Air Force Institute of Technology

AFIT Scholar

Theses and Dissertations | Student Graduate Works

3-15-2007

Comparative Analysis of a High Bypass Turbofan using a Pulsed Detonation Combustor

Ionio Q. Andrus

Follow this and additional works at: https://scholar.afit.edu/etd

Part of the Propulsion and Power Commons

Recommended Citation

Andrus, Ionio Q., "Comparative Analysis of a High Bypass Turbofan using a Pulsed Detonation Combustor" (2007). *Theses and Dissertations*. 2957.
https://scholar.afit.edu/etd/2957

This Thesis is brought to you for free and open access by the Student Graduate Works at AFIT Scholar. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AFIT Scholar. For more information, please contact AFIT.ENWL.Repository@us.af.mil.

Comparative Analysis of a High Bypass Turbofan Using a Pulsed Detonation Combustor

THESIS

Ionio Q. Andrus, Captain, USAF

AFIT/GAE/ENY/07-M02

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GAE/ENY/07-M02

Comparative Analysis of a High Bypass Turbofan Using a Pulsed Detonation Combustor

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Aeronautical Engineering

Ionio Q. Andrus, B.S.M.E.

Captain, USAF

March 2007

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT/GAE/ENY/07-M02

Comparative Analysis of a High Bypass Turbofan Using a Pulsed Detonation Combustor

Ionio Q. Andrus, B.S.M.E.
Captain, USAF

Approved:

/signed/	15 Mar 2007
Dr. Paul I. King (Chairman)	date
/signed/	15 Mar 2007
Dr. Milton E. Franke (Member)	date
/signed/	15 Mar 2007
Dr. Michael L. Heil (Member)	date

Abstract

It has been proposed that the implementation of a pulsed detonation combustor in a high-bypass turbofan engine would result in an engine that is both more efficient and more reliable. The validity of the performance claims are evaluated based on a comparison between the baseline and hybrid turbofans as modeled in the Numerical Propulsion System Simulation (NPSS). The engine cycle of the baseline high-bypass turbofan is evaluated and compared using both the Aircraft Engine Design System (AEDsys) and NPSS programs. The baseline engine agreed to within 1% on the net thrust calculation and specific fuel consumption between the two programs. Differences are traceable to the variation in specific heat models and to different methods of calculating temperature across the turbine and compressor components. The hybrid pulsed detonation engine model shares a common architecture with the baseline turbofan model. Inlet mass flow and core mass flow are maintained, but the combustor of the baseline engine is replaced with a pulsed detonation combustor for the hybrid engine. The effect of detonation on the core air flow is calculated using a closed form solution of the Chapman-Jouguet Mach number with a total energy correction applied. Cycle time is calculated to provide a reasonable estimate of frequency for the user input geometry. Effects of sub-component design choices within the pulsed detonation combustor are evaluated using simple parametric studies. These studies are used to select an optimal architecture for the combustor. The effects of detonation are accounted for by applying pressure and temperature losses to the fluid exiting the combustor. A parametric study was performed to demonstrate what level of loss continues to yield a more efficient engine. Results show improvement in thrust specific fuel consumption with careful selection of combustor design parameters. There is a definite level of acceptable loss that if surpassed makes pulsed detonation combustion a good candidate for inclusion into a hybrid turbofan engine.

Acknowledgements

This thesis represents more than one man's work. Their names may not appear on the cover page, but it belongs as much to those who have helped as to me.

To Dr. King goes a special note of thanks. You provided the quiet and consistent guidance needed to move this work forward. Thank you for the hours of your time spent listening and questioning.

Many professors and teachers have contributed to my education. From Mrs. Black who taught me phonics in Kindergarten to the professors at AFIT, one concept at a time, motivating one person to go a little farther.

Dr. Kuprowicz and Dr. Schauer who spend their days cultivating a technology that may some day revolutionize aircraft propulsion.

Thanks to the ENY-2 lunch crowd who kept the 18 months in the proper Air Force frame of reference.

I shall not ignore the magnificent support from the NPSS team at NASA/GRC. This work would not be possible without them.

And to The Almighty, who I believe has guided me in this path thus far. To Him I owe the greatest degree of gratitude, without Him I would not be.

My children have a debt of gratitude. You sacrificed time with your Dad, weekend trips to the zoo and museum, bedtime stories, and one-on-one time so I could do homework and a thesis. Only time will tell if the sacrifice you made in these early years will give you more of the father that you need.

Thank you Mrs. Andrus; no man could ask for a better wife. You've been willing to sacrifice your time with me, your health, your rest, and your spare time so that I could work on this. Soon your husband will be able to spend his evenings with you instead of his books.

Ionio Q. Andrus

Table of Contents

Page

Abstract iv

Acknowledgements v

List of Figures viii

List of Tables x

List of Symbols xi

List of Abbreviations xiv

I. Introduction 1
- 1.1 Purpose 1
- 1.2 Procedure 2
  - 1.2.1 Turbofan Comparison in NPSS and AEDsys . . 2
  - 1.2.2 Comparison of PDC Hybrid with Baseline Turbofan 2

II. Prior Work on Pulsed Detonation Combustors 4
- 2.1 Thermodynamics of Pulsed Detonation Engines 4
  - 2.1.1 Chapman Jouguet Detonation 4
  - 2.1.2 The Zeldovich-Von Neumann- Döring and Multi-Dimensional Detonation Models 6
  - 2.1.3 Thermodynamic Performance Models 8
- 2.2 Pulsed Detonation Combustor Models 10
  - 2.2.1 Pulsed Detonation Combustor Performance Maps 10
  - 2.2.2 Exhaust into Turbines 11
  - 2.2.3 Converting Pulsed Flow into Steady State Flow 12

III. Baseline and Hybrid Models 14
- 3.1 Baseline High Bypass Turbofan 14
  - 3.1.1 Baseline Engine Configuration 16
- 3.2 Baseline Model Similarity Between AEDsys and NPSS . 18
  - 3.2.1 Thermodynamic Models 23
- 3.3 Hybrid Turbofan - Pulsed Detonation Combustor Engine 31
  - 3.3.1 Turbofan Configuration 31
  - 3.3.2 Pulsed Detonation Combustor 33

Page

IV. Results . . . 51
4.1 Comparison of Baseline Turbofan . . . 51
4.1.1 Baseline Engine On-Design Results . . . 51
4.1.2 Baseline Turbofan Comparison Off-Design . . . 51
4.1.3 Explanation of Differences Between AEDsys and NPSS . . . 57
4.2 PDC Hybrid Results . . . 61
4.2.1 Sizing the PDC . . . 62
4.2.2 Effects of the Transition to Steady State . . . 74

V. Conclusions . . . 80
5.1 Baseline Engine Comparison . . . 80
5.1.1 On-Design Comparison . . . 80
5.1.2 Off-Design Comparison . . . 80
5.2 Hybrid Engine Performance . . . 80
5.2.1 Geometric Constraints . . . 81
5.2.2 Promise of a More Efficient Engine . . . 81
5.3 Recommendations . . . 81

Appendix A. NPSS Model Code . . . 83

Appendix B. Fortran unFAIR Code . . . 96

Appendix C. Pulsed Detonation Combustor Code . . . 99

Bibliography . . . 121

Vita . . . 125

Index . . . 126

List of Figures

Figure		Page
2.1.	Hugoniot Curve	5
2.2.	Diagram of the Detonation Wave Effects	6
2.3.	Diagram of the Von-Neumann Spike	7
2.4.	Temperature-Entropy Diagram of the PDE Cycle	9
3.1.	High Bypass Stations.	15
3.2.	High Bypass Flow Chart.	15
3.3.	Ramifications of Specific Heat Model	24
3.4.	Specific Heat Differences - Pure Air	27
3.5.	Enthalpy Differences - Pure Air	28
3.6.	Enthalpy Differences - Pure Air, 1atm - Detail	28
3.7.	Specific Heat Differences - FAR = 0.0253	29
3.8.	Enthalpy Differences - FAR = 0.0253	30
3.9.	Specific Heat Differences - FAR = 0.0676	31
3.10.	Enthalpy Differences - FAR = 0.0676	32
3.11.	Hybrid Engine Configuration	32
3.12.	Pulsed Detonation Combustor Configuration	34
3.13.	Pulsed Detonation Combustor Configuration	34
3.14.	Calculated Chapman-Jouguet Mach Number	37
3.15.	Entropy Error Affecting Pressure and Temperature	38
3.16.	Calculated Chapman-Jouguet Mach Number at High Pressure	38
3.17.	Temperature-Entropy Diagram of the PDE Cycle	39
3.18.	Calculated Pressure Ratio Behind Shock	40
3.19.	Calculated Temperature Ratio Behind Shock	40
3.20.	Blowdown Time vs Pressure Ratio	46
4.1.	Throttle Hook for Baseline Engine	56

Figure Page

4.2. Erroneous Throttle Hook Using janaf 56

4.3. Low Pressure Spool Efficiencies Off-Design 61

4.4. High Pressure Spool Efficiencies Off-Design 62

4.5. Effect of Tube Diameter on Hybrid Performance 65

4.6. Effect of Tube Length on Hybrid Performance 66

4.7. Effect of Number of Tubes on Hybrid Performance 68

4.8. Effect of Valve Inlet Area Ratio on Hybrid Performance 69

4.9. Effect of Valve Inlet Mach Number on Hybrid Performance . . 70

4.10. Effect of Compressor Pressure Ratio on Hybrid Performance . . 72

4.11. Effect of Mixer Inlet Mach Number on Hybrid Performance . . 73

4.12. Fill/Purge and Equivalence Ratio Carpet Plot: TSFC 73

4.13. Fill/Purge and Equivalence Ratio Carpet Plot: Fn 74

4.14. Transition to Steady State Effects: TSFC 75

4.15. Transition to Steady State Effects: Fn 78

List of Tables

Table		Page
3.1.	Baseline Engine Parameters	14
3.2.	AEDsys Input - Reference Conditions	16
3.3.	AEDsys Input - Thermodynamics and Efficiencies	17
3.4.	AEDsys Input - Constraints	18
3.5.	AEDsys and NPSS Input Differences	19
3.6.	Coefficients Used for the FAIR Subroutine	25
4.1.	Baseline Engine On-Design AEDsys Engine Test Output	52
4.2.	Baseline Engine On-Design NPSS Engine Test Output	53
4.3.	Baseline Engine On-design Component Interface AEDsys	54
4.4.	Baseline Engine On-Design Component Interface Using NPSS	55
4.5.	Power Differences Due to Temperature Calculations for Baseline Engines	58
4.6.	Power Differences for Baseline Engines Based on Specific Heat Model Variations	59
4.7.	Comparison of Thrust After Power Extraction - GasTbl	60
4.8.	Comparison of Thrust After Power Extraction - janaf	60
4.9.	Initial Hybrid PDC Configurations	63
4.10.	Improved PDC Configuration	75
4.11.	Baseline Engine On-Design Component Interface Using NPSS	76
4.12.	Hybrid Engine On-Design Component Interface Using NPSS	77

