPCDP.HPT.S_map.effDes	9073	7567	8684	4.5%	-12.9%
LPCDP.HPT.S_map.parmNcDes	0.1349	0.07037	0.1045	29.1%	-32.7%
LPCDP.LPT.S_map.parmMap	0.1040	0.05065	0.07471	39.2%	-32.2%
LPCDP.Splitter.BPR	0.04218	0.01848	0.02773	52.1%	-33.4%
LPCDP.Bld3.Cool1.fracW	0.03710	0.00350	0.01247	197.4%	-72.0%
LPCDP.Bld3.Cool2.fracW	0.4044	0.3659	0.3860	4.8%	-5.2%
LPCDP.HPC.Cool1.fracBldW	0.8793	0.7538	0.7860	11.9%	-4.1%
LPCDP.HPC.Cool2.fracBldW	1.133	0.9451	1.085	4.5%	-12.9%
LPCDP.Inlet.Fl_O.MN	19.85	11.04	18.30	8.5%	-39.7%
LPCDP.Fan.Fl_O.MN	2.275	1.470	1.777	28.0%	-17.3%
LPCDP.Splitter.Fl_02.MN	0.4614	0.2638	0.3912	17.9%	-32.6%
LPCDP.Splitter.Fl_01.MN	0.9166	0.8839	0.9119	0.5%	-3.1%
LPCDP.Duct4.Fl_O.MN	6.993	3.487	5.867	19.2%	-40.6%
LPCDP.LPC.Fl_O.MN	6.993	3.487	5.867	19.2%	-40.6%
LPCDP.Duct6.Fl_O.MN	100.7	77.59	93.89	7.3%	-17.4%
LPCDP.Burner.Fl_O.MN	0.6499	0.6499	0.6499	0.0%	0.0%
LPCDP.HPC.Fl_O.MN	2843	1973	2425	17.3%	-18.6%
LPCDP.HPT.Fl_O.MN	0.4270	0.2961	0.3776	13.1%	-21.6%
LPCDP.Duct11.Fl_O.MN	0.9136	0.8914	0.9025	1.2%	-1.2%
LPCDP.BypBld.Fl_O.MN	1.581	1.096	1.233	28.3%	-11.1%
LPCDP.Duct13.Fl_O.MN	5.613	3.627	4.328	29.7%	-16.2%
LPCDP.Duct15.Fl_O.MN	8.336	4.942	6.035	38.1%	-18.1%
LPCDP.HP_Shaft.Nmech	0.3278	0.3270	0.3278	0.0%	-0.3%
LPCDP.LP_Shaft.Nmech	0.4788	0.3958	0.4352	10.0%	-9.1%

Table 80. Range of Solutions for HPCDP Independents for Experiment 5

Independent	Max Value	Min Value	Base Solution	% Above	% Below
HPCDP.Ambient.W	2507	1172	1923	30.4%	-39.1%
HPCDP.Burner.FAR	0.02372	0.01046	0.01516	56.5%	-31.0%
HPCDP.Fan.S_map.RlineMap	1.658	1.383	1.530	8.3%	-9.6%
HPCDP.HP_Shaft.Nmech	8497	7430	8078	5.2%	-8.0%
HPCDP.HPC.S_map.RlineMap	2.000	2.000	2.000	0.0%	0.0%
HPCDP.HPT.S_map.parmMap	7.328	3.615	6.246	17.3%	-42.1%
HPCDP.LP_Shaft.Nmech	2613	1165	1847	41.5%	-37.0%
HPCDP.LPC.S_map.RlineMap	2.784	1.813	2.257	23.4%	-19.7%
HPCDP.LPT.S_map.parmMap	3.991	1.707	2.597	53.7%	-34.3%
HPCDP.Splitter.BPR	8.079	4.979	5.891	37.2%	-15.5%

Table 81. Range of Solutions for TOC1 Independents for Experiment 5

Independent	Max Value	Min Value	Base Solution	% Above	% Below
TOC1.Ambient.W	1213	983.1	1065	13.9%	-7.7%
TOC1.Burner.FAR	0.02961	0.02158	0.02483	19.3%	-13.1%
TOC1.Fan.S_map.RlineMap	2.227	2.059	2.131	4.5%	-3.4%
TOC1.HP_Shaft.Nmech	9461	7663	9005	5.1%	-14.9%
TOC1.HPC.S_map.RlineMap	2.241	1.100	1.686	32.9%	-34.7%
TOC1.HPT.S_map.parmMap	7.001	3.458	5.853	19.6%	-40.9%
TOC1.LP_Shaft.Nmech	3070	2041	2564	19.7%	-20.4%
TOC1.LPC.S_map.RlineMap	2.159	1.871	1.999	8.0%	-6.4%
TOC1.LPT.S_map.parmMap	5.657	3.648	4.369	29.5%	-16.5%
TOC1.Splitter.BPR	8.143	4.733	5.834	39.6%	-18.9%

Table 82. Range of Solutions for TOC2 Independents for Experiment 5

Independent	Max Value	Min Value	Base Solution	% Above	% Below
TOC2.Ambient.W	1469	1190	1290	13.9%	-7.7%
TOC2.Burner.FAR	0.02958	0.02160	0.02484	19.1%	-13.1%
TOC2.Fan.S_map.RlineMap	2.227	2.059	2.131	4.5%	-3.4%
TOC2.HP_Shaft.Nmech	9480	7697	9022	5.1%	-14.7%
TOC2.HPC.S_map.RlineMap	2.212	1.064	1.669	32.5%	-36.3%
TOC2.HPT.S_map.parmMap	7.006	3.465	5.856	19.6%	-40.8%
TOC2.LP_Shaft.Nmech	3086	2051	2576	19.8%	-20.4%
TOC2.LPC.S_map.RlineMap	2.136	1.857	1.988	7.4%	-6.6%
TOC2.LPT.S_map.parmMap	5.656	3.647	4.368	29.5%	-16.5%
TOC2.Splitter.BPR	8.156	4.734	5.838	39.7%	-18.9%

Table 83. Range of Solutions for TKO Independents for Experiment 5

Independent	Max Value	Min Value	Base Solution	% Above	% Below
TKO.Ambient.W	3398	2851	3047	11.5%	-6.4%
TKO.Burner.FAR	0.03363	0.02589	0.02880	16.8%	-10.1%
TKO.Fan.S_map.RlineMap	1.897	1.697	1.815	4.5%	-6.5%
TKO.HP_Shaft.Nmech	10216	8371	9755	4.7%	-14.2%
TKO.HPC.S_map.RlineMap	2.181	0.8001	1.578	38.2%	-49.3%
TKO.HPT.S_map.parmMap	7.307	3.530	6.015	21.5%	-41.3%
TKO.LP_Shaft.Nmech	3218	2085	2660	21.0%	-21.6%
TKO.LPC.S_map.RlineMap	2.398	1.997	2.068	16.0%	-3.4%
TKO.LPT.S_map.parmMap	5.079	3.480	3.965	28.1%	-12.2%
TKO.Splitter.BPR	8.060	4.749	5.893	36.8%	-19.4%

Table 84. Range of Solutions for HHDTKO Independents for Experiment 5

Independent	Max Value	Min Value	Base Solution	% Above	% Below
HHDTKO.Ambient.W	2955	2469	2647	11.6%	-6.8%
HHDTKO.Burner.FAR	0.03422	0.02586	0.02915	17.4%	-11.3%
HHDTKO.Fan.S_map.RlineMap	2.009	1.735	1.882	6.7%	-7.8%
HHDTKO.HP_Shaft.Nmech	10210	8313	9766	4.5%	-14.9%
HHDTKO.HPC.S_map.RlineMap	2.137	0.6686	1.540	38.8%	-56.6%
HHDTKO.HPT.S_map.parmMap	7.265	3.508	5.986	21.4%	-41.4%
HHDTKO.LP_Shaft.Nmech	3299	2118	2718	21.4%	-22.1%
HHDTKO.LPC.S_map.RlineMap	2.225	1.956	2.031	9.6%	-3.7%
HHDTKO.LPT.S_map.parmMap	5.219	3.535	4.068	28.3%	-13.1%
HHDTKO.Splitter.BPR	7.960	4.646	5.777	37.8%	-19.6%

Table 85. Range of Solutions for Cutback Independents

Independent	Max Value	Min Value	Base Solution	% Above	% Below
CUTBACK.Ambient.W	3022	2548	2711	11.5%	-6.0%
CUTBACK.Burner.FAR	0.02967	0.02307	0.02558	16.0%	-9.8%
CUTBACK.Fan.S_map.RlineMap	1.806	1.664	1.755	2.9%	-5.2%
CUTBACK.HP_Shaft.Nmech	9731	8103	9308	4.5%	-12.9%
CUTBACK.HPC.S_map.RlineMap	2.207	1.295	1.695	30.2%	-23.6%
CUTBACK.HPT.S_map.parmMap	7.207	3.554	5.999	20.1%	-40.8%
CUTBACK.LP_Shaft.Nmech	2951	1959	2469	19.5%	-20.7%
CUTBACK.LPC.S_map.RlineMap	2.518	1.954	2.138	17.8%	-8.6%
CUTBACK.LPT.S_map.parmMap	4.822	3.359	3.777	27.7%	-11.1%
CUTBACK.Splitter.BPR	8.179	4.957	6.047	35.2%	-18.0%

Table 86. Range of Solutions for SLSI Independents for Experiment 5

Independent	Max Value	Min Value	Base Solution	% Above	% Below
SLSI.Ambient.W	3243	2706	2934	10.5%	-7.8%
SLSI.Burner.FAR	0.03401	0.02565	0.02901	17.2%	-11.6%
SLSI.Fan.S_map.RlineMap	1.876	1.617	1.772	5.9%	-8.8%
SLSI.HP_Shaft.Nmech	10221	8352	9776	4.6%	-14.6%
SLSI.HPC.S_map.RlineMap	2.168	0.6894	1.546	40.2%	-55.4%
SLSI.HPT.S_map.parmMap	7.330	3.523	6.021	21.7%	-41.5%
SLSI.LP_Shaft.Nmech	3179	2071	2663	19.4%	-22.2%
SLSI.LPC.S_map.RlineMap	2.481	2.012	2.074	19.6%	-3.0%
SLSI.LPT.S_map.parmMap	5.042	3.460	3.931	28.3%	-12.0%
SLSI.Splitter.BPR	7.901	4.725	5.803	36.1%	-18.6%

Table 87. Range of Solutions for SLSU Independents for Experiment 5

Independent	Max Value	Min Value	Base Solution	% Above	% Below
SLSU.Ambient.W	3453	2784	3059	12.9%	-9.0%
SLSU.Burner.FAR	0.03247	0.02338	0.02700	20.3%	-13.4%
SLSU.Fan.S_map.RlineMap	1.878	1.641	1.796	4.6%	-8.6%
SLSU.HP_Shaft.Nmech	9927	8193	9475	4.8%	-13.5%
SLSU.HPC.S_map.RlineMap	2.133	0.7766	1.548	37.8%	-49.8%
SLSU.HPT.S_map.parmMap	7.227	3.511	5.976	20.9%	-41.2%
SLSU.LP_Shaft.Nmech	3072	2131	2620	17.3%	-18.6%
SLSU.LPC.S_map.RlineMap	2.135	1.957	2.030	5.2%	-3.6%
SLSU.LPT.S_map.parmMap	5.190	3.467	3.980	30.4%	-12.9%
SLSU.Splitter.BPR	7.864	4.761	5.799	35.6%	-17.9%

APPENDIX H

CYCLE RESULTS FOR INITIAL ITERATE DEVELOPMENT

SUMMARY OUTPUT DATA

MN	alt	dTamb	W	Fn	TSFC	Wfuel	BPR	VTAS	OPR	T4	T41	T49
0.8	35000	0	1373.11	16324.5	0.5588	9122.08	8.7633	778.35	31.848	2729.8	2568.3	1834.9
Thermal Eff:	0.533		Propulsive Eff:	0.6571		Overall Eff:	0.3502					

