PARAMETRIC CYCLE ANALYSIS OF ADAPTIVE CYCLE ENGINE

By

HARINKUMAR RAJENDRABHAI PATEL

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE IN AEROSPACE ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

MAY 2016

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompany the success of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned my efforts with success.

First, I would like to express my sincere gratitude to my advisor Dr. Donald R. Wilson for his continuous guidance, support of my Master's thesis study and related research, for his patience, motivation, and immense knowledge. The door to Dr. Wilson's office was always open whenever I ran into a trouble spot or had a question about my research or writing. I could not have imagined having a better advisor and mentor for my thesis work.

Besides my advisor, I would like to acknowledge and thank my thesis defense committee: Dr. Frank K Lu and Dr. Brian Van Lear, for their insightful comments, critique and encouragement.

I would also like to thank my dear friends and colleagues, for their constant support. Furthermore, I would like to express my very profound gratitude to my parents and family members for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Last, but not the least, I would also like to thank the Mechanical and Aerospace Department for providing a healthy environment for successfully completion of my thesis.

April 5, 2016

ABSTRACT

PARAMETRIC CYCLE ANALYSIS OF ADAPTIVE CYCLE ENGINE

Harinkumar Rajendrabhai Patel, MS

The University of Texas at Arlington, 2016

Supervising Professor: Donald R. Wilson

One of the main area of research currently in air-breathing propulsion is increasing the fuel efficiency of engines. Increasing fuel efficiency of an air-breathing engine will be advantageous for civil transport as well as military aircraft. This objective can be achieved in several ways. Present design models are developed based on their uses: commercial transport, high range rescue aircraft, military aircraft. One of the main priorities of military aircraft is possessing high thrust but increasing fuel efficiency will also be advantageous resulting in more time in combat. Today's engine design operates best at their design point and has reduced thrust and high fuel consumption values in off-design. The adaptive cycle engine concept was introduced to overcome this problem.

The adaptive cycle engine is a variable cycle engine concept equipped with an extra bypass (3rd bypass) stream. This engine varies the bypass ratio and the fan pressure ratio, the two main parameters affecting thrust and fuel consumption values of the engine. In cruise, more flow will flow through the third stream resulting in the high bypass engine giving lower fuel consumption. On the other hand, the engine will act as a low bypass turbofan engine producing more thrust by allowing more air to flow through core while in combat.

The simulation of this engine was carried out using the Numerical Propulsion System Simulation (NPSS) software. The effect of the bypass ratio and the fan pressure ratio along with variation in Mach number were studied. After the parametric variation study, the mixture configuration was also studied. Once the effects of the parameters were understood, the best design operating point configuration was selected and then the engine performance for off-design was calculated. Optimum values of bypass ratio and fan pressure ratio were also obtained for each altitude selected for off-design performance.

The data obtained from this study shows that the adaptive cycle engine is more advantageous over the other present engines in use because of its ability to vary parameters to get optimum thrust and fuel consumption values according to flight conditions.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

ADVENT = Adaptive Versatile Engine Technology.
VAATE = Versatile Affordable Advanced Turbine Engines.
USAF = United States Air Force
AEDT = Adaptive Engine Technology Development
NPSS = Numerical Propulsion System Simulation
SFC = Specific Fuel Consumption
DOD = Department Of Defense
CFD = Computational Fluid Dynamics
JANAF = Joint Army Navy Air Force
CEA = Chemical Equilibrium with Applications
NIST = National Institute of Standards and Technology
T_{t4}, TET = Turbine Entry Temperature
BPR = By-Pass Ratio
F_g = Gross Thrust
$\dot{m}_{exit}$ = Mass of exit air
V_{exit} = Exit Velocity
C_{noz} = Nozzle coefficient
P_{exit} = Exit Pressure
P_{amb} = Ambient Pressure
A_{exit} = Exit Area
Wfuel = Mass flow rate of Fuel
Wair = Mass flow rate of air

$\dot{m}$ = Mass flow rate

ρ = Density

A = Area

V = Velocity

P_t = Total Pressure

T_t = Total Temperature

γ = Specific heat ratio

R = Universal Gas Constant

M = Mach number

P_{t2} = Total pressure at compressor inlet

P_{t0} = Total pressure at inlet

η_i = Inlet efficiency.

CHAPTER 1

INTRODUCTION

1.1 Motivation

While contemporary aircraft have improved vastly over earlier generation of jets there are certain aspects which need to be improved to meet future challenges. A key area where continuous improvement is required is propulsion. Improving the efficiency of the air-breathing engine has tremendous advantages in both civil and military aviation. The engine of the future must allow aircraft to fly faster, higher and farther than before while consuming less fuel, contributing to savings in overall cost. From the propulsion point of view, obtaining these mission objectives is mutually exclusive. Higher thrust results in high amount of flow through the core, but will also require high temperature at turbine entry which in turn results in high fuel consumption. On the other hand, a decrease in fuel consumption as required in long range flight will result in low thrust value. The issue of combining both capabilities, low fuel consumption and high thrust, into one propulsion system has become the next challenge. How can one solve this problem? Can it be solved? How? These are a few of the questions that engine makers in today’s world are faced with. Researchers and engineers working under and with the U.S. Air Force Research Laboratory (AFRL) came up with a solution to this problem called the Adaptive Engine Cycle. The next step is to test this adaptive engine cycle concept for a new generation of engines that will have varying fan pressure ratio and bypass ratio according to the flight profile. This technology will not only meet the demand of the present world but will also introduce a new technology of adaptive cycle, for sixth-generation aircraft engines. Studying this technology and simulating its result is the basic aim of this study.

Adaptable engines use the concept of variable geometry where the fan pressure ratio and overall bypass ratio (main factors contributing towards fuel consumption and thrust) can be altered according to the condition and requirement of flight. To alter the bypass ratio, variable-cycle engines add a third airflow stream outside of both the standard bypass duct and core. The third stream provides an extra source of airflow that, depending on the phase of the mission, can be adapted to provide either additional bypass flow for increased propulsive efficiency and lower fuel burn or increased mass flow through the core to increase thrust. The second stream has a mechanism which controls the air flow rate through the core flow. These mechanisms can be used for allowing less air flow through core and hence giving characteristics of high by-pass engine or to provide additional core flow for higher thrust.

1.2 History

1.2.1 Turboprop

These engines were developed to overcome the problems faced by the piston engine. A turboprop engine is a gas turbine engine driving a propeller. In these types of engines, the propeller is located at the front of engine. The propeller converts the power generated by the engine into forward aircraft motion as effectively as possible and it does this under all operating conditions. The turboprop engine has a core shaft which is connected to a hub and a reducer gear to drive the propeller at lower RPM. Many low speed private jets or small range transport aircraft use a turboprop propulsion system.

The large amount of thrust produced by the turboprop engines is achieved by increasing the velocity of the large mass of air passing through the propeller. Propellers are one of the ways of producing thrust with low fuel consumption.

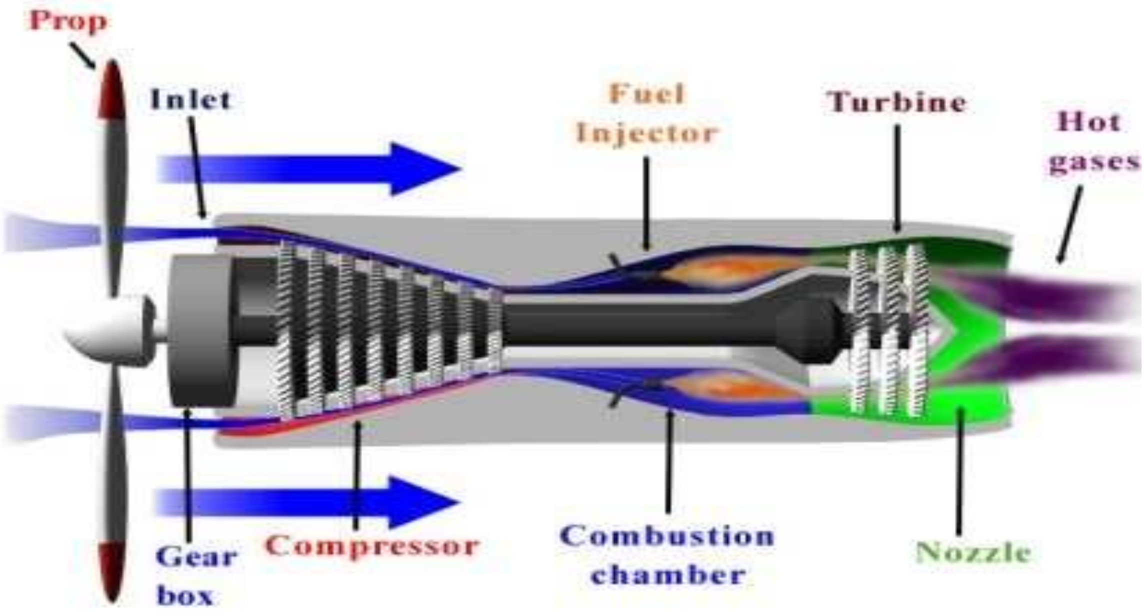


Figure 1-1 Turbo-prop Engine [18]

Turboprop engines have two main parts - the propeller and the core. The core is used to expand the small amount of air that passes through the combustion chamber and nozzle so as to produce jet thrust.

1.2.2 Turbojet

The turbojet engine is the simplest type of gas turbine engine. It consists of an inlet, compressor, combustion chamber, turbine and a propelling nozzle. Turbojet engines have good performance, higher speeds, and longer range when compared with turboprop engines, resulting in replacement of turboprops. Turbojets have become common in medium range cruise flights as they have high exhaust speed, small size and are simple in construction. Turbojet engines are generally limited for use in aircraft only, as the performance is mostly good for high-speed vehicles.

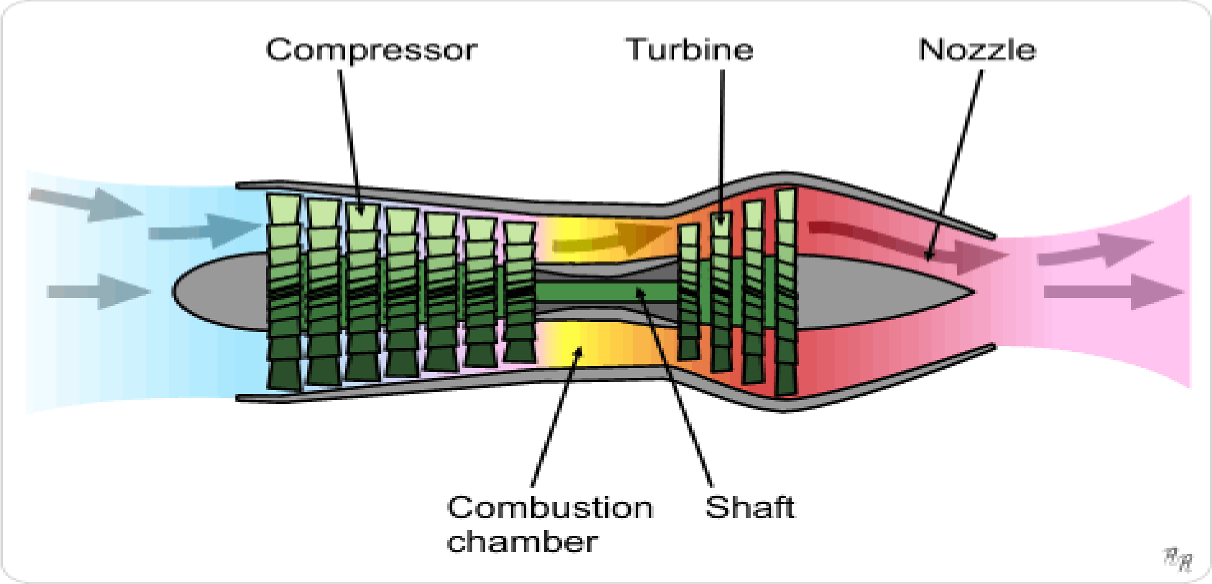


Figure 1-2 Turbojet[20]

In operation, the engine takes in a large amount of surrounding air through the inlet. The function of the inlet is to increase the pressure and reduce the velocity of the air according to the compressor requirements. The air after passing through the inlet enters the compressor. Compressor compresses the air thereby increasing the pressure and reducing the velocity significantly. This high-pressure air then enters the combustion chamber where the air is mixed with fuel and burnt. Burning results in raising the temperature of the air and slightly reducing its pressure. The air then passes through the turbine section. The turbine expands the air resulting a decrease in the temperature of the gas. The decrease in temperature in converted into shaft work which is used to rotate the compressor. The gases are further expanded in the nozzle section to produce high kinetic energy (velocity) to propel the airplane forward.

1.2.3 Turbofan

This engine configuration is similar to that of the turboprop, but a ducted fan is used instead of the propeller. This large fan at the front provides thrust in the same way as a propeller. The turbofan engine has a front fan which runs at the same speed as the low-pressure compressor and a fan turbine (low pressure turbine) located at the back to drive the fan. High-pressure compressor and high-pressure turbine are connected to high pressure shaft. Many of the large transport aircraft today use turbofan engines because they produce high thrust, have lower fuel consumption and create low engine-noise.

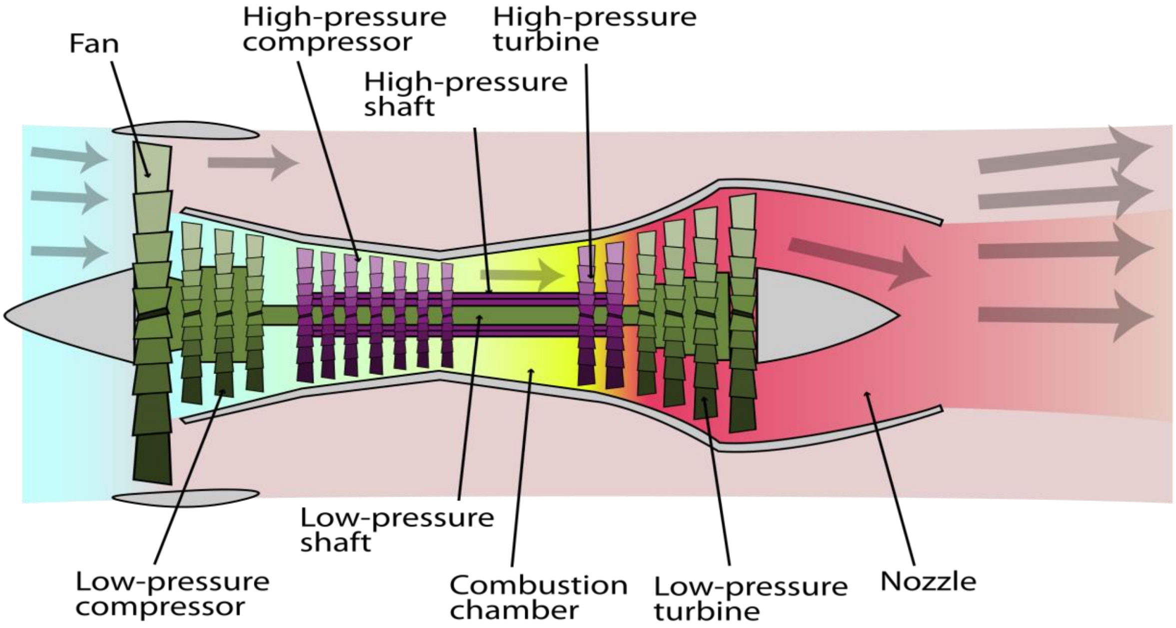


Figure 1-3 Turbofan[19]

Since the amount of air intake is larger, this implies that a larger quantity of air to be compressed and accelerated by the fan. Certain amount of air will entirely bypass the core of the engine (burner and turbine sections) and pass through the fan duct. These engines have improved propulsion efficiency which is achieved by accelerating a large quantity of flow at comparatively lower increase in velocity to produce thrust. Turbofan engine also has lower engine noise levels when compared to turbojet engine. They also requires reduced takeoff distance and yields higher thrust during steep climb than a turbojet of approximately the same size. The higher thrust produced by turbofan powered aircraft allows it to take off at a much larger gross weight.

1.3 Background and Development of Adaptive Cycle Engines

Adaptive Versatile Engine Technology (ADVENT) is one of the approaches towards development of improved engine cycles being carried out under the Air Force's Versatile Affordable Advanced Turbine Engines (VAATE) program. The ADVENT program was announced in mid-2007. In August of the same year General Electric (GE Aviation) and Rolls Royce were awarded Phase I contracts. Phase I of the program included the investigation of the new engine concept that was suggested, development and testing of some of the important components of the engine; and they were asked to begin with the preliminary designs of an engine.[2][3] For Phase II of the ADVENT program, Rolls-Royce was given the authorization to proceed with the development of the engine at the end of 2009. Phase II included contracts of component testing and integration of the concept advancements into physical models to demonstrate the technology studied.[5] " GE Aviation was also awarded funds to continue developing the engine with their technology demonstration core which was unaccepted as only one contractor had to be selected for the Phase II of ADVENT program".[2][5] At first the engine was been developed as a backup for F-35 but, USAF officials have denied such allegations.[5] With the threat from the GE/RR F136, Pratt & Whitney has funded their own adaptive fan variant of its F135. After their recent advancement in research they may qualify for the follow-on Adaptive Engine Technology Development (AETD) program under the US Air Force Research Laboratory.[5] GE Aviation in 2012 was selected to continue its ADVENT work into the AETD program.[5] Pratt & Whitney defeated Rolls Royce to combine with GE and continue the AETD program to develop fuel-efficient and high-thrust power-plants.[5][2] Currently GE has started working on the Phase III ADVENT program in which they will test the inlets, nozzles, engines and thermal technologies.

1.4 Objectives

The objectives of this study are;

- To model Adaptive Cycle Engine using Numerical Propulsion System Simulation (NPSS)[7].
- To evaluate the performance of Adaptive Cycle Engine by using NPSS.
- To analyze the performance of the engine in Design and Off-Design conditions.
- To obtain the best possible values of specific fuel consumption (SFC) (i.e., as low as possible) and thrust (i.e., as high as possible according to the requirement) for this design.

- Compare the results with engine used in the Lockheed Martin F22 (i.e. Pratt & Whitney F119).

Chapter 2

LITERATURE REVIEW

2.1 Adaptive Cycle Engine

This chapter briefly describes some of the prior works/studies conducted in the development of adaptive engines. One of the main focuses of the Department of Defense (DOD) is to develop a new engine which could travel longer distance without using as much fuel. GE and the Department of Defense have been heavily involved in the development of an advanced and complicated design based on an adaptive cycle engine, which will have enhanced performance characteristics and which can go longer distances without consuming as large amount of fuel. The adaptive cycle engine in operation optimizes engine performance according to different flight conditions by varying the two main propulsion parameters that are mostly responsible for amount thrust produced and specific fuel consumption - bypass ratio and fan pressure ratio. The adaptive cycle improved capabilities include increased thrust, improved thermal management, reduced fuel consumption, and good performance at low level flights. Other advantages also include high pressure compressor improvement, use of advanced materials, and incorporation of an effective cooling system to obtain higher temperatures entering the turbine which will in turn result in increased thrust.[2] Reduced fuel consumption will reduce dependency on refueling tankers for long range flights.[2] “The engine will have a three-stream adaptive airflow control that results in a 25 percent enhancement in fuel efficiency, an increase of more than 30 percent in aircraft operating range and more than a 10 percent enhancement in thrust compared to today’s most advanced military combat engines”.[2] This project was granted to General Electric (GE) who is working with the United States Air Force (USAF) and the Department of Defense (DOD) in the pursuit of advanced technologies and design concepts that will maintain the US’s technological edge well into the future[3]. Adaptive cycle engine programs will create a significant value to the country’s defense, as it can increase the combat capability of war-fighters. This program will also help in reducing dependency on non-renewable energy through significant reductions in fuel consumption [6]. The adaptive cycle engine is an excellent platform to learn and conduct research towards developing new technologies and to develop advanced manufacturing processes.

CHAPTER 3

SOFTWARE – AN OVERVIEW

3.1 Introduction

The Numerical Propulsion System Simulation (NPSS) code, which is designed by the National Aeronautics and Space Administration (NASA) Glenn Research Center in support with the U.S. aero-propulsion industry and the Department of Defense, is used to analyze the adaptive engine in this project work. [7] The physical interactions within an engine can be genuinely modelled by NPSS which in turn speeds up the concept-to-production development time and also decrease the requirement of costly full-scale tests and experiments. It can also be used to simulate 1D, 2D and 3D models.