List of Symbols

Symbol		Page
ρ	Density	4
p	Pressure	4
γ	Ratio of Specific Heats	4
q	Heat Flux	4
h°	Heat of Reaction	4
u_{CJ}	Chapman-Jouguet Detonation Wave Velocity	4
FAR	Fuel-to-Air Ratio	4
h_{pr}	Fuel Lower Heating Value	4
u	Fluid Velocity	5
$\dot{m}$	Mass Flow Rate	5
a	Speed of Sound in a Fluid	6
$\tilde{q}$	non-dimensional Heat Addition	8
C_P	Constant Pressure Specific Heat	8
f	Mass Fuel-to-Air Ratio	8
h_{pr}	Fuel Lower Heating Value	8
Ψ	Non-dimensional Temperature Ratio	8
M_{CJ}	Chapman-Jouguet Mach Number	8
S	Entropy	8
η_b	Burning Efficiency	8
π_b	Burner Pressure Ratio	8
η_N	nozzle efficiency	9
C_v	Nozzle Gross Thrust Coefficient	10
h_0	Enthalpy at Beginning of Cycle	10
q_{add}	Heat Flux Into System	10
Fn	Net Thrust	14

Symbol		Page
BPR	Bypass Ratio	14
T_{t4}	Total Temperature at Turbine Inlet	18
$\frac{\Delta P}{P}$	Unscaled Normalized Pressure Drop	18
CTOL	Low Pressure Spool Power Take-off Coefficient	19
CTOH	High Pressure Spool Power Take-off Coefficient	19
π_{dmax}	Maximum Inlet Pressure Ratio $\left(\frac{P_{t2}}{P_{t1}}\right)$	19
π_b	Burner Pressure Ratio$\left(\frac{P_{t4}}{P_{t3}}\right)$	19
π_n	Main Nozzle Inlet Pressure Ratio$\left(\frac{P_{t9}}{P_{t7}}\right)$	19
π_{nf}	Fan Nozzle Pressure Ratio $\left(\frac{P_{t19}}{P_{t17}}\right)$	19
η_c	Compressor Adiabatic Efficiency	20
η_{tb}	Turbine Adiabatic Efficiency	20
e_c	Compressor Polytropic Efficiency	20
e_{tb}	Turbine Polytropic Efficiency	20
$\dot{W}$	Work Rate or Power	20
P_r	Reduced Pressure	21
π_c	Compressor Total Pressure Ratio	21
h	Enthalpy	21
π_{tb}	Total Pressure Ratio across the Turbine	21
τ_{tb}	Total Temperature Ratio Across the Turbine	21
h_{tbexit}	Turbine Exit Enthalpy	21
P_t	Total Pressure	22
ϕ	Entropy Function	25
Δh	Difference in Enthalpy Between Two Models	29
pf	Purge Fraction	41
ff	Fill Fraction	41
A_{tube}	Detonation Tube Cross-Sectional Area	42
l_{tube}	Tube Length	42

Symbol		Page
$V_{purge\,air}$	Purge Air Volume Per Cycle	42
$V_{fuel-air\,mixture}$	Fuel-Air Mixture Volume per Cycle	42
m_{purge}	Purge Air Mass	42
$m_{fuel-air\,mixture}$	Fuel-Air Mass for Fill	42
ρ_t	Total Density	42
$m_{fill\,air}$	Mass of Pure Air Mixed with Fuel During Fill	43
m_{fuel}	Mass of Fuel Used for One Fill Cycle	43
$\dot{m}_{air\,total}$	Mass Flow Rate of Air Through Detonation Tubes	43
f	Filling or Cycle Frequency	43
$\dot{m}_{vo}$	Maximum Mass Flow Rate when Detonation Valve is Open	43
MFP	Mass Flow Parameter	43
τ_{vo}	Valve Open Time Fraction $\frac{t_{valve\,open}}{t_{cycle}}$	44
t_{cycle}	Cycle Time	44
DDT	Deflagration to Detonation Time	44
$t_{Blowdown}$	Blowdown Time	45
t_{purge}	Purge Time	46
t_{fill}	Fill Time	46
$\dot{m}_{iBPR}$	Air Mass Flow Rate Through the Internal Bypass Duct	47
$\dot{m}_{core\,air}$	Mass Flow Rate Through Detonation Tubes and Internal Bypass	47
Fn	Net Thrust	67
TSFC	Thrust Specific Fuel Consumption	67

List of Abbreviations

Abbreviation		Page
PDE	Pulsed Detonation Engine	1
PDC	Pulsed Detonation combustor	1
SFC	Specific Fuel Consumption	1
NPSS	Numerical Propulsion System Simulation	2
AEDsys	Aircraft Engine Design System	2
CJ	Chapman-Jouguet	4
ZND	Zeldovich, Von Neumann, and Döring	6
CVCCE	Constant Volume Combustion Cycle Engine	10
TIT	Turbine Inlet Temperature	14
FPR	Fan Pressure Ratio	14
OPR	Overall Pressure Ratio	14
PR	Pressure Ratio	16
LPC	Low Pressure Compressor	17
HPC	High Pressure Compressor	17
HPT	Low Pressure Turbine	17
LPT	High Pressure Turbine	17
LP	Low Pressure	17
HP	High Pressure	17
RPM	Revolutions Per Minute	18
PTO	Power Take Off	19
VSH	Variable Specific Heats	20
FAR	Mass Fuel-to-Air Ratio	29
iBPR	Internal Bypass Ratio	47
TSFC	Thrust-Specific Fuel Consumption	51
NIST	National Institute of Standards and Technology	60

Comparative Analysis of a High Bypass Turbofan Using a Pulsed Detonation Combustor

I. Introduction

A significant effort has been made in the past ten years to develop pulsed detonation engines (PDE) as a means of aircraft propulsion. Detonation combustion holds the promise of a more efficient engine due to the simultaneous pressure increase and combustion. Additionally, fewer moving parts mean lower maintenance costs in association with this engine.

Efficiency is the impetus for development, but PDEs still face several key technical hurdles. Noise is a significant issue for any aviation engine, and *detonation* creates more of it than previous aircraft engines.

It is hoped that a marriage of the PDE with traditional turbomachinery will result in an engine that is more efficient than turbine engines and quieter than a pure PDE. The engine configuration of interest to this work involves replacing the burner or high pressure spool of a turbofan with a pulsed detonation combustor (PDC). The studies of Petters and Felder [1] and Smith et al. [2] have indicated that specific fuel consumption (SFC) might be improved by 5 to 15% with the inclusion of a PDC. These claims are significant, and they require explanation through some sort of thermodynamic modeling.

1.1 Purpose

This thesis was generated in order to better understand how PDC technology will impact future engines. Specifically, the objective of this thesis was to generate a model for a PDC-turbofan hybrid engine, and to compare the hybrid model to a conventional (or "baseline") turbofan engine. Pursuing the hybrid engine cycle only

makes sense if the comparison demonstrates an increase in propulsive efficiency. Thus, the goal of the comparison is to test the claims that a hybrid engine cycle is more efficient than the baseline technology.

1.2 Procedure

Two steps were taken to ensure results demonstrated only the change in combustion technology. First, the baseline high bypass turbofan with separate exhaust streams was modeled in two programs to corroborate the engine cycle solutions. Then a PDC hybrid turbofan was modeled and compared with the baseline engine.

1.2.1 Turbofan Comparison in NPSS and AEDsys. The baseline high bypass turbofan engine was modeled in the Numerical Propulsion System Simulation (NPSS)and in the Aircraft Engine Design system (AEDsys). The engine cycle calculation result comparison assured us that the engine configuration was identical between the two programs and acceptable to the propulsion community.

Each piece of software was selected for a different reason. In order to allow the models generated for this thesis to be useful to the sponsor (AFRL/PRTA), NPSS was mandated for the both the baseline and hybrid engines. NPSS was developed within the cooperative framework established between industry and NASA, and is becoming an industry standard. AEDsys was selected as the second piece of software because it is easy to use, available, and well documented. AEDsys was developed by Mattingly [3] for use in preliminary and academic situations. The comparison was not intended to demonstrate the accuracy of either code, nor to provide a qualitative judgment of either code. Instead, the intent was to ensure that the results generated by the model constructed in NPSS are reasonably correct.

1.2.2 Comparison of PDC Hybrid with Baseline Turbofan. Once the baseline engine was well-established the thermodynamic performance of a PDC was coded into an NPSS element based on the work of on the work of Heiser and Pratt [4] and

Dyer and Kaemming [5]. This NPSS element was then called by the NPSS hybrid engine model to perform cycle analysis calculations. This hybrid model required some minor configuration changes to the baseline engine's high pressure section in order to accommodate the new element.

An acceptable pulsed detonation configuration had to be identified before the baseline and hybrid engines could be compared. This configuration was determined through parametric studies which showed how various PDC parameters affected the overall engine performance. Once an acceptable PDC configuration was identified, a component impulse analysis was performed using NPSS for both the hybrid and baseline engines at design conditions. The effects of the unsteady and noisy flow generated by the pulsed detonation in the hybrid cycle performance was then simulated through the application of pressure and temperature losses to the flow exiting the combustor.

The results indicate that a hybrid engine can be more efficient than a conventional high-bypass turbofan engine. Care must be taken to prevent significant total pressure loss as the flow is transitioned to a steady state in order to see the improvement. Detonation tube fill time has a significant impact on the density of the engine - shorter tube fill times will result in fewer tubes being required to provide the same thrust.

II. Prior Work on Pulsed Detonation Combustors

A significant amount of work has been performed to model pulsed detonation combustors. This chapter reviews some of the detonation theories, detonation engine performance work, PDC models, and the results generated from those models. The models are central to the pulsed detonation process, which in turn is central to the hybrid engine cycle.

2.1 Thermodynamics of Pulsed Detonation Engines

The models seeking to mimic actual pulsed detonation are based on the underlying thermodynamics. Both Kuo [6] and Glassman [7, 221-265] have good explanations of two basic models of detonation summarized in sections 2.1.1 and 2.1.2.

2.1.1 Chapman Jouguet Detonation. Only a few brief points on Chapman-Jouguet (CJ) theory are given here since Kuo [6] provides a detailed description and derivation. CJ theory builds on the (Rankine-) Hugoniot relation defined by:

$$\frac{\gamma}{\gamma-1}\left(\frac{p_2}{\rho_2}-\frac{p_1}{\rho_1}\right)-\frac{1}{2}\left(p_2-p_1\right)\left(\frac{1}{\rho_1}+\frac{1}{\rho_2}\right)=q \tag{2.1}$$

where ρ is density, p is pressure, and γ. When exit pressure p_2 is plotted as a function of specific volume $\frac{1}{\rho_2}$ for a constant heat flux (q) value, the result is the Hugoniot curve. Recall that q is defined as

$$q = h_1^\circ - h_2^\circ \tag{2.2}$$

where h° is the heat of reaction. So we can control the offset of the Hugoniot curve (and CJ wave velocity u_{CJ}) by changing the q value through fuel-to-air ratio (FAR) or fuel lower heating value (h_{pr}).