FLOW STATION DATA

		W	Pt	Tt	ht	FAR	Wc	Ps	Ts	Aphy	MN	gamt
FS_1	Ambient.Fl_O	1373.1	5.270	444.23	106.324	0.00000	3543.91	3.458	393.85	10719.7	0.8000	1.3997
FS_2	Inlet.Fl_O	1373.1	5.254	444.23	106.324	0.00000	3554.57	3.953	409.59	11759.7	0.6508	1.3997
FS_21	Fan.Fl_O	1373.1	8.130	507.00	121.378	0.00000	2454.02	7.096	487.66	10449.9	0.4449	1.4001
FS_13	Splitter.Fl_02	1232.5	8.130	507.00	121.378	0.00000	2202.67	7.129	488.31	9508.5	0.4372	1.4001
FS_22	Splitter.Fl_01	140.64	8.130	507.00	121.378	0.00000	251.35	7.627	497.83	1475.8	0.3033	1.4001
FS_23	Duct4.Fl_O	140.64	8.049	507.00	121.378	0.00000	253.89	7.532	497.47	1464.8	0.3093	1.4001
FS_24	LPC.Fl_O	140.64	11.623	568.52	136.151	0.00000	186.18	10.483	552.01	886.78	0.3869	1.3997
FS_25	Duct6.Fl_O	140.64	11.530	568.52	136.151	0.00000	187.68	10.367	551.53	883.01	0.3927	1.3997
FS_3	HPC.Fl_O	128.91	167.322	1275.15	310.839	0.00000	17.75	157.42	1254.54	106.44	0.3003	1.3638
FS_36	Bld3.Fl_O	104.41	167.322	1275.15	310.839	0.00000	14.38	157.42	1254.54	86.22	0.3003	1.3638
FS_4	Burner.Fl_O	106.95	159.793	2729.82	737.25	0.02430	22.56	158.96	2726.65	442.92	0.0900	1.2876
FS_45	HPT.Fl_O	131.44	38.165	1869.64	479.67	0.01970	96.09	35.851	1841.52	570.77	0.3090	1.3193
FS_46	ATD_Bleed.Fl_O	131.44	38.165	1869.64	479.67	0.01970	96.09	35.851	1841.52	570.77	0.3090	1.3193
FS_48	Duct11.Fl_O	131.44	37.783	1869.64	479.67	0.01970	97.06	36.835	1858.18	877.58	0.1965	1.3193
FS_5	LPT.Fl_O	139.17	6.238	1215.76	300.517	0.01850	502.01	5.886	1197.41	3081.2	0.2937	1.3548
FS_7	Duct13.Fl_O	139.17	6.188	1215.76	300.517	0.01850	506.06	5.792	1194.85	2927.8	0.3138	1.3548
FS_9	Core_Nozz.Fl_O	139.17	6.188	1215.76	300.517	0.01850	506.06	3.458	1042.09	1491.4	0.9601	1.3548
FS_15	BypBld.Fl_O	1226.3	8.130	507.00	121.378	0.00000	2191.66	7.141	488.54	9508.9	0.4343	1.4001
FS_17	Duct15.Fl_O	1226.3	7.983	507.00	121.378	0.00000	2231.83	6.970	487.69	9508.6	0.4446	1.4001
FS_19	Byp_Nozz.Fl_O	1226.3	7.983	507.00	121.378	0.00000	2231.83	4.233	422.52	6482.8	1.0000	1.4001

TURBOMACHINERY PERFORMANCE DATA

	Wc	PR	eff	Nc	TR	efPoly	pwr	SMN	SMW	s_Re
Fan	3554.57	1.547	0.9544	2217.59	1.141	0.9561	-29247.3	26	18.49	1
LPC	253.89	1.444	0.9047	2075.77	1.121	0.9105	-2939.4	36.26	22.71	1
HPC	187.68	14.512	0.8926	8014.11	2.243	0.9236	-32875.1	19.23	19.19	1
HPT	22.56	4.187	0.9001	160.59	1.338	0.8838	33125.1			
LPT	97.06	6.057	0.9145	47.46	1.498	0.8941	32186.7			

TURBOMACHINERY MAP DATA

	WcMap	PRmap	effMap	NcMap	R/Parm	s_WcDes	s_PRdes	s_effDes	s_NcDes
Fan	3237.17	1.593	0.9118	1.000	1.914	1.0980	0.9231	1.0468	0.0005
LPC	226.53	1.223	0.9056	0.943	1.954	1.1208	1.9920	0.9991	0.0005
HPC	223.58	15.144	0.8718	0.993	1.985	0.8394	0.9554	1.0238	0.0001
HPT	22.99	4.888	0.9197	100.56	4.888	0.9815	1.2199	0.9787	0.0275
LPT	150.58	5.415	0.9264	96.47	5.415	0.6441	0.8731	0.9864	0.0892

INLETS	eRam	Afs	Fram
Inlet	0.997	10719.74	33217.9

DUCTS	dPnorm	MN	Aphy		BLEEDS - interstg	Wb/Win	BldWk	BldP	W	Tt	ht	Pt
Duct4	0.010	0.3033	1475.83		ATDbleed HPC.A>	0.000	1.000	1.000	0.000	1275.15	310.84	167.322
Duct6	0.008	0.3869	886.78		CstmrBld HPC.C>	0.0284	0.35	0.1465	4.000	821.43	197.29	34.353
Duct11	0.010	0.309	570.77		C_LPTexit HPC.C>	0.0200	0.35	0.1465	2.813	821.43	197.29	6.238
Duct13	0.008	0.2937	3081.15		C_LPTinlt HPC.C>	0.0350	0.35	0.1465	4.922	821.43	197.29	37.783
Duct15	0.018	0.4343	9508.89									

SPLITTERS	BPR	dP/P1	dP/P2		BLEEDS - output	Wb/Win	hscale	Pscale	W	Tt	ht	Pt
Splitter	8.7633	0.000	0.000		C_HPTexit Bld3.>	0.0750	1.000	1.000	9.6678	1275.15	310.84	38.165
					C_HPTinlt Bld3.>	0.1150	1.000	1.000	14.824	1275.15	310.84	159.793
SHAFTS	Nmech	trq in	pwr in		Duct15Lk BypBl>	0.0050	1.000	1.000	6.1624	507	121.38	8.13
HP_Shaft	8390.4	20735.1	33125.1									
LP_Shaft	2052.3	82370.9	32186.7									

BURNERS	TtOut	eff	dPnorm	Wfuel	FAR	EINOx
Burner	2729.82	0.995	0.045	2.53391	0.02427	13.936

NOZZLES	PR	Cfg	CdTh	Cv	Ath	MNth	Vact	Fg
Core_Nozz	1.789	0.9925	0.9521	0.9955	1566.4	0.96	1490	6445.1
Byp_Nozz	2.309	0.9934	0.9665	0.9946	6707.7	1	998.9	43097.3

Figure 98. LPCDP Off-design Cycle Data for Establishing Initial Iterate

```
                                                           SUMMARY OUTPUT DATA
      MN            alt         dTamb          W             Fn          TSFC        Wfuel         BPR        VTAS         OPR          T4          T41          T49
       0              0             0        2930.04       73975.8      0.2859     21150.48      8.5287          0        27.176       2893.8      2729.2       1944.2
 Thermal Eff:      0.4569                Propulsive Eff:       0                    Overall Eff:      0

                                                                               FLOW STATION DATA
                                 W            Pt            Tt           ht          FAR           Wc          Ps           Ts          Aphy          MN         gamt
FS_1   Ambient.Fl_O           2930.0       14.696       518.67      124.181      0.00000     2930.07      14.696      518.67       -----.--      0.0000      1.3998
FS_2   Inlet.Fl_O             2930.0       14.593       518.67      124.181      0.00000     2950.73      12.417      495.27       11759.7       0.4859      1.3998
FS_21  Fan.Fl_O               2930.0       20.946       580.39      138.999      0.00000     2174.61      18.933      563.88       10449.9       0.3828      1.3997
FS_13  Splitter.Fl_02         2622.5       20.946       580.39      138.999      0.00000     1946.39      19.006      564.50       9508.5        0.3753      1.3997
FS_22  Splitter.Fl_01         307.50       20.946       580.39      138.999      0.00000     228.22       19.892      571.89       1475.8        0.2726      1.3997
FS_23  Duct4.Fl_O             307.50       20.737       580.39      138.999      0.00000     230.52       19.653      571.56       1464.8        0.2779      1.3997
FS_24  LPC.Fl_O               307.50       27.542       634.54      152.021      0.00000     181.48       24.992      617.22       886.78        0.3754      1.3980
FS_25  Duct6.Fl_O             307.50       27.322       634.54      152.021      0.00000     182.94       24.722      616.72       883.01        0.3809      1.3980
FS_3   HPC.Fl_O               286.58       396.579      1409.27     345.619      0.00000     17.51        373.79      1387.52      106.44        0.2966      1.3557
FS_36  Bld3.Fl_O              232.13       396.579      1409.27     345.619      0.00000     14.18        373.79      1387.52      86.22         0.2966      1.3557
FS_4   Burner.Fl_O            238.01       378.733      2893.77     789.012      0.02530     21.81        376.89      2890.67      442.92        0.0872      1.2823
FS_45  HPT.Fl_O               292.46       86.252       1978.92     511.301      0.02050     97.33        80.875      1948.63      570.77        0.3141      1.3146
FS_46  ATD_Bleed.Fl_O         292.46       86.252       1978.92     511.301      0.02050     97.33        80.875      1948.63      570.77        0.3141      1.3146
FS_48  Duct11.Fl_O            292.46       85.389       1978.92     511.301      0.02050     98.32        83.190      1966.60      877.58        0.1996      1.3146
FS_5   LPT.Fl_O               309.37       18.498       1371.72     342.081      0.01940     399.7        17.853      1359.29      3081.2        0.2303      1.3446
FS_7   Duct13.Fl_O            309.37       18.350       1371.72     342.081      0.01940     402.92       17.627      1357.63      2927.8        0.2453      1.3446
FS_9   Core_Nozz.Fl_O         309.37       18.350       1371.72     342.081      0.01940     402.92       14.696      1295.55      1444.0        0.5819      1.3446
FS_15  BypBld.Fl_O            2609.4       20.946       580.39      138.999      0.00000     1936.66      19.028      564.69       9508.9        0.3730      1.3997
FS_17  Duct15.Fl_O            2609.4       20.569       580.39      138.999      0.00000     1972.16      18.607      564.01       9508.6        0.3812      1.3997
FS_19  Byp_Nozz.Fl_O          2609.4       20.569       580.39      138.999      0.00000     1972.16      14.696      527.27       6250.4        0.7100      1.3997
TURBOMACHINERY PERFORMANCE DATA
                  Wc           PR           eff           Nc           TR         efPoly        pwr          SMN          SMW          s_Re
     Fan       2950.73       1.435       0.9106       1979.40       1.119       0.9162     -61429.3       24.25        12.56          1
     LPC        230.52       1.328       0.9067       1871.20       1.093       0.9096      -5665.4       42.34        21.93          1
     HPC        182.94      14.515       0.8954       7967.16       2.221       0.9255     -80503.9       16.23        16.2           1
     HPT         21.81       4.391       0.9007        163.82       1.346       0.8841      80753.9
     LPT         98.32       4.616       0.9206         44.50       1.407       0.9052      67094.7

TURBOMACHINERY MAP DATA
                 WcMap        PRmap        effMap       NcMap        R/Parm      s_WcDes      s_PRdes     s_effDes     s_NcDes
     Fan       2687.25       1.472       0.8699        0.893        1.522       1.0980       0.9231      1.0468       0.0005
     LPC        205.68       1.165       0.9076        0.850        1.960       1.1208       1.9920      0.9991       0.0005
     HPC        217.94      15.147       0.8746        0.987        1.813       0.8394       0.9554      1.0238       0.0001
     HPT         22.23       5.137       0.9203       102.58        5.137       0.9815       1.2199      0.9787       0.0275
     LPT        151.67       4.157       0.9274        90.44        4.157       0.6441       0.8731      0.9864       0.0892

   INLETS        eRam          Afs          Fram
    Inlet        0.993       -----.--        0

   DUCTS        dPnorm         MN           Aphy                 BLEEDS - interstg     Wb/Win       BldWk        BldP          W            Tt           ht           Pt
    Duct4        0.010       0.2726       1475.83               ATDbleed HPC.A>       0.000        1.000       1.000       0.000       1409.27       345.62      396.579
    Duct6        0.008       0.3754       886.78                CstmrBld HPC.C>       0.0130       0.35        0.1465      4.000        913.17       219.78       81.418
    Duct11       0.010       0.3141       570.77               C_LPTexit HPC.C>       0.0200       0.35        0.1465      6.150        913.17       219.78       18.498
    Duct13       0.008       0.2303       3081.15              C_LPTinlt HPC.C>       0.0350       0.35        0.1465      10.762       913.17       219.78       85.389
    Duct15       0.018       0.373        9508.89

  SPLITTERS      BPR          dP/P1         dP/P2                BLEEDS - output       Wb/Win       hscale       Pscale        W            Tt           ht           Pt
   Splitter     8.5287        0.000         0.000               C_HPTexit Bld3.>      0.0750       1.000       1.000       21.4938      1409.27       345.62       86.252
                                                                C_HPTinlt Bld3.>      0.1150       1.000       1.000       32.9572      1409.27       345.62      378.733
                                                                Duct15Lk BypBl>       0.0050       1.000       1.000       13.1127       580.39       139          20.946
   SHAFTS       Nmech         trq in        pwr in
  HP_Shaft      8812.3       48129.4       80753.9
  LP_Shaft      1979.4      178028.7       67094.7

  BURNERS       TtOut          eff          dPnorm        Wfuel        FAR          EINOx
   Burner      2893.77        0.995         0.045        5.87513      0.02531      25.651

  NOZZLES         PR           Cfg          CdTh          Cv           Ath          MNth         Vact          Fg
  Core_Nozz      1.249        0.993        0.9219        0.996        1566.4        0.582       1001.2       9627.2
  Byp_Nozz       1.400        0.9928       0.9318        0.9958       6707.7        0.71        793.4        64348.6
```