NPSS was selected for this analysis as it is specially created for gas turbine engine analysis and is also well suitably applicable to any other complex thermodynamic systems. Moreover, NPSS is presently one of the modelling tools used by engine manufacturing companies. Using NPSS, it is relatively easy to congregate an extensive range of components into various different system arrangements. The NPSS package contains a collection of standard components of gas turbine engines like compressors, combustors, turbines, nozzle, etc. NPSS is written in C++ and is an object-oriented tool which can perform cycle design, off-design, steady state and transient performance prediction, data matching and other tasks of engine cycle simulation which makes it an easy tool for conceptual design and performance estimation of gas turbine engine. The user can define new components in any level of detailing as desired which can be as complex as connecting to external CFD simulations and as simple as table look up. [8] The information (pressure, enthalpy, composition, temperature, etc.) between linked components is stored and passed on in NPSS via a port data structure. To find the flow state based on some combination of temperature, pressure, enthalpy and entropy NPSS will execute functions with several built-in thermodynamics packages. The modelling demonstrated in this project uses the simple gas property code developed by Pratt and Whitney, which is a thermodynamics package capable of executing chemical equilibrium calculations every time these built-in functions are called. Other thermodynamics package options comprise a restricted chemical equilibrium calculation (‘JANAF’ package) and Chemical Equilibrium with Applications (CEA).

3.2 Advantages of NPSS

- It can be used for transient mode analysis.
- It has thermodynamic property packages, which enables a wide range of cycle analysis calculations.
- Object oriented simulation which makes it easy to run.
- Easy to create different and versatile components according to the engine requirements.

3.3 Methodology

The NPSS model can be defined in one or more input files. In NPSS, engine components such as inlets, compressors, combustors, turbines and nozzles are represented as element objects. These elements can be combined into assemblies and each assembly can have one solver. Various engine components modelled as elements can be linked to each other through ports and have individual linkages defined. These linkages are used to display outputs at each station.[7] To support air-breathing engine analysis (aircraft and industrial gas turbine engines), various thermodynamic gas property packages are provided with NPSS. With the help of pre-defined commands, the operator can choose the desired package at the run time. One of the packages named "JANAF" provides flexibility and equals the NIST standards (NIST−JANAF, Revision 3) at the cost of some computational speed. A second package named "CEA" (Chemical Equilibrium with Applications) thermodynamic package is an implementation of the NASA chemical equilibrium code. The "GasTbl" by Pratt & Whitney is based on NASA's "Therm", includes chemical equilibrium proficiencies along with humidity calculations. "allFuel", from General Electric, contains properties of both gasses and fuels. A “FPT”, fluid property table package is accessible as well which allows operators to designate the thermodynamic properties of the fluid in the form of NPSS tables and/or functions. [7]

The user must set design conditions after the engine is modelled. Design conditions are the reference conditions at some altitude and are defined as properties at the design point. This design point is used to size the engine - each component of the engine is sized after running the model several times and comparing the results with similar results previously obtained from practical tests. The performance can be calculated for any operating and flight conditions defined in advance after the engine is sized.

While designing a gas turbine engine one must consider its different variable parameters. A jet engine design has three design parameters - high and low compressor pressure ratios and turbine entry temperature (T_{t4}). A turbofan engine has five design parameters - high and low compressor pressure ratios, fan pressure ratio, turbine entry temperature (T_{t4}) and bypass ratio (BPR) – while an adaptive engine has seven, with two fan pressure ratios and bypass ratios. These parameters decide the engine performance and hence, these are varied to obtain the desired results of specific thrust and fuel consumption. The analysis carried out is an ideal cycle analysis unless losses are included. Some of the losses are included by defining duct pressure losses, nozzle velocity coefficients[7], compressor and turbine efficiencies, shaft mechanical efficiencies, etc. Mattingly[9] lists approximate values of the –loss parameters for 1st , 2nd, 3rd and 4th generation aircraft engines which can be used as estimates in the absence of detailed component designs. It is often the case that if one loss parameter increases, the others also follow; depending on the requirement, some compromises have to be made to obtain optimum results. For this reason, cycle design parameters are varied over a range of values within material design limits to yield the optimum propulsion system from a set of many engine designs for specific aircraft application[10].

After the design point values are obtained and the engine is sized, off-design parameters are evaluated. In case of off-design flight conditions the altitude, Mach number and throttle setting are not the same as the design conditions, and are specified to obtain performance of the engine for various flight envelopes which the aircraft will undergo during its entire flight. Hence, this is the reason off-design analysis is also called performance analysis. Off-design analysis is carried out for different points and the results are checked with the existing results, if any. In the case of compressors and turbines, pre-built graphs are accessible that use scaling laws and matching to deliver turbo-machinery performance at off-design. All the performance calculation in NPSS are done in solver. The solver is an important part of NPSS that propels the model to a valid solution. NPSS automatically creates a solver for the user, one solver is always created for the top-level assembly. Once the user provides the run command it is responsibility of the solver to iteratively adjusting the values of independent variables so that the dependent conditions of the system are satisfied. User specifies the number of iterations to be carried out. If the solution does not converge within a specified number of iterations, an error message is returned for that case. Transient mode analysis can also be performed using NPSS. For transient simulations, the solver is also responsible for controlling the increment in time within the run, providing a converged solution at each point in time[7]. The analysis carried until now was steady state and time was not considered. Transient simulations, in which the time-varying behavior of a system is sought over some time interval, are performed by finding a series of solutions at discrete time steps spanning the interval desired [7]. The solution at each time step is similar to a steady-state solution: it is represented by a system of equations that can be solved iteratively. The transient problem is unique, however, in that some of the equations to be solved represent integrations. Suppose, the transient response of a gas turbine engine to a ramp in fuel flow needs to be found, the change in high pressure spool is calculated and then the results are evaluated. Time step is the most important parameter of the analysis as transient mode uses explicit method to calculate parameters.

3.4 Solution Methods

To model a complex thermodynamic system, the user must first understand how the system works, what the requirements are and how to achieve them. Once the user understands the system he or she must specify all the dependent conditions for independent variables in order to solve for stable operating point.[8] There are different methods for approaching the required solution. The aim of this section is not to give a review of possible methods of solution but rather to prescribe some of those most related to that of the work done: finding best possible properties of adaptive engines. Calculating solutions for complicated models such as gas turbine is not very easy. Therefore, iterative methods are required which starts with an initial guess and better approximations are calculated for every iteration until a solution is found[7]. Methods such as finite differenced Newton's method are preferred for systems with less number of equations and low function computation cost. In Newton's methods, the partial derivative matrix (Jacobian) is calculated and updated for each iteration via finite differencing. These methods are adopted because they can perform relatively large number of calculations per iteration. The calculations used in this study are small (several equations) so the approaches which are specialized for small system calculations are taken into consideration.

The user has to define a set of dependent and independent variables for the NPSS solver to run. The solver varies and controls the independent variables given by the user. In the system to be analyzed any parameter can be defined as an independent variable. In operation independent variables are varied for each iteration using the Jacobian matrix until the solution converges to meet all the dependent conditions. The solver has the capability to iterate and satisfy all the independent conditions together, so the user can define several independents and dependents to obtain the required results. One example of a dependent-independent relationship is changing the air flow rate to match the required thrust value or vice versa. The number of dependents and independents must be same.

Many elements in NPSS have predefined dependents and/or independents. If the user chooses to use the "default" solver setup, then the solver automatically uses the predefined dependents/independent to solve the cycle. Users can also define their own dependents and independents based on the requirement of the analysis instead of using the defaults provided. The convergence of the solution of the system is described by quasi-Newton’s method discussed previously. The complicated set of inter-relationships between the dependents and independents are described by independent variables that are used to model the Jacobian matrix of partial derivative. Corrections required in independent variables for residuals to become zero are approximated by solution algorithm depending on error in conditions for the dependent and Jacobian. After the first iteration is completed, a new Jacobian matrix is only formed when the specified convergence criteria are not met[7].

Solvers:

A set of equations can be considered to consist of a set of N independent variables, the values of which can be set however desired, and a set of N dependent conditions, the state of which is completely determined by the values of the independent variables. For example, consider the following single equation:

$$x^4 - 10 = x^1 - 5 \qquad \ldots\ldots (3.1)$$

In this case, there is one independent variable named *x*. For any given value of *x*, the value of the equation's left-hand side is determined (x^4 – 10), as is the value of the equation's right-hand side (x - 5). The dependent condition is that the left-hand side equals the right-hand side. It is customary to express this condition as an error term whose value is zero when the condition is satisfied. For example, if the equation left-hand side is denoted by eq_lhs and the right-hand side by eq_rhs, an error term could be defined as follows:

$$\text{error} = \text{eq\_lhs} - \text{eq\_rhs} \qquad \ldots\ldots (3.2)$$

Some tolerance would be specified, meaning that the equation is considered satisfactorily solved when the error term is less than the tolerance value. In the example above, the tolerance specified would be called an absolute tolerance since it is compared to the absolute difference between eq_lhs and eq_rhs. The value of absolute tolerances can be as low as 10^{-5}.

Users can also create the solver using their own commands. The top-level assembly has a Solver object named solver created automatically for the user. “Each Solver object in the model is distinct, and can have its own set of attribute values that differ from those in other Solver objects” [7]. A solver object must exist before Independent and Dependent objects can be added to it.

CHAPTER 4

DESIGN AND MODELLING

4.1 Modelling in NPSS

In this section, the basic building blocks of a typical engine simulation model will be explained. Each part of the model will be discussed in detail in the sections below. “The model is read as a single continuous input stream and may be constructed in a single file” [7]. This file will describe the whole model, i.e. the inputs and how they are linked, the tables and maps that are required for simulation, the output, and the description of the conditions at which the model is to be run.

The model file is lengthy and can get complicated to understand. Hence, the model file can be divided into smaller separate files. These files can be called into main model file whenever needed by using the #include directive. This makes the model more easily readable, editable and understandable, hence allowing easy data transfer from one department to another.

The model file types may be identified by their file extensions:

File description	**Extension**	**Example**
Study executive	*.run	fanfan.run
Model description	*.mdl	fanfan.mdl
Case definition	*.case	desOD.case
Functions	*.fnc	bleed_macros.fnc
Interpreted components	*.int	EngPerf.int, controls.int
Component maps	*.map	fanE3.map, hptE3.map
Tables	*.tbl	pressLoss.tbl
Output viewers	*.view	ncp.view, Transient.view

Table 4.1- Recommended file extensions [7]

The basic structure of NPSS code is to define the required element components that can be connected to each other to create models of any desired and complicated systems. The simplicity with which these components can be linked makes NPSS comparably easy and fast environment for creating different system models and examining several systems and configurations[8]. The flow flowing into the component and out of a component in NPSS is described by ‘Flow stations’: ‘Fluid Ports’ and ‘Bleed Ports’. All the important information of the fluid stream like pressure, flow rate, ratio of specific heat, temperature, Mach number, enthalpy, velocity, flow area etc. are stored in flow stations. The information about extracting or transferring shaft power to the components is controlled by Shaft Ports.

All the Fluid Input Ports have to be linked to a Fluid Output Port, which can be easily done by using the ‘linkPorts’ command in NPSS, hence creating a line/connection between the components. Shaft and Fuel Ports are also linked similarly. The figure 4.1 below shows a simple model in NPSS (a simplified turbojet) and its relation with the system solver. For this example, the link between Fluid Ports is shown. The Inlet Start element is linked to Inlet element which is further linked to Compressor to Burner to Turbine to Nozzle to flow end components[7]. Compressor and turbine components are also connected by Shaft and Bleed Ports.

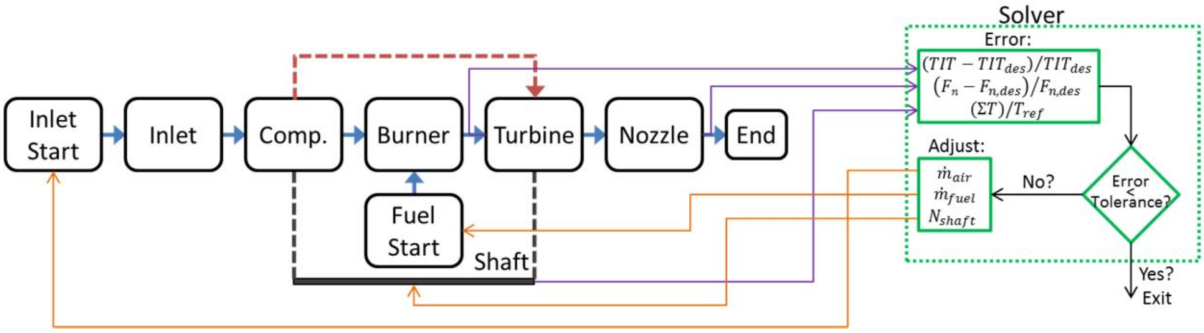


Figure 4-1 NPSS Model of Simple Turbojet and Solver[8]

Figure 4.2 below shows that a component in NPSS can have n number of ‘Fluid Input Ports’, ‘Bleed Input Ports’, ‘Fluid Output Ports’, ‘Bleed Output Ports’, ‘Shaft Ports’, and ‘Fuel Ports’.

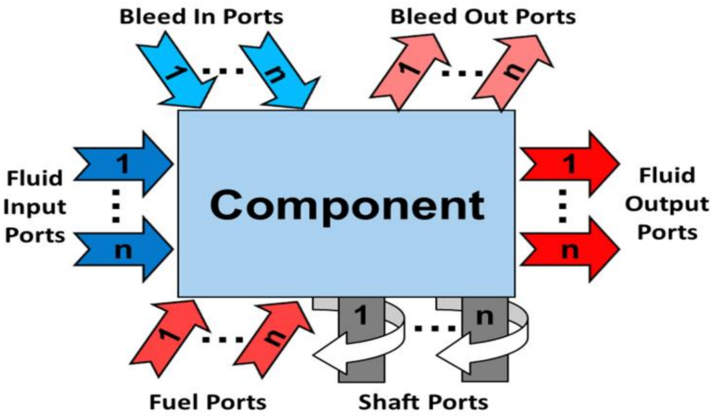


Figure 4-2 Generic Element Diagram[8]

4.2 Component Modelling in NPSS

The main components of an adaptive engine will be shown and discussed in the following sections. To provide an overview, the element types used are listed below.

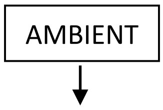
Element Name(s)	**Element**	**Type Description**
Amb	Ambient	Sets ambient conditions, establishes a source of airflow into engine (does not have an outlet port and requires the InletStart).
FsEng	InletStart	Sets the flow that enters a following Inlet.
InEng	Inlet	Sets conditions at engine inlet, after ram recovery.
SpltFan1, SpltFan2	Splitter	Divides flow between the bypass stream and core stream.

CmpFSec1,CmpFSec2, CmpL, CmpH	Compressor	Compresses flow.
FusEng	FuelStart	Establishes a source of fuel for the burner.
BrnPri	Burner	Burns fuel/air mixture.
TrbH, TrbL	Turbine	Extracts work from flow.
FMixerC, Fmixer	Mixer	Mixes two different flows into one.
D130, D230	Duct	Defines pressure loss between components.
NozPri	Nozzle	Delivers flow to the atmosphere at engine exit.
FePri	FlowEnd	Establishes a sink for a fluid stream.
ShH, ShL	Shaft	Transfers torque between rotating components.
Perf	EngPerf	Calculates overall performance parameters.

Table 4.2- Adaptive engine elements

4.2.1 Ambient Element

The Ambient element is used to define flight condition attributes such as altitude, ambient temperature, ambient pressure, Mach number, air speed, etc. as user defined inputs which are then fed to an Inlet Start element (explained in inlets). All fluid streams must start with an Ambient. A single ambient element may provide freestream conditions for several Inlet streams. It initiates ambient conditions but has no output fluid ports. The element takes various combinations of inputs to calculate the flight condition.



socket: S_customDay
socketType: TDAYCUSTOM
returns: TsDay

Figure 4-3 Ambient Element [11]

4.2.2 Inlet Start

The inlet start element, based on the fight condition and data provided by the ambient element, will set the initial conditions for the inlet. The balance on the inlet actual or corrected flow that passes through inlet is handled by Inlet Start component.[11]. Inlet Start element does not have any independent variables or dependent conditions.

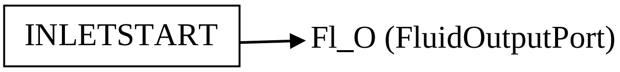


Figure 4-4 Inlet Start Element

4.2.3 Inlet

The Inlet element calculates the change in properties as the flow passes through it. The ram recovery factor has to be specified by user as an input. The user can provide an input value or an input calculation function. The division of total pressure in the free-stream by the 'recovered' pressure in the inlet is defined as the ram pressure recovery (π_{inlet}.). Even ram drag is calculated by the inlet element.[8] Inlet element does not have any dependent conditions or independent variables. This element does not give any warning or provisional errors.[11] The inlet component has single inlet and output flow ports.

Figure 4-5 Inlet Element

4.2.4 Splitter

The splitter element takes one entering stream and splits it into two streams, with a pressure loss applied to each flow separately. The splitter element can only be used when the flow is flowing in one direction at all times and there is no recirculation (open cycle only)[11]. The By-Pass Ratio (BPR) is assigned as a design condition or as an initial guess for the first case run and it also specifies the ratio of secondary to primary flow.

Temperature and pressure values are assumed to remain constant through the element[11]. As the temperature, composition and pressure remain constant through the splitter, this implies that energy is preserved in the splitter element.

Figure 4-6 Splitter Element

4.2.5 Compressor

Pressure ratio (PR) and efficiency (eff) are parameters used for determination of compressor performance. The Compressor element raises the pressure of air flowing through it based on the pressure ratio required and efficiency of the compressor values provided by the user. Compressor map sub element usually contains solver independents and dependents, which plugs into the socket. The solver is often used to drive the scaled map corrected weight flow (which includes audit factors, if they are used) so that the actual corrected compressor inlet flow is reached[11]. The shaft output port is used to share information between the compressor and the shaft element to which it is connected. The Compressor element CmpFSec models the tip portion of the fan and receives the secondary flow from the splitter. Compressors CmpL and CmpH are modelled similarly[11].

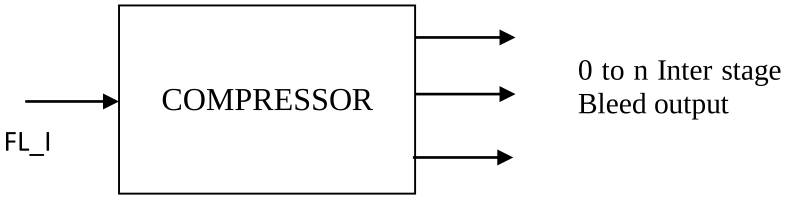


Figure 4-1 Compressor Element

Figure 4-8 shows an example compressor map where the black lines show the contours of constant corrected speed and the contours of constant efficiency are shown by colored lines. In order to obtain the mass flow and speed of the shaft values from the performance map, the user must uniquely define the steady-state operating condition, obtained by defining operating line parameters. This operating point parameter defines the positions of an operating line on the compressor map[8] .