The Hugoniot curve is traditionally broken into five regions, as shown in Figure 2.1. Consider the Rayleigh line equation in region V where both density (ρ) and

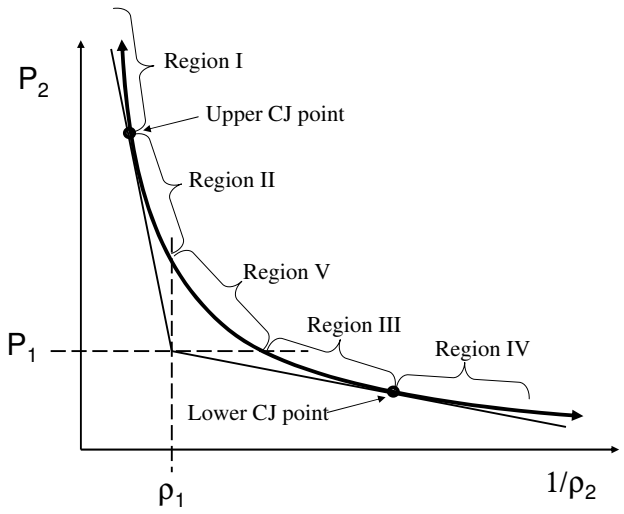


Figure 2.1: Diagram of Hugoniot curve. Combustion is generally only observed in region III and at the upper Chapman-Jouguet points.

pressure (p) at station 2 are greater than at station 1:

$$\rho_1^2 u_1^2 = \frac{p_2 - p_1}{\frac{1}{\rho_1} - \frac{1}{\rho_2}} = \dot{m}^2 \tag{2.3}$$

For these conditions, equation 2.3 implies an imaginary solution for the fluid velocity (u) or mass flow rate ($\dot{m}$). Region I is rarely observed, requiring an overdriven shock. Region II is likewise rarely observed, requiring a fast-reacting mixture. Region IV will require that the gas be accelerated transonically through the deflagration wave and is therefore defined as forbidden. The lower CJ point bordering region IV is also not seen experimentally and is therefore excluded. This leaves only region III - referred to as weak-deflagration - and the upper Chapman-Jouguet point. Since detonation is the region of interest to this work we will ignore deflagration and focus on the upper CJ point.

The Chapman-Jouguet points are located where the following equality holds true:

$$\frac{p_2 - p_1}{\frac{1}{\rho_2} - \frac{1}{\rho_1}} = -\gamma \rho_2 p_2 \tag{2.4}$$

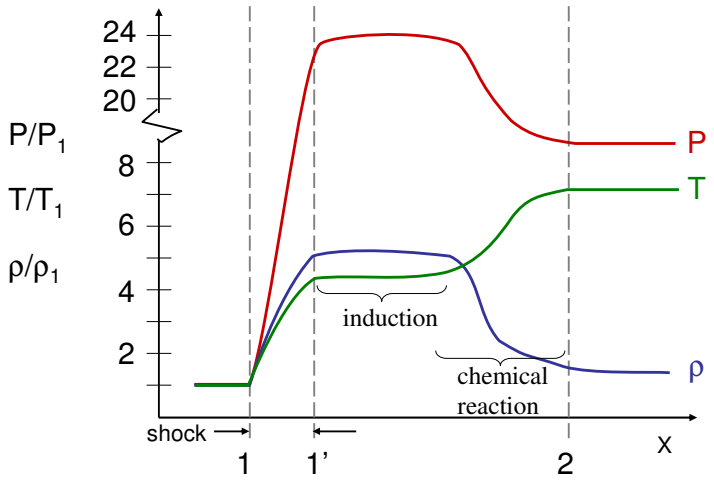


Figure 2.2: Diagram of the 1-dimensional detonation model used for ZND detonation theory.

This equality can be combined with equation 2.3 to yield equation 2.5.

$$u_2^2 = \frac{\gamma p_2}{\rho_2} = a_2^2 \; or \; |u_2| = a_2 \tag{2.5}$$

where a is the speed of sound in the fluid. This implies that at the CJ points, $M_2 = 1$. The upper CJ point corresponds to the minimum detonation wave speed. Since commonly observed detonation corresponds to the upper CJ point, we seek to obtain the corresponding velocity. There are several methods of doing this that are iterative in nature, but a closed-form solution has also been obtained and will be discussed in section 2.1.3.

2.1.2 The Zeldovich-Von Neumann- Döring and Multi-Dimensional Detonation Models. Zeldovich, Von Neumann, and Döring (ZND) independently made the same assumption that detonation flow is one dimensional. The result is the ZND model of detonation, extending the Chapman-Jouguet theory of detonation. Their model treats the detonation as a shock wave propagating into the quiescent mixture that pre-heats the reactants to a level that allows the ensuing combustive reaction to travel at the speed of the shock. This is easily seen in Figure 2.2. As the unreacted

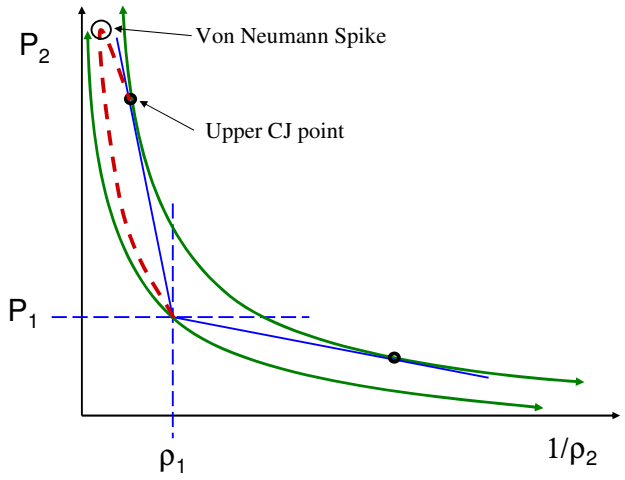


Figure 2.3: As the detonation initiates, it drives the fluid properties along the dashed path, causing a pressure spike first seen by von Neumann.

gas ($q = 0$) passes through the shock, the pressure rise drives it along the Hugoniot curve to the point described in Figure 2.3 as the von Neumann spike. It is at this point that combustion begins to occur, releasing heat and moving the fluid to a new Hugoniot curve with a different value of q. Height of the spike depends on the rate of chemical kinetics, with faster chemical interactions resulting in a smaller spike. If the spike is not high enough, or non-existent, the detonation may not have enough power to sustain itself.

These models demonstrate the underlying effects of detonation , but the reality is always more complex. Detonation occurs in three dimensions, and is a complex interaction where pressure, temperature, and density vary in all directions as shown in Kuo [6] and Strehlow [8]. Detonations are cellular in nature, so the geometry of the detonation tube can have a dramatic impact on the ability to reliably detonate reacting mixtures. Indeed, for a cylindrical tube, the detonation wave prefers to travel in a spiral whose pitch for the leading transverse wave is defined by the first helical acoustic mode.

2.1.3 Thermodynamic Performance Models. Heiser and Pratt [4] presented a simplified model for thermodynamic performance calculation of a PDE. Their method is based on solving for the Chapman-Jouguet Mach number of the detonation wave flowing into the quiescent mixture. The CJ Mach number is then used to calculate entropy and pressure rise across the detonation volume. With these properties, they can then calculate all fluid properties of the fluid exiting the PDE. The closed form algebraic solutions developed by Shapiro and others [8–10] are summarized here.

First, a non-dimensional heat addition term is calculated

$$\tilde{q} \equiv \frac{q_{supplied}}{C_p T_0} = \frac{f h_{pr}}{C_p T_0} \tag{2.6}$$

where $\tilde{q}$ is the non-dimensional heat addition, C_P, f is the mass fuel-air ratio, and h_{pr} is the lower heating value of the fuel. Having calculated $\tilde{q}$ it is possible to calculate the Chapman-Jouguet Mach number using equation 4 from Heiser and Pratt [4]:

$$M_{CJ}^2 = (\gamma+1)\left(\frac{\tilde{q}}{\Psi}\right) + 1 + \sqrt{\left[(\gamma+1)\left(\frac{\tilde{q}}{\Psi}\right)+1\right]^2 - 1} \tag{2.7}$$

where Ψ equal $\frac{T_3}{T_0}$, the ratio of the compressed static temperature to the free-stream static temperature. Having calculated the Chapman-Jouguet Mach number (M_{CJ}) it is possible to calculate the entropy S generated using equation 5 from Heiser and Pratt [4]:

$$\frac{s_4 - s_3}{C_p} = -\ln\left[M_{CJ}^2\left(\frac{\gamma+1}{1+\gamma M_{CJ}^2}\right)^{\frac{\gamma+1}{\gamma}}\right] \tag{2.8}$$

For a real PDE cycle, the non-dimensional heat addition term, $\tilde{q}$, is multiplied by a burning efficiency (η_b) and a pressure ratio (π_b) is calculated across the tubes using equation 18 from Heiser and Pratt [4]

$$\frac{p_4}{p_0} = \frac{1+\gamma M_{CJ}^2}{\gamma+1}\frac{p_3}{p_0} \geq 1 \tag{2.9}$$

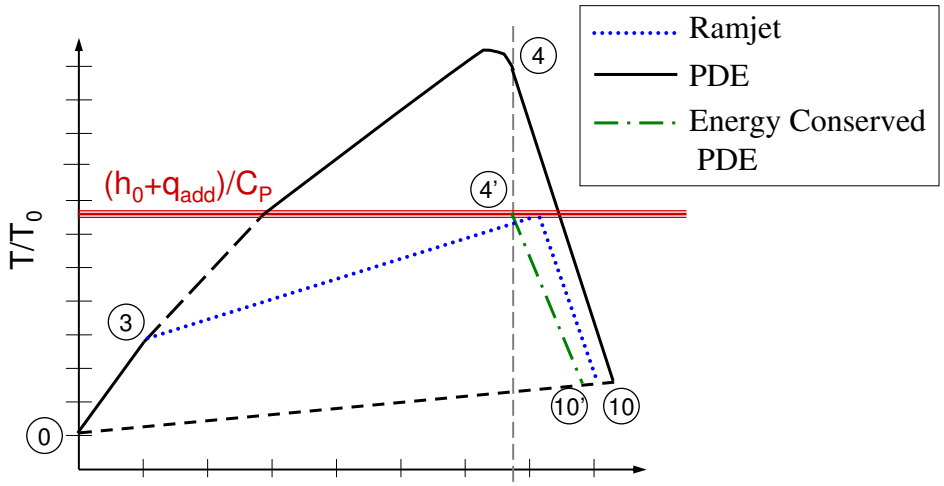


Figure 2.4: The T-S diagram shown in Dyer and Kaemming paper [5, fig 10] with the proposed correction to the PDE thermodynamic cycle. Conservation of energy will prevent incorrect calculation of entropy as the fluid exits the engine.

With both the entropy and pressure of the fluid post-combustion known it is possible to solve for all the other fluid properties through look-up tables or linear approximations.

Harris et al. [11] faulted the method of Heiser and Pratt [4] as overly optimistic because it neglects the fill portion of the cycle. Neglecting the fill portion does not invalidate the solutions obtained for pressure and entropy, but it does change the time averaged value of the properties. There is, however, a more significant drawback to using the cycle solution proposed by Heiser and Pratt [4]: energy is not conserved.