Figure 99. HPCDP Off-design Cycle Data for Establishing Initial Iterate

SUMMARY OUTPUT DATA

MN	alt	dTamb	W	Fn	TSFC	Wfuel	BPR	VTAS	OPR	T4	T41	T49
0.85	39000	0	1225.37	15800	0.5953	9405.82	8.2693	822.84	35.355	2900.0	2727.3	1961.8
Thermal Eff:	0.5486		Propulsive Eff:	0.6335		Overall Eff:	0.3475					

FLOW STATION DATA

		W	Pt	Tt	ht	FAR	Wc	Ps	Ts	Aphy	MN	gamt
FS_1	Ambient.Fl_O	1225.4	4.559	446.27	106.815	0.00000	3663.87	2.854	389.97	10856.4	0.8500	1.3997
FS_2	Inlet.Fl_O	1225.4	4.545	446.27	106.815	0.00000	3674.89	3.292	407.00	11759.7	0.6951	1.3997
FS_21	Fan.Fl_O	1225.4	7.273	517.66	123.937	0.00000	2473.69	6.331	497.54	10449.9	0.4495	1.3998
FS_13	Splitter.Fl_02	1093.2	7.273	517.66	123.937	0.00000	2206.82	6.374	498.51	9508.5	0.4381	1.3998
FS_22	Splitter.Fl_01	132.20	7.273	517.66	123.937	0.00000	266.87	6.761	506.97	1475.8	0.3246	1.3998
FS_23	Duct4.Fl_O	132.20	7.200	517.66	123.937	0.00000	269.56	6.673	506.54	1464.8	0.3312	1.3998
FS_24	LPC.Fl_O	132.20	10.800	588.51	140.949	0.00000	191.61	9.669	570.21	886.78	0.4007	1.3996
FS_25	Duct6.Fl_O	132.20	10.714	588.51	140.949	0.00000	193.16	9.560	569.67	883.01	0.4068	1.3996
FS_3	HPC.Fl_O	120.93	160.703	1330.61	325.151	0.00000	17.71	151.23	1309.35	106.44	0.3000	1.3604
FS_36	Bld3.Fl_O	97.95	160.703	1330.61	325.151	0.00000	14.35	151.23	1309.35	86.22	0.3000	1.3604
FS_4	Burner.Fl_O	100.56	153.472	2900.00	792.367	0.02670	22.77	152.66	2896.62	442.92	0.0911	1.2812
FS_45	HPT.Fl_O	123.54	37.295	1999.49	517.855	0.02160	95.58	35.060	1970.20	570.77	0.3079	1.3131
FS_46	ATD_Bleed.Fl_O	123.54	37.295	1999.49	517.855	0.02160	95.58	35.060	1970.20	570.77	0.3079	1.3131
FS_48	Duct11.Fl_O	123.54	36.922	1999.49	517.855	0.02160	96.54	36.006	1987.55	877.58	0.1959	1.3131
FS_5	LPT.Fl_O	130.81	5.999	1298.65	322.899	0.02040	507.02	5.654	1278.87	3081.2	0.2977	1.3484
FS_7	Duct13.Fl_O	130.81	5.951	1298.65	322.899	0.02040	511.1	5.562	1276.11	2927.8	0.3182	1.3484
FS_9	Core_Nozz.Fl_O	130.81	5.951	1298.65	322.899	0.02040	511.1	3.182	1102.43	1507.0	1.0000	1.3484
FS_15	BypBld.Fl_O	1087.7	7.273	517.66	123.937	0.00000	2195.79	6.385	498.75	9508.9	0.4353	1.3998
FS_17	Duct15.Fl_O	1087.7	7.142	517.66	123.937	0.00000	2236.04	6.232	497.87	9508.6	0.4456	1.3998
FS_19	Byp_Nozz.Fl_O	1087.7	7.142	517.66	123.937	0.00000	2236.04	3.787	431.39	6493.1	1.0000	1.3998

TURBOMACHINERY PERFORMANCE DATA

	Wc	PR	eff	Nc	TR	efPoly	pwr	SMN	SMW	s_Re
Fan	3674.89	1.600	0.9100	2371.75	1.160	0.9140	-29685.7	22.49	17.18	1
LPC	269.56	1.500	0.8950	2202.16	1.137	0.9025	-3181.8	34.58	24.5	1
HPC	193.16	15.000	0.8900	8073.59	2.261	0.9221	-32543.6	18.65	18.59	1
HPT	22.77	4.115	0.9000	159.70	1.327	0.8841	32793.6			
LPT	96.54	6.154	0.9200	49.20	1.499	0.9010	32867.5			

TURBOMACHINERY MAP DATA

	WcMap	PRmap	effMap	NcMap	R/Parm	s_WcDes	s_PRdes	s_effDes	s_NcDes
Fan	3346.75	1.650	0.8694	1.070	2.050	1.0980	0.9231	1.0468	0.0005
LPC	240.51	1.251	0.8959	1.000	2.050	1.1208	1.9920	0.9991	0.0005
HPC	230.11	15.654	0.8693	1.000	2.050	0.8394	0.9554	1.0238	0.0001
HPT	23.20	4.800	0.9196	100.00	4.800	0.9815	1.2199	0.9787	0.0275
LPT	149.88	5.500	0.9327	100.00	5.500	0.6441	0.8731	0.9864	0.0892

INLETS	eRam	Afs	Fram
Inlet	0.997	10856.39	31338.3

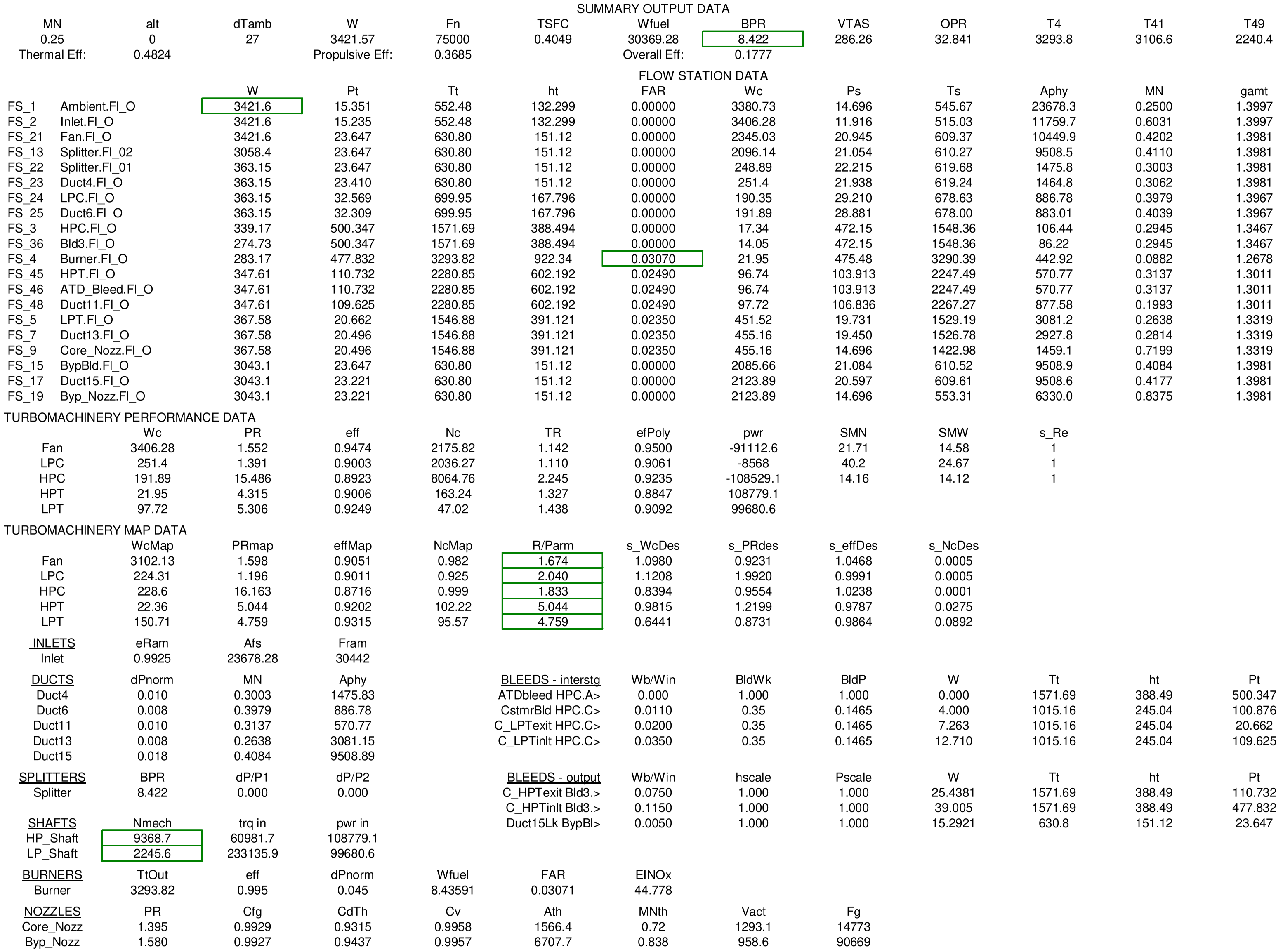
DUCTS	dPnorm	MN	Aphy		BLEEDS - interstg	Wb/Win	BldWk	BldP	W	Tt	ht	Pt
Duct4	0.010	0.3246	1475.83		ATDbleed HPC.A>	0.000	1.000	1.000	0.000	1330.61	325.15	160.703
Duct6	0.008	0.4007	886.78		CstmrBld HPC.C>	0.0303	0.35	0.1465	4.000	854.68	205.42	32.687
Duct11	0.010	0.3079	570.77		C_LPTexit HPC.C>	0.0200	0.35	0.1465	2.644	854.68	205.42	5.999
Duct13	0.008	0.2977	3081.15		C_LPTinlt HPC.C>	0.0350	0.35	0.1465	4.627	854.68	205.42	36.922
Duct15	0.018	0.4353	9508.89									

SPLITTERS	BPR	dP/P1	dP/P2		BLEEDS - output	Wb/Win	hscale	Pscale	W	Tt	ht	Pt
Splitter	8.2693	0.000	0.000		C_HPTexit Bld3.>	0.0750	1.000	1.000	9.0694	1330.61	325.15	37.295
					C_HPTinlt Bld3.>	0.1150	1.000	1.000	13.9064	1330.61	325.15	153.472
SHAFTS	Nmech	trq in	pwr in		Duct15Lk BypBl>	0.0050	1.000	1.000	5.4659	517.66	123.94	7.273
HP_Shaft	8600	20027.4	32793.6									
LP_Shaft	2200	78465.5	32867.5									

BURNERS	TtOut	eff	dPnorm	Wfuel	FAR	EINOx
Burner	2900	0.995	0.045	2.61273	0.02667	16.069

NOZZLES	PR	Cfg	CdTh	Cv	Ath	MNth	Vact	Fg
Core_Nozz	2.085	0.9924	0.9620	0.9952	1566.4	1	1592.9	6971
Byp_Nozz	2.502	0.9920	0.9680	0.9940	6707.7	1	1008.8	40167.3