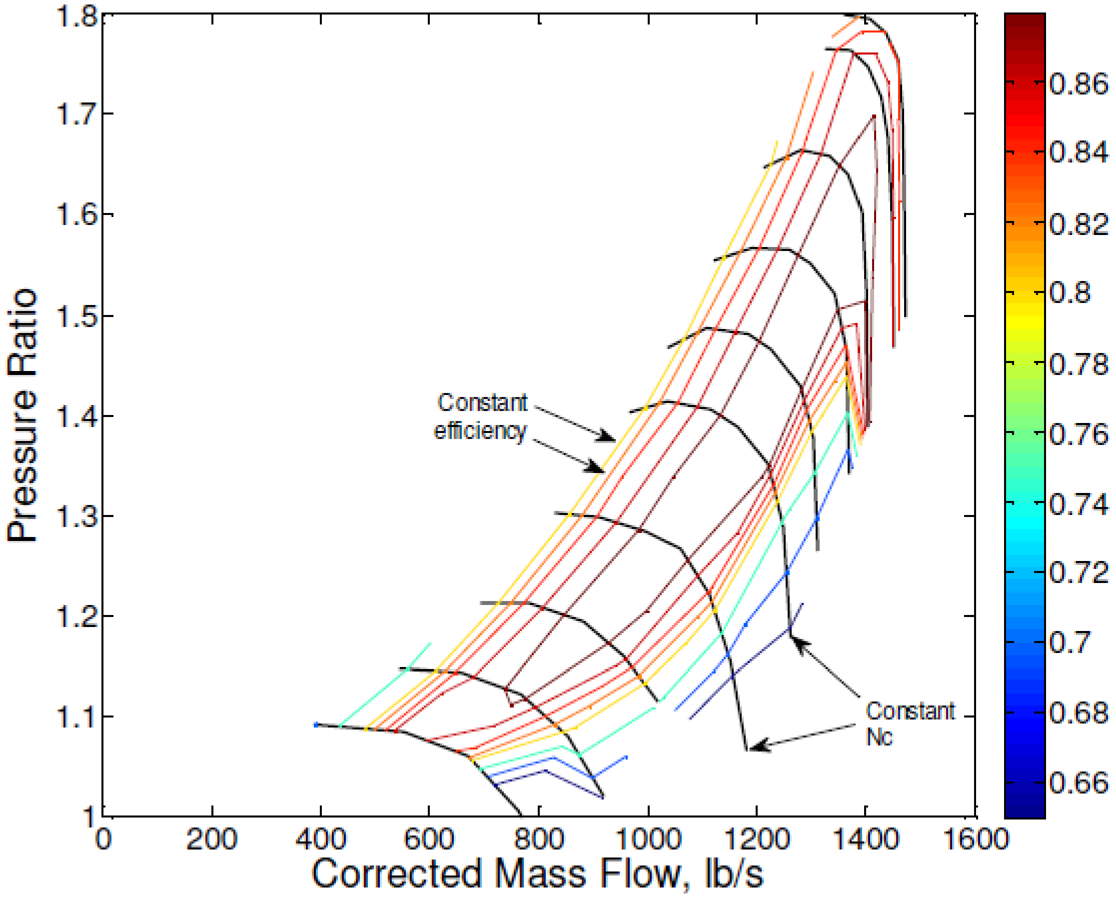


Figure 4-2 Compressor Performance Map [8]

The process of matching the values of efficiencies to get the desired performance values is illustrated in Figure 4.9. The design point on the map is denoted by the purple dot. For example, suppose the pressure ratio, mass flow and efficiency on performance map at the design point are 1.7, 1400 lb/s and 87% respectively. The scaling of map will take place to obtain the user defined design point which has a pressure ratio of 1.2 and efficiency of 85%. The map will be scaled in such a way that the design corrected mass flow value provided by the user will be targeted[8]. For example, the developer gives a design corrected mass flow rate value of 1100 lb/s, the map is scaled accordingly (shown in the fig:4.9 right side plot). Linear scaling of all three axis is performed. This allows for a single performance map to have performance for a wide range of conditions[8]. For the off-design mode, the map obtained in design mode analysis is fixed and scaling of the compressor away from the design point is done.

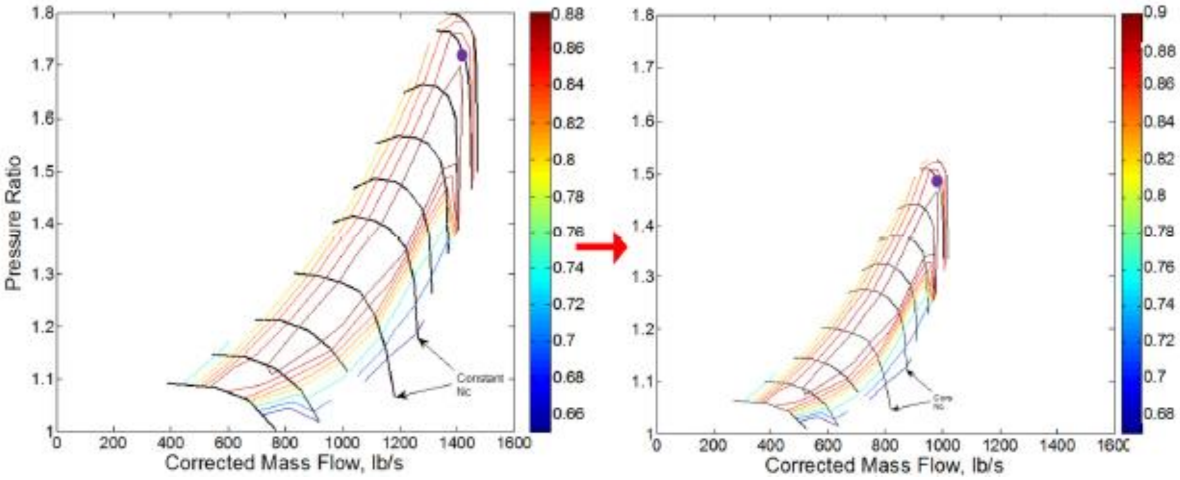


Figure 4-3 Compressor Performance Scaling Map [8]

4.2.6 Burner

The burner element combines air and fuel, and calculates the result of burning the mixture.[11] The burner will receive information such as temperature and pressure from the compressor element, and fuel properties from the fuel start element. The burner element is used to determine the fuel flow, fuel/air ratio, or exit temperature of the flow. In process, the value of the fuel flow rate is guessed and provided as an input; this value will change according to the required conditions. Before the combustion calculations are performed pressure drop is applied to the burner element. The thermodynamic package will define the nature of the combustion calculations selected for simulation process[11]. In this work, the GasTbl thermodynamics package is used to calculate the properties.

Figure 4-40 Burner Element

4.2.7 Turbine

The turbine element is used to expand the incoming flow and extract work from the gases based on the pressure ratio and efficiency provided by the user. This information is then passed to the nozzle and shaft elements. The values of pressure ratio and turbine efficiency can either be taken from the defined turbine performance map or can be assigned directly by the user. The turbine element operates in the same way as that of the compressor. For Design mode calculations, NPSS software scales linearly on the performance map in order to obtain the user-defined design point values, turbine corrected shaft speed and turbine corrected mass flow. This process of scaling is similar to the one studied for compressor element.

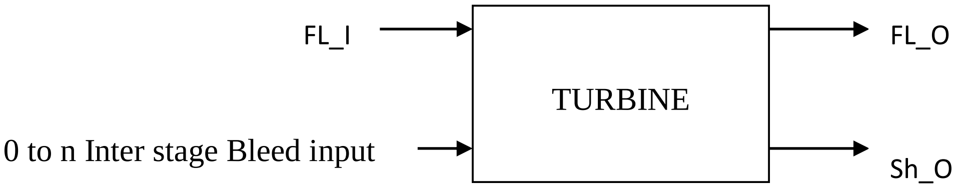


Figure 4-5 Turbine Element

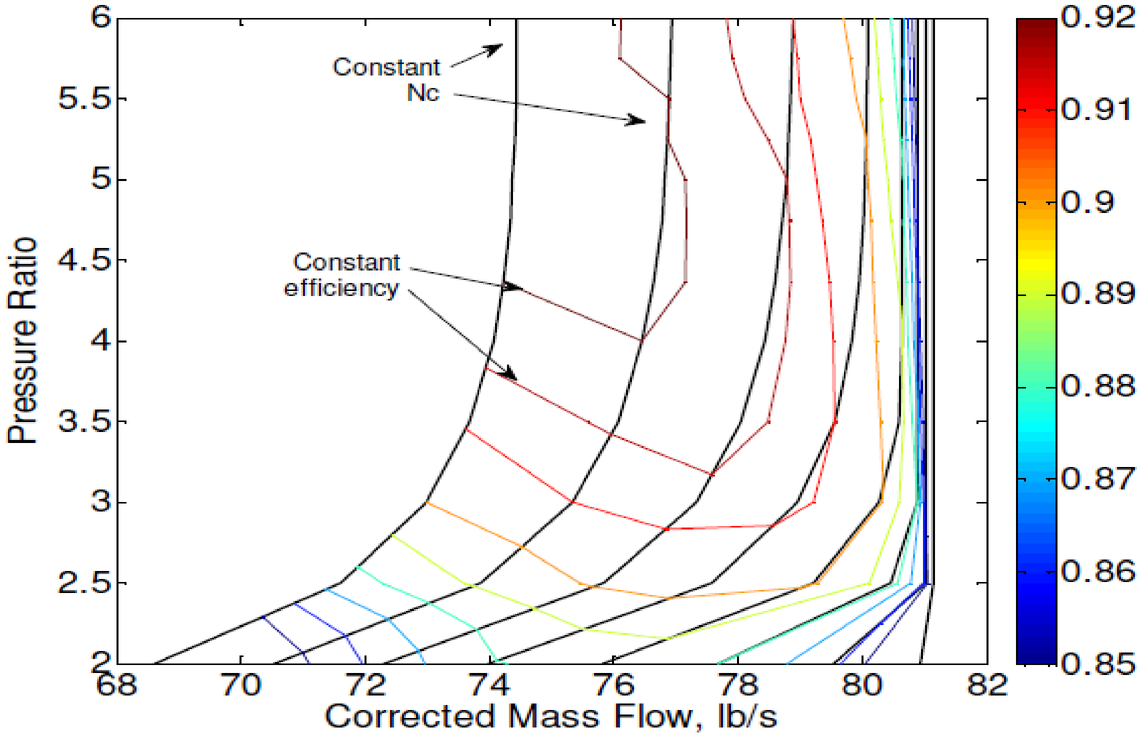


Figure 4-6 Turbine Performance Map [8]

4.2.8 Mixer

The mixer element combines two flows and gives one single flow at the end. The energy, continuity, and momentum of the flow are conserved when two streams are mixed into a single stream.[11] For the design point calculations, the flow entrance Mach number of the primary flow selected has to be specified by the user. This Mach number is used to calculate the primary entrance area of the flow. The area of the secondary entrance is calculated by varying the entrance Mach number until the static pressure of the two streams becomes equal. The area at the exit is determined by combining the two entrance areas together (constant area mixer).

Figure 4-7 Mixer Element

The mixer element has an inbuilt socket which calculates the correct value of thrust due to partial mixing. There is also another socket which calculates an adder term that is applied to the incoming impulse terms (both momentum and pressure force)[11]. The number obtained can be used to compensate for the losses occurring in the mixer. Losses in velocity can also be given as input conditions.

4.2.9 Nozzle

The Nozzle element performs calculations for convergent and/or convergent-divergent nozzles with variable or fixed exit areas. In this model, the flows from two fan ducts (bypass air) and turbine (main stream air) are combined using mixer and are fed to the nozzle. By default, the nozzle element calculates an ideally expanded converging-diverging nozzle. The ambient condition pressure value is obtained from the ambient element. Condition of choked throat is controlled by setting the Mach number at the flow station to 1.0 and using the NPSS function for determining overall state of the flow.[8] The choked condition of the flow is known when the calculated pressure at the throat is less than the ambient pressure. If the choked condition is not reached then the flow exiting the converging-diverging nozzle will be subsonic. In Design mode, the station pressure is given input by the user which, along with the NPSS function that iterates the exit Mach number to meet the pressure value, is used for obtaining flow conditions at the exit and overall flow state. The Mach number and pressure conditions are also used for the determination of exit area. In off- design mode, nozzle geometry is kept fixed and the values of exit Mach number and pressure are calculated by NPSS[11]. In either case (design or off-design), the gross thrust obtained can be calculated as:

$$F_g = \dot{m}_{exit} V_{exit} C_{noz} + (P_{exit} - P_{amb})A_{exit} \qquad \ldots\ldots (3.2)$$

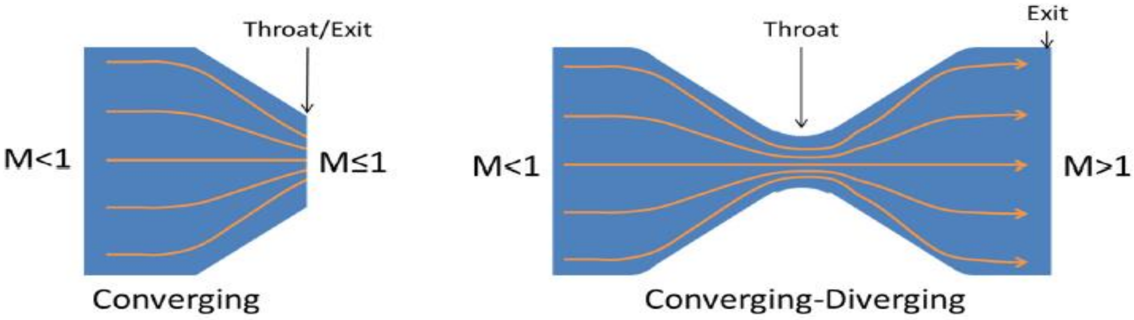


Figure 4-8 Converging and Converging-Diverging Nozzle [8]

For a variable area nozzle, the pressure at the exit of the nozzle and the ambient pressure are matched explicitly, so that the second term in thrust equation becomes zero, giving maximum possible thrust. The area at the throat is calculated for the mass flow at inlet of the nozzle element. The system level solver adjusts mass flow rate into the component in subsequent iterations of the entire model. Variation in mass flow rate can be achieved by changing the amount of air flowing into the entire engine or by changing the bypass ratio of the engine to adjust the air flow rate into the core of the engine. In case of the thrust vectoring nozzle (as in this case), a convergent or converging-diverging nozzle (used for analysis) can be used as per the requirement[8].

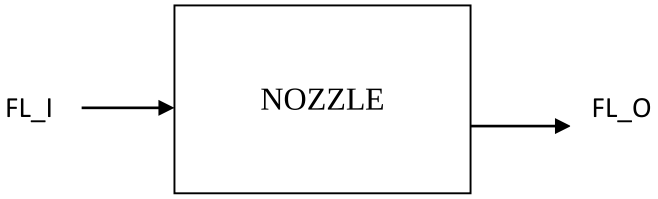


Figure 4-9 Nozzle Element

4.2.10 Duct

In NPSS, the pressure drop in an engine can be calculated by using the duct element. The percent drop in pressure can be given as an input value or can be calculated from a sub-element. Similarly, heat loss values also can either be given as an input or sub-element can be used to calculate its value.[11] The duct element does not have any independent variables or dependent conditions.

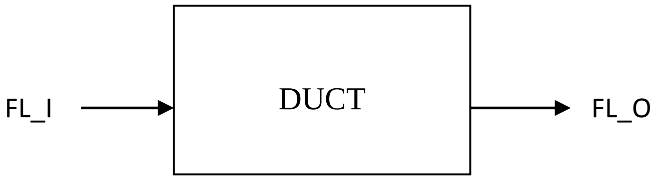


Figure 4-10 Duct Element

4.2.11 Shaft

The Shaft element provides basic mechanical connections between rotating elements such as turbines, compressors and fan. The power balance between all the components in a system that are connected to the shaft is given by the shaft element. The shaft element can have any number of mechanical ports attached to it.

Figure 4-11 Shaft Element

4.3 Modelling of Gas Turbine Engines

All of the gas turbine engine models described in this work are ‘rubber engines’. A rubber engine means that the engines do not have any limit to their size or power levels and these properties depends on the study being performed. This helps in studying a broad range of conditions, however it is limited by the fact that the modelled engines will not always directly correspond to any currently available engine. The alternative approach to rubber engine is to perform a ‘fixed engine’ analysis. In fixed engine analysis, a specific engine which has the same features as that of the required engine, say similar size and power level is assumed for all studies. This approach has the advantage of comparing the result to an existing engine, but will have less flexibility. For this study, the ‘rubber engine’ approach will be used as it allows more flexibility in design.

4.3.1 Turbojet Engine

A turbojet model in NPSS is shown in figure 4.18. Elements are represented using boxes. Connection between elements and shaft is represented by dash lines. Flow direction is given by direction of the arrows.

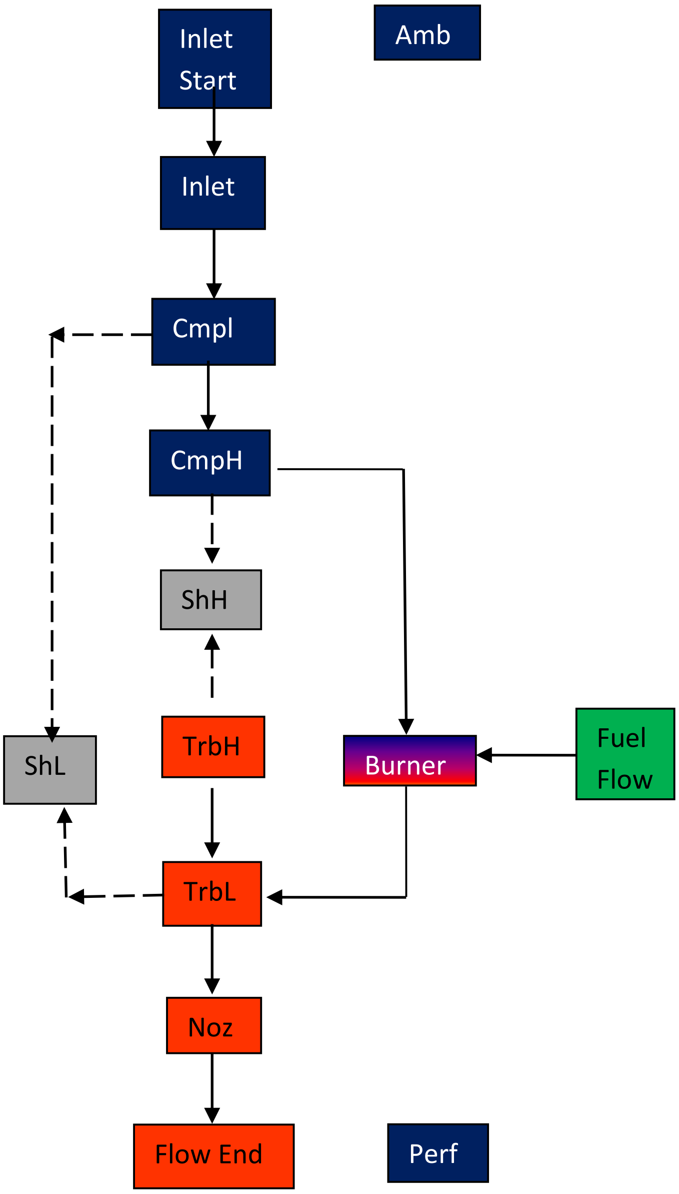


Figure 4-128 NPSS Schematic Model of Turbojet Engine

Inlet start is the first element through which flow initiation takes place. The function of inlet start is to collect data from the ambient element to set the inlet condition parameters such as pressure, Mach number, velocity and temperature. The total flow properties obtained by the inlet element are determined after imposing an inlet pressure recovery loss. In the compressor element the pressure and temperature rise is calculated using a compressor performance map (this procedure has been previously described) which relates corrected mass flow and shaft speed to pressure ratio and isentropic efficiency. The compressor and shaft elements exchange information of shaft speed and torque. Initiation of fuel flow for ignition process is done by the fuel start element. Fuel is spread in the burner element which calculates the amount of heat released when fuel/air reacts with each other. The expansion process resulting in temperature and pressure drop is calculated by turbine element on the bases of its performance map as described. Here, speed of shaft and torque information are exchanged between the elements, i.e. the turbine and shaft elements. Based on the nozzle used converging/converging-diverging the nozzle element is used to calculate the acceleration of the flow. From the values obtained, performance of the engine such as specific thrust, specific fuel consumption, etc. can be calculated.

4.3.2 Turbofan

A combined exhaust and dual spool (i.e., two shafts) turbofan engine is a more advanced engine than the turbojet. This design may be more complex than the turbojet engine, but has advantage over it as it uses multiple shafts which rotate at different speeds, in turn allowing turbo-machinery to be rotated at different speeds depending on the shaft they are mounted on. The low pressure compressor and high pressure compressors are connected to low pressure turbine and high pressure turbine through low speed and high speed shafts respectively. The NPSS model of the turbofan is shown in the figure 4-19 below. NPSS elements are represented by boxes, arrows represent direction of the flow and flow port connections, and shaft port connections are represented by dashed arrows. All the elements operate similarly as they did in the turbojet model described earlier. The blue color box represent cold flow and orange color represents hot flow i.e. flow with high temperature.