Dyer and Kaemming [5] proposed a different thermodynamic performance estimation and compared their results with ramjet thermodynamic performance. Their work is very similar to the work of Heiser and Pratt [4], but differs in how it accesses the fluid properties at station 4 (detonation tube exit). Figure 2.4 shows a temperature-entropy diagram of three cycles: a ramjet cycle, the PDE cycle as proposed by Heiser and Pratt, and the energy conserved PDE cycle. They realized that using a nozzle efficiency (η_N) accesses the kinetic energy at station 4, an energy level that is not truly available since the velocity will be "paid back" when gasses expand

back to static conditions. Instead, they use the nozzle gross thrust coefficient (C_v) that utilizes the entropy at tube exit and ambient pressure - not the energy. The result is that less entropy is calculated between generation from tube exit (station 4) and system exit (station 10). They propose that an accurate representation of the available energy is to use the known CJ entropy as calculated by Heiser and Pratt [4] coupled with the known system enthalpy calculated as ($h_0 + q_{add}$) where h_0 is the enthalpy at the beginning of the cycle and q_{add} is the heat flux into the system.

2.2 Pulsed Detonation Combustor Models

Embedding pulsed detonation within a turbine engine has been examined by groups from NASA [1,12,13], G.E. Global Research [14–20], the U.S. Air Force [4,21], and Indiana University and Liberty Works [2,22]. The results of this work have generated hope that a marriage between pulsed detonation and turbomachinery will result in a more efficient engine that is simultaneously mechanically simpler. Work initiated at NASA has been moved to industry through the Constant Volume Combustion Cycle Engine program (CVCCE)

2.2.1 Pulsed Detonation Combustor Performance Maps. Work done at General Electric Global Research [17] under the CVCCE program implemented a CFD code to achieve a converged pulse detonation cycle that described fluid elements and cycle timing. CFD results were ported into a performance map as described by Paxson [13]. This approach uses the CFD cycle results for inlet temperature, fraction of time that the flow contains a reacting mixture, and mixture stoichiometry to calculate PDC performance. The map returns total enthalpy and total pressure ratios given the three input parameters of 1) fill fraction 2) purge fraction and 3)non-dimensional heat of reaction (q_0). There were some minor modifications to Paxson's [13, 23] formulae, but the process remained unchanged. This method was nearly selected for implementation in this work because of the ability to base a performance map on experimental data, but time limitations drove the work in a different direction.

The CFD simulation was a 1-D MacCormack predictor-corrector time marching algorithm, second order accurate in time and space. A 1-D simulation does not capture all the effects of inlet valves and combustor exhaust, however, it does provide information about change in fluid properties at combustor exit. Also, cycle time calculations that provided accurate detonation frequency and mass flow rates. Significantly, a parametric study to demonstrate the effect of constant specific heats was performed and included in the report by Tangirala et al. [17]. It demonstrated that maintaining a constant specific heat through a PDC analysis results in a flawed prediction of total pressure ratios and cycle times.

Additionally, the stated geometry of the detonation tube used for the PDC was not realistic: its length was only 3 diameters long. This is not long enough to allow for a detonation cycle to occur. Although not stated, the assumption is that since the model is one dimensional only the length of the tube, and total cross-sectional area are important to find a solution.

2.2.2 Exhaust into Turbines. One major hurdle to the implementation of a PDC into hybrid engines is the interaction between the cyclic pulsation of the detonations and the flow through the turbines. In order to maintain the efficiency of the baseline engine, a pulsed engine will require efficient power extraction from the combusted fluid. A good understanding of the pulsed detonation exhaust is required if current turbine technology is to be used.

A significant body of work exists for PDE applications, and results have converged to agreement over the past five years or so. The data that has converged to agreement for unconfined venting of PDE exhaust. A PDC, on the other hand, vents into a volume confined by the detonation tube exhaust plane, splitter plates, and the stator/turbine assembly. Schauer et al [21] noted that when venting into a turbine, blowdown times of a single PDE tube increased from 3 ms to more than 10 ms, often extending into the fill portion of the next cycle. This delay subsequently impacts the

fill negatively. Recognizing this fundamental difference, Rasheed et al. [14] performed some research to better understand this process.

Under the NASA CVCCE program, GE Global Research [16,18–20] performed a series of tests on an eight-tube PDC exhausting into a turbine taken from a locomotive turbocharger. The detonation tubes used in this experiment were un-valved; flow entered the inlet plenum perpendicular to the detonation tubes, and tube cooling air was flowed through a separate inlet. These conditions differ from those that were modeled for our work, but the results gained from this effort provide valuable insight into how a PDC will work in conjunction with a turbine.

In an effort to better understand how a pulsed detonation engine affects the flow through the turbine, the G.E. Global Research group [14] also performed a series of CFD and experimental studies examining how the detonation waves interacted with a 2-dimensional cascade of stator blades. Their work shows that there is significant fluctuation in mass flow rate, pressure, and temperature across the turbine during the pulse detonation cycle. The fluctuations induced by the detonation are passed through the turbine and will affect the other rotor stages as well.

2.2.3 Converting Pulsed Flow into Steady State Flow. The results reported by Schauer et al. [21] and G.E. Global Research [19, 20] indicate that current turbine technology will not perform as expected unless the pulsed nature of the PDC is somehow balanced to produce steady-state or near steady-state flow into the turbines.

The machinery required between the pulsed flow generated by a PDC and the steady flow desired for optimal turbine performance has not yet been identified. It may involve ejectors, capacitive tanks, diffusers or a combination of these items combined with new ideas. It seems reasonable to assume that whatever the machinery, some pressure and enthalpy loss to the fluids that traverse them. The level of losses is not known at this time, but certainly there will be a definite efficiency required before the hybrid technology can present an improvement over a baseline high bypass turbofan engine.

We are left with the broader question: what level of losses can be incurred in the transition to steady state before the benefit gained by pulsed detonation is lost?

III. Baseline and Hybrid Models

Models of the baseline and the hybrid high bypass turbofan were generated with great care to ensure consistency. The baseline model was evaluated using both AEDsys and NPSS. This secured a good understanding of the high bypass turbofan and provided a solid foundation for comparing it to the hybrid engine. Once the baseline model was defined, the hybrid model implementing the PDC could be completed .

3.1 Baseline High Bypass Turbofan

The modeled engine was based on public information of the TF-39-GE-1C engine currently used on the C-5 Galaxy to certify that it was both relevant to the Air Force and easily verifiable. Table 3.1 shows a comparison of available information on the TF-39-GE-1C and the engine model generated for this exercise. The engine model was created without implementation of a requirements analysis and remains notional at this point. Since the component efficiencies were unknown, they were selected from Mattingly's table 4.4 [3, 107], using a level 4 technology. Level 4 technology corresponds not to the engine technology currently used, but to the technology level projected ten to twenty years in the future. The level 4 technology also projects a higher turbine inlet temperature (TIT) than the engines currently found in service.

Table 3.1: Fundamental parameters for baseline engine.

Engine	TF39-GE-1C	Notional Baseline
Aircraft	C-5	N/A
Fn (lb)	40805	41500
Weight (lb)	7186	N/A
Length (in)	100	N/A
Max Diam (in)	203	N/A
BPR	8	8
FPR	1.56	1.56
OPR	26	26
TSFC (1/h)	0.315	0.325
Airflow (lbm/s)	1549	1500
TIT (F)	2350	2830

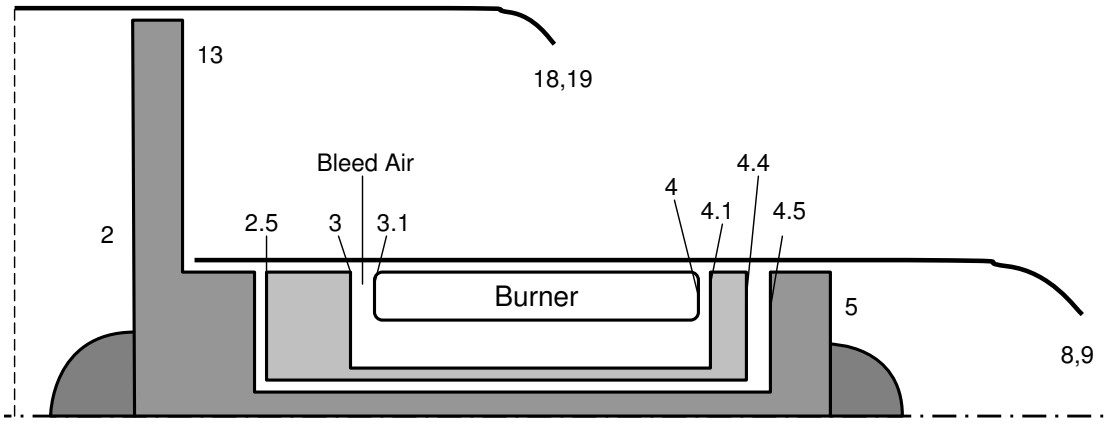


Figure 3.1: High bypass separate stream turbofan engine configuration and station numbering.

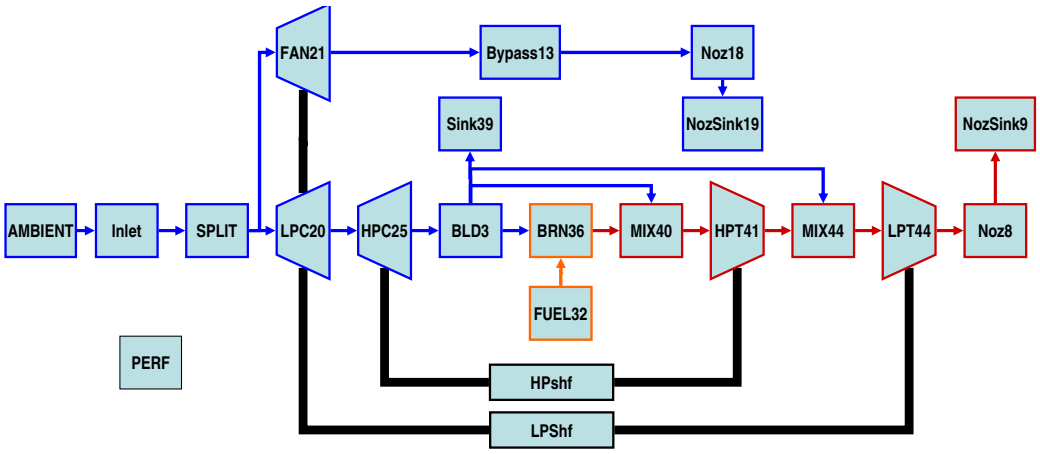


Figure 3.2: High bypass turbofan engine configuration in NPSS. Shown here are the named components and how the fluid is passed from one component to the other.

3.1.1 Baseline Engine Configuration. The engine model coded in NPSS is constrained by the fixed architecture of the high bypass split stream turbofan engine described by Mattingly [3, 569-587]. Indeed, the engine was first modeled using AEDsys with the input parameters given in Tables 3.2, 3.3, and 3.4. This high bypass turbofan engine is described in the most fundamental level by figure 3.1. The engine employs a separate exhaust stream for the bypass. A mixer for combining the fluid exiting the burner and one of the bleeds is implemented between stations 4 and 4.1. A similar mixer is implemented between stations 4.4 and 4.5. Fixed conical nozzle assumptions are used, and the fan is the outer portion of the first stage on the low-pressure compressor disk. Figure 3.1 shows the bypass and core airflows splitting at the fan exit, but the NPSS architecture required the split at the fan entry for correct implementation of separate fan and low-pressure compressor maps. The model file that defines the NPSS engine can be found in Appendix A. A component diagram of the engine is shown in Figure 3.2

Table 3.2: AEDsys reference design input variables for the baseline turbofan engine and thermodynamic package selection.