Figure 100. TOC 1 Off-design Cycle Data for Establishing Initial Iterate



```
                                                              SUMMARY OUTPUT DATA
      MN           alt        dTamb          W            Fn         TSFC        Wfuel        BPR        VTAS        OPR          T4          T41         T49
     0.25           0           27        3421.57       75000       0.4049     30369.28      8.422      286.26      32.841      3293.8      3106.6      2240.4
 Thermal Eff:     0.4824                Propulsive Eff:  0.3685                 Overall Eff:  0.1777

                                                                         FLOW STATION DATA
                                W           Pt           Tt          ht         FAR          Wc          Ps          Ts         Aphy          MN         gamt
FS_1   Ambient.Fl_O          3421.6       15.351       552.48     132.299     0.00000     3380.73      14.696      545.67     23678.3      0.2500      1.3997
FS_2   Inlet.Fl_O            3421.6       15.235       552.48     132.299     0.00000     3406.28      11.916      515.03     11759.7      0.6031      1.3997
FS_21  Fan.Fl_O              3421.6       23.647       630.80     151.12      0.00000     2345.03      20.945      609.37     10449.9      0.4202      1.3981
FS_13  Splitter.Fl_02        3058.4       23.647       630.80     151.12      0.00000     2096.14      21.054      610.27      9508.5      0.4110      1.3981
FS_22  Splitter.Fl_01        363.15       23.647       630.80     151.12      0.00000      248.89      22.215      619.68      1475.8      0.3003      1.3981
FS_23  Duct4.Fl_O            363.15       23.410       630.80     151.12      0.00000      251.4       21.938      619.24      1464.8      0.3062      1.3981
FS_24  LPC.Fl_O              363.15       32.569       699.95     167.796     0.00000      190.35      29.210      678.63      886.78      0.3979      1.3967
FS_25  Duct6.Fl_O            363.15       32.309       699.95     167.796     0.00000      191.89      28.881      678.00      883.01      0.4039      1.3967
FS_3   HPC.Fl_O              339.17      500.347      1571.69     388.494     0.00000       17.34      472.15     1548.36      106.44      0.2945      1.3467
FS_36  Bld3.Fl_O             274.73      500.347      1571.69     388.494     0.00000       14.05      472.15     1548.36       86.22      0.2945      1.3467
FS_4   Burner.Fl_O           283.17      477.832      3293.82     922.34      0.03070       21.95      475.48     3290.39      442.92      0.0882      1.2678
FS_45  HPT.Fl_O              347.61      110.732      2280.85     602.192     0.02490       96.74     103.913     2247.49      570.77      0.3137      1.3011
FS_46  ATD_Bleed.Fl_O        347.61      110.732      2280.85     602.192     0.02490       96.74     103.913     2247.49      570.77      0.3137      1.3011
FS_48  Duct11.Fl_O           347.61      109.625      2280.85     602.192     0.02490       97.72     106.836     2267.27      877.58      0.1993      1.3011
FS_5   LPT.Fl_O              367.58       20.662      1546.88     391.121     0.02350      451.52      19.731     1529.19      3081.2      0.2638      1.3319
FS_7   Duct13.Fl_O           367.58       20.496      1546.88     391.121     0.02350      455.16      19.450     1526.78      2927.8      0.2814      1.3319
FS_9   Core_Nozz.Fl_O        367.58       20.496      1546.88     391.121     0.02350      455.16      14.696     1422.98      1459.1      0.7199      1.3319
FS_15  BypBld.Fl_O           3043.1       23.647       630.80     151.12      0.00000     2085.66      21.084      610.52      9508.9      0.4084      1.3981
FS_17  Duct15.Fl_O           3043.1       23.221       630.80     151.12      0.00000     2123.89      20.597      609.61      9508.6      0.4177      1.3981
FS_19  Byp_Nozz.Fl_O         3043.1       23.221       630.80     151.12      0.00000     2123.89      14.696      553.31      6330.0      0.8375      1.3981

TURBOMACHINERY PERFORMANCE DATA
                Wc           PR          eff          Nc          TR        efPoly        pwr         SMN         SMW        s_Re
     Fan     3406.28       1.552       0.9474      2175.82      1.142       0.9500     -91112.6      21.71       14.58         1
     LPC       251.4       1.391       0.9003      2036.27      1.110       0.9061        -8568       40.2       24.67         1
     HPC      191.89      15.486       0.8923      8064.76      2.245       0.9235    -108529.1      14.16       14.12         1
     HPT       21.95       4.315       0.9006       163.24      1.327       0.8847     108779.1
     LPT       97.72       5.306       0.9249        47.02      1.438       0.9092      99680.6

TURBOMACHINERY MAP DATA
               WcMap        PRmap       effMap       NcMap       R/Parm      s_WcDes     s_PRdes     s_effDes    s_NcDes
     Fan     3102.13       1.598       0.9051       0.982       1.674       1.0980      0.9231      1.0468      0.0005
     LPC      224.31       1.196       0.9011       0.925       2.040       1.1208      1.9920      0.9991      0.0005
     HPC       228.6      16.163       0.8716       0.999       1.833       0.8394      0.9554      1.0238      0.0001
     HPT       22.36       5.044       0.9202      102.22       5.044       0.9815      1.2199      0.9787      0.0275
     LPT      150.71       4.759       0.9315       95.57       4.759       0.6441      0.8731      0.9864      0.0892

   INLETS      eRam          Afs         Fram
    Inlet     0.9925     23678.28       30442

   DUCTS      dPnorm          MN          Aphy               BLEEDS - interstg     Wb/Win      BldWk        BldP          W           Tt          ht          Pt
    Duct4      0.010       0.3003      1475.83               ATDbleed HPC.A>       0.000       1.000       1.000       0.000     1571.69      388.49     500.347
    Duct6      0.008       0.3979       886.78               CstmrBld HPC.C>       0.0110       0.35       0.1465       4.000     1015.16      245.04     100.876
    Duct11     0.010       0.3137       570.77               C_LPTexit HPC.C>      0.0200       0.35       0.1465       7.263     1015.16      245.04      20.662
    Duct13     0.008       0.2638      3081.15               C_LPTinlt HPC.C>      0.0350       0.35       0.1465      12.710     1015.16      245.04     109.625
    Duct15     0.018       0.4084      9508.89

 SPLITTERS      BPR         dP/P1        dP/P2               BLEEDS - output       Wb/Win      hscale      Pscale         W           Tt          ht          Pt
  Splitter     8.422        0.000        0.000               C_HPTexit Bld3.>      0.0750      1.000       1.000      25.4381     1571.69      388.49     110.732
                                                             C_HPTinlt Bld3.>      0.1150      1.000       1.000      39.005      1571.69      388.49     477.832
  SHAFTS       Nmech        trq in       pwr in              Duct15Lk BypBl>       0.0050      1.000       1.000      15.2921      630.8      151.12      23.647
  HP_Shaft    9368.7      60981.7     108779.1
  LP_Shaft    2245.6     233135.9      99680.6

  BURNERS      TtOut          eff        dPnorm       Wfuel        FAR         EINOx
   Burner     3293.82       0.995        0.045      8.43591     0.03071      44.778

  NOZZLES       PR            Cfg         CdTh          Cv          Ath         MNth        Vact          Fg
 Core_Nozz     1.395        0.9929       0.9315      0.9958      1566.4        0.72       1293.1       14773
  Byp_Nozz     1.580        0.9927       0.9437      0.9957      6707.7       0.838        958.6       90669
```

Figure 101. TKO Off-design Cycle Data for Establishing Initial Iterate

APPENDIX I

MULTIVARIATE PLOTS OF ACTIVE CONSTRAINTS FOR EXPERIMENT 2

The following figures show multivariate plots with histograms for each of the constraints in experiment 2. Every data point represents the value of the design variables when the constraint is active. Examination of the multivariate plots and histograms can reveal several characteristics about the constraint including; how often the constraint is active, which design variables most effect the likelihood of the constraint activation, and the values of the design variables that tend to activate the constraint.