High and low pressure compressors are compressor elements with different compressor maps and high and low pressure turbines are turbine elements with different turbine maps. The Inlet flow will pass through the splitter element and is divided into core and bypass flows with same flow properties. The amount of mass flowing through each stream is determined by the bypass ratio. The core flow then passes through the low pressure compressor connected to the low speed shaft and then through the high pressure compressor connected to the high speed shaft. It then passes through the Burner, followed by a high pressure turbine which is also connected to the high speed shaft, and then through a low pressure turbine connected to the low speed shaft.

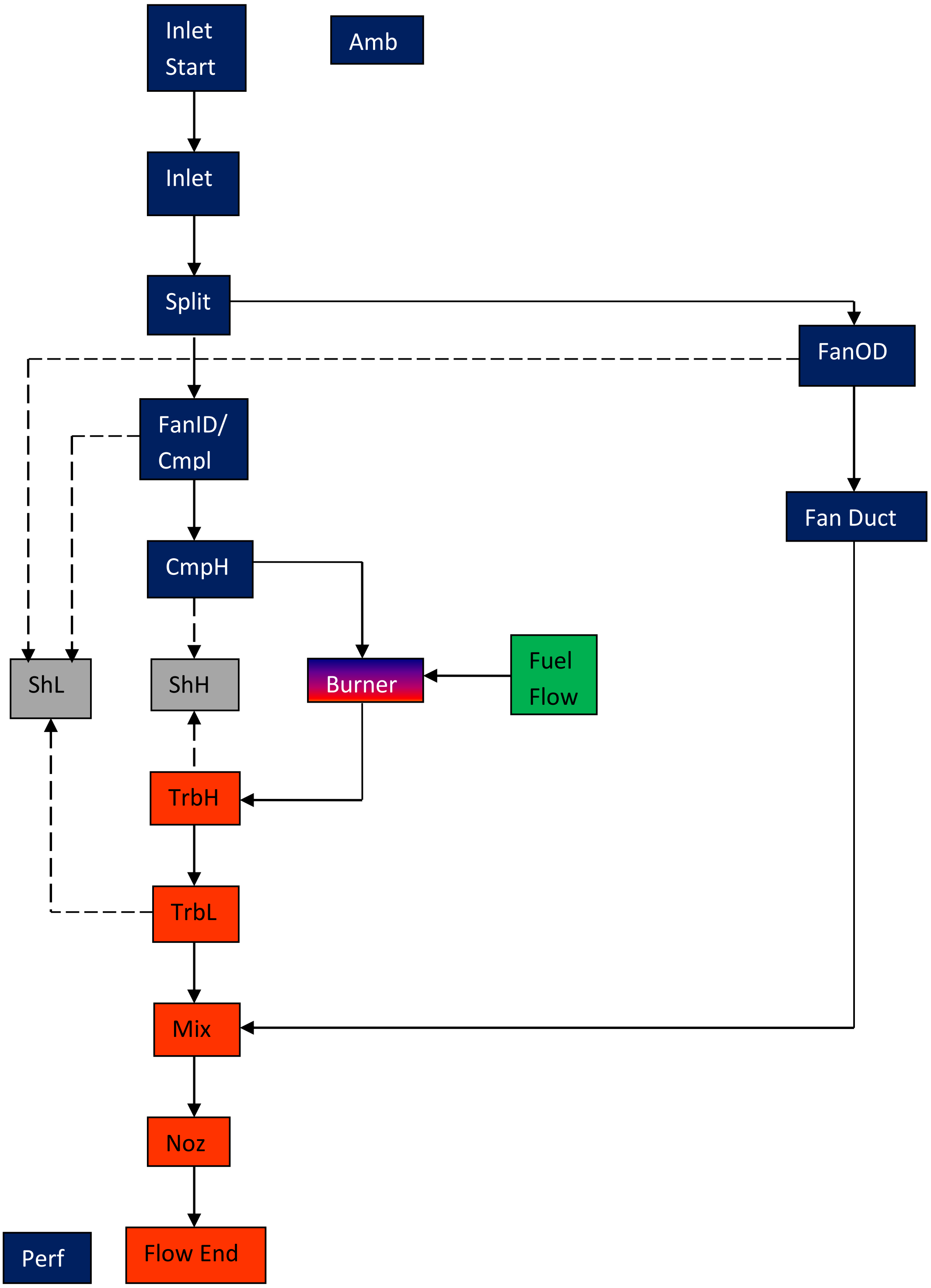


Figure 4-19 NPSS Model Schematic of Low BPR Turbofan.

The mixer element is used to combine the bypass flow stream and core flow after the core flow passes through the turbine. The combined flow will then pass through the nozzle and then the flow end element.

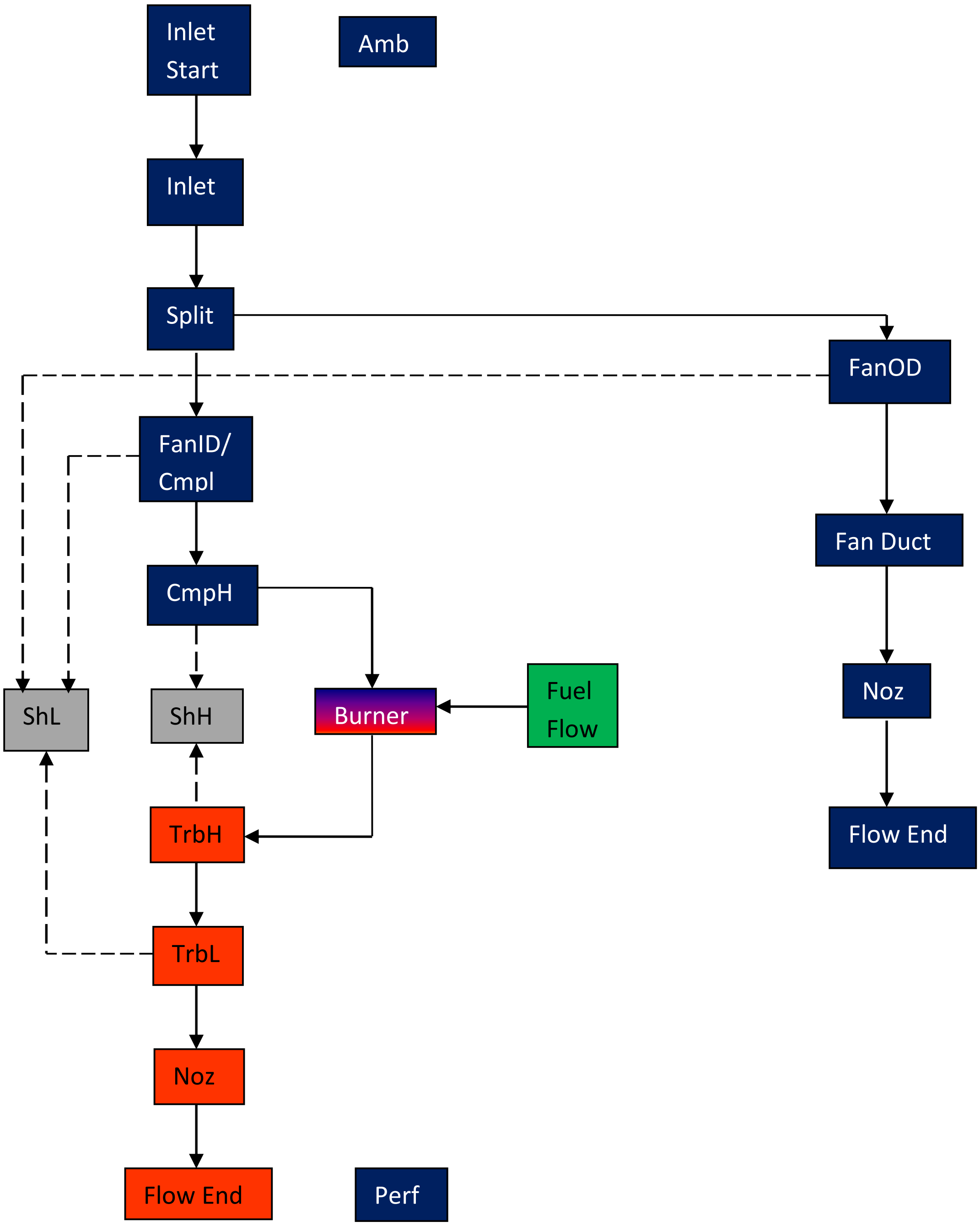


Figure 4-13 NPSS Model Schematic of High BPR Turbofan.

The fig above (4.20) shows the configuration of turbofan engine with different nozzle exits. The function of the elements are same as explained before, only that there is no mixing of flows considered here hence, the mixer element is not used.

4.3.3 Adaptive Cycle Engine

The adaptive cycle engine is one of the latest engine technology under development. This design type is suited for military aircraft. The NPSS model of the adaptive cycle engine is shown in figure 4.22. Elements are represented by boxes, arrows represent flow direction and port connections. Shaft ports connections are represented by dashed lines. The functions of all the elements are similar to that of the turbojet and bypass turbofan models. The first splitter divides the flow from the inlet into 3rd bypass stream and core flow. The core flow is further divided by the second splitter into core flow and 2nd bypass. The 2nd bypass air flow passes through the fan which is connected to low speed shaft and is then combined with the core flow before entering the nozzle with the help of a mixer element. The 3rd bypass air flow passes through the fan which is also connected to low speed shaft and is then combined with the core and 2nd flow stream before entering the nozzle with the help of the mixer element. This engine configuration is very versatile. One can even have different configurations for mixer elements. The core flow flows through the low pressure compressor which is connected to low speed shaft, the high pressure compressor which is connected to high speed shaft, then through the burner, the high pressure turbine (connected to the high-speed shaft) and the low pressure turbine (connected to the low-speed shaft), ultimately to be combined with the flow from the other two streams and to pass through nozzle and the end element. Figure 4-21 shows the diagram of adaptive cycle engine, where the outer bypass flow is represented by white line (top line), the inner bypass flow by green line (two middle lines) and the core flow by red line (center line).

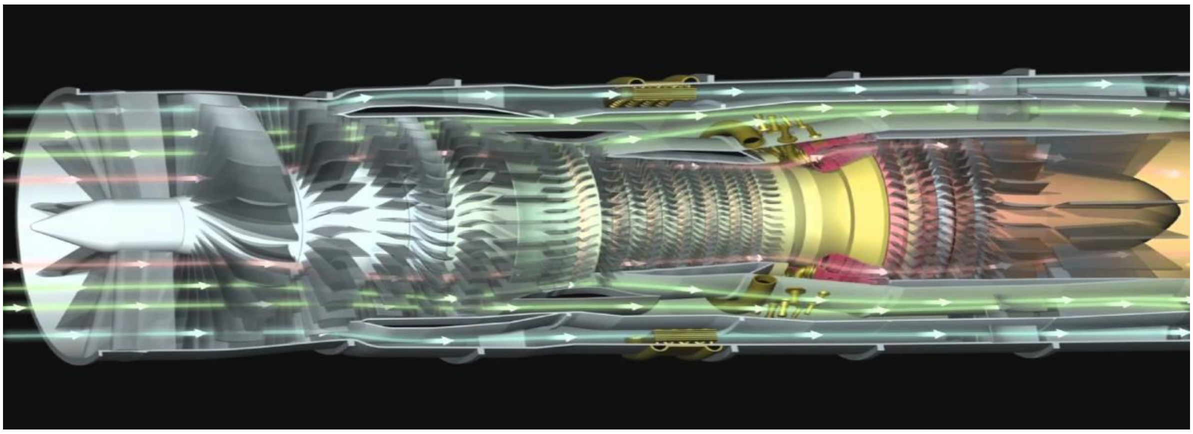


Figure 4-14 Adaptive Engine [3]

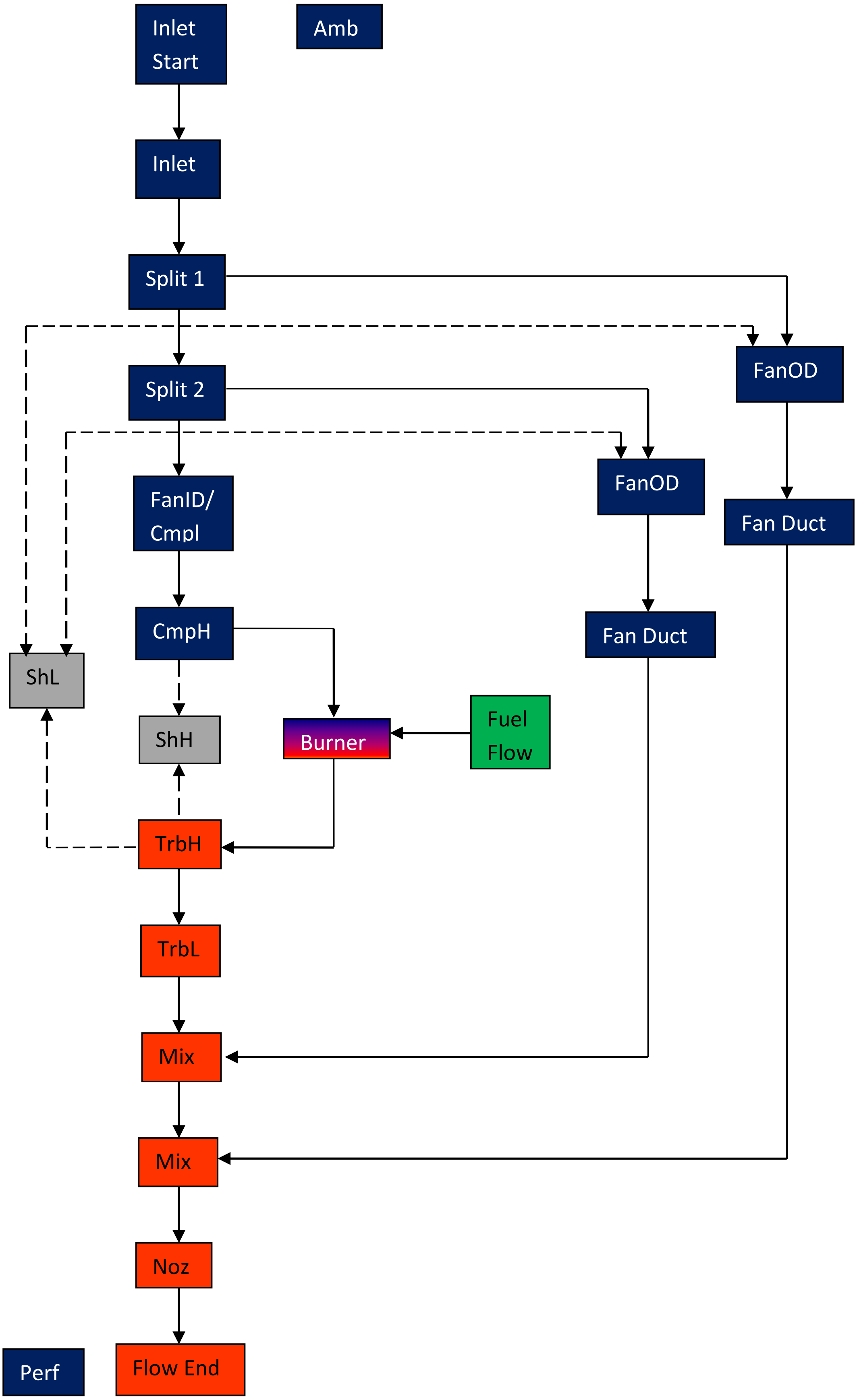


Figure 4-15 NPSS Model Schematic of Adaptive Cycle Engine.

CHAPTER 5

RESULTS

This chapter presents the results obtained for different simulation runs performed for various design and off design points. The analysis was done for design and three off-design points. The effect of different parameters on the engine were also studied and the designed engine was compared with the existing F119.

5.1 Effect of Bypass Ratio and Fan Pressure Ratio.

First the effect of bypass ratio and fan pressure ratio was studied for thrust and specific fuel consumption (SFC) values. In doing so, the values of bypass ratio are varied keeping everything else constant. This study was done by taking sea level static as the design point. The variation obtained was plotted in the form of thrust vs SFC. The plot obtained is shown below.

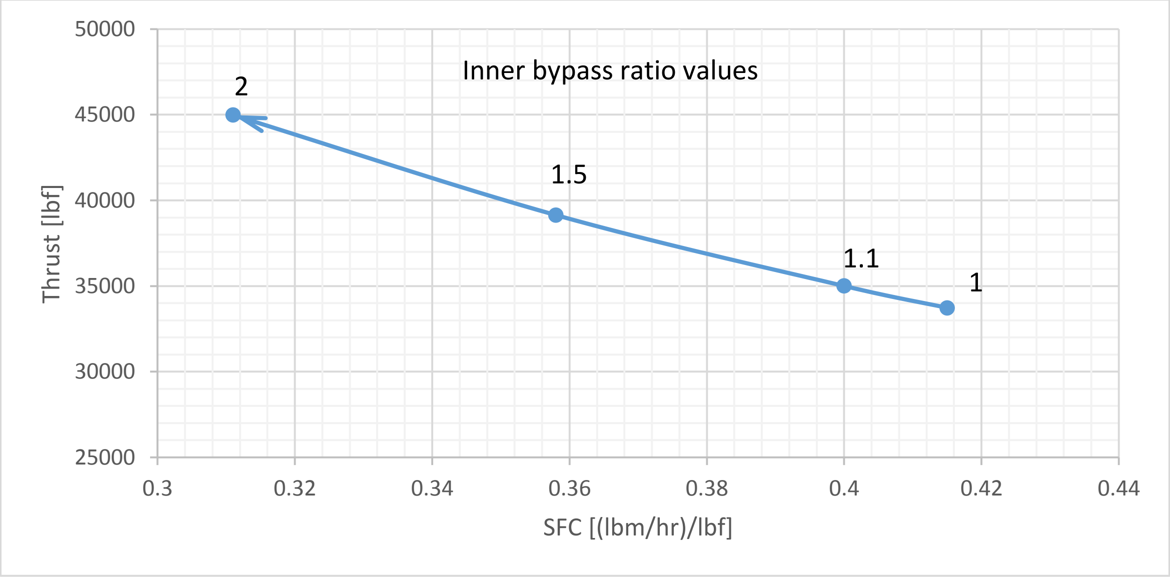


Figure 5-1 Variation of SFC and Gross Thrust by Varying Inner Bypass Ratio

For the above plot the values of bypass ratio, fan pressure ratio and compressor pressure ratio were taken at random. The values of inner bypass ratio, outer bypass ratio, first fan pressure ratio, second fan pressure ratio, high pressure compressor pressure ratio and turbine entry temperature taken for this analysis were 2, 2, 1.5, 1.5, 10 and 3000 Rankine respectively. After that then the value of just the inner bypass ratio was varied (from 1 to 2). From the plot it can be observed that as we increase the value of the inner bypass ratio, the SFC decreases, whereas the thrust increases. Along with inner bypass ratio the effect of Mach number was also studied. In this case the inner bypass ratio was set to a specific value (1, 1.1, 1.5. 2) and Mach number was varied (from 0 to 1.5) for that value of bypass ratio. The results obtained are shown in Fig. 5.2.