Input Variables	Value	Units
Gas Model		
Constant,Modified,Variable Specific Heat	MSH	
Reference Engine Design		
Flight Conditions		
Mach Number	0	
Altitude	0	ft
Temperature	518.67	R
Pressure	14.696	psia
Size		
Mass Flow Rate	1500	lbm/s
Design Limits		
total temperature leaving combustor	2900	R
Design Variables		
Compressor Pressure Ratio (Pt3/Pt2)	26	
LPC PR (Pt2.5/Pt2)	1.56	
Fan PR (Pt13/Pt2)	1.56	
Bypass ratio	8	

Table 3.3: AEDsys thermodynamic inputs for the modified specific heat option.

Input Variables	Value	Units
Fuel/Gas properties, Component Efficiency		
Fuel/Gas Properties		
Fuel Heating Value	18400	BTU/lbm
C_P compressor	0.2415	BTU/(lbm R)
gamma c	1.3986	
C_P turbine	0.295	BTU/(lbm R)
gamma turbine	1.2957	
Bleed air/ Coolant air		
Bleed Air	1	%
Coolant Air	5	%
Coolant Air 2	5	%
Power Takeoff		
C_{TOL}	0.00	
C_{TOH}	0.0005330	
LP Spool Mechanical Efficiency	0.99	
HP Spool Mechanical Efficiency	0.99	
Component total Pressure Ratio		
π diffuser max (Pt2/Pt1)	0.995	
π burner (Pt4/Pt3)	0.96	
π nozzle (Pt9/Pt7)	0.985	
π fan nozzle	0.98	
Polytropic Efficiencies		
Fan	0.89	
LPC	0.89	
HPC	0.90	
HPT	0.89	
LPT	0.89	
Component Efficiencies		
Burner	0.995	
Mechanical shaft LP spool	0.99	
Mechanical shaft HP spool	0.99	

Table 3.4: AEDsys engine control inputs.

Input Variables	Value	Units
Controls / Install Model/ # Engines		
Engine Controls		
Max Temperature at Station 4 (T_{t4}	3200	R
Max Compressor Pressure Ration	32	
Max Pressure at station 3	650	psia
Max Temperature at station 3	1860	R
Max % Ref RPM - LP Spool	110	
Max % Ref RPM - HP Spool	110	

3.2 Baseline Model Similarity Between AEDsys and NPSS

Although the baseline turbofan engine shares a common architecture in both programs, the two softwares require different methods of characterizing the components. Great care had to be taken to ascertain that the configurations not only had the same form, but also shared the same definition of component performance. Inevitably, differences in each program's solution process will produce some variation in the results. This section describes the greatest contributor to solution differences.

Design parameters were carefully ported into the NPSS model from the AEDsys program to eliminate differences due to component performance inputs. NPSS handles efficiencies of several types differently than AEDsys, so a translation had to be made in order to confirm models were similar. For some components, efficiencies input in AEDsys had to be translated to a normalized pressure drop term $\frac{\Delta P}{P}$ for NPSS. For other components, the calculated values of efficiency output from AEDsys had to be copied and input into the NPSS model. Table 3.5 shows the translation between the two sets of inputs. Also note that mechanical efficiencies from AEDsys translate to fractional losses in NPSS. The most significant item in this table is that NPSS requires adiabatic efficiency input whereas AEDsys requires polytropic efficiency for the compressors and turbines as an input.

Engine control inputs are not included in Table 3.5 because there is not a simple element that will impose restrictions on the engine. Instead, the control limitations

Table 3.5: AEDsys inputs differ slightly from NPSS inputs. This table should provide a method to correlate the inputs required by each program.

AEDsys Input	NPSS Element	Name in Appendix	Variable	Translation
Gas Model	Thermo Package	N\A	GasTbl\janaf\etc.	NPSS' thermo translates to AEDsys' VSH
Mach number	FlightConditions	AMBIENT	MN	Same
Altitude	FlightConditions	AMBIENT	alt	Same
Temperature	FlightConditions	AMBIENT	ZTt and ZTs	must set both
Pressure	FlightConditions	AMBIENT	ZPt and ZPs	must set both
Mass Flow Rate	FlightConditions	AMBIENT	W	Same
T_{t4}	Burner	BRN36	TtCombOut	set switchBurn=TEMPERATURE
Compressor PR	Compressor	HPC25	S_map.PRdes	CPR = HPC_{PR}*LPC_{PR}
LPC PR	Compressor	LPC20	S_map.PRdes	Same
Fan PR	Compressor	Fan21	S_map.PRdes	Same
Bypass Ratio	Splitter	SPLIT	BPR	Same
Fuel Heating Value	FuelStart	FUEL32	LHV	Depends on thermodynamic package.
Bleed Air (% $\dot{m}$)	Bleed	BLD3	port name.fracW	Divide AEDsys value by 100
Coolant Air 1 (% $\dot{m}$)	Bleed	BLD3	port name.fracW	Divide AEDsys value by 100
Coolant Air 2 (% $\dot{m}$)	Bleed	BLD3	port name.fracW	Divide AEDsys value by 100
CTOL	Shaft	LPShf	HPX	HPX = CTO*$h_0 * \dot{m}_0/\eta_m$
CTOH	Shaft	HPShf	HPX	(may also use AEDsys engine test output)
LP Spool PTO Efficiency η_m	Shaft	LPShf	fracLoss	Included in HPX calc
HP Spool PTO Efficiency η_m	Shaft	HPShf	fracLoss	if different than mechanical efficiency
π_{dmax} $\left(\frac{P_{t2}}{P_{t1}}\right)$	Inlet	INLET	eRamBase	Same
π_b $\left(\frac{P_{t4}}{P_{t3}}\right)$	Burner	BRN36	dPqPBase	$\frac{\Delta P}{P} = 1.0 - \pi_b$
π_n $\left(\frac{P_{t9}}{P_{t7}}\right)$	Nozzle	Noz8	dPqP	$\frac{\Delta P}{P} = 1.0 - \pi_b$
π_{nf} $\left(\frac{P_{t19}}{P_{t17}}\right)$	Nozzle	Noz18	dPqP	$\frac{\Delta P}{P} = 1.0 - \pi_b$
Fan polytropic efficiency	Compressor	Fan21	S_map.effDes	(Use AEDsys output isentropic efficiency)
LPC polytropic efficiency	Compressor	LPC20	S_map.effDes	(Use AEDsys output isentropic efficiency)
HPC polytropic efficiency	Compressor	HPC25	S_map.effDes	(Use AEDsys output isentropic efficiency)
HPT polytropic efficiency	Compressor	HPT41	S_map.effDes	(Use AEDsys output isentropic efficiency)
LPT polytropic efficiency	Compressor	LPT45	S_map.effDes	(Use AEDsys output isentropic efficiency)
η_b	Burner	BRN36	effBase	Same
η_m LP Spool	Shaft	LPShf	fracLoss	fracLoss = $1.0 - \eta_m$
η_m HP Spool	Shaft	HPShf	fracLoss	fracLoss = $1.0 - \eta_m$

are imposed through the use of independent and dependent variable declarations. These declared variables are then added to the NPSS solver. More information can be found in the NPSS user guide [24, sec 6.8].

With two different types of input for efficiency, a significant certification effort ensured that the turbines and compressors had the same performance. The polytropic and adiabatic efficiencies could not be matched simultaneously between the engines because the two programs work the progression from the input efficiency to the calculated efficiency in reverse order from each other.

Recognizing the need to relate the polytropic and isentropic (or adiabatic) efficiencies, a review of a fundamental turbomachinery text was performed. At a simplified level, Oates [25, 214 & 222] and Wilson and Korakianitis [26] recorded the formulae for computing the adiabatic efficiencies (η_c and η_{tb}) given the polytropic efficiencies (e_c and e_{tb}):

$$\eta_c = \frac{\pi_c^{\frac{\gamma_c - 1}{\gamma_c}} - 1}{\pi_c^{\frac{\gamma_c - 1}{e_c \gamma_c}} - 1} = \frac{\dot{W}_{in\ ideal}}{\dot{W}_{in\ actual}} \tag{3.1}$$

$$\eta_{tb} = \frac{1 - \pi_{tb}^{e_{tb}\frac{(\gamma_{tb} - 1)}{\gamma_{tb}}}}{1 - \pi_{tb}^{\frac{\gamma_{tb} - 1}{\gamma_{tb}}}} = \frac{1 - \tau_{tb}}{1 - \tau_{tb}^{\frac{1}{e_{tb}}}} = \frac{\dot{W}_{out\ actual}}{\dot{W}_{out\ ideal}} \tag{3.2}$$

where $\dot{W}$ is the power put into or taken out of the component. However, this relationship is observed in AEDsys only for the constant and modified specific heat thermodynamic cycles. Using this relationship to calculate the adiabatic efficiency does not account for the variation of specific heat calculated with the AEDsys variable specific heats (VSH) option or NPSS thermodynamic packages. The formulae for variable specific heats become more complicated since the variation of temperature across the components requires a complete thermal properties solution at exit.

With the goal of understanding the difference between AEDsys and NPSS, an examination of the methods employed by these programs was made. Mattingly [3, 106] chose to use the polytropic efficiency as input for his VSH because it is a measure of

technology. The polytropic efficiency is used to obtain the reduced pressure (P_r) at compressor exit:

$$P_{r\,exit} = P_{r\,inlet} \cdot \pi_c^{1/e_f} \tag{3.3}$$

$$P_{r\,exit\,ideal} = P_{r\,inlet} \cdot \pi_c \tag{3.4}$$

where π_c is the total pressure ratio across the compressor. With the reduced pressure and a fuel fraction of 0, the FAIR subroutine (described in more detail in section 3.2.1) is called to obtain all remaining fluid properties (including enthalpy (h)) at compressor exit.

A slightly different variation is employed to get the total pressure ratio across the turbine (π_{tb}). Assuming the conditions at burner exit (station 4) are known, the temperature ratio across the turbine (τ_{tb}) is identified using the required power across the shaft. The enthalpy drop across the turbine is calculated:

$$h_{tb\,exit} = h_{tb\,inlet}\tau_{tb} \tag{3.5}$$

The enthalpy at turbine exit(h_{tbexit}) is then passed into FAIR to obtain turbine exit conditions. This allows the calculation of the turbine pressure ratio:

$$\pi_{tb} = \left(\frac{P_{r\,tb\,exit}}{P_{r\,tb\,inlet}}\right)^{1/e_{tb}} \tag{3.6}$$

With the pressure ratio across the turbine we can get the ideal conditions using FAIR if we first calculate the reduced pressure:

$$P_{r\,tb\,exit\,ideal} = \pi_{tb} P_{r\,tb\,inlet} \tag{3.7}$$

Using the exit ideal and non-ideal conditions, the adiabatic efficiency is calculated for both the compressor and the turbine using total enthalpy differences:

$$\eta_c = \frac{h_{t\,ex\,ideal} - h_{t\,in}}{h_{t\,ex\,actual} - h_{t\,in}} = \frac{\dot{W}_{in\,ideal}}{\dot{W}_{in\,actual}} \tag{3.8}$$

$$\eta_{tb} = \frac{h_{t\,in} - h_{t\,ex\,actual}}{h_{t\,in} - h_{t\,ex\,ideal}} = \frac{\dot{W}_{out\,actual}}{\dot{W}_{out\,ideal}} \tag{3.9}$$

The adiabatic efficiency is stored as an output for on-design conditions, and is used in the off-design conditions.