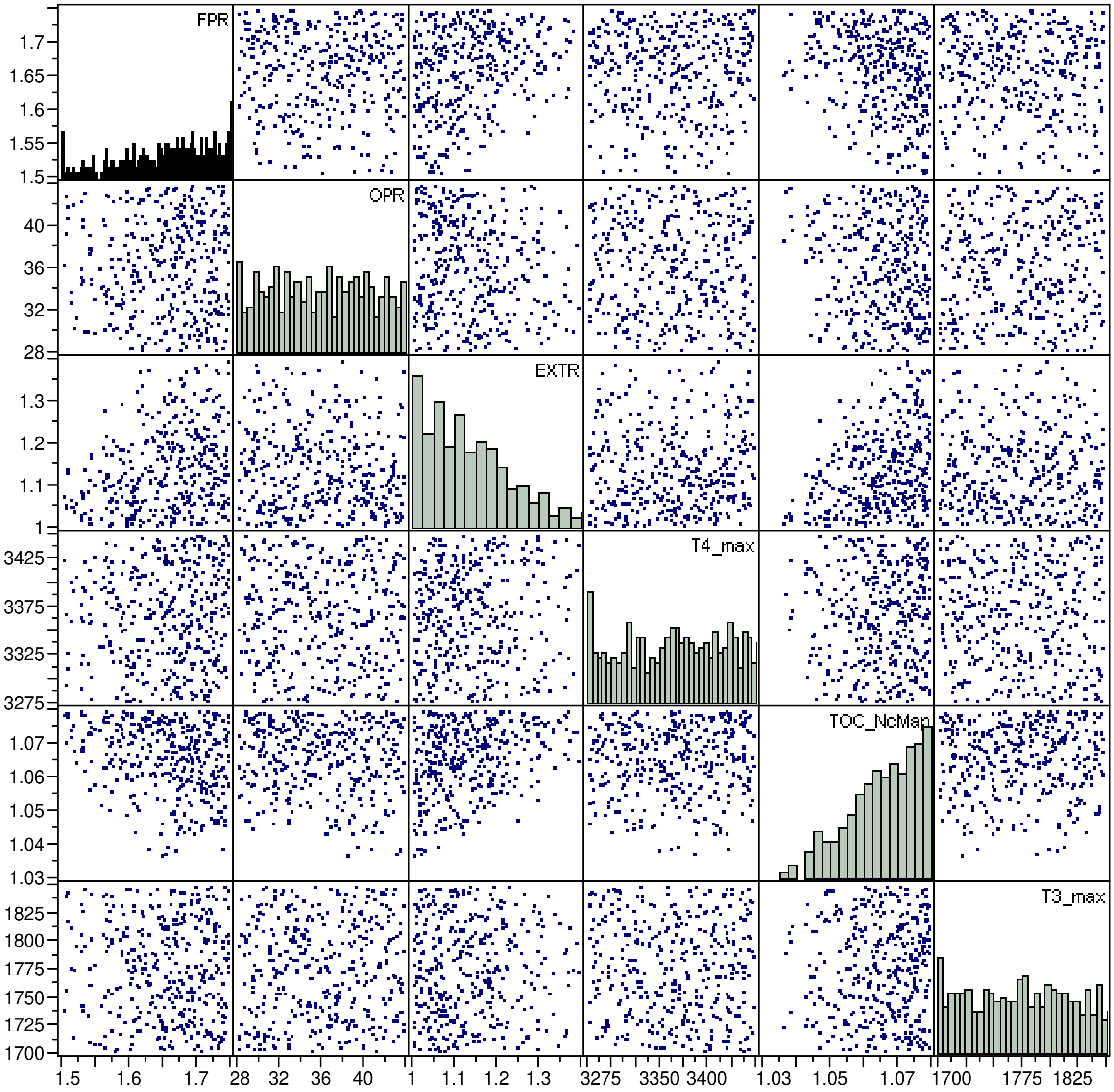


Figure 102. Multivariate Plot of Active Constraint HHDTKO_HPT_BleedFlow1

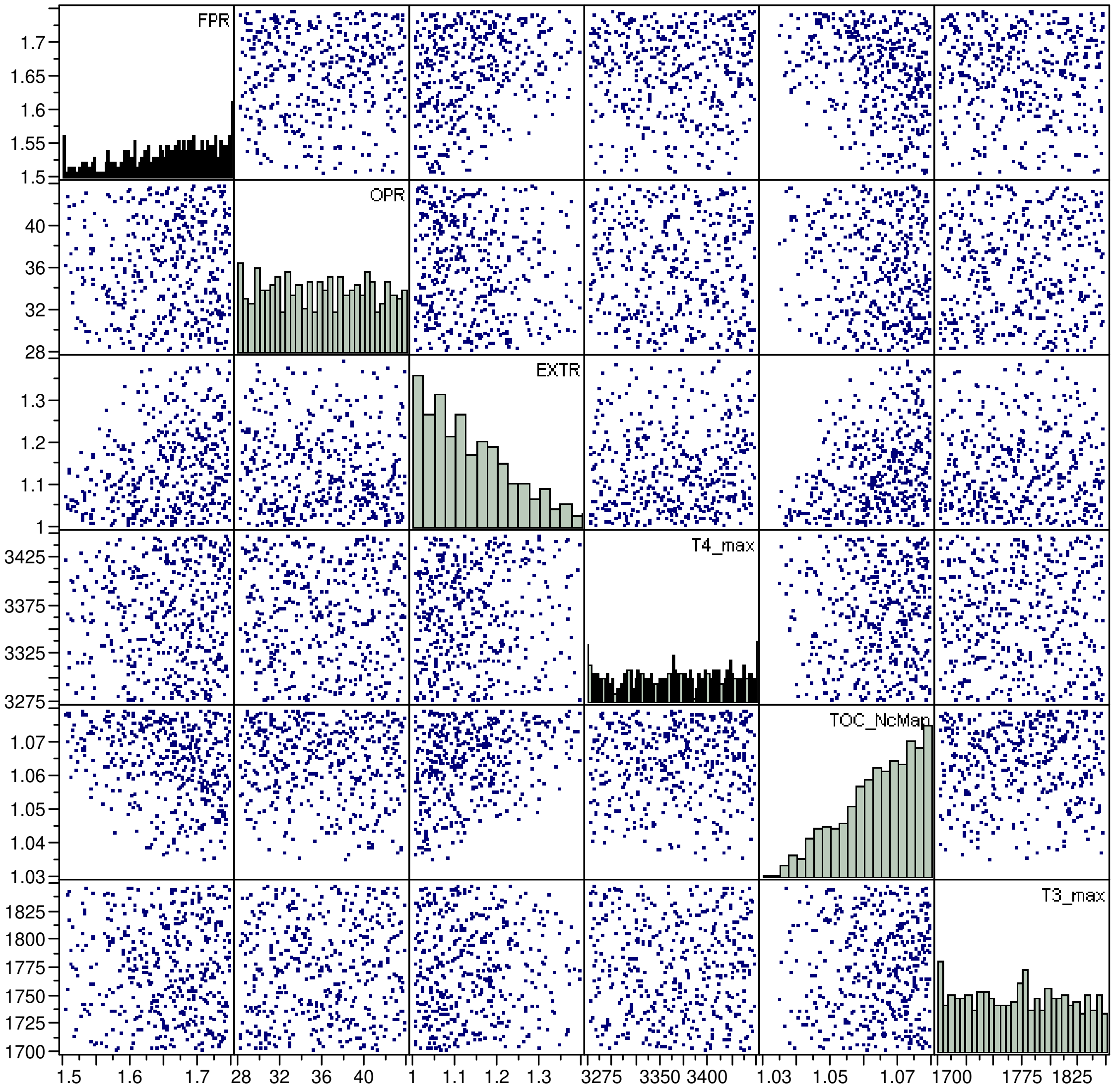


Figure 103. Multivariate Plot of Active Constraint HHDTKO_HPT_BleedFlow2

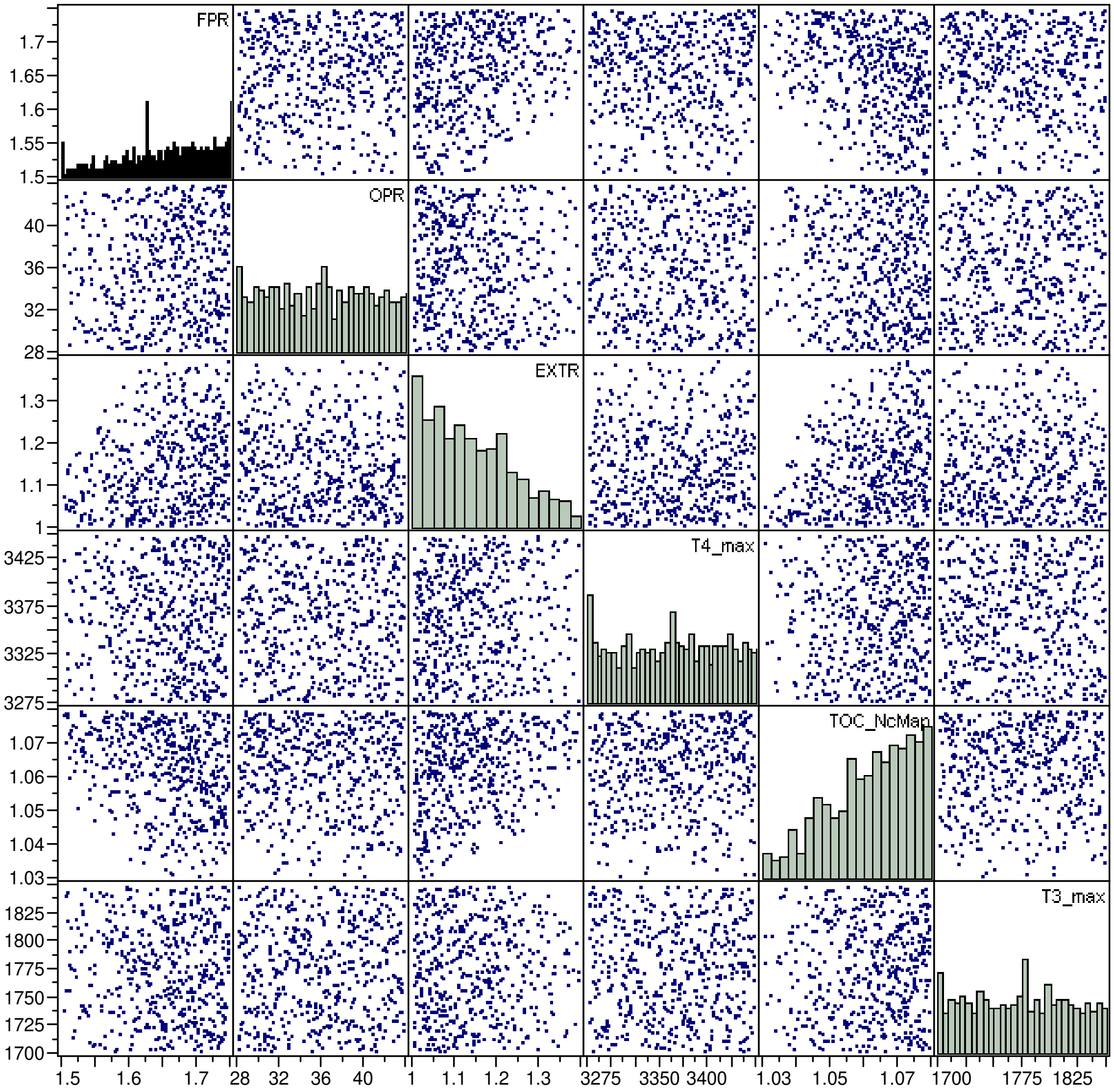


Figure 104. Multivariate Plot of Active Constraint HHDTKO_LPT_BleedFlow1

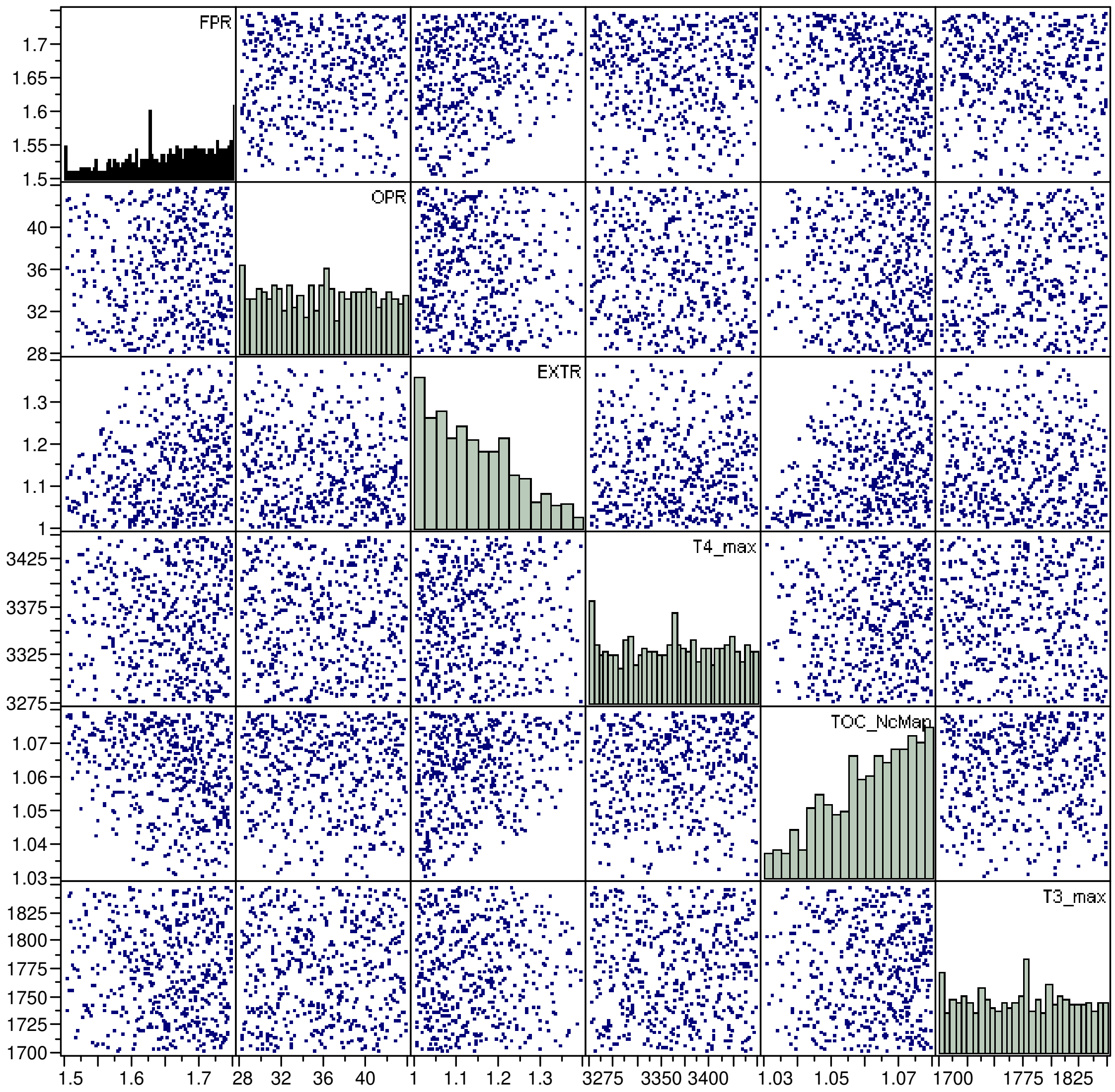


Figure 105. Multivariate Plot of Active Constraint HHDTKO_LPT_BleedFlow2

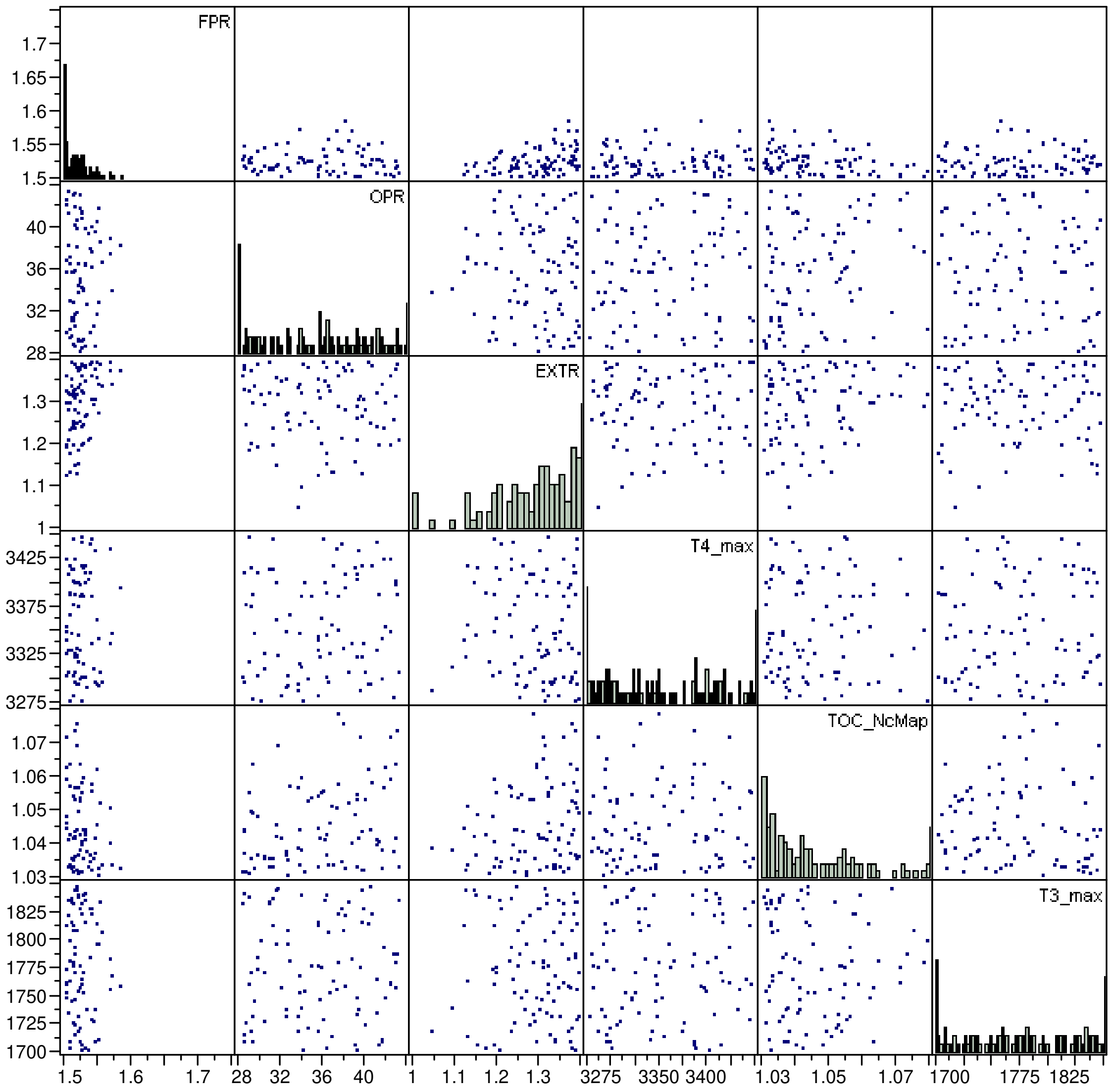


Figure 106. Multivariate Plot of Active Constraint Max_Fan_Diameter

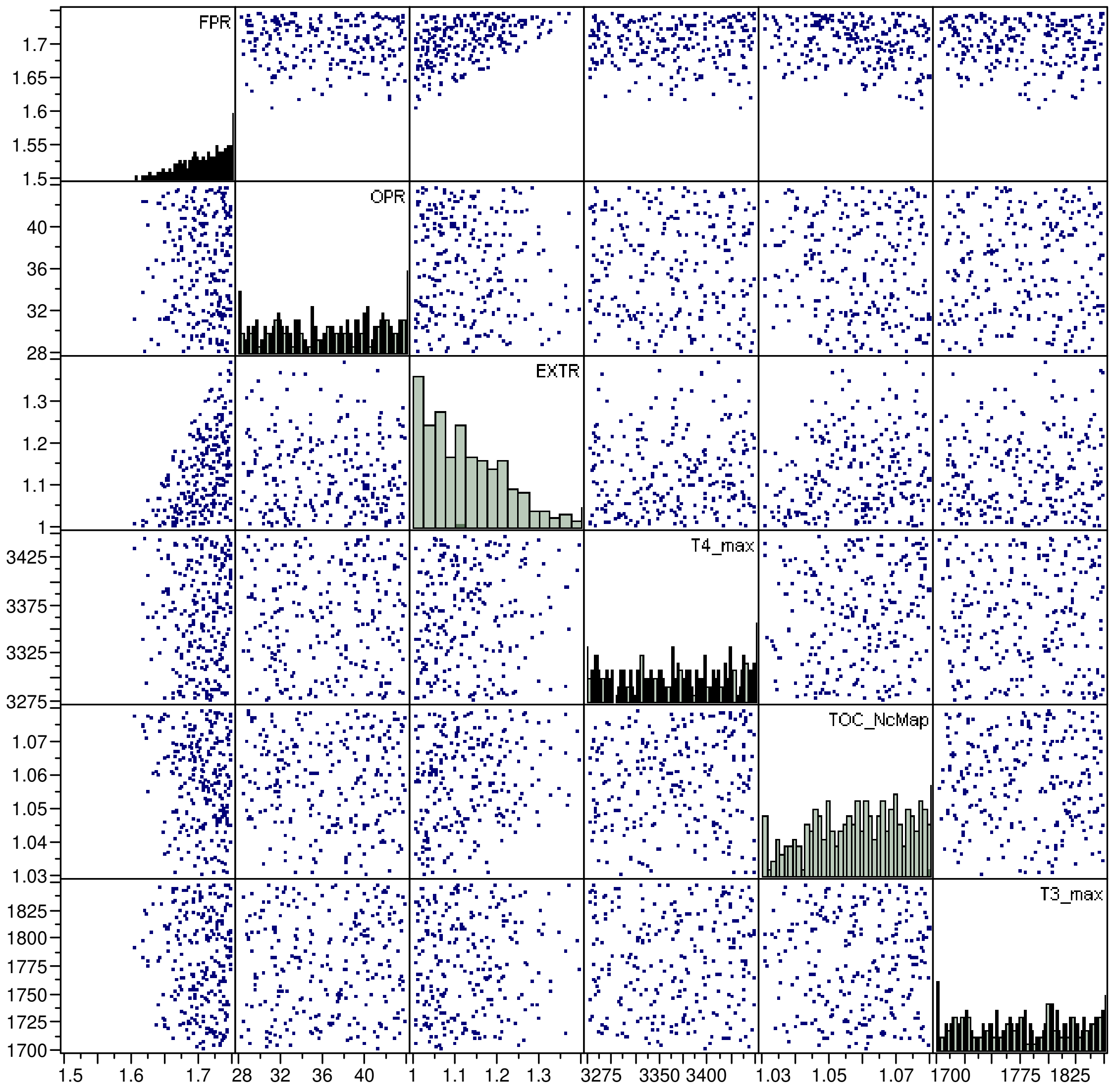


Figure 107. Multivariate Plot of Active Constraint Cutback_Vmix

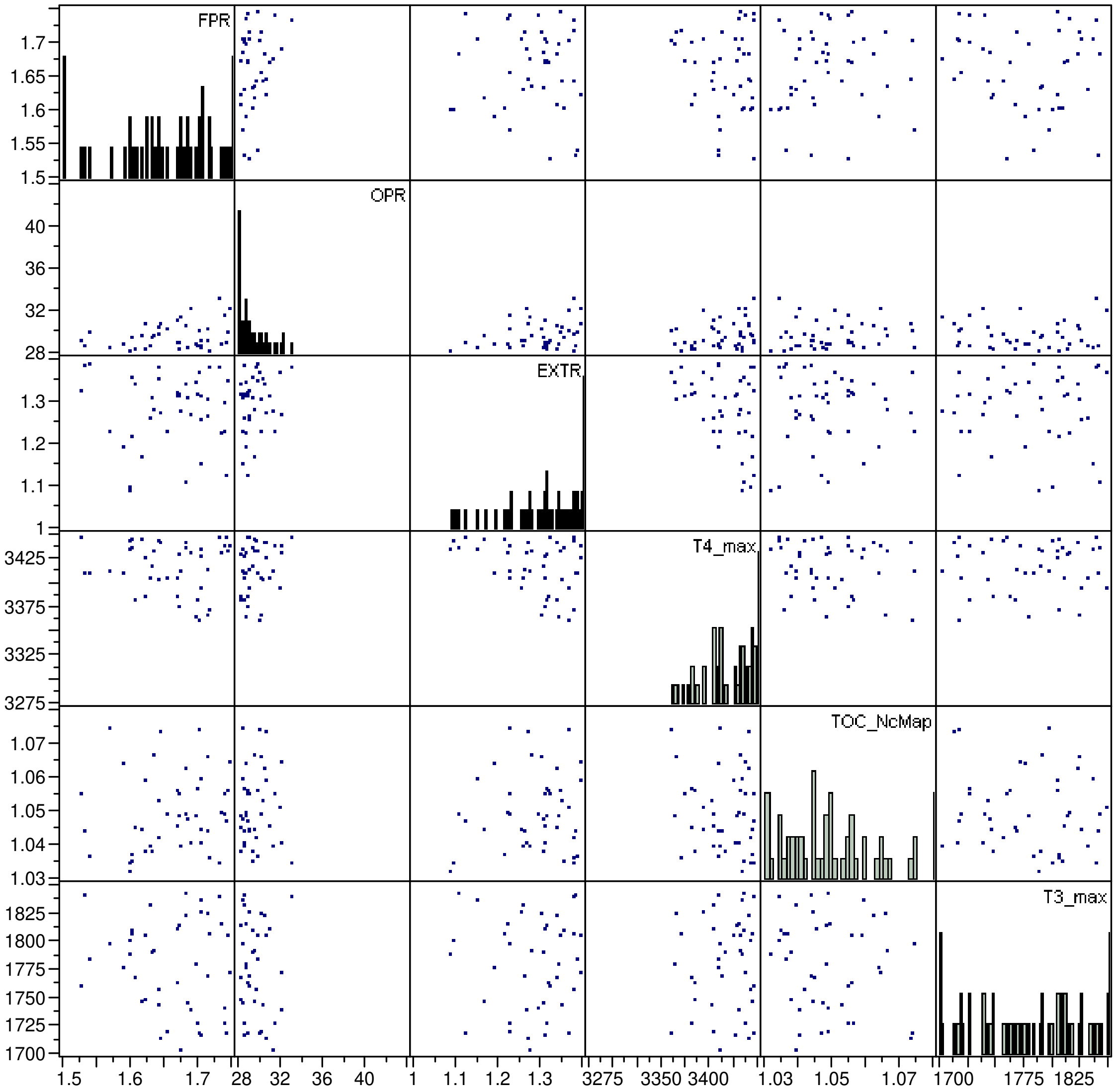


Figure 108. Multivariate Plot of Active Constraint LPCDP_LPCPRmin

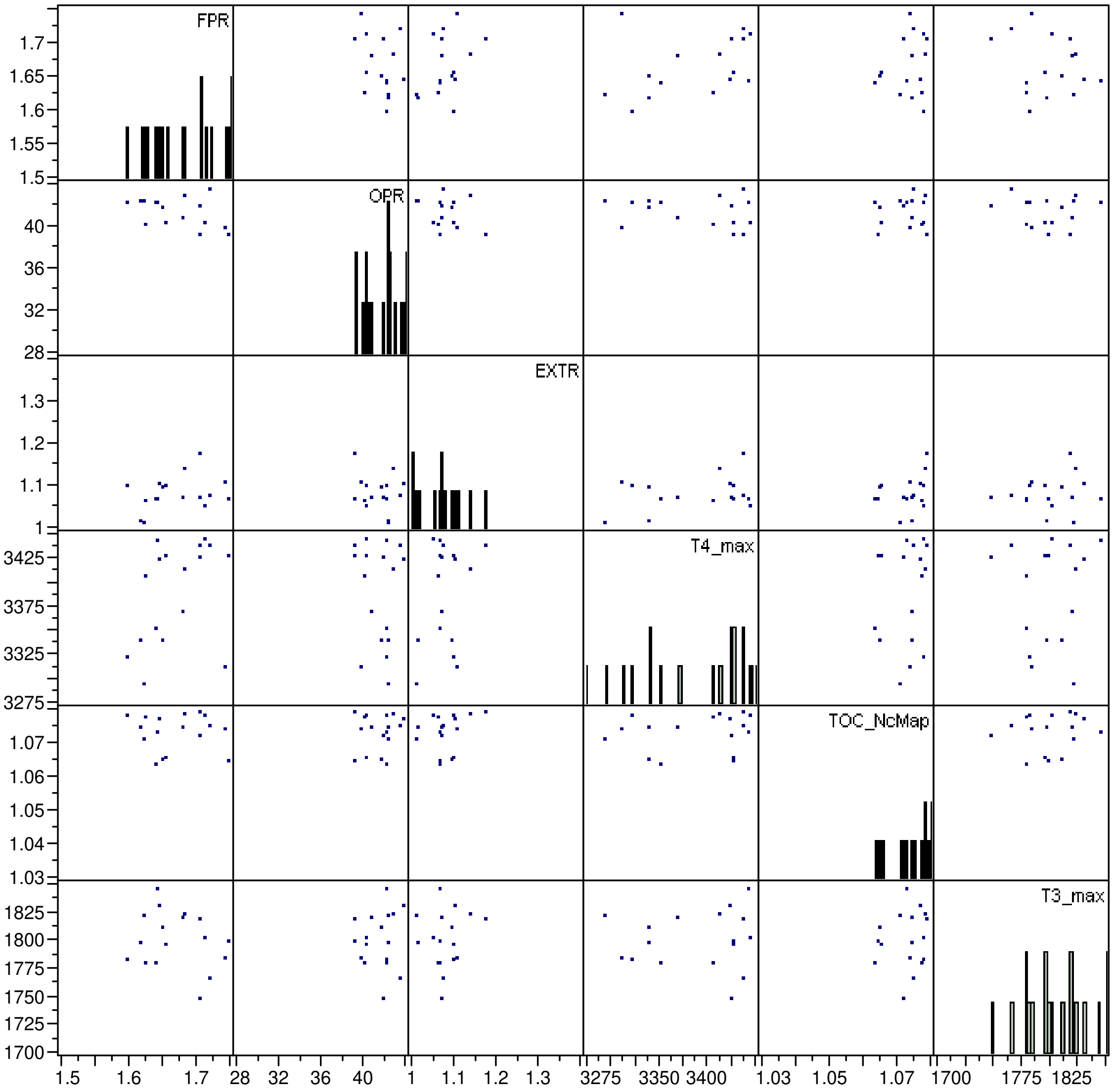


Figure 109. Multivariate Plot of Active Constraint HHDTKO_T3

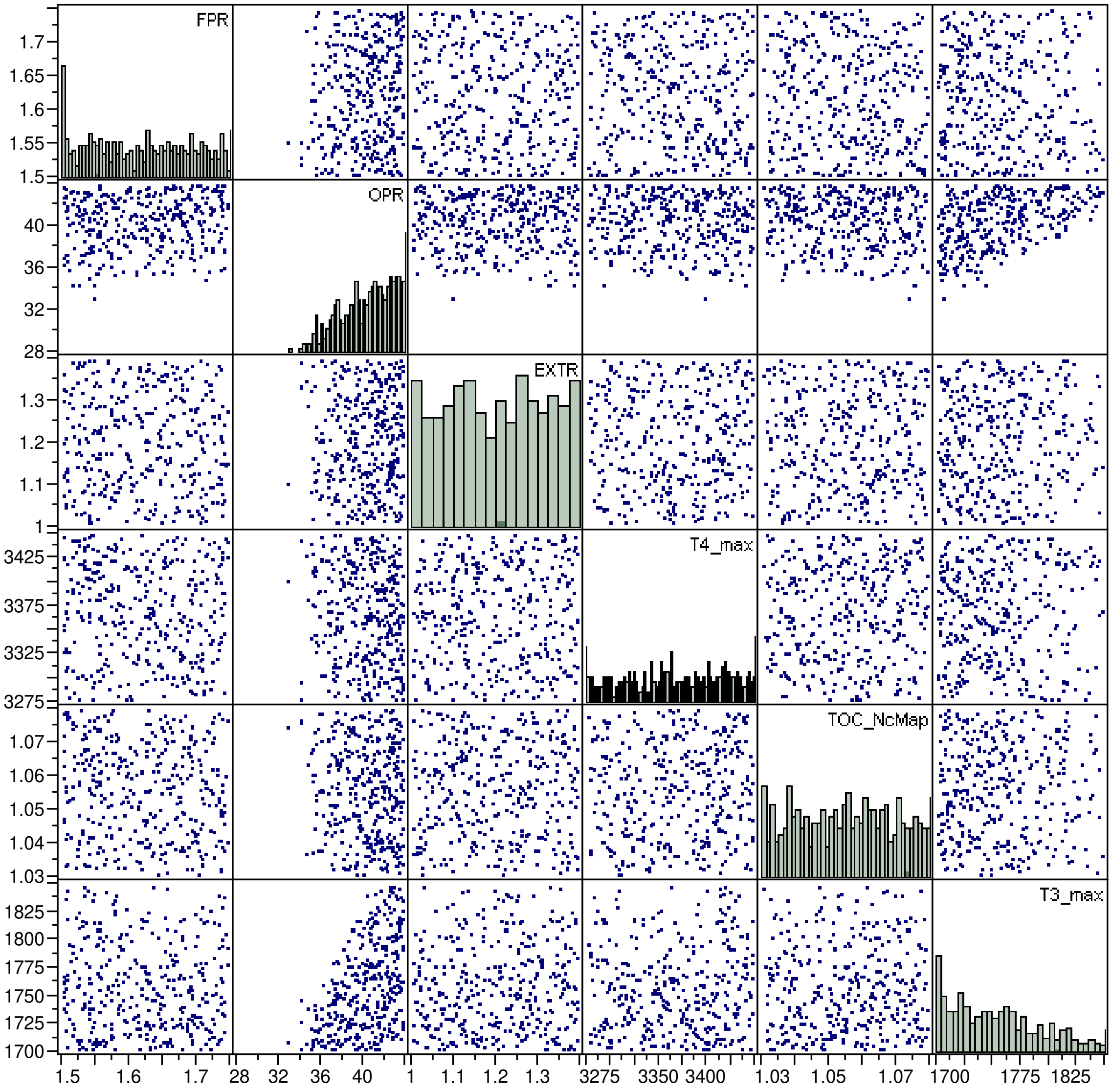