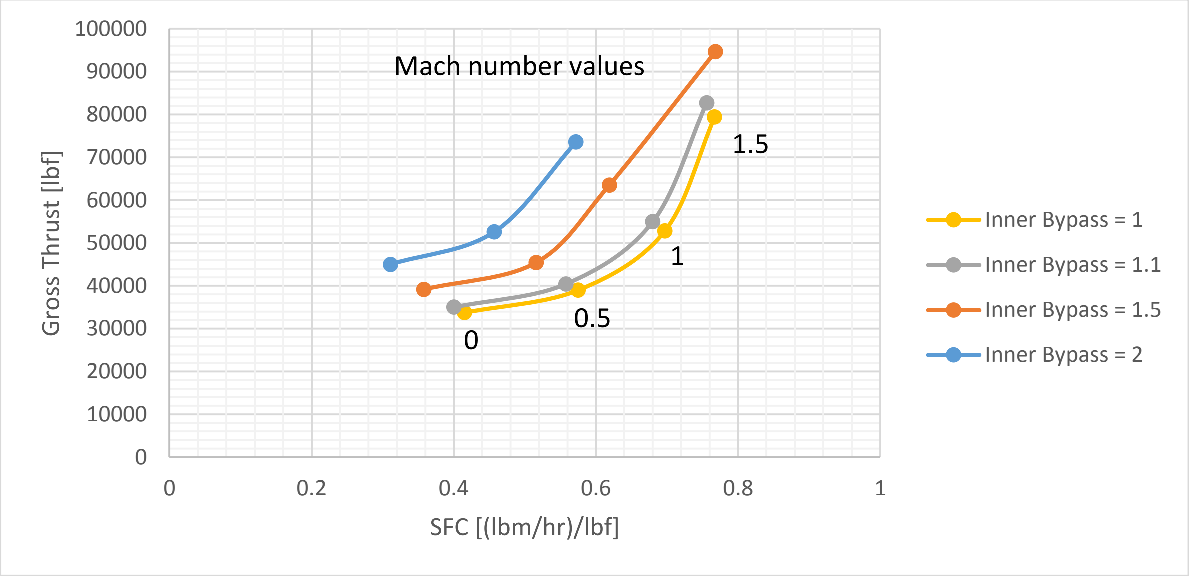


Figure 5-2 Variation of SFC and Gross Thrust with Mach Number and Inner Bypass Ratio

The plot shows similar results as observed in Fig 5.1 for increasing bypass ratio. In the Figure 5-2 above the yellow color line (bottom line) is for bypass ratio of 1, grey line (2nd from bottom) is for bypass ratio 1.1, orange line indicates for bypass ratio 1.5 and blue line is for bypass ratio of 2. As observed earlier the thrust increases as we increase the value of inner bypass ratio. It is observed that both SFC and thrust increase as we increase Mach number. It can be noted that the values shown in the above plot are for gross thrust, the values of net thrust will actually decrease as we increase Mach number due to the effect of ram drag. Ram drag will increase as we increase Mach number. The variation for net thrust is shown in Fig 5.3 below. Similar to the above plot, each line is for a specified value of bypass ratio (bottom 1.1 to top 2). For the analysis carried out for selecting design and off-design point cycle parameters, net thrust will be considered. The plots for varying outer bypass ratio were also studied. These plots are shown in Fig 5.4. The results obtained were similar to those seen for inner bypass ratio. Fig 5.5 shows the variation of outer bypass ratio along with Mach number and it is also the same as observed for inner bypass ratio. The values of engine properties and Mach number range are same as used for the analysis above. Here also, the lines are for constant values of outer bypass ratio. The results for both cases were observed to be the same because by increasing or decreasing the values of either the inner or outer bypass ratio we are actually increasing or decreasing the overall bypass ratio for whole system. So, it can be concluded that increasing bypass ratio of the system will increase the amount of thrust produced by the system and reduce the amount of fuel required.

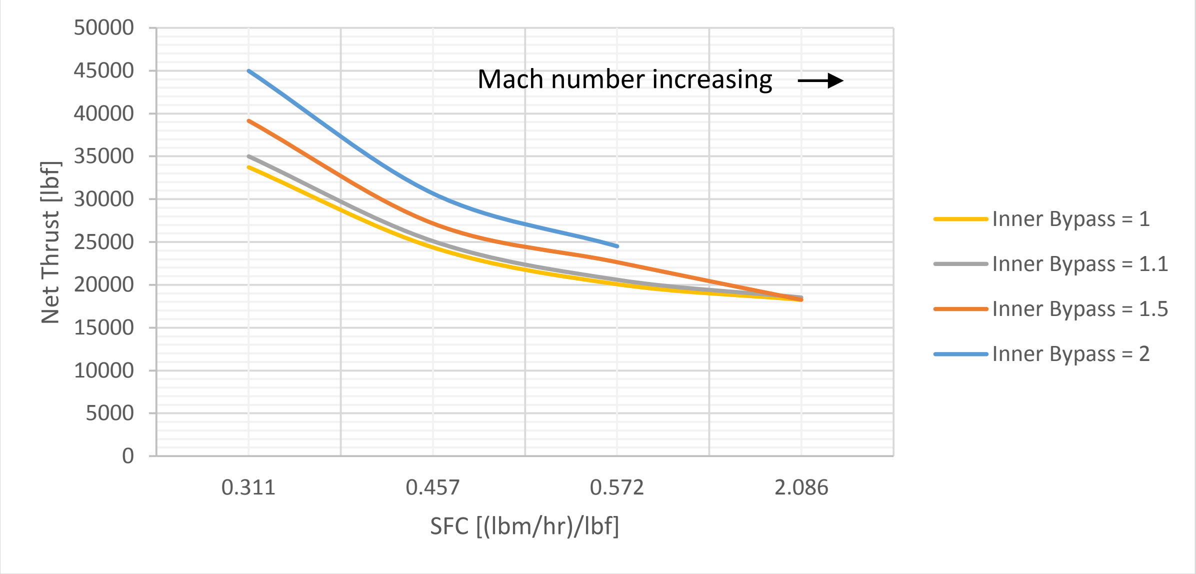


Figure 5-3 Variation of SFC and Net Thrust for Different Values of Inner Bypass Ratio and Mach Number.

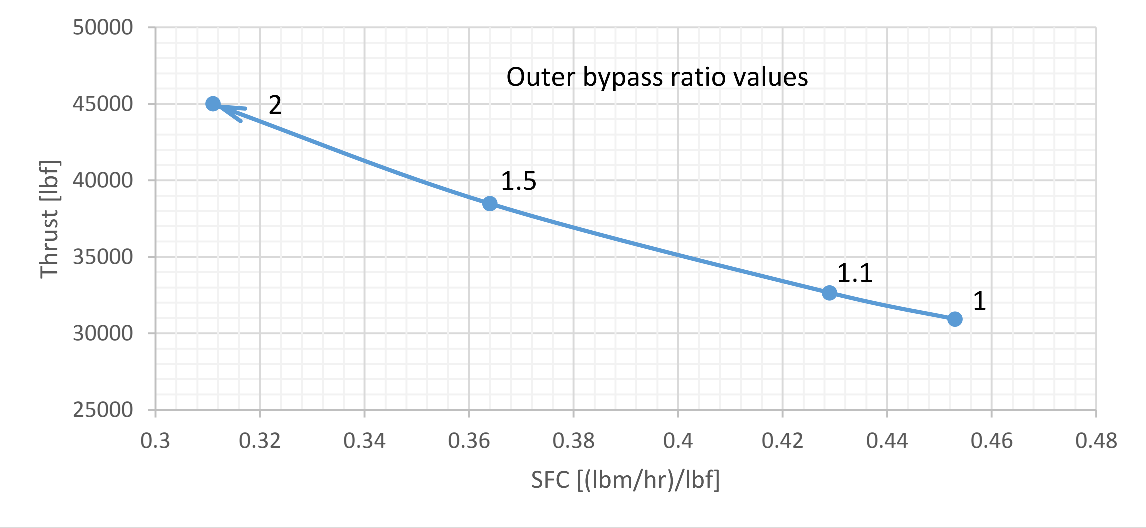


Figure 5-4 Variation of SFC and Gross Thrust by Varying Outer Bypass Ratio.

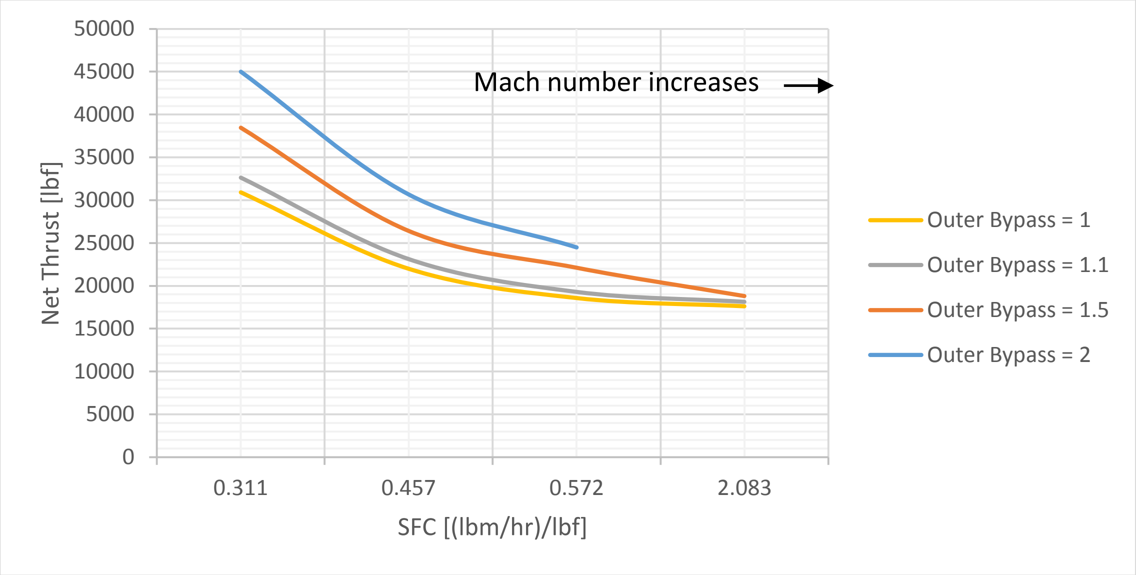


Figure 5-5 Variation of SFC and Gross Thrust for Different Values of Outer Bypass Ratio and Mach Number

After the variation of bypass ratio, the effect on engine performance due to variation in second fan pressure ratio was investigated. This study was also done in a similar manner as the one used for bypass ratio variation. A two fan configuration was considered and an arbitrary design point with inner bypass ratio, outer bypass ratio, first and second fan pressure ratio, high pressure compressor pressure ratio and turbine entry temperature of 0.4, 1.4, 3.3, 1.8, 6, 4000 Rankine was taken for the study. Then the value of pressure ratio for the second fan was varied (from 1 to 2). The variation of SFC and thrust is shown in Fig. 5.6 below.

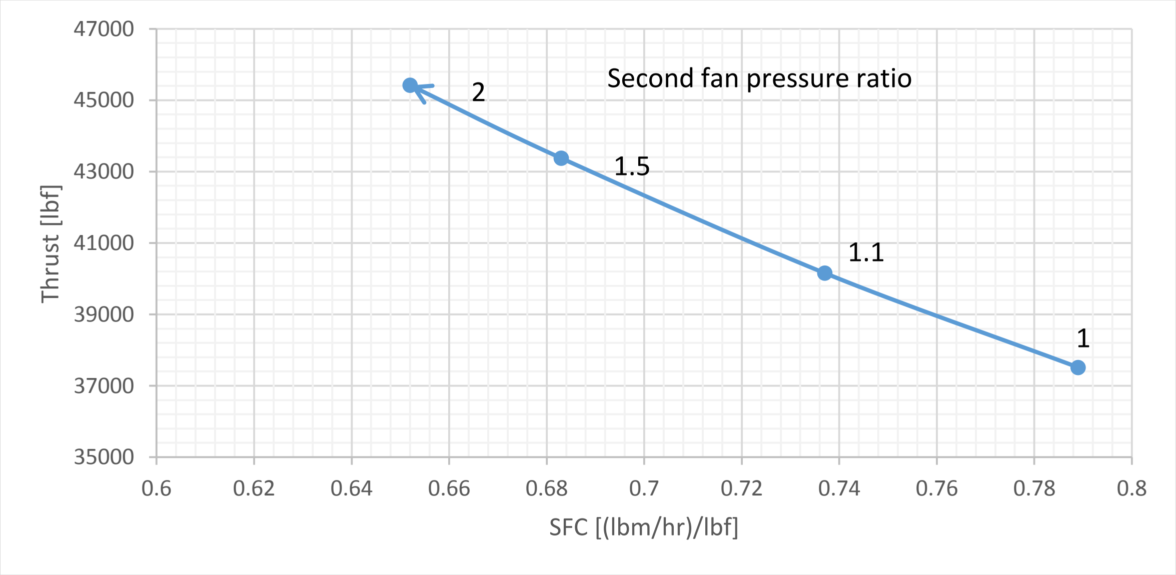


Figure 5-6 Variation of SFC and Thrust When Varying Second Fan Pressure Ratio.

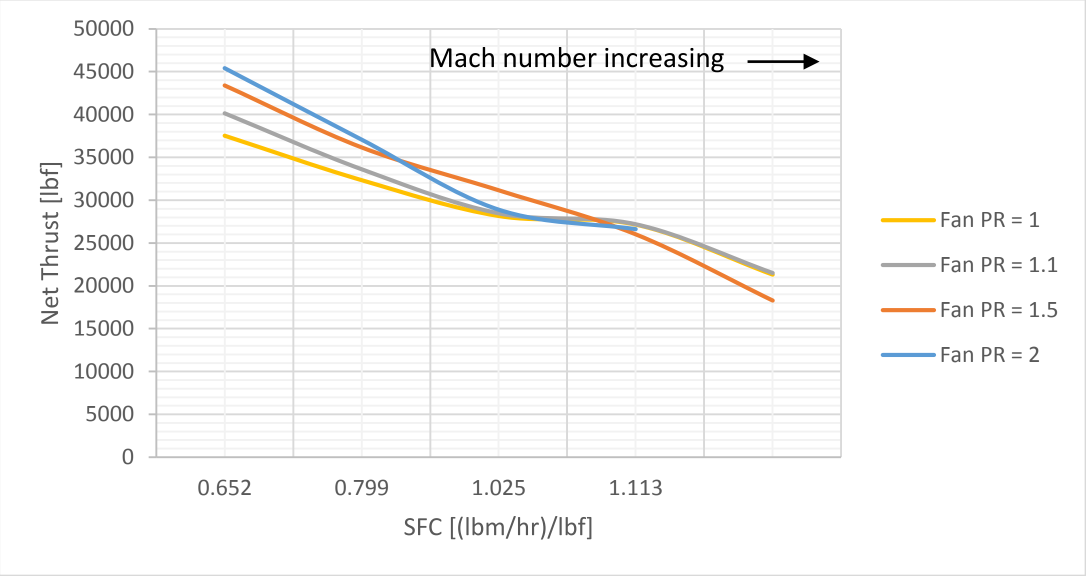


Figure 5-7 Variation of SFC and Net Thrust When Varying Second Fan Pressure Ratio

It was observed that the value of thrust increases and SFC decreases as we increase second fan pressure ratio. Fig 5.7 shows the variation of the parameters with varying value of second fan pressure ratio and Mach number (from 0 to 1.5). The lines in the above plot are for constant value of fan pressure ratio. The bottom line is for fan pressure ratio of 1, second from bottom is 1.1, second from top is 1.5 and the top is for the pressure ratio of 2. The value of net thrust decreases as we increase value of Mach number and value of SFC will increase with increase in Mach number.

5.2 Mixer Combination

In the adaptive cycle design there is only a single exhaust nozzle. Hence, the flows coming out of the two secondary streams and the core flow will mix before entering the CD nozzle. NPSS has an inbuilt mixer element to combine two streams into a single stream as explained earlier. In order to combine all three streams two mixer elements were used. Hence, there can be two combinations possible. The combinations are listed below.

1. First the core flow mixes with the first (inner) stream and then the mixed flow coming out of the mixer combines with the second (outer) stream.(used for thesis analysis).
2. First the flow coming out of both bypasses will mix and then that mixed flow will mix with the core flow and into the nozzle.

This analysis was carried out for inner and outer bypass ratios, the first and second fan pressure ratios, high pressure compressor pressure ratio and turbine entry temperature of 0.6, 1.2, 1.6, 1.4, 11, 4000 Rankine respectively. The results obtained are tabulated below.

Altitude (ft.)	Mach No.	Wair (lbm/s)	SFC (lbm/hr lbf)	Gross Thrust (lbf)	Net Thrust (lbf)
0	0	778	0.867	45000	45000
35000	0.9	335	0.903	22142	13021

Table 5.1- Values for first case

Altitude	Mach No.	Wair (lbm/s)	SFC (lbm/hr lbf)	Gross Thrust (lbf)	Net Thrust (lbf)
0	0	670	0.928	45001	45001
35000	0.9	311	1.009	21940	13456

Table 5.2- Values for second case

The study was done for the design point at sea level static and the off design point at 35000 ft and a Mach of 0.9. It was observed that for both mixing cases, the values of thrust and SFC values were almost same for both design and off-design points. Along with thrust and SFC, the values of mass flow rate, which is one of the parameters considered in the design analysis, as it determines the size of the engine, was also observed to be similar. It was inferred that the configuration/placement of mixer would not have much effect the result and hence, one can use either configuration in the design analysis. In this parametric analysis, the first case, i.e. first the core flow mixes with the first stream and then the mixed flow coming out of the mixer combines with the outer (second) stream, was used.

5.3 Design Point

The next step in the study was to decide on the design point cycle configuration. The method used for selecting the design point configuration is described here. There are six parameters to be taken into account for this design, two bypass ratios, two fan pressure ratios, high-pressure compressor, and turbine entry temperature (TET, T_{t4}). The value of TET was taken as 4000 Rankine, based on the literature survey[2,3,4]. All the other parameters were varied to obtain an "optimum" design configuration. The results were obtained for different fan and compressor pressure ratios, and bypass ratio. The design with three different nozzles i.e. without mixing the flows from all the streams and passing each stream through different individual nozzles, was used in the initial analysis to select the design point configuration. First, the outer bypass ratio (from 0 to 2) and first fan pressure ratio (from 1.4 to 6.1) were varied keeping other variables constant, and a plot was obtained as shown below.

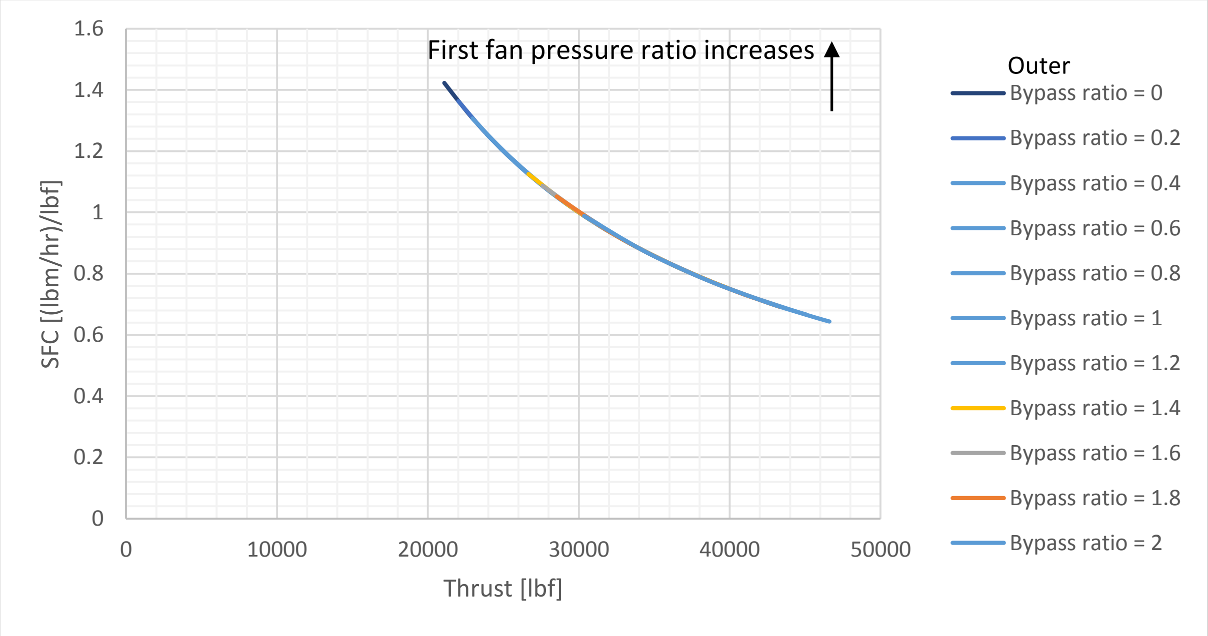


Figure 5-8 Variation of Parameters Due to Variation In Outer Bypass and First Fan Pressure Ratio

In the Fig-5.8, the values of first fan pressure ratio and outer bypass ratio were varied. In the graph the lines are for constant values of outer bypass ratios. The top line is for outer bypass ratio of 0 and the bottom line is for outer bypass ratio of 2. Each line has an increment of 0.2 bypass ratio as we go down. The graph indicates increase in thrust and decrease in SFC as we increase fan pressure ratio and bypass ratio (this study also explained in 5.1). In the graph obtained lines are overlapping and making it difficult to understand hence, the SFC lines were multiplied by a factor (0.9, 0.8, 0.7) to explain the effect of outer bypass ratio and first fan pressure ratio. From the design point of view the engine should have high bypass ratio for cruise condition to obtain higher range by decreasing fuel consumption, and low bypass for combat or take-off condition to obtain more thrust. From the analysis and engine requirements of adjustable bypass ratio, the bypass ratio value for outer stream was taken in the range of 0.1 to 0.6. As when the plane is in combat, the inner bypass, which is switchable, will have low or zero bypass ratio resulting in lower overall bypass ratio; and for cruise condition the value of inner bypass will be high resulting in higher overall bypass ratio. The points were selected according to the thrust requirements. From the plots shown below only those points were selected which produced the thrust value of 45,000lb (engine requirement). Similar plots were obtained for different values of inner bypass ratios, and are shown in Fig-5.9.