NPSS takes a different approach to calculating the properties across compressors and turbines. As with AEDsys, inlet properties are known and the user specifies the pressure ratio across the component. The total pressure (P_t) at exit is calculated as

$$P_{t\,exit} = P_{t\,inlet} \cdot \pi_c \tag{3.10}$$

or

$$P_{t\,exit} = \frac{P_{t\,inlet}}{\pi_{tb}} \tag{3.11}$$

However, in order to obtain the enthalpy at exit, the ideal fluid conditions must first be calculated. The thermodynamic packages in NPSS requires two properties besides the fuel fraction when calculating all other fluid properties. Having the ideal total pressure, we still need one more ideal fluid attribute to obtain all fluid properties. In an ideal component there is no entropy change from inlet to exit so the fluid property solution can be obtained using the inlet entropy and calculated total exit pressure from equation 3.10 or 3.11. Non-ideal total enthalpy at component exit can then be calculated using the enthalpies and user input adiabatic efficiencies:

$$h_{t\,exit\,c} = \frac{h_{t\,ex\,ideal} - h_{t\,inlet}}{\eta_c} + h_{t\,inlet} \tag{3.12}$$

$$h_{t\,exit\,tb} = h_{t\,inlet} - (h_{t\,inlet - h_{t\,ex\,ideal}}) \cdot \eta_{tb} \tag{3.13}$$

This equation is the same as Equation 3.9; it has only been re-arranged. Subsequently, the thermodynamics routine is called using the non-ideal exit total pressure and total enthalpy as calculated above to obtain the fluid attributes.

Once the actual inlet and exit properties are known the polytropic efficiency is calculated.

$$e_c = \frac{R_{t\,inlet} \cdot \ln(\pi_c)}{R_{t\,inlet} \cdot \ln(\pi_c) + S_{exit} - S_{inlet}} \tag{3.14}$$

$$e_{tb} = \frac{R_{t\,inlet} \cdot \ln(\pi_{tb})}{R_{t\,inlet} \cdot \ln(\pi_{tb}) + S_{exit} - S_{inlet}} \tag{3.15}$$

There are two reasons to match the adiabatic efficiency between the two programs: first, the adiabatic (isentropic) efficiency is implemented by both programs identically using some form of equation 3.8. The second is that both programs utilize this efficiency for off-design calculations.

When the fluid properties at component exit are being calculated, small variations in the thermodynamic codes will result in a fluid properties discrepancy between programs. The resulting differences in the entropy across the component causes a small change in calculated efficiencies and a small temperature divergence. Small differences in temperature between the two programs can result in sizeable variations of reported power since, as Çengal and Boles put it [27]:

$$-\dot{W} = \dot{m}\left[(h_2 - h_1) - \left(V_2^2 - V_1^2\right) - g\,(z_2 - z_1)\right] \tag{3.16}$$

and enthalpy (h) not only dominates the other terms in the power equation, but is also a function of temperature.

To better understand how the small temperature variations between NPSS and AEDsys result in a difference of thrust, we must take a closer look at the thermodynamic models that are the basis of the two programs.

3.2.1 Thermodynamic Models. Two types of error might be encountered when comparing thermodynamic codes. As shown in Figure 3.3, the first type of error

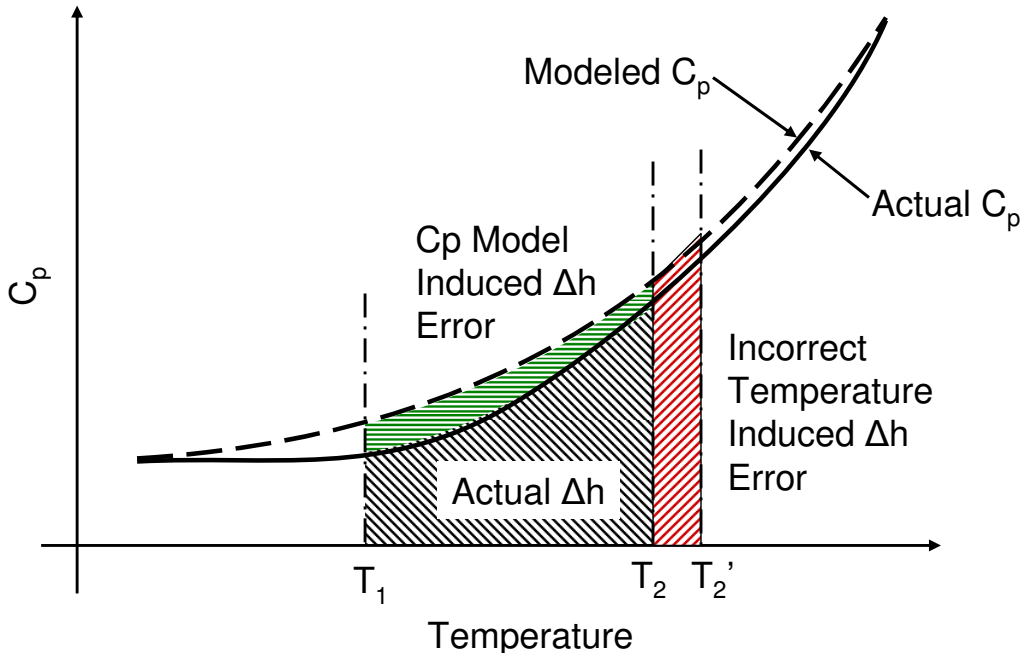


Figure 3.3: Two ways that errors might be introduced into a solution through use of the thermodynamics package.

is introduced when the specific heat model deviates from the true characteristics of the gas or other models. The area between the specific heat models is integrated to obtain the enthalpy at the given temperature using the formula:

$$h(T) = \int_{T}^{T_{ref}} C_p \, dT \tag{3.17}$$

Thus, any variation included in the integrated specific heat will yield a change in the area. Changing the area results in changing the enthalpy.

The second type of error is generated when temperature calculations across components report discrepant results. When this happens, the specific heat is integrated over different intervals. So, even if the thermodynamic model is shared the enthalpy will differ. The lesson here is that small alterations in specific heat modeling and fractions of a degree can translate into noticeable differences in final results.

To better understand the deviations in baseline engine results between AEDsys and NPSS, the thermodynamic packages of each were evaluated in order to mea-

Table 3.6: The coefficients used for the FAIR subroutine. FAIR provides the variable specific heat thermodynamic cycle calculations in AEDsys.

Pure Air		Vitiated Air	
A_0	$2.5020051x10^{-1}$	A_0	$7.3816638x10^{-2}$
A_1	$-5.1536879x10^{-5}$	A_1	$1.2258630x10^{-3}$
A_2	$6.5519486x10^{-8}$	A_2	$-1.3771901x10^{-6}$
A_3	$-6.7178376x10^{-12}$	A_3	$9.9686793x10^{-10}$
A_4	$-1.5128259x10^{-14}$	A_4	$-4.2051104x10^{-13}$
A_5	$7.6215767x10^{-18}$	A_5	$1.0212913x10^{-16}$
A_6	$-1.4526770x10^{-21}$	A_6	$-1.3335668x10^{-20}$
A_7	$1.0115540x10^{-25}$	A_7	$7.2678710x10^{-25}$
h_{ref}	$-1.7558886\frac{Btu}{lbm}$	h_{ref}	$30.58153\frac{Btu}{lbm}$
ϕ_{ref}	$0.0454323\frac{Btu}{lbm\,\deg R}$	ϕ_{ref}	$0.6483398\frac{Btu}{lbm\,\deg R}$

sure and correct any mismatches in reported enthalpy. The AEDsys thermodynamic package is a subroutine termed FAIR. FAIR is an 8th order polynomial fit to JANAF specific heat data for pure air, and CEA data for vitiated air according to Mattingly [28, 89-91]. First recorded by Capt. McKinney [29, 30], enthalpy and entropy functions are calculated using the relationships

$$C_p = A_0 + A_1T + A_2T^2 + A_3T^3 + A_4T^4 + A_5T^5 + A_6T^6 + A_7T^7 \tag{3.18}$$

$$h = h_{ref} + A_0T + \frac{A_1}{2}T^2 + \frac{A_2}{3}T^3 + \frac{A_3}{4}T^4 + \frac{A_4}{5}T^5 + \frac{A_5}{6}T^6 + \frac{A_6}{7}T^7 + \frac{A_7}{8}T^8 \tag{3.19}$$

$$\phi = \phi_{ref} + A_0 \ln T + A_1T + \frac{A_2}{2}T^2 + \frac{A_3}{3}T^3 + \frac{A_4}{4}T^4 + \frac{A_5}{5}T^5 + \frac{A_6}{6}T^6 + \frac{A_7}{7}T^7 \tag{3.20}$$

the same coefficients used for the calculation of specific heat (C_P) are also used for calculation of enthalpy (h) and entropy function (ϕ). The coefficients are recorded in Table 3.6. Since FAIR was not available as a stand alone routine a simplified routine titled unFAIR was written using Fortran90. The unFAIR subroutine is less robust than FAIR since it is only able to calculate the fluid properties if the temperature is

passed in. Fluid property output matches the tabulated data reported by Mattingly [28, 812-813]. A copy of this subroutine may be found in Appendix B.

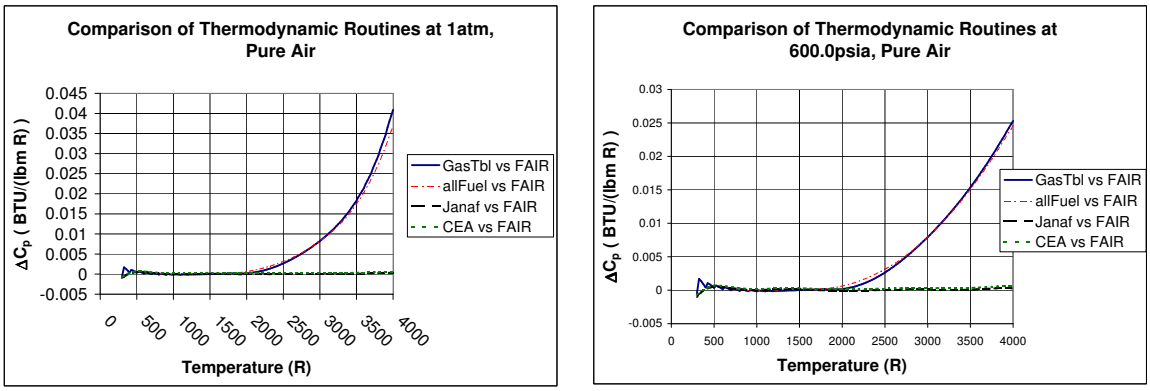
NPSS is more flexible than AEDsys. It has several thermodynamic packages available [24, sec 4.1]. Four mentioned in this study are:

- GasTbl, Package developed by Pratt & Whitney based on Therm, but adding humidity calculations and some chemical equilibrium capabilities.
- allFuel, Package developed by General Electric that contains gas properties and fuel properties.
- janaf, Implementation of the National Institute of Standards and Technology gas properties prepared by Honeywell
- CEA, Implementation of NASA's Chemical Equilibrium code; however, it is not a full implementation and lacks the capability to solve detonation.