Figure 110. Multivariate Plot of Active Constraint TKO_T3

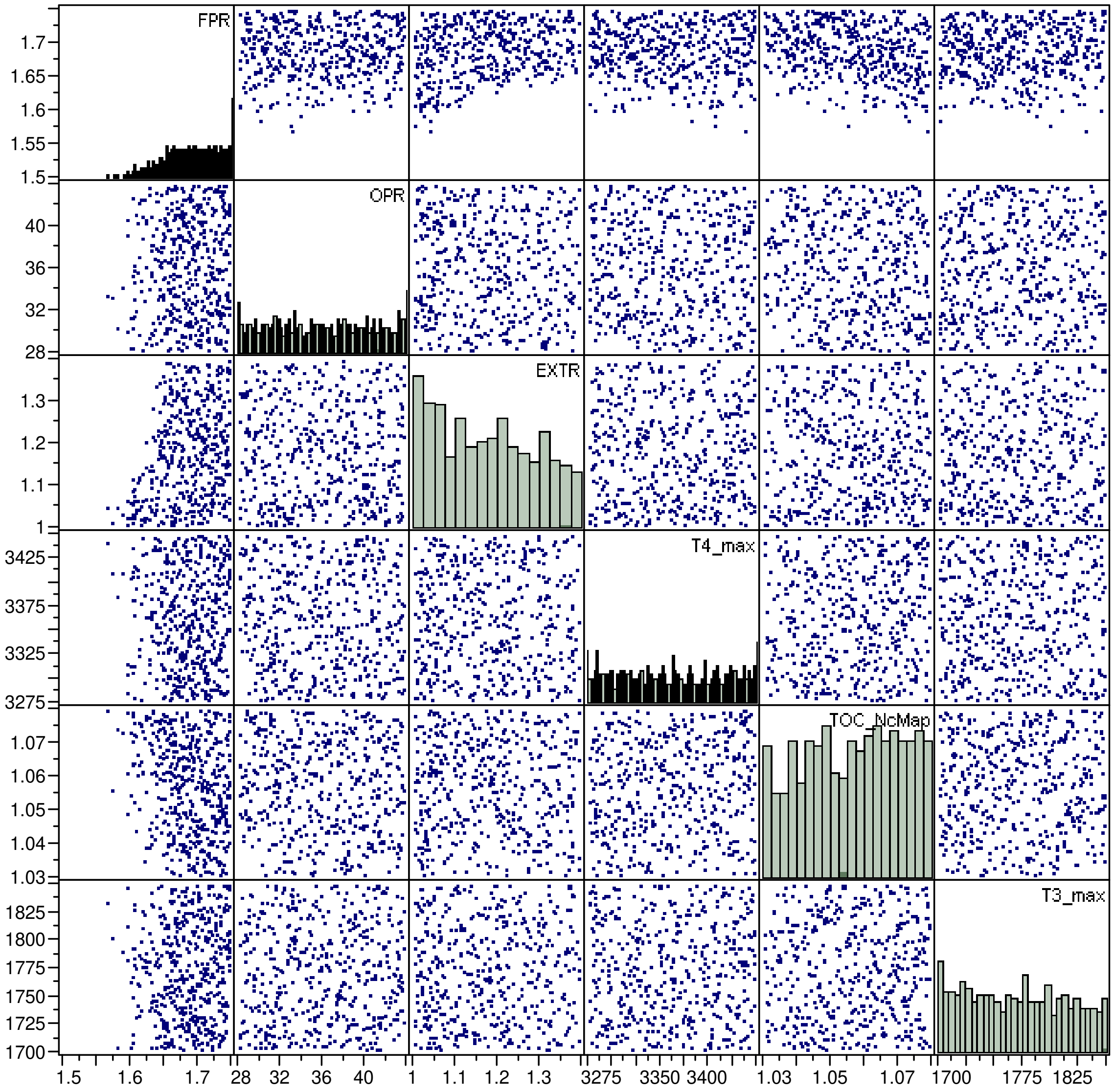


Figure 111. Multivariate Plot of Active Constraint TOC2_Fan_Nc

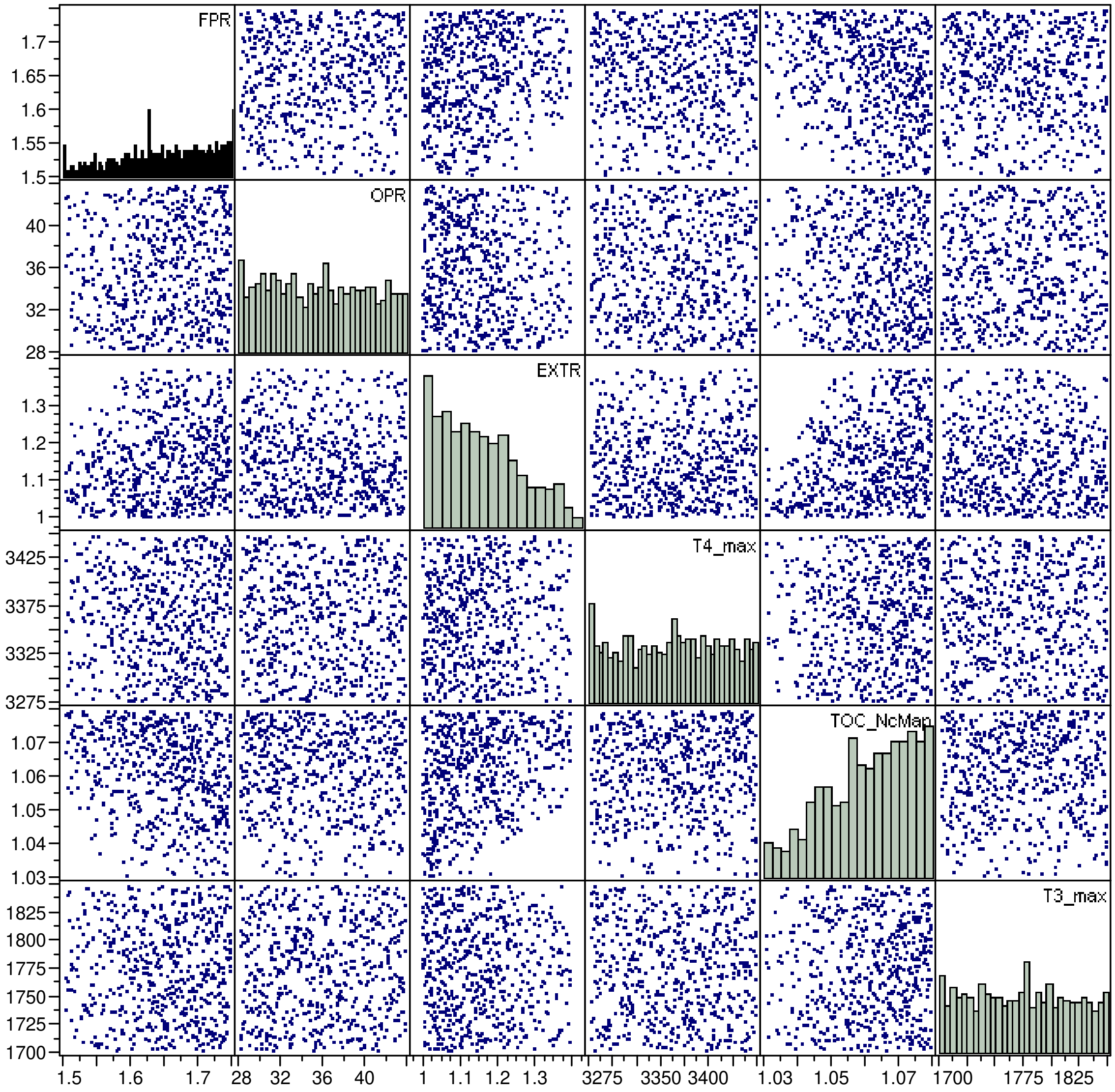


Figure 112. Multivariate Plot of Active Constraint HHDTKO_T4max

REFERENCES

[1] Gunsalus, J. (2007) *Airbus unlikely to sign to use .G.E engines on .A350 this year*

[2] *The house that jack built.* Economist, 1999. **352**(8137): p. 23-26.

[3] Layne, R. and Gunsalus, J. (2007) *G.E may develop engine to compete for business jets (update1)*

[4] Kerrebrock, J.L., *Aircraft engines and gas turbines*. 1977: MIT Press Cambridge, Mass.

[5] Plourde, G.A., Sims, R.E., and Cea, R.A., *Fighter engine cycle selection*, in *AIAA/SAE/ASME 19th Joint Propulsion Conference*. 1983: Seattle, WA.

[6] Willis, D., *Raptor – fighter supreme.* Aviation News, 2005(January).

[7] Jacquet, J.-M. and Seiwert, D.L., *Methodology for commercial engine/aircraft optimization* in *AIAA/SAE/ASME 19th Joint Propulsion Conference29th Joint Propulsion Conference*. 1993, AIAA: Monterey, CA.

[8] Cohen, H., Rogers, G.F.C., and Saravanamuttoo, H.I.H., *Gas turbine theory*. 1996: Longman, London.

[9] Mattingly, J.D., *Elements of gas turbine performance*. 1996: McGraw-Hill, Inc.

[10] Oates, G.C., *Aerothermodynamics of gas turbine and rocket propulsion*. 1997: AIAA.

[11] Simmons, J.R., *Preliminary design engine thermodynamic cycle selection for advanced fighters*, in *AlAA Aircraft Design, Systems and Technology Meeting*. 1983, AIAA: Fort Worth, TX.

[12] Bevilaqua, P.M., *Joint strike fighter dual-cycle propulsion system.* Journal of Propulsion and Power, 2005. **21**(5): p. 778-783.

[13] Liew, K.H., Urip, E., and Yang, S.L., *Parametric cycle analysis of a turbofan engine with an interstage turbine burner.* Journal of Propulsion and Power, 2005. **21**(3): p. 546-551.