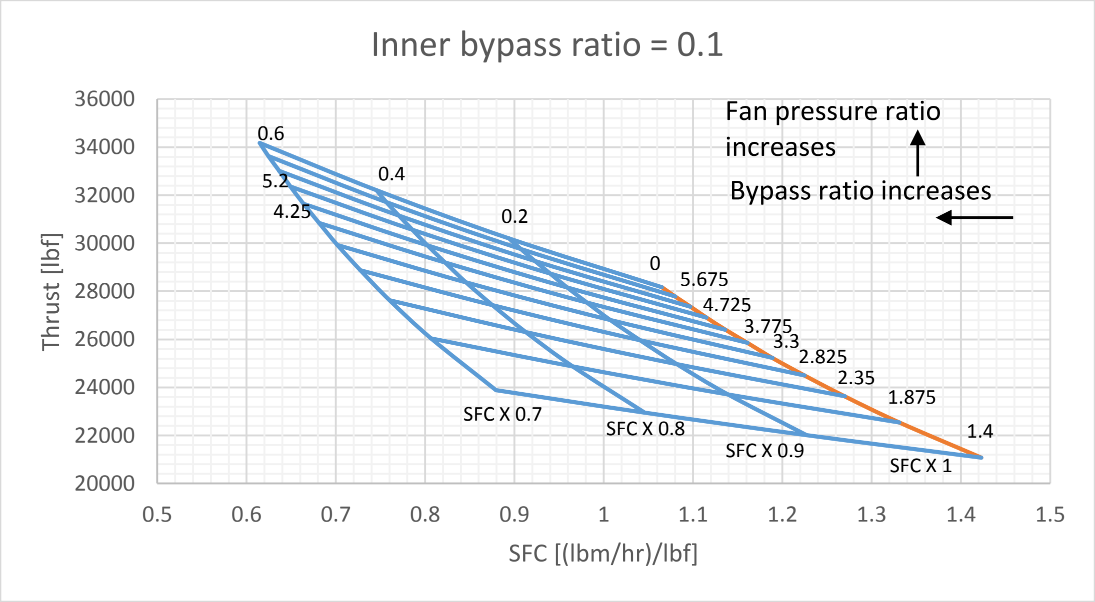


Figure 5-9 Variation of Parameters for Varying Outer Bypass Ratio and First Fan Pressure Ratio for Inner Bypass Ratio of 0.1

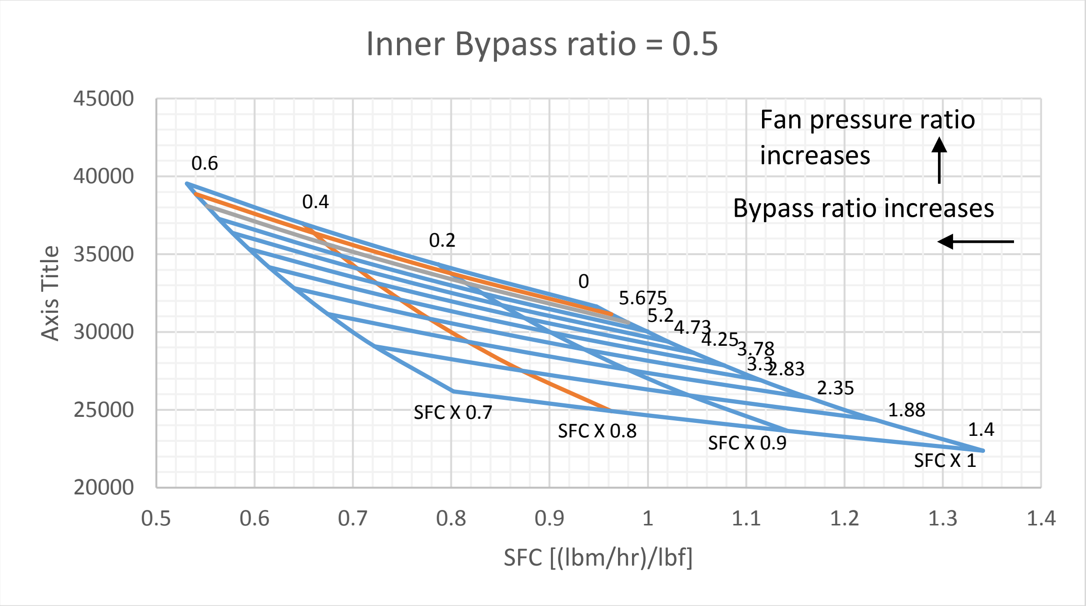


Figure 5-10 Variation of Parameters for Varying Outer Bypass Ratio and First Fan Pressure Ratio for Inner Bypass Ratio of 0.5

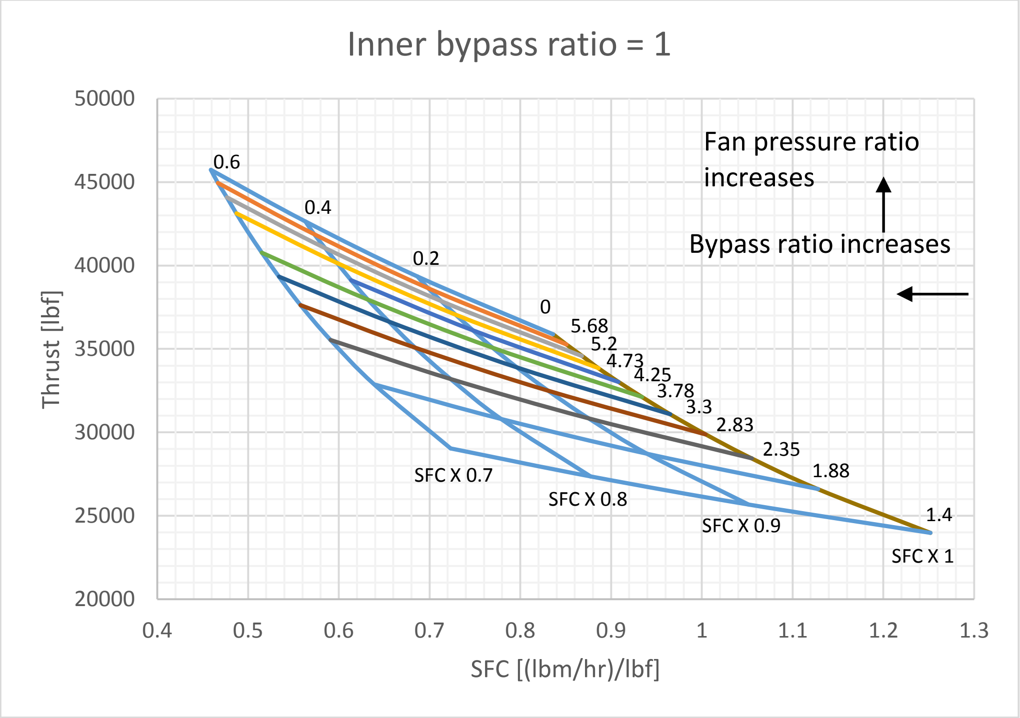


Figure 5-11 Variation of Parameters for Varying Outer Bypass Ratio and First Fan Pressure Ratio for Inner Bypass Ratio of 1

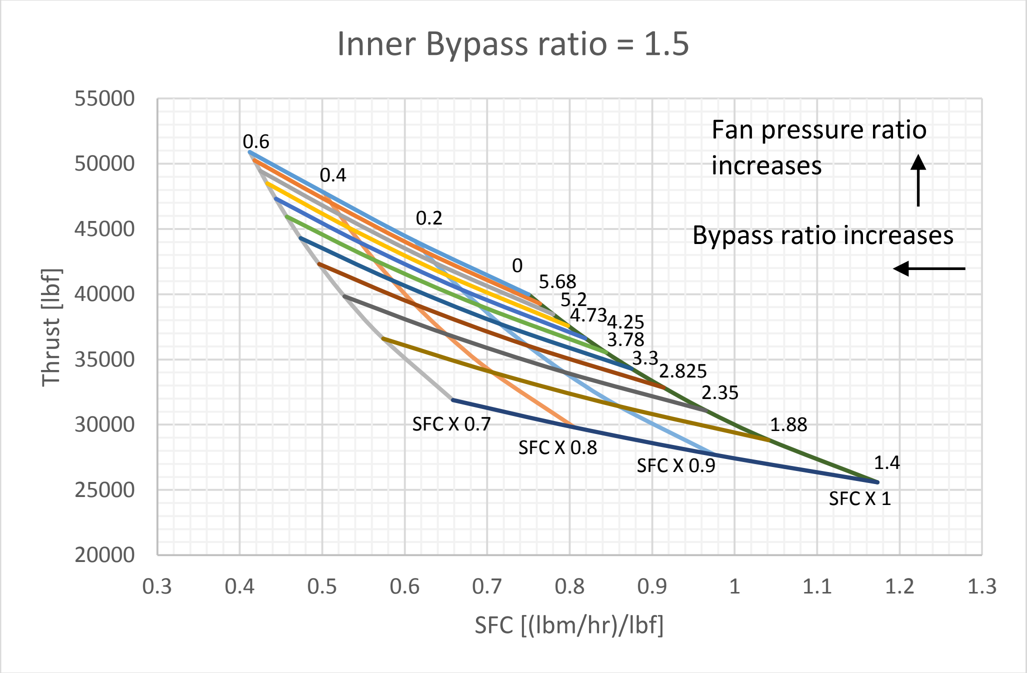


Figure 5-12 Variation of Parameters for Varying Outer Bypass Ratio and First Fan Pressure Ratio for Inner Bypass Ratio of 1.5

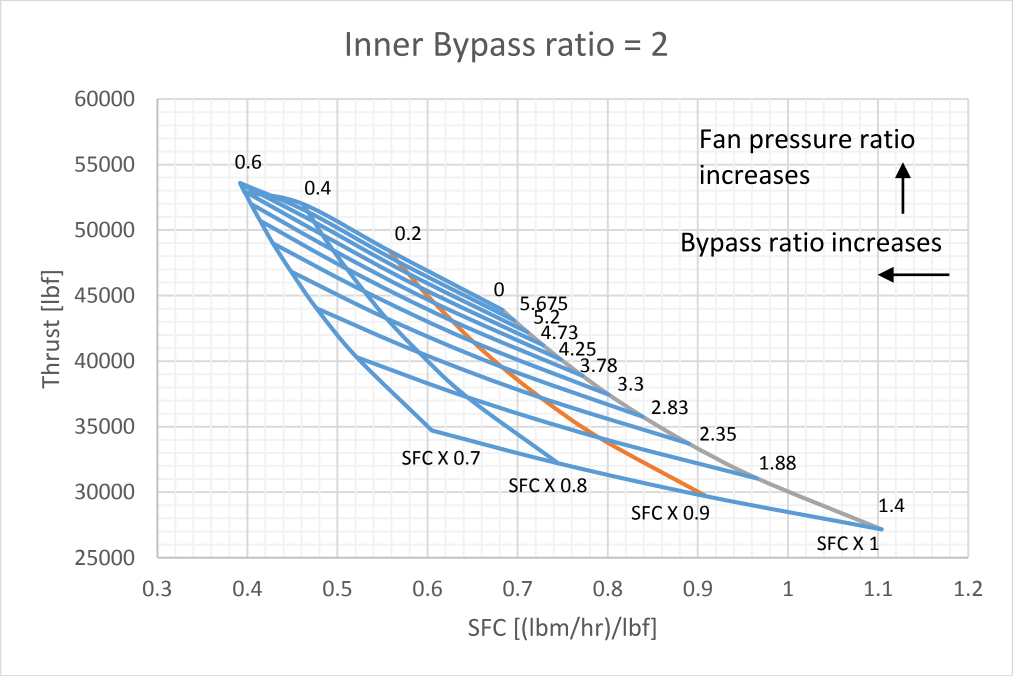


Figure 5-13 Variation of Parameters for Varying Outer Bypass Ratio and First Fan Pressure Ratio for Inner Bypass Ratio of 2

A similar plot was obtained for varying the second fan pressure ratio. From the values obtained the combination of inner and outer bypass ratios, first and second fan pressure ratios and high-pressure compressor pressure ratio, which gave net thrust of 45,000lb were selected. The next thing taken into consideration was mass flow rate, as it determines the size of an engine. Mass flow rate was calculated for the given thrust required by using mass flow parameter formula.

$$mass\ flow\ rate, \dot{m} = \rho AV \qquad \ldots\ldots (5.1)$$

where, ρ is density, A is area and V is velocity,

$$\dot{m} = \frac{A*Pt}{(Tt)^{\frac{1}{2}}}\left(\frac{\gamma}{R}\right)^{\frac{1}{2}} M\left(1+\frac{\gamma-1}{2}*M^2\right)^{-\frac{\gamma+1}{2(\gamma-1)}} \qquad \ldots\ldots (5.2)$$

The selected points were then checked for approximate correct mass flow rate values. The effect of bypass ratio and fan pressure ratio on mass flow rate is shown below.

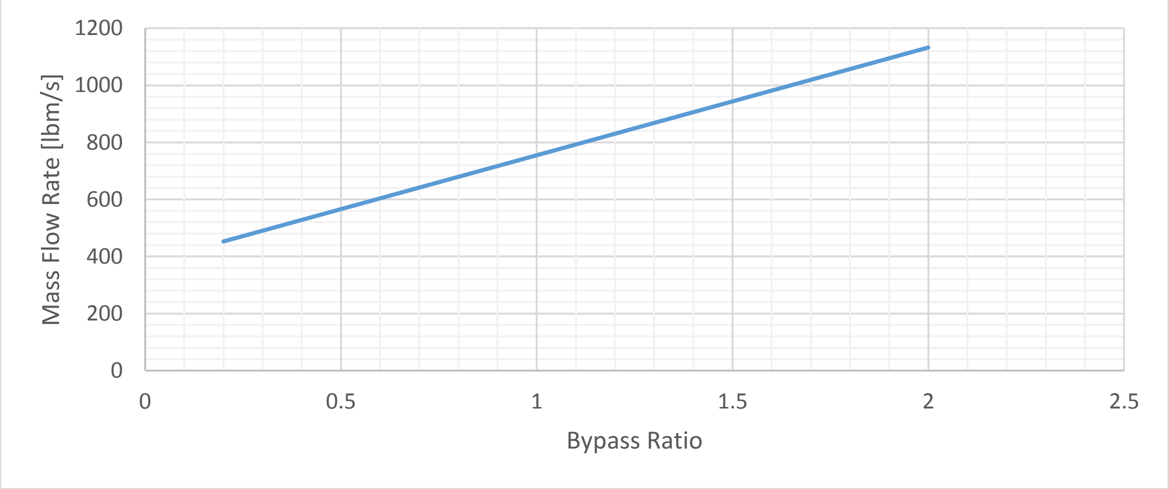


Figure 5-14 Variation of Mass Flow Rate with Outer Bypass Ratio

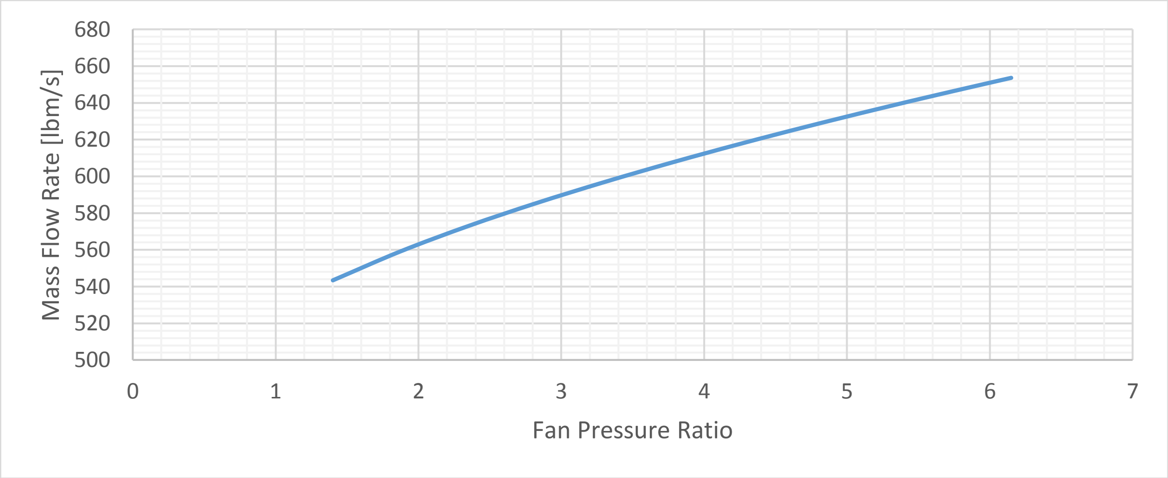


Figure 5-15 Variation of Mass Flow Rate with First Fan Pressure Ratio

It can be seen that increasing either bypass ratio or fan pressure ratio will increase the amount of total air flow into the engine. Furthermore, increase in mass flow rate will increase the size of the engine. So, considering the thrust required, mass flow rate, turbine entry temperature and values of outer bypass (between: 0 to 0.5) range the following points were selected. The pressure ratio value of first fan is higher, as there are two fan stages considered for that fan. We can even incorporate a map for both fan stages, i.e. total of three fan maps and model that in NPSS. But, that will only increase the number of parameters to be considered. The performance at multiple design points was analyzed because of the fact that if the cycle has good performance at design point that does not necessarily indicate that it will also have good performance at off-design points.

Outer Bypass Ratio	Inner Bypass Ratio	First Fan Pressure Ratio	Second Fan Pressure Ratio	High Pressure Compressor	Turbine Entry Temperature (Rankine)	SFC (lbm/hr lbf)	-Thrust (lbf)
0.4	1.4	3.3	1.8	6	4000	0.698	44999
0.4	1.4	3.2	1.9	6	4000	0.697	45000
0.4	1.4	3.4	1.8	6	4000	0.693	45000
0.3	1.6	3.4	1.8	6	4000	0.694	45000
0.4	1.4	3.3	1.6	7	4000	0.697	44999
0.4	1.4	3.3	1.9	6	4000	0.691	45000
0.4	1.4	3.4	1.6	7	4000	0.691	44999
0.4	1.4	3.4	1.9	6	4000	0.686	45000
0.3	1.6	3.4	1.9	6	4000	0.687	44999
0.4	1.4	3.3	1.7	7	4000	0.689	45000

Table 5.3- Selected design point cycle configurations

Once a few design points were selected, the mixers were sized according to the thrust requirement. As discussed above, the combination selected for this study was first mixing the core flow with the inner bypass and then combining the outer bypass (3rd stream) with the mixed flow. All the flows are combined before entering the nozzle. It should be noted that in this study heat losses, afterburner etc. are not considered. Mixer elements were sized for each of the design points selected.

5.4 Off-Design Performance

Once the sizing of the mixers for all the design point configurations was done, all the selected design points were analyzed for off-design performance. Off-design cases taken in this study are altitude, Mach number: 500 ft., 0.8; 35,000 ft., 0.8 and 40,000 ft., 2.2. The 500 ft., 0.8 Mach point was selected as a low altitude transonic dash case, used while in combat. The 35,000 ft., 0.8 Mach point was selected as high altitude cruise case, used for travelling from one point to another; and the 40000 ft., 2.2 case was selected as high altitude high dash case, used in combat. For off-design analysis, values of turbine entry temperature were specified. A temperature value of 4000 R was used for 500 ft. and 40000 ft operation, as the aircraft will be in combat and it will operate at max thrust condition. While for cruise at 35000 ft a turbine entry temperature of 3600 R was used. One of the main features of the adaptive engine cycle is the ability to mechanically vary bypass ratio and fan pressure ratio according to the operational situation to obtain maximum performance for wide range of conditions. So, along with specifying turbine entry temperature values, a range of fan pressure ratios of the second stage fan and inner stream bypass ratios were studied for each case. Once the parameters and conditions were known, off-design simulation of the engine was carried out and multiple data sets were studied. From the data generated, the following three design point cycle configurations were selected.