Unlike AEDsys, NPSS is designed to allow the user to control the engine architecture and implementation of controls. Responsibility for configuration does not come without a price: more control given requires more control be exercised. Careful attention to the inputs and a good understanding of the thermodynamic models should prevent the generation of incorrect solutions that appear to be valid.

In order to understand the difference in results between AEDsys and NPSS, a study was performed to measure the variations between the two programs' thermodynamic subroutines. Since specific heat is the fundamental property that defines the other thermodynamic values, the specific heat was differenced between the AEDsys and NPSS thermodynamic routines.

Figure 3.4 shows differences of the NPSS solution from the AEDsys solution for specific heat in pure air (no fuel added) for conditions at 1 atm and 40.8 atm. Above 2000 °R and 1 atm the GasTbl and allFuel values diverge from FAIR, janaf, and CEA to approximately 15% difference at 4000 °R.There is a slight difference between the two plots at the higher temperatures: note that the high-pressure data from NPSS



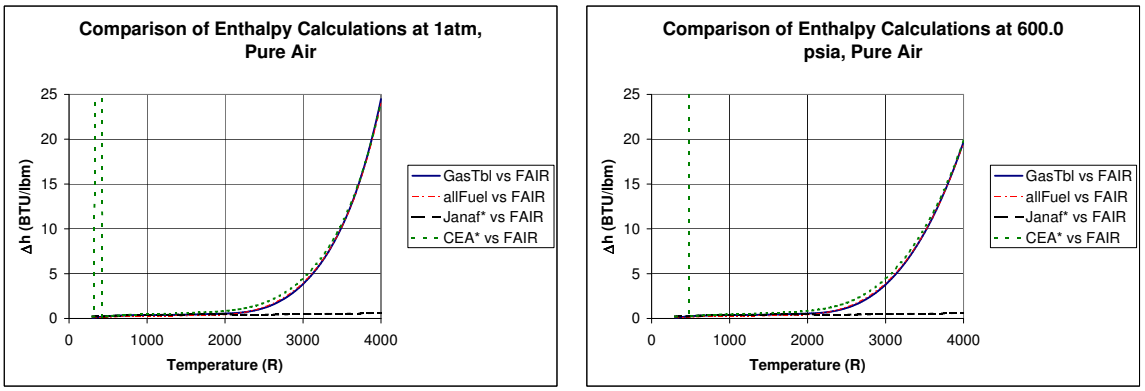
(a) (b)

Figure 3.4: Differences between FAIR and four thermodynamic routines found in NPSS for specific heat of pure air.(a) 1 atm (b) 40.8 atm. ($C_{p\ NPSS} - C_{p\ FAIR}$)

is closer to the FAIR data. Below 2000 °R the variations are small, with the peak at 500 °R representing a 2% difference.

The differences in the specific heat indicate differences in enthalpy as well. Since there was no documentation about the specific relationship between enthalpy and specific heat, it was decided to plot enthalpy differences as well. Figures 3.5 and 3.6 show that for pure air, the enthalpies calculated between the two programs diverge (with the exception of janaf.) The change in enthalpy appears to be minor, but by looking at the change in Δh that occurs between 518 °R and 598 °R (temperature change across the fan and low-pressure compressor), the difference in computed power can be calculated. If the Δh values were constant (and the line shown in Figure 3.6 had no slope) then the programs would suffer from a simple bias that would have very little effect upon the power calculation (such is the case of janaf above about 1500 °R for pure air). Consider the GasTbl enthalpy change across the fan and low pressure compressor: the changing Δh value between 525 and 600 °R results in a difference between the enthalpy change $(h_2 - h_1)$ calculations of 0.05 BTU/lbm. Recall that power for a compressor might be written as:

$$-\dot{W} = \dot{m}\left[(h_2 - h_1) - \left(V_2^2 - V_1^2\right) - g\left(z_2 - z_1\right)\right] \tag{3.21}$$



(a) (b)

Figure 3.5: Difference between the NPSS calculation of enthalpy and FAIR calculation at (a)1 and (b) 40.8 atm. FAIR calculates h independent of pressure, but is in better agreement when the NPSS calculations use high pressures.

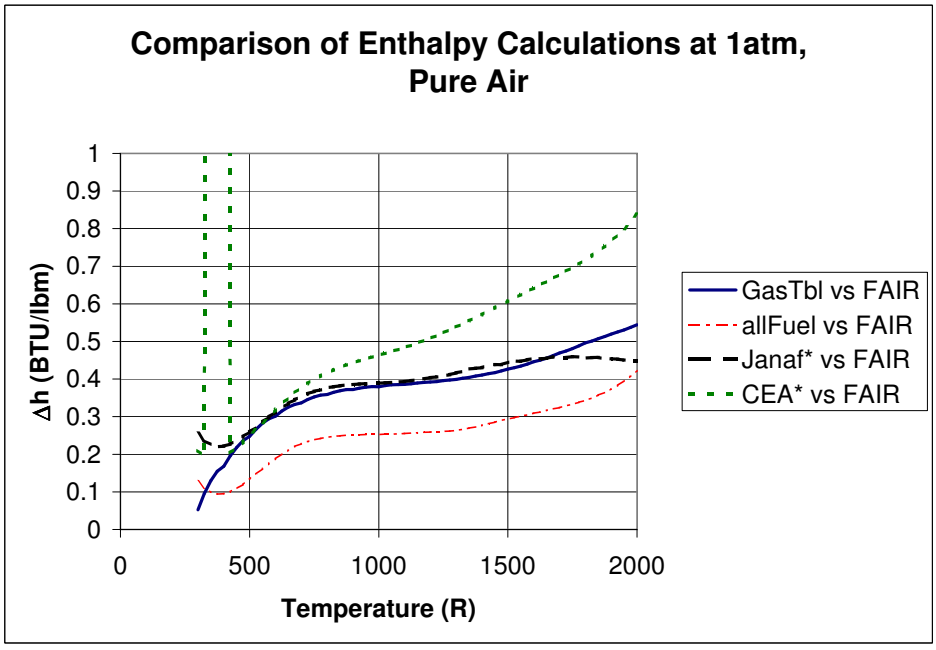


Figure 3.6: Detailed plot of the differences between FAIR and NPSS thermodynamics package calculation of enthalpy for un-reacted air at 1 atm. Although the changes are very small, when they are multiplied by a large mass flow, they can create noticeable deviations.

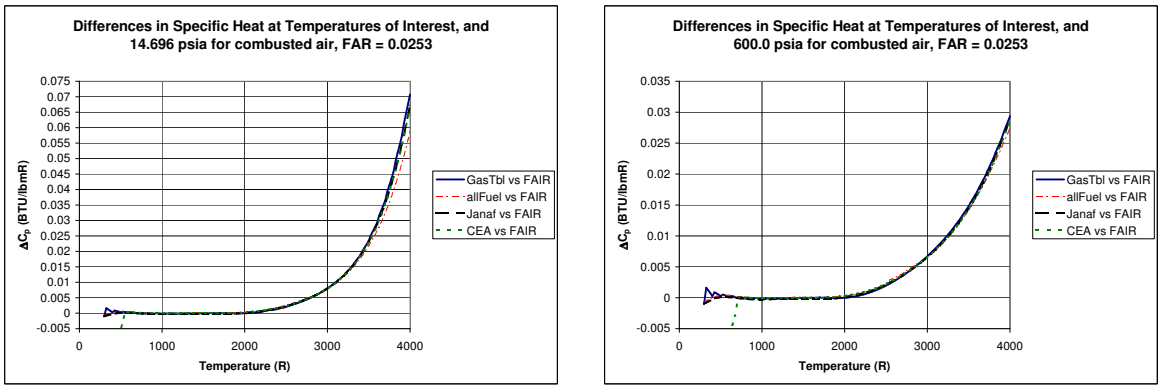


Figure 3.7: Difference in specific heats for combusted air (FAR = 0.0253) between FAIR and AEDsys routines at (a) 1 atm and (b) 40.8 atm.

so the difference in computed power becomes:

$$-\dot{W} = \dot{m}\,(\Delta h_2 - \Delta h_1) \tag{3.22}$$

where Δh is the difference in enthalpy between NPSS and AEDsys. For our low pressure compressor the change in power might be approximated as:

$$-\dot{W} = 1500\frac{lbm}{s}(0.301082 - 0.266627)\frac{BTU}{lbm} = 51.68\frac{BTU}{s} = 73.12hp \tag{3.23}$$

For the high pressure compressor, the temperature rises from approximately 600 °R to 1425 °R, and the $-\dot{W}$ is computed as

$$-\dot{W} = 166.67\frac{lbm}{s}(0.414196 - 0.301082)\frac{BTU}{lbm} = 18.98\frac{BTU}{s} = 26.85hp \tag{3.24}$$

We note that the errors in computed power will be greatest where the slope of the Δh line is the greatest, and that even relatively small differences in calculated enthalpy can be magnified by a large mass flow through the component.

Total property calculations for combusted gases follow similar trends as pure air. Data was run for a FAR of 0.0253, and the difference between specific heats is shown in Figure 3.7. Here the differences at high temperatures is notable because the

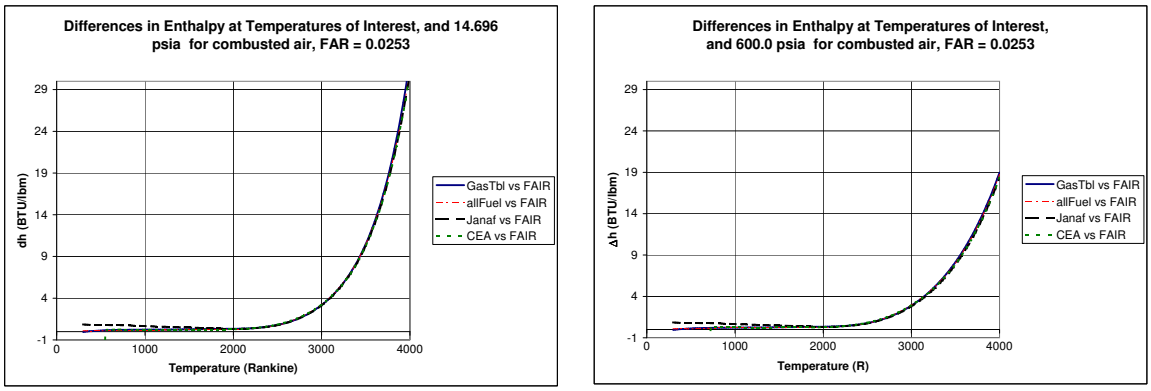


Figure 3.8: Differences in calculated enthalpies for combusted air with a FAR of 0.0253 at (a) 1 atm and (b) 40.8 atm. Agreement between the programs is much better when NPSS enthalpies are calculated using high pressure

combusted air is often above the 2000 °R point. The extreme difference at 4000 °R, 1 atm represents an 18.2% difference between the two programs. At 3000 °R, where the peak engine temperatures for this study were limited, the C_P is approximately 2.5% different. Even more concerning is that this trend is repeated in the enthalpy differences as seen in Figure 3.8. Again, when the NPSS data is generated assuming a high pressure, the data at high temperatures enjoys better agreement. At 3000 °R, there is approximately 5% reduction in the differences between the programs.