[14] Liew, K.H., et al., *Performance cycle analysis of turbofan engine with interstage turbine burner.* Journal of Propulsion and Power, 2006. **22**(2): p. 411-416.

[15] Wood, A. and Pilidis, P., *A variable cycle jet engine for astovl aircraft.* Aircraft Engineering and Aerospace Technology, 1997. **69**(6): p. 534.

[16] Herteman, J.P. and Goutines, M., *Design principles and methods for military turbojet engines*, in *Design Principles and Methods for Aircraft Gas Turbine Engines*. 1999, NATO.

[17] Timnat, Y.M., *Advanced airbreathing propulsion*. 1996, Malabar, FL: Krieger Publishing Co.

[18] Papamoschou, D., *Engine cycle and exhaust configuration for quiet supersonic propulsion.* Journal of Propulsion and Power, 2004. **20**(2): p. 255-262.

[19] Callender, B., Gutmark, E., and Martens, S., *Far-field acoustic investigation into chevron nozzle mechanisms and trends.* AIAA Journal, 2005. **43**(1): p. 87-95.

[20] Viswanathan, K., *Elegant concept for reduction of jet noise from turbofan engines.* Journal of Aircraft, 2006. **43**(3): p. 616-626.

[21] Saiyed, N.H., Mikkelsen, K.L., and Bridges, J.E., *Acoustics and thrust of quiet separate-flow high-bypass-ratio nozzles.* AIAA Journal, 2003. **41**(3): p. 372-378.

[22] Steinmetz, R.B. and Wagner, M.J., *Turbofan engine cycle design selection-year 2000.* AIAA, AHS, and ASEE, aircraft systems, design and technology meeting, 1986. **20**.

[23] *Current market outlook brochure*. 2007, The Boeing Company.

[24] *Boeing current market outlook database*. 2007, The Boeing Company.

[25] Deluis, J., *A process for the quantification of aircraft noise and emissions interdependencies*. 2008, Georgia Institute of Technology: Atlanta, GA.

[26] Kraemer, K.L. and King, J.L., *Computer-based models for policy making: Uses and impacts in the us federal government.* Operations Research, 1986. **34**(4): p. 501-512.

[27] Halstead, D., *Environmental tradeoffs in commercial aircraft design: A.I.A .E.D.S feasibility test and lessons learned.* 2004, F.A.A Transportation Research Board.

[28] Shakariyants, S.A., et al. *Generic airplane and aero-engine simulation procedures for exhaust emission studies*. in *ASME Turbo Expo 2007: Power for Land, Sea and Air*. 2007. Montreal.

[29] Morden, D.B., *Engine/airframe performance matching*, in *Aircraft propulsion systems technology and design*, G.C. Oates, Editor. 1989, AIAA.

[30] Walsh, P.P. and Fletcher, P., *Gas turbine performance*. 2nd ed. 2004: Blackwell Science Ltd.

[31] Boyce, M.P., *Gas turbine engineering handbook*. 3rd ed. 2006: Gulf Professional Publishing.

[32] Moran, M.J. and Shapiro, H.N., *Fundamentals of engineering thermodynamics*. 1988: John Wiley & Sons, Inc.

[33] Roth, B.A. and Mavris, D.N., *A work availability perspective of turbofan engine performance.* 2001.

[34] Roth, B. and Mavris, D., *A comparison of thermodynamic loss models applied to the j-79 turbojet engine.* 36th Joint Propulsion Conference and Exhibit, Huntsville, AL, July, 2000.

[35] Bullock, R.O., *Design and development of aircraft propulsion systems*, in *Aircraft propulsion systems technology and design*, G.C. Oates, Editor. 1989, AIAA.

[36] *Aircraft propulsion system performance station designation and nomenclature*, in *Aerospace Recommended Practice (ARP) 755B*. 1994, Society of Automotive Engineers Warrendale, PA.

[37] Turner, M., et al., *Multi-fidelity simulation of a turbofan engine with results zoomed into mini-maps for a zero-d cycle simulation*. 2004, NASA.

[38] Stricker, J.M., *The gas turbine engine conceptual design process - an integrated approach*, in *Design Principles and Methods for Aircraft Gas Turbine Engines*. 1999, NATO.

[39] Jones, M.J., Bradbrook, S.J., and Nurney, K., *A preliminary engine design process for an affordable capability*, in *Reduction of Military Vehicle Acquisition Time and Cost through Advanced Modelling and Virtual Simulation*. 2003, NATO.

[40] Raymer, D.P., ed. *Aircraft design: A conceptual approach*. Aiaa education series, ed. J. S. Przemieniecki. 1999, AIAA: Reston, VA.

[41] Fowler, T.W., ed. *Jet engines and propulsion systems for engineers*. 1989, GE Aircraft Engines: Cincinnati, OH.

[42] Healy, M.J., Kowalik, J.S., and Ramsay, J.W., *Airplane engine selection by optimization on surface fit approximations.* Journal of Aircraft, 1975. **12**(7): p. 593-599.

[43] Lord, W.K., Macmartin, D.G., and Tillman, T.G., *Flow control opportunities in gas turbine engines.* AIAA Paper, 2000. **2234**: p. 19-22.

[44] Mavris, D.N., Macsotai, N.I., and Roth, B.A., *A probabilistic design methodology for commercial aircraft engine cycle selection.* 1998.

[45] Mavris, D., Roth, B., and Macsotai, N. *A method for probabilistic sensitivity analysis of commercial aircraft engines*. in *ISABE - 14th International Symposium on Air Breathing Engines*. 1999. Florence, Italy: International Society for Air Breathing Engines - ISOABE.

[46] Roth, B., et al., *Adaptive selection of pareto optimal engine technology solution sets*, in *ICAS*. 2002.

[47] *The aircraft gas turbine engine and its operation*. 1988: United Technologies Corporation, P&W Oper. Instr. 200.

[48] Hill, P.G. and Peterson, C.R., *Mechanics and thermodynamics of propulsion*. 2nd ed. 1992: Addison-Wesley Publishing Company.

[49] Cohen, H. and Rogers, G.F.C., *Gas turbine theory*. 1951: Longmans, Green and CO.

[50] Saravanamuttoo, H.I.H., Rogers, G.F.C., and Cohen, H., *Gas turbine theory*. 2001: Prentice Hall.

[51] Cumpsty, N.A., *Jet propulsion: A simple guide to the aerodynamic and thermodynamic design and performance of jet engines*. 2003: Cambridge University Press.

[52] Reed, J. and Afjeh, A. *A comparative study of high and low fidelity fan models for turbofan engine system simulation*. in *IASTED International Conference on Applied Modelling and Simulation*. 1997. Banff, Canada.

[53] Sanghi, V., *Technical notes: Sensitivity of design point choice on engine cycle selection.* Journal of Propulsion and Power, 1999. **15**(5).

[54] Bathie, W., *Fundamentals of gas turbines*. 2nd ed. 1996: John Wiley and Sons, New York.

[55] Ypma, T.J., *Historical development of the newton-raphson method.* SIAM Review, 1995. **37**(4): p. 531-551.

[56] *Standard atmosphere*, in *ISO 2533:1975*. 1975, International Organization for Standardization.

[57] Mattingly, J.D., Heiser, W.H., and Pratt, D.T., *Aircraft engine design*. 2002: AIAA.

[58] *N.P.S.S user guide, software release: N.P.S.S_1.6.4, rev: Q*. 2006, NASA.

[59] William, H.P., et al., *Numerical recipes in fortran (2nd ed.): The art of scientific computing*. 1992: Cambridge University Press. 963.

[60] Kharab, A. and Guenther, R., *An introduction to numerical methods: A matlab approach*. 2006: Chapman & Hall/CRC.

[61] Visser, W., Broomhead, M., and Van Der Vorst, J., *Terts, a generic real-time gas turbine simulation environment.* ASME Paper, 2001: p. 0446.

[62] Chappell, M.A. and Mclaughlin, P.W., *Approach to modeling continuous turbine engine operation from startup to shutdown.* Journal of Propulsion and Power, 1993. **9**(3): p. 466-471.

[63] Kim, J.H., et al., *Dynamic simulation of full startup procedure of heavy-duty gas turbines.* Journal of Engineering for Gas Turbines and Power, 2002. **124**(3): p. 510-516.

[64] Visser, W.P.J., Kogenhop, O., and Oostveen, M., *A generic approach for gas turbine adaptive modeling.* Journal of Engineering for Gas Turbines and Power, 2006. **128**(1): p. 13-19.

[65] Press, W., et al., *Numerical recipes in c: The art of scientific programming*, in *Cambridge U Press, Cambridge, England*. 1992.

[66] Li, Y., *Performance-analysis-based gas turbine diagnostics: A review.* Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy, 2002. **216**(5): p. 363-377.

[67] Ogaji, S.O.T., et al., *Parameter selection for diagnosing a gas-turbine's performance-deterioration.* Applied Energy, 2002. **73**(1): p. 25-46.

[68] Quarteroni, A., Sacco, R., and Saleri, F., *Numerical mathematics*. 2007: Springer Verlag.

[69] Dennis Jr, J.E. and Schnabel, R.B., *Numerical methods for unconstrained optimization and nonlinear equations (classics in applied mathematics, 16).* 1996.

[70] Kelley, C.T., *Solving nonlinear equations with newton's method.* 2003: Society for Industrial Mathematics.

[71] Eldén, L. and Wittmeyer-Koch, L., *Numerical analysis: An introduction*. 1990: Academic Press Professional, Inc. San Diego, CA, USA.

[72] Broyden, C.G., *The convergence of single-rank quasi-newton methods.* Mathematics of Computation, 1970. **24**(110): p. 365-382.

[73] Broyden, C.G., *A class of methods for solving nonlinear simultaneous equations.* Mathematics of Computation, 1965. **19**(92): p. 577-593.

[74] Gay, D.M., *Some convergence properties of broyden's method.* SIAM Journal on Numerical Analysis, 1979. **16**(4): p. 623-630.

[75] Dennis, J.E., Jr., *On the convergence of broyden's method for nonlinear systems of equations.* Mathematics of Computation, 1971. **25**(115): p. 559-567.

[76] Mathews, J.H., *Numerical methods for computer science, engineering and mathematics*. 1987: Prentice-Hall, Inc. Englewood Cliffs, NJ.

[77] Lytle, J., et al., *Numerical propulsion system simulation (npss) 1999 industry review [r]*, NASA TM-200-209795, 200.

[78] *N.P.S.S developer's guide, software release: N.P.S.S_1.6.4, rev: Q*. 2006, NASA.

[79] *N.P.S.S reference sheets, software release: N.P.S.S_1.6.4, rev: Q*. 2007, NASA.

[80] Curlett, B.P. and Felder, J.L., *Object-oriented approach for gas turbine engine simulation*. 1995, NASA TM-106970.

[81] Jones, S.M., *An introduction to thermodynamic performance analysis of aircraft gas turbine engine cycles*
using the numerical propulsion system simulation code. 2007, NASA TM 2007-214690.

[82] Lylte, J.K., *The numerical propulsion system simulation: An overview*. 2000, NASA TM 2000-209915.

[83] Lylte, J.K., *The numerical propulsion system simulation: A multidisciplinary design system for aerospace vehicles*. 1999, NASA TM 1999-209194.

[84] Lylte, J.K., *2001 numerical propulsion system simulation review*. 2002, NASA TM 2002-211197.

[85] Kowalski, E.J. and Atkins Jr., A., *A computer code for estimating installed performance of aircraft gas turbine engines*. 1979, NASA CR 159693.

[86] Creason, T. and Baghdadi, S., *Design and test of a low aspect ratio fan stage* in *24th Joint Propulsion Conference and Exhibit, Boston, MA, July*. 1988.

[87] Cline, S.J., et al., *Energy efficient engine. Fan and quarter-stage component performance report* 1983, NASA CR 168070.

[88] Converse, G.L. and Giffin, R.G., *Extended parametric representation of compressor fans and turbines. Volume 1: Cmgen user's manual* 1984, NASA CR 174645.

[89] Oates, G.C., *The aerothermodynamics of aircraft gas turbine engines*. 1978, AFAPL-TR-78-52.

[90] Groeneweg, J.F. and Rice, E.J., *Aircraft turbofan noise.* Journal of Turbomachinery, 1987. **109**: p. 130-141.

[91] Soares, C., *Gas turbines: A handbook of air, land and sea applications*. 2007, Butterworth-Heinemann.

VITA

Jeff Schutte was born in Louisville Kentucky where he attended the University of Louisville obtaining his Bachelors of Science degree in mechanical engineering in 1996 and a Masters of Engineering degree in mechanical engineering the following year. Upon graduation he worked 6 years for Cummins Inc. located in Columbus Indiana, initially developing fuel system components such as actuators and injectors and then developing natural gas engine performance calibrations for power generation. In 2003, Jeff left Cummins to pursue his Ph.D. in aerospace engineering at Georgia Tech.