Outer Bypass Ratio	Inner Bypass Ratio	First Fan Pressure Ratio	Second Fan Pressure Ratio	High Pressure Compressor Pressure Ratio
0.4	1.4	3.3	1.8	6
0.4	1.4	3.4	1.8	6
0.4	1.4	3.3	1.9	6

Table 5.4- Selected design point parameters from off-design performance analysis

The configurations were selected on the engine's overall performance basis. For overall performance evaluation, the performance of each selected design point configuration was studied for all three off-design points. The points which gave good performance results for all the three points were selected. The off-design performance values for the selected design points are tabulated below.

Parameter	Altitude/Mach number		
	500 ft./0.8	35000 ft./0.8	40000 ft./2.2
Wair	741.9 lbm/s	244.3 lbm/s	534.8 lbm/s
2nd Bypass	1.456	1.331	1.672
1st Bypass	0.398	0.392	0.383
Fan 1 PR	2.822	3.409	1.970
Fan 2 PR	1.759	1.807	1.597
CompH PR	5.756	5.964	4.791
Shaft L RPM	4704	4871	4205
Shaft H RPM	10220	9407	10770
T_4	4000 R	3600 R	4000 R
Wfuel	9.875 lbm/s	3.064 lbm/s	6.219 lbm/s
SFC	0.933 lbm/(hr*lbf)	0.839 lbm/(hr*lbf)	1.204 lbm/(hr*lbf)
Gross Thrust	58683 lbf	19061 lbf	60217 lbf
Net Thrust	38119 lbf	13148 lbf	18597 lbf

Table 5.5- Values of off-design performance for configuration 1

Parameter	Altitude/Mach number		
	500 ft./0.8	35000 ft./0.8	40000 ft./2.2
Wair	737.6 lbm/s	243.6 lbm/s	528.5 lbm/s
2nd Bypass	1.457	1.330	1.665
1st Bypass	0.398	0.392	0.382
Fan 1 PR	2.898	3.514	1.806
Fan 2 PR	1.759	1.806	1.597
CompH PR	5.763	5.962	4.812
Shaft L RPM	4699	4873	4191
Shaft H RPM	10218	9408	10760
T_4	4000 R	3600 R	4000 R
Wfuel	9.776 lbm/s	3.040 lbm/s	6.136 lbm/s
SFC	0.925 lbm/(hr*lbf)	0.832 lbm/(hr*lbf)	1.198 lbm/(hr*lbf)
Gross Thrust	58478 lbf	19047 lbf	53443 lbf
Net Thrust	38033 lbf	13152 lbf	18446 lbf

Table 5.6- Values of off-design performance for configuration 2

Parameter	Altitude/Mach Number		
	500 ft./0.8	35000 ft./0.8	40000 ft./2.2
Wair	738.3 lbm/s	243.9 lbm/s	528.8 lbm/s
2nd Bypass	1.458	1.330	1.679
1st Bypass	0.397	0.391	0.378
Fan 1 PR	2.816	3.410	1.959
Fan 2 PR	1.852	1.907	1.665
CompH PR	5.764	5.961	4.824
Shaft L RPM	4697	4872	4185
Shaft H RPM	10216	9409	10753
T_4	4000 R	3600 R	4000 R
Wfuel	9.740 lbm/s	3.032 lbm/s	6.097 lbm/s
SFC	0.923 lbm/(hr*lbf)	0.830 lbm/(hr*lbf)	1.196 lbm/(hr*lbf)
Gross Thrust	58445 lbf	19047 lbf	53368 lbf
Net Thrust	37981 lbf	13145 lbf	18349 lbf

Table 5.7- Values of off-design performance for configuration 3

For calculating the values of thrust, the inlet ram recovery effect was also included. Mil. Spec. was used for calculating inlet pressure recovery. The formula used for pressure recovery is shown below.

$$\frac{Pt_2}{Pt_0} = \eta_i(1 - 0.075(M - 1)^{1.35}) \qquad \text{For M > 1} \qquad \text{…… (5.3)}$$

$$\frac{Pt_2}{Pt_0} = \eta_i \qquad \text{For M < 1} \qquad \text{…… (5.4)}$$

After the three configurations were selected, all the data were studied in detail and a final engine cycle was selected. The performance shown in the tables 5.5 to 5.7 are just the off-design performance for the three points. The final configuration was selected based on thrust produced and specific fuel consumption values. It was also observed from the data that lower values of the 1st bypass produced high thrust and higher value of the 1st bypass resulted in lower specific fuel consumption values. So, high bypass ratio was selected for cruise condition and lower values of bypass ratio were selected for combat or high thrust case. From the three points selected, the first configuration had better overall performance when compared to other two configurations. Once the design configuration was selected, best possible combination for inner bypass ratio and second fan pressure ratio for each off-design point were studied and obtained. The recommended values of inner bypass ratio and 2nd fan pressure ratio when operating at the different altitude/Mach number flight conditions are tabulated below

Parameter	500 ft., 0.8 Mach	35000 ft., 0.8 Mach	40000 ft., 2.2 Mach
2nd Fan Pressure ratio	1.7	1.4	1.6
Inner bypass ratio	0.1	1.2	0

Table 5.8- Values of bypass ratio and fan pressure ratio for best performance at different altitudes.

The value of the inner bypass ratio from the above table determines the amount of air flowing through the core. For the case of 500 ft. most of the air will flow through the core and only 1/10th of the air will flow through the second bypass. For 40000 ft. all the air will flow through the core and there will be no flow through the inner bypass stream. On the other hand for transonic cruise case, a high amount of flow will pass through the inner bypass resulting in higher overall bypass ratio. The off-design performance of the engine for the selected design point configuration, with optimum values of fan pressure ratio and bypass ratio are tabulated below.

Parameter	Altitude/Mach number		
	500 ft./0.8	35000 ft/0.8	40000 ft./2.2
SFC	0.831 lbm/(hr*lbf)	0.671 lbm/(hr*lbf)	1.753 lbm/(hr*lbf)
Gross Thrust	61358 lbf	24132 lbf	65567 lbf
Net Thrust	41501 lbf	15120 lbf	19756 lbf

Table 5.9- Values of fuel consumption and thrust for different off-design points

It can be observed that the values of thrust in each case after using best possible combination of inner bypass ratio and second fan pressure ratio, is greater than running the designed engine in off-design mode by allowing each parameter to scale by themselves (Table 5.5). Along with thrust values reduction in specific fuel conception value was also observed except for 40000 ft. where an increase was seen. The engine will be now compared to existing engine.

5.5 Comparison of The Adaptive Cycle and Conventional Cycle Engine.

The final variable cycle engine design was compared to the existing Pratt & Whitney F-119 (used in the Lockheed-Martin F-22 aircraft) engine. The published SLS cycle parameters of the F-119 were used and a NPSS simulation was carried out to predict performance at the sea level design point and the same three off-design points. While running the code for the F-119, the value of target thrust at sea level was set to 45000 lb so that proper comparison of the engines can take place. Now, when both engines are of the same thrust class, with different values of bypass ratio's and pressure ratio's, the comparison is tabulated below.

Adaptive cycle engine.				F-119			
Altitude	Mach No.	Thrust	SFC	Altitude	Mach No.	Thrust	SFC
0	0	44999 lbf	0.698 lbm/(hr*lbf)	0	0	45000 lbf	1.121 lbm/(hr*lbf)
500 ft.	0.8	41501 lbf	0.831 lbm/(hr*lbf)	500 ft.	0.8	36964 lbf	1.304 lbm/(hr*lbf)
35000 ft.	0.8	15120 lbf	0.671 lbm/(hr*lbf)	35000 ft.	0.8	16143 lbf	1.136 lbm/(hr*lbf)
45000 ft.	2.2	19756 lbf	1.753 lbm/(hr*lbf)	45000 ft.	2.2	11018 lbf	1.771 lbm/(hr*lbf)

Table 5.10- Comparison of adaptive cycle engine with F-119

It can be observed that adaptive cycle engine produces lower SFC values at all operating conditions when compared to the F-119. It also produces more thrust than the F-119 in combat conditions. The F-119 has comparable thrust to the adaptive cycle engine for transonic cruise but has very large value of SFC when compared to the adaptive cycle engine. So, it can be said that the liberty of changing the parameters in off-design conditions will result in a more complicated mechanical design, but will also have tremendous advantage for obtaining increase in thrust and decrease in fuel consumption and hence, overall a good performance gas turbine engine.

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

Numerical simulation of a variable cycle aircraft engine was performed using Numerical Propulsion System Simulation (NPSS) software in an attempt to evaluate the performance of the new adaptive cycle engine to determine how the engine will operate for different flight conditions.

Design point analysis

- Variation of specific thrust and fuel consumption was observed for different values of Mach number, fan pressure ratio and bypass ratio.
- The thrust increased with increase in bypass ratio while the specific fuel consumption decreased with increase in bypass ratio.
- Similar to that of bypass ratio the thrust increases as we increase fan pressure ratio and specific fuel consumption reduces with increase in fan pressure ratio.
- The values of gross thrust increased, net thrust decreased and fuel consumption increased as we increased the Mach number, keeping all other parameters constant (including altitude).
- If the engine has a single exhaust, the design can have two configuration of mixers; first, mixing the core flow with the second stream and the mixing the flow from third stream with the mixed flow from mixer 1, and second, mixing both bypass steams first and the mixing the combined flow with the core flow and into the nozzle. Both configurations will give essentially very similar results, i.e. almost the same values of thrust and fuel consumption.

Off-design performance analysis

- The engine can be designed for one design point and depending on the mission profile the off-design parameters can be calculated. For the adaptive cycle engine, the highest thrust is obtained by allowing almost all of the flow through the inner bypass to be diverted into the core, implying almost no flow through the inner (2^{nd}) bypass. However for the transonic cruise condition, allowing less flow through the core results in a higher bypass engine, reducing fuel consumption.
- As adaptive cycle engine has variable bypass ratios and fan pressure ratios. The optimum configuration of these parameters were obtained for each off-design case.
- Adaptive cycle engine produces more thrust and requires less fuel when compared to conventional fixed bypass engine designs.
- Adaptive cycle engine technology is an advanced concept that can be used in advanced military airplanes.

6.2 Future Work

- Although the work done in this study provides basic analysis of the adaptive engine cycle, many aspects of its performance need to be studied further. The simulation of the engine for the design point (SLS) and three off-design points is studied but the aerodynamic effects affecting the performance have not been addressed here. So, the aerodynamics effects on adaptive cycle engine can be a focus area for future research.
- This study deals with only one nozzle. But, different exhaust conditions can be taken into account; such as three separate exhaust nozzle configurations or two exhaust nozzle configurations.
- Study of the adaptive engine with afterburner can also be carried out as afterburner was not considered for this study.
- Full analysis of the adaptive cycle engine which includes heat transfer analysis, afterburner, use of heat exchangers, etc. can also be done.
- This project only considers parametric study of adaptive cycle engine. The data obtained can be validated with the experimental data.

- Transient mode analysis of an engine was not carried out in this study but, it is an important analysis to be considered.
- Best configuration of bypass ratio and fan pressure ratio for different altitudes and conditions in off-design can be tabulated for future reference.

APPENDIX A

NPSS Code For Modeling of Adaptive Cycle Engine

```
//---------------------------------------------------------------------
//                        Model Definition
//---------------------------------------------------------------------

// Instantiate the Ambient element
// This element sets the ambient conditions
// and is refered to by the InletStart and Nozzle elements

Element Ambient Amb {
switchMode = "ALDTMN"; // Set ambient conditions based on altitude, Mach
                                              //number, and standard day delta T
  alt_in = 35000.; // ft, input altitude
  MN_in = 0.8;// input, Mach number
  dTs_in = 0.; // Rankine, input temperature delta from standard day conditions
} // END Amb

// Instantiate the InletStart element
Element InletStart FsEng {
  AmbientName = "Amb"; // Name of the Ambient element
  W_in = 100.; // lbm/s, input air flow flow rate
} // End InletStart

// Instantiate the Inlet element
Element Inlet InEng {
  eRamBase = 0.995; // Ram pressure recovery
} // END InEng
```

```
// Instantiate the bypass splitter
Element Splitter SpltFan1 {
        BPRdes = 5.;
} // END SpltFan

// Instantiate the fan OD 2 element
Element CompressorA CmpFSec2 {
        // Set compressor design point values
        PRdesA  = 1.5; // design point pressure ratio
        effDesA = 0.85; // design point efficiency

// Load file that instantiates a subelement and plugs into compressor socket (S_map) and contains
// compressor performance map
#include "fanE3F.map"; // when scoping, refer to the subelement by its socket name: S_map
} // End CmpFSec

// Instantiate the fan duct for the bypass flow in duct 2
Element Duct D230 {
        switchDP = "INPUT"; // allow the user to input the relative pressure drop (deltP / Pin)
        dPqP_in = 0.015; // user-input relative pressure drop
} // END Dfan

Element SplitterA SpltFan2 {
        BPRdesA = 5.;
} // END SpltFan

// Instantiate the fan OD 1 element
Element Compressor CmpFSec1 {
```

```
    // Set compressor design point values
    PRdes  = 1.5; // design point pressure ratio
    effDes = 0.85; // design point efficiency

    // Load file that instantiates a subelement and plugs into compressor socket (S_map) and contains
    // compressor performance map
    #include "fanE3.map"; // when scoping, refer to the subelement by its socket name: S_map
} // End CmpFSec

// Instantiate the fan duct for the bypass flow
Element Duct D130 {
    switchDP = "INPUT"; // allow the user to input the relative pressure drop (deltP / Pin)
    dPqP_in = 0.015; // user-input relative pressure drop
} // END Dfan

// Instantiate the Compressor element
Element CompressorA CmpL {
    // Set compressor design point values
    PRdesA  = CmpFSec1.PRdes * CmpFSec2.PRdesA; // design point pressure ratio
    effDesA = 0.85; // design point efficiency

    // Load file that instantiates a subelement and plugs into compressor socket (S_map) and contains
    // compressor performance map
    #include "lpcE3.map"; // when scoping, refer to the subelement by its socket name: S_map

} // End CmpL
```

```
Element Compressor CmpH {
    // Set compressor design point values
    PRdes  = 10.0; // design point pressure ratio
    effDes = 0.89; // design point efficiency

    // Load file that instantiates a subelement and plugs into compressor socket (S_map) and contains
    // compressor performance map
    #include "hpcE3.map"; // when scoping, refer to the subelement by its socket name: S_map

} // End CmpH

// Start the flow of fuel
Element FuelStart FusEng {
    Wfuel = 2.0; // lbm/s, fuel flow rate (Used ONLY when Burner switchBurn = WFUEL)
    //LHV = 18000; // BTU/lbm, user input fuel lower heating value (LHV).  Default is 18400 BTU/lbm
} // End FusEng

// Instantiate the Burner element
Element Burner BrnPri {
    //dPqPfBase        = 0.05; // user input friction relative pressure drop (Pin - Pout)/Pin
    dPqP_dmd        = 0.05; // user input friction relative pressure drop (Pin - Pout)/Pin
    effBase         = 0.98; // user input burner adiabatic efficiency

    // The value for switchBurn determines how burner fuel flow rate is set
    // if switchBurn = FUEL, then use Wfuel that is set in the burner element
    // if switchBurn = WFUEL, then use fuel start element flow rate to set Wfuel (Fu_I.Wfuel inherited from fuel start Fu_O.Wfuel)
```

```
	// if switchBurn = FAR, then use air inlet FAR value (Fl_I.FAR) to calculate Wfuel
	switchBurn = "WFUEL"; // FUEL, WFUEL, or FAR
	// Wfuel = 0.2; // lbm/s, user input fuel flow rate.  Used only whey switchBurn = FUEL
} // End BrnPri


// Instantiate the High Pressure Turbine element
Element Turbine TrbH {
	// Set turbine design point efficiency and pressure ratio initial guess
	effDes = 0.92;  // user-specified Design efficiency
	PRbase = 4.5;  // pressure ratio initial guess
	// Load file that instantiates a subelement and plugs into turbine socket (S_map) and contains
	// turbine performance map
	#include "hptE3.map"; // when scoping, refer to the subelement by its socket name: S_map
} // End TrbH


// Instantiate the High Pressure Turbine element
Element Turbine TrbL {
	// Set turbine design point efficiency and pressure ratio initial guess
	effDes = 0.92;  // user-specified Design efficiency
	PRbase = 4.5;  // pressure ratio initial guess
	// Load file that instantiates a subelement and plugs into turbine socket (S_map) and contains
	// turbine performance map
	#include "lptE3.map"; // when scoping, refer to the subelement by its socket name: S_map
} // End TrbL


// Instantiate the Nozzle element for the primary flow
```

```
Element Mixer FMixer {
	// declare some extra variables of interest
	real Area, AreaDes, ExtractionRatio;
	Fl_I1.MN = 0.1; // primary stream Mach number
	//Fl_I2.MN = F150.MN; // secondary stream Mach number
	s_V1 = 0.970; // primary stream velocity loss coefficient
	s_V2 = 0.970; // secondary stream velocity loss coefficient
	// calculate mixer total entrance area and extraction ratio
	void postexecute() {
	Area = Fl_I1.Aphy + Fl_I2.Aphy;
	ExtractionRatio = Fl_I2.Pt / Fl_I1.Pt;
	}
}


Element Mixer FMixerC {
	// declare some extra variables of interest
	real Area, AreaDes, ExtractionRatio;
	//Fl_I1.MN = 0.9; // primary stream Mach number
	Fl_I1.MN = 0.3; // secondary stream Mach number
	s_V1 = 0.970; // primary stream velocity loss coefficient
	s_V2 = 0.970; // secondary stream velocity loss coefficient
	// calculate mixer total entrance area and extraction ratio
	void postexecute() {
	Area = Fl_I1.Aphy + Fl_I2.Aphy;
	ExtractionRatio = Fl_I2.Pt / Fl_I1.Pt;
	}
}
```

```
Element Nozzle NozPri {
  PsExhName = "Amb.Ps"; // Model variable for ambient static pressure
  switchType = "CONIC"; // conic nozzle
} //END NozPri


// Instantiate the Nozzle element for the bypass flow
/*
Element Nozzle NozSec1 {
  PsExhName = "Amb.Ps"; // Model variable for ambient static pressure
  switchType = "CONIC"; // conic nozzle
} //END NozSec


Element Nozzle NozSec2 {
  PsExhName = "Amb.Ps"; // Model variable for ambient static pressure
  switchType = "CONIC"; // conic nozzle
} //END NozSec
*/
// Instantiate the Low Pressure Shaft element


Element Shaft ShL {
  //  Mechanical Ports.  These are created as needed on the shaft.
ShaftInputPort Sh_ICmpFSec1; // create a shaft port for a mechanical connection between the shaft and the fan OD 1 element
ShaftInputPort Sh_ICmpFSec2; // create a shaft port for a mechanical connection between the shaft and the fan OD 2 element
ShaftInputPort Sh_ICmp; // create a shaft port for a mechanical connection between the shaft and the low pressure compressor
ShaftInputPort Sh_ITrb; // create a shaft port for a mechanical connection between the shaft and the low pressure turbine
 Nmech = 5000.; // rpm, shaft speed
} // End ShL
```

```
// Instantiate the Low Pressure Shaft element
Element Shaft ShH {
  //  Mechanical Ports.  These are created as needed on the shaft.
ShaftInputPort Sh_ICmp; // create a shaft port for a mechanical connection between the shaft and the high pressure compressor
ShaftInputPort Sh_ITrb; // create a shaft port for a mechanical connection between the shaft and the high pressure turbine
Nmech = 10000.; // rpm, shaft speed
} // End ShH



// End the flow of air for primary stream
Element FlowEnd FePri {
} // End FrPri

// End the flow of air for secondary stream
/*
Element FlowEnd FeSec1 {
} // End FeSec

Element FlowEnd FeSec2 {
} // End FeSec
*/
// Instantiate the EngPerf element
// This element makes some basic engine performance calculations

Element EngPerf Perf {
} //End Perf
//--------------------------------------------------------------------
```

```
//                      Component Linkages
//---------------------------------------------------------------------

// Link Fluid Ports

// Primary Hot Section
linkPorts( "FsEng.Fl_O"   ,              "InEng.Fl_I"      ,       "F0" );
linkPorts( "InEng.Fl_O"        ,         "SpltFan1.Fl_I"   ,       "F020" );
//linkPorts( "CmpFSec1.Fl_O" ,           "SpltFan1.Fl_I"  ,        "F120" );
linkPorts( "SpltFan1.Fl_O1" ,            "SpltFan2.Fl_IA"   ,      "F022" );
//linkPorts( "CmpFSec2.Fl_O"  ,          "SpltFan2.Fl_I"  ,        "F220" );
linkPorts( "SpltFan2.Fl_O1A"   ,         "CmpL.Fl_IA"       ,      "F025" );
linkPorts( "CmpL.Fl_OA",                 "CmpH.Fl_I"       ,       "F026" );
linkPorts( "CmpH.Fl_O" ,                 "BrnPri.Fl_I"     ,       "F030" );
linkPorts( "BrnPri.Fl_O" ,               "TrbH.Fl_I"       ,       "F040" );
linkPorts( "TrbH.Fl_O",                  "TrbL.Fl_I"       ,       "F045" );
linkPorts( "TrbL.Fl_O" ,                 "FMixerC.Fl_I2"   ,       "F050" );
linkPorts( "FMixer.Fl_O" ,               "NozPri.Fl_I"     ,       "F055" );
linkPorts( "NozPri.Fl_O" ,               "FePri.Fl_I"     ,        "F090" );

// Fan duct 1 section
linkPorts( "SpltFan2.Fl_O2A",            "CmpFSec1.Fl_I",        "F120"   );
linkPorts( "CmpFSec1.Fl_O",              "D130.Fl_I",              "F130"   );
linkPorts( "D130.Fl_O",                  "FMixerC.Fl_I1",          "F170"   );
linkPorts( "FMixerC.Fl_O",               "FMixer.Fl_I2",           "F175"   );
//linkPorts( "NozSec1.Fl_O",             "FeSec1.Fl_I",            "F190"   );

// Fan duct 2 section
linkPorts( "SpltFan1.Fl_O2",             "CmpFSec2.Fl_IA",   “F220"   );
```

```
linkPorts( "CmpFSec2.Fl_OA",        "D230.Fl_I",          "F230"   );
linkPorts( "D230.Fl_O",             "FMixer.Fl_I1",       "F270"   );
//linkPorts( "NozSec2.Fl_O",        "FeSec2.Fl_I",        "F290"   );


// Link Fuel Ports
linkPorts( "FusEng.Fu_O" ,          "BrnPri.Fu_I" ,       "Fu_In" );


// Link Shaft Ports
linkPorts( "CmpFSec1.Sh_O" ,        "ShL.Sh_ICmpFSec1"  ,       "MeCmpFSec1" );
linkPorts( "CmpFSec2.Sh_OA",        "ShL.Sh_ICmpFSec2"  ,       "MeCmpFSec2" );
linkPorts( "CmpL.Sh_OA" ,           "ShL.Sh_ICmp"       ,       "MeCmpL" );
linkPorts( "TrbL.Sh_O",             "ShL.Sh_ITrb"       ,       "MeTrbL" );


linkPorts( "CmpH.Sh_O",             "ShH.Sh_ICmp",              "MeCmpH" );
linkPorts( "TrbH.Sh_O",             "ShH.Sh_ITrb",              "MeTrbH" );


//------------------------------------------------------------------------
//                      Solver Sequence
//------------------------------------------------------------------------


// If solver execution sequence is not set by the user (below), the default sequence will be the order of the element instantiation above
/*
solver.executionSequence = {
                    "Amb",
                    "InletStart",
                    "Cmp"
}
*/
```

```
//----------------------------------------------------------------------
//                         Solver Variables
//----------------------------------------------------------------------


real T4_req = 2500; // Rankine, requested turbine inlet temp.  Used in solver dependent dep_T4
real Fn_req = 10000; // lbf, requested net thrust.  Used in solver dependent dep_Fn


real T4_max = 2700; // Rankine, requested turbine inlet temp.  Used in solver dependent dep_T4
real NmechL_max = 5200; // rpm, low pressure shaft max allowable speed
real NmechH_max = 11000; // rpm, high pressure shaft max allowable speed


// Declare a solver independent variable that varies fuel flow rate
Independent ind_Wfuel{
            varName = "FusEng.Wfuel";
}


// Declare a solver dependent variable that targets a specified turbine inlet temperature, T4 (F04.Tt = Turb.Fl_I.Tt),
Dependent dep_T4 {
        eq_lhs = "F040.Tt"; // model value
        eq_rhs = "T4_req"; // target value
}


// Declare a solver independent variable that varies engine air flow rate
Independent ind_Wair{
            varName = "FsEng.W_in";
}


// Declare a solver dependent variable that targets a specified engine net thrust, Perf.Fn
```

```
Dependent dep_Fn {
        eq_lhs = "Perf.Fn"; // model value
        eq_rhs = "Fn_req"; // target value
}

Dependent Target_DesignMixerEXTR1 { // this is the dependent name
eq_lhs = "FMixer.ExtractionRatio"; // this is the left side of an equality
eq_rhs = "1.02"; // this is the right side of an equality
}

Dependent Target_DesignMixerEXTR2 { // this is the dependent name
eq_lhs = "FMixerC.ExtractionRatio"; // this is the left side of an equality
eq_rhs = "1.02"; // this is the right side of an equality
}

Dependent dep_errPs {
    eq_lhs = "Fl_I1.Ps";
    eq_rhs = "Fl_I2.Ps";
    description = "Pressure balance error";
  }

//--------------------------------------------------------------------
//                          Solver Constraints
//--------------------------------------------------------------------

// Declare a solver dependent that limits the turbine inlet temperature
Dependent dep_T4_max {
        eq_lhs = "F040.Tt"; // variable to be constrained
        eq_rhs = "T4_max"; // set maximum allowable value
```

```
}

// Declare a solver dependent that limits the shaft speed
Dependent dep_NmechL_max {
        eq_lhs = "ShL.Nmech"; // variable to be constrained
        eq_rhs = "NmechL_max"; // set maximum allowable value
}

// Declare a solver dependent that limits the shaft speed
Dependent dep_NmechH_max {
        eq_lhs = "ShH.Nmech"; // variable to be constrained
        eq_rhs = "NmechH_max"; // set maximum allowable value
}

SecantSolver iterMN {
    description = "solver for exit total pressure";
    maxDx = 0.5;
    tolerance = 1.E-5;  // tolPs
    perturbSize = 0.01;
    maxIters = 50;
  }

 SecantSolver iterPt {
    description = "solver for exit total pressure";
    maxDx = 5;
    tolerance =1.E-5;  // tolPs
    perturbSize = 0.01;
    maxIters = 50;
  }
```

```
// Apply the dep_T4_max constraint to the dep_T4 dependent and set the constraint as an upper limit (MAX)

dep_Fn.addConstraint( "dep_T4_max", "MAX");

// Apply the dep_NmechL_max constraint to the dep_Fn dependent and set the constraint as an upper limit (MAX)

dep_Fn.addConstraint( "dep_NmechL_max", "MAX");

// Apply the dep_NmechH_max constraint to the dep_Fn dependent and set the constraint as an upper limit (MAX)

dep_Fn.addConstraint( "dep_NmechH_max", "MAX");
```

APPENDIX B

NPSS Code For Running Adaptive Cycle Engine

```
//   File Name: turbofan_example.run

//   Date: January 2010

//   Author: P. Johnson

//

//   Description: Model to demonstrate a twin spool turbofan

//

//

//------------------------------------------------------------------------



#include <InterpIncludes.ncp> // file contains unit names, socket types, error statements, and some constants


//------------------------------------------------------------------------

//                      Set Thermodynamic Package

//------------------------------------------------------------------------


setThermoPackage("GasTbl");  // air properties, developed by Pratt and Whitney

//setThermoPackage("Janaf"); // air properties, developed by Honeywell

//setThermoPackage("FPT"); // custom fluid property tables, developed by the user


//------------------------------------------------------------------------

//                      User-Defined Elements

//------------------------------------------------------------------------
```

```
//----------------------------------------------------------------------
//                  User-Defined Tables and Functions
//----------------------------------------------------------------------


// User-defined functions
// printResults() prints some model results to the command window
// setupDesign() configures the model to run in design mode
// setupOffDsign() configures the model to run in off-design mode
#include "user.fnc";


//----------------------------------------------------------------------
//                      User-Defined Variables
//----------------------------------------------------------------------




//----------------------------------------------------------------------
//                         Model Definition
//----------------------------------------------------------------------


// Load the that file contains the model definition (element instantiations, links, and solver variables)
#include "Turbofan.mdl";


// Load the CaseRowViewer file
#include "EngResultsRow.view";
// Save run results in output file EngDesign.txt
```

```
os_EngResultsRow.filename = "EngResults.txt"; // set the viewer output file name


//------------------------------------------------------------------------
//                      Running the Model
//------------------------------------------------------------------------
cout << endl;
cout << "======================================== \n";
cout << "=====    RUNNING DESIGN POINT     ===== \n";
cout << "======================================== \n";
cout << endl;


// Call function that sets up design mode
setupDesign();


//setOption("solutionMode", "ONE_PASS");


// Load the design point values
#include "Design.inp";


// Run the model
run();


// Print model results to the command window
printResults();
EngResultsRow.update();


/*
```

```
runMaxFnHook( 0., 0, 2, 21, 1.4, 3.3, 20, 1.2, 1.9, 8, 5, 24, 20 );
runMaxFnHook( 0.1, 0, 2, 21, 1.4, 3.3, 20, 1.2, 1.9, 8, 5, 24, 20 );
runMaxFnHook( 0.2, 0, 2, 21, 1.4, 3.3, 20, 1.2, 1.9, 8, 5, 24, 20 );
runMaxFnHook( 0.3, 0, 2, 21, 1.4, 3.3, 20, 1.2, 1.9, 8, 5, 24, 20 );
runMaxFnHook( 0.4, 0, 2, 21, 1.4, 3.3, 20, 1.2, 1.9, 8, 5, 24, 20 );
runMaxFnHook( 0.5, 0, 2, 21, 1.4, 3.3, 20, 1.2, 1.9, 8, 5, 24, 20 );
runMaxFnHook( 0.6, 0, 2, 21, 1.4, 3.3, 20, 1.2, 1.9, 8, 5, 24, 20 );
runMaxFnHook( 0.7, 0, 2, 21, 1.4, 3.3, 20, 1.2, 1.9, 8, 5, 24, 20 );
runMaxFnHook( 0.8, 0, 2, 21, 1.4, 3.3, 20, 1.2, 1.9, 8, 5, 24, 20 );
runMaxFnHook( 0.9, 0, 2, 21, 1.4, 3.3, 20, 1.2, 1.9, 8, 5, 24, 20 );
runMaxFnHook( 1.0, 0, 2, 21, 1.4, 3.3, 20, 1.2, 1.9, 8, 5, 24, 20 );
runMaxFnHook( 1.1, 0, 2, 21, 1.4, 3.3, 20, 1.2, 1.9, 8, 5, 24, 20 );
runMaxFnHook( 1.2, 0, 2, 21, 1.4, 3.3, 20, 1.2, 1.9, 8, 5, 24, 20 );
runMaxFnHook( 1.3, 0, 2, 21, 1.4, 3.3, 20, 1.2, 1.9, 8, 5, 24, 20 );
runMaxFnHook( 1.4, 0, 2, 21, 1.4, 3.3, 20, 1.2, 1.9, 8, 5, 24, 20 );
runMaxFnHook( 1.5, 0, 2, 21, 1.4, 3.3, 20, 1.2, 1.9, 8, 5, 24, 20 );
runMaxFnHook( 1.6, 0, 2, 21, 1.4, 3.3, 20, 1.2, 1.9, 8, 5, 24, 20 );
runMaxFnHook( 1.7, 0, 2, 21, 1.4, 3.3, 20, 1.2, 1.9, 8, 5, 24, 20 );
runMaxFnHook( 1.8, 0, 2, 21, 1.4, 3.3, 20, 1.2, 1.9, 8, 5, 24, 20 );
runMaxFnHook( 1.9, 0, 2, 21, 1.4, 3.3, 20, 1.2, 1.9, 8, 5, 24, 20 );
runMaxFnHook( 2.0, 0, 2, 21, 1.4, 3.3, 20, 1.2, 1.9, 8, 5, 24, 20 );
*/
cout << endl;
cout << "======================================= \n";
cout << "=====  RUNNING OFF DESIGN POINT   ===== \n";
cout << "======================================= \n";
cout << endl;
```

```
// Call function (located in file user.fnc) to setup off-design mode
setupOffDesign();

//#include "Off Design.inp";
// Engine constraints (used for off-design)
T4_max = 4200; // Rankine, maximum allowable turbine inlet temperature
NmechL_max = 5200; // rpm, low pressure shaft max allowable speed
NmechH_max = 11000; // rpm, high pressure shaft max allowable speed
SpltFan2.BPRdesA = 0.1; // bypass split ratio, bypass 1/ core
CmpFSec2.PRdesA = CmpFSec1.PR * 1.7; // fan OD 2 design pressure ratio
CmpL.PRdesA = CmpFSec2.PRdesA; // low pressure compressor design pressure ratio

// Set some off design operating values
Amb.alt_in = 500; // ft, altitude
Amb.MN_in = 0.8; // Mach number
//InEng.eRamBase = 0.9014;
//Fn_req = 4000; // lbm, requested thrust
T4_req = 3600;
//solver.addIndependent("SpltFan1.ind_BPR");

// Run the model
run();
printResults();
EngResultsRow.update();
*/
/*
cout << endl;
```

```
cout << "======================================= \n";
cout << "=====  RUNNING OFF DESIGN POINT   ===== \n";
cout << "======================================= \n";
cout << endl;


// Call function (located in file user.fnc) to setup off-design mode
setupOffDesign();

//#include "Off Design.inp";
// Engine constraints (used for off-design)
T4_max = 3800; // Rankine, maximum allowable turbine inlet temperature
NmechL_max = 5200; // rpm, low pressure shaft max allowable speed
NmechH_max = 11000; // rpm, high pressure shaft max allowable speed
SpltFan2.BPRdesA = 1.2; // bypass split ratio, bypass 1/ core
CmpFSec2.PRdesA = CmpFSec1.PR * 1.4; // fan OD 2 design pressure ratio
CmpL.PRdesA = CmpFSec2.PRdesA; // low pressure compressor design pressure ratio

// Set some off design operating values
Amb.alt_in = 35000; // ft, altitude
Amb.MN_in = 0.8; // Mach number
//InEng.eRamBase = 0.9014;
//Fn_req = 43000; // lbm, requested thrust
T4_req = 3600;
//solver.addIndependent("SpltFan1.ind_BPR");

// Run the model
```

```
run();
printResults();
EngResultsRow.update();
*/
/*
cout << endl;
cout << "======================================== \n";
cout << "=====  RUNNING OFF DESIGN POINT   ===== \n";
cout << "======================================== \n";
cout << endl;

// Call function (located in file user.fnc) to setup off-design mode
setupOffDesign();

//#include "Off Design.inp";
// Engine constraints (used for off-design)
T4_max = 3800; // Rankine, maximum allowable turbine inlet temperature
NmechL_max = 5200; // rpm, low pressure shaft max allowable speed
NmechH_max = 11000; // rpm, high pressure shaft max allowable speed
SpltFan2.BPRdesA = 0; // bypass split ratio, bypass 1/ core
CmpFSec2.PRdesA = CmpFSec1.PR * 1.6; // fan OD 2 design pressure ratio
CmpL.PRdesA = CmpFSec2.PRdesA; // low pressure compressor design pressure ratio

// Set some off design operating values
Amb.alt_in = 40000; // ft, altitude
Amb.MN_in = 2.2; // Mach number
//Fn_req = 50000; // lbm, requested thrust
```

```
InEng.eRamBase = 0.9014;

T4_req = 4000;


//solver.addIndependent("SpltFan1.ind_BPR");


// Run the model

run();


// Print model results to the command window

printResults();

// Update the viewer with the last case run

EngResultsRow.update();


//quit();

*/

//runMaxFnHook( real alfa2, real start_alfa1, real end_alfa1, int numPts )

//runMaxFnHook( 0, 0, 2, 20 );

/*

runMaxFnHook( 0.1, 0, 2, 20 );

runMaxFnHook( 0.2, 0, 2, 20 );

runMaxFnHook( 0.3, 0.0, 2, 20 );

runMaxFnHook( 0.4, 0.0, 2, 20 );

runMaxFnHook( 0.5, 0.0, 2, 20 );

runMaxFnHook( 0.6, 0.0, 2, 20 );

runMaxFnHook( 0.7, 0.0, 2, 20 );

runMaxFnHook( 0.8, 0.0, 2, 20 );

runMaxFnHook( 0.9, 0.0, 2, 20 );
```

```
runMaxFnHook( 1.0, 0.0, 2, 20 );
runMaxFnHook( 1.1, 0.0, 2, 20 );
runMaxFnHook( 1.3, 0.0, 2, 20 );
runMaxFnHook( 1.4, 0.0, 2, 20 );
runMaxFnHook( 1.5, 0.0, 2, 20 );
runMaxFnHook( 1.6, 0.0, 2, 20 );
runMaxFnHook( 1.7, 0.0, 2, 20 );
runMaxFnHook( 1.8, 0.0, 2, 20 );
runMaxFnHook( 1.9, 0.0, 2, 20 );
runMaxFnHook( 2.0, 0.0, 2, 20 );
*/
```

REFERENCES

1. Jack L. Kerrebrock., “Aircraft Engine and Gas Turbines”, the MIT press, 1992, second edition.

2. “GE details Sixth-Generation Adaptive Fighter Engine Plan”, Aviation week and Space Technology, February 2-15, 2015.

3. “GE Adaptive Cycle Engine”, Aviation week and Space Technology, [online database], URL: http://aviationweek.com/GEadaptivecycle [cited Jan 16, 2015].

4. “GE aviation engine sets record in latest test”, The Business Journal, [online database], URL:http://www.bizjournals.com/cincinnati/news/2013/07/29/ge-aviation-engine-sets-record-in.html [cited Jul 29, 2013].

5. “The ADVENT of a better Jet engine”, Defense Industry Daily, Defense program acquisition news, budget data, market briefing, [online database], URL: http://www.defenseindustrydaily.com/the-advent-of-a-better-jet-engine-03623/ [cited June 25, 2015].

6. “2015 Aviation Week Program Excellence Initiative”, Program: Adaptive Cycle Engine.

7. “NPSS Users Guide”, Wolverine Ventures, software release 2.6.1, revision 4, December 20, 2013.

8. Daniel Francis Waters, “modeling of gas turbine-solid oxide fuel cell system for combined propulsion and power an aircraft”, University of Maryland, dissertation 2015.

9. Mattingly, J. D. and Ohain, H Van, “Elements of Propulsion: Gas Turbine and Rockets”, AIAA Virginia, 2006.

10. Schott M, Jones "Steady-State Modeling of Gas Turbine Engines Using The Numerical Propulsion System Simulation Code," *ASME Turbo Expo 2010: Power for Land, Sea and Air*, ASME, Glasgow, UK, June 14-18, 2010.
11. "NPSS User Guide Reference Sheets", Software release 2.6.2, December 17, 2013.
12. R. Douglas Archer and Maido Saarlas, "An Introduction to Aerospace Propulsion", Prentice Hall College Div; Facsimilie edition, March 20, 1998.
13. Sissenwine, N., Dubin, M., and Wexler, H., "The U.S. Standard Atmosphere, 1962," *Journal of Geophysical Research*, Vol. 67, No. 9, 1962, pp. 3627–3630.
14. "Climatic Information to Determine Design and Test Requirements for Military Equipment," U.S. Department of Defense, Washington, DC, MIL-STD-210A, 1958.
15. "Global Climatic Data for Developing Military Products," U.S. Department of Defense, Washington, DC, MIL-HDBK-310, 1997.
16. George P. Sutton and Oscar Biblard, "Rocket Propulsion Elements".
17. Hatim Soeb Rangwala, "Simulation of Ejector Nozzle In A Low-Bypass Turbofan Engine Using NPSS", University of Texas at Arlington, May 2015.
18. US study initiative, [online database], URL: http://www.ustudy.in/node/5110
19. Technology Section Bilingual English, Turbojet, [online database], URL: http://www.petervaldivia.com/turbo-jet/
20. Schematic diagram of the operation of a axial flow turbojet engine, [online database], source: Turbojet_operation-_axial_flow.png