Since the baseline turbofan engines operate with a mass fuel-to-air ratio of approximately 0.0253, we can use this data to calculate differences in turbine power as well as compressor power. For the high pressure turbine work is performed between 2825 and 2125 °R. Our power difference between NPSS using GasTbl and AEDsys' FAIR is calculated as:

$$-\dot{W} = 160.42\frac{lbm}{s}(0.358752 - 1.9993)\frac{BTU}{lbm} = -263.176\frac{BTU}{s} = -372.3hp \quad (3.25)$$

For the low pressure turbine, work is calculated between 2075 and 1475 °R. The power difference between the programs for this temperature range is:

$$-\dot{W} = 160.42\frac{lbm}{s}(0.264505 - 0.344959)\frac{BTU}{lbm} = -13.577\frac{BTU}{s} = -19.21hp \quad (3.26)$$

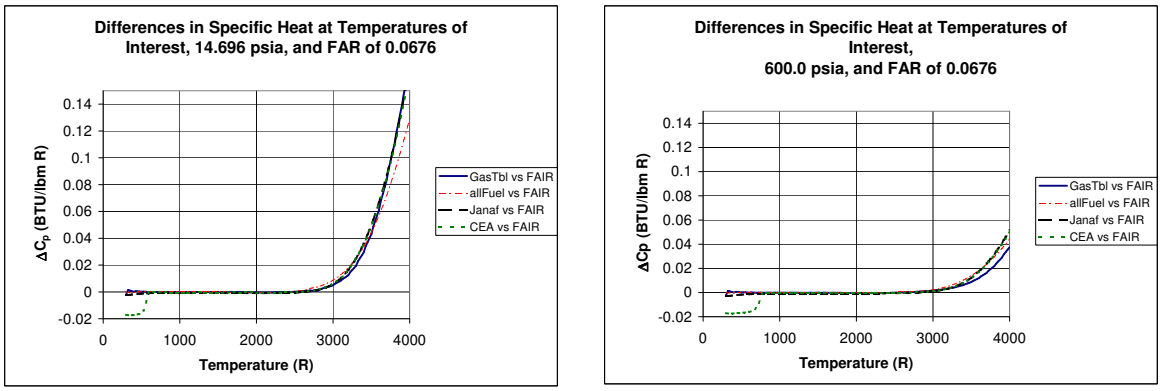


Figure 3.9: Difference in specific heats for combusted air (FAR = 0.0676) between FAIR and AEDsys routines at (a) 1 atm and (b) 40.8 atm.

By averaging the power extracted at the compressor with the power generated at the turbine, we obtain an average power difference across a given shaft due to thermodynamic model differences. From the preceding analysis, we see that AEDsys calculated approximately 50 hp net less power on the low pressure shaft and 200 hp net less power done on the high-pressure shaft. If this were the only thermodynamic difference between the two programs, it would be possible to apply the net power difference to the AEDsys shaft to achieve a net thrust that was virtually identical to that of the NPSS calculation. However, it will be shown in the results section that the AEDsys results are lower than NPSS results to begin with. This is due to the other thermodynamic difference - calculation of the temperature across a component, as was discussed in 3.2.

It is also interesting to note that for an equivalence ratio near one, the thermodynamic models again follow the same trend. As seen in figures 3.9 and 3.10, the quantities match each other more closely at low temperatures and high pressures.

3.3 Hybrid Turbofan - Pulsed Detonation Combustor Engine

3.3.1 Turbofan Configuration. The hybrid model contains the compressors, fans, and turbines of the baseline high bypass turbofan, but the burner section is replaced with a pulsed detonation combustor as seen in Figure 3.11. As much

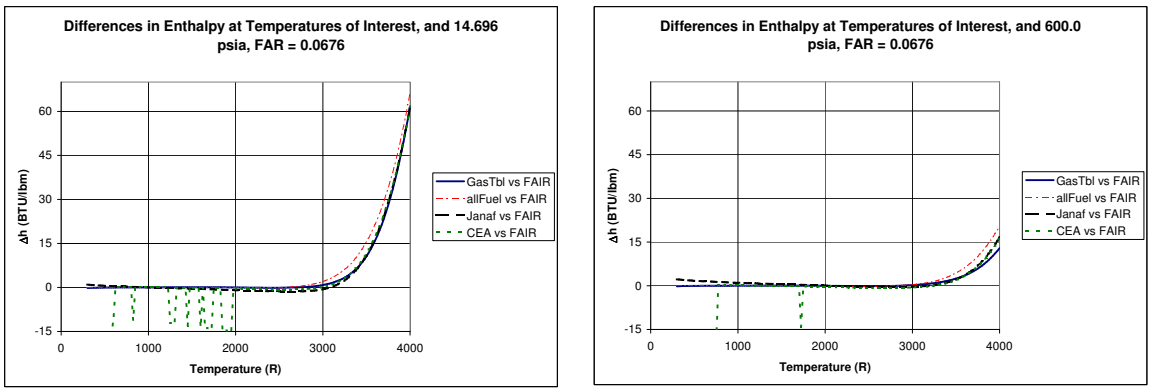


Figure 3.10: Differences in calculated enthalpies for combusted air with a FAR of 0.0676 at (a) 1 atm and (b) 40.8 atm. Agreement between specific heat models is much closer when NPSS properties are calculated using high pressure.

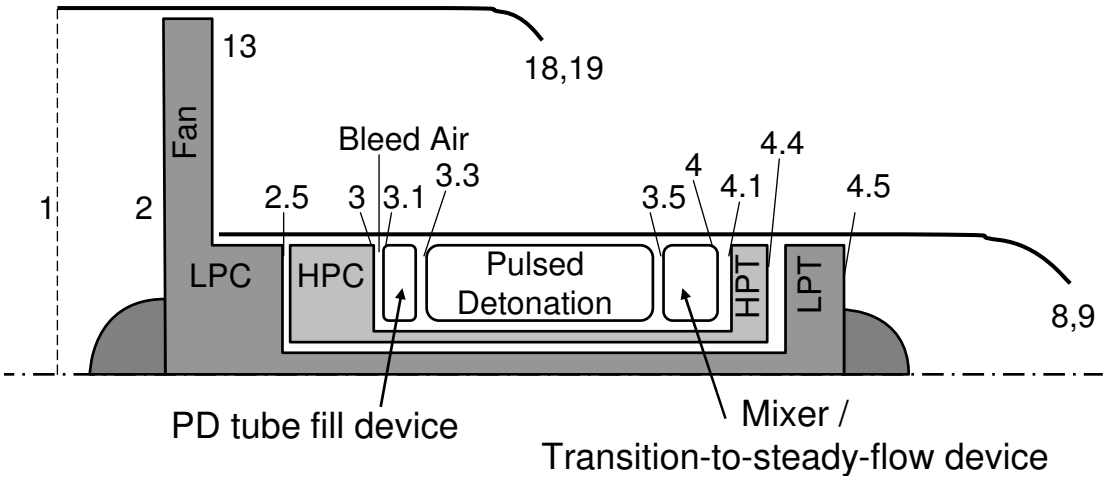


Figure 3.11: Configuration of the hybrid engine evaluated in this thesis.

of the baseline configuration as possible was retained so that the change in engine performance could be measured. The flow control at the detonation tube inlet was modeled as a pressure loss $\frac{\Delta P}{P}$ term between inlet and detonation tubes. This pressure loss matched the dry-duct pressure loss experienced by the conventional combustor (which is overly optimistic for the PDC). Mass inlet and bypass ratios were maintained constant between the hybrid and baseline models to mask the effect of thrust augmentation due to increased bypass flow. Compressor pressure ratios were also held constant, causing shaft power to remain identical to the baseline engine for all results with the exception of one of the trade off studies.

A mixer was introduced to combine the tube and internal bypass flows exiting the combustor. Utilizing the mixer introduced some difficulties since it occasionally allows NPSS to arrive at an incorrect converged solution without generating an error. The erroneous solution can be identified by a negative mach number for the secondary incoming flow (the internal bypass flow for this model). All solutions included in this thesis were checked for a correct mixer solution, with no data being reported for cycle solutions with incorrect mixer data.

3.3.2 Pulsed Detonation Combustor. The mathematics that define the detonation within the PDC model are taken from the work of Heiser and Pratt [4] with a Dyer and Kaemming correction [5]. The detonation occurs cyclically, introducing an unsteady flow. In order to assume steady flow through the same turbines as the baseline engine, a correction was made for transitioning the pulsing flow to steady state flow. No attempt was made to define an architecture for accomplishing the transition, instead, an arbitrary adjustment was made to the total temperature and pressure to account for the thermodynamic changes effected by the transition.

The pulsed detonation combustor was conceptually divided into three sections as shown in Figure 3.12:

- Flow control at detonation tube inlet

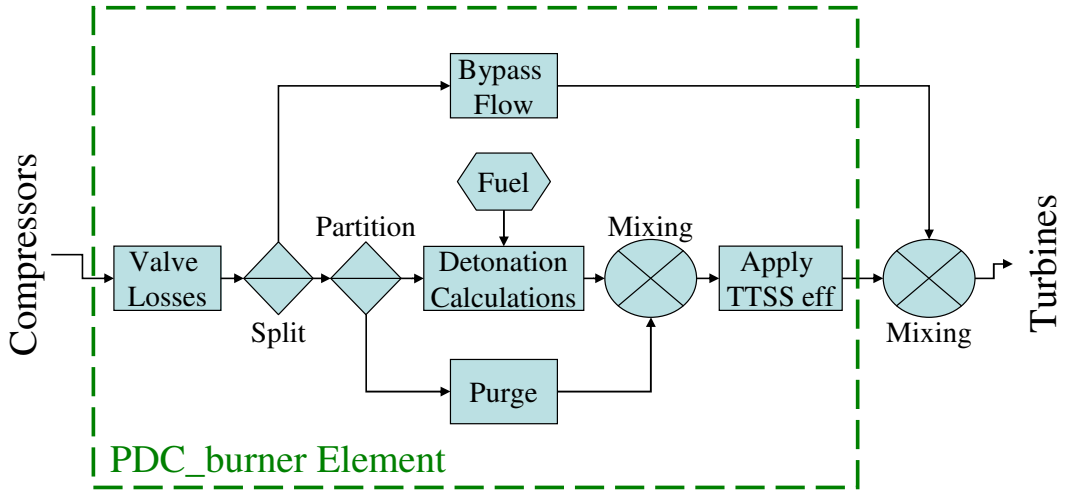


Figure 3.12: Configuration of the pulsed detonation combustor used for the hybrid engine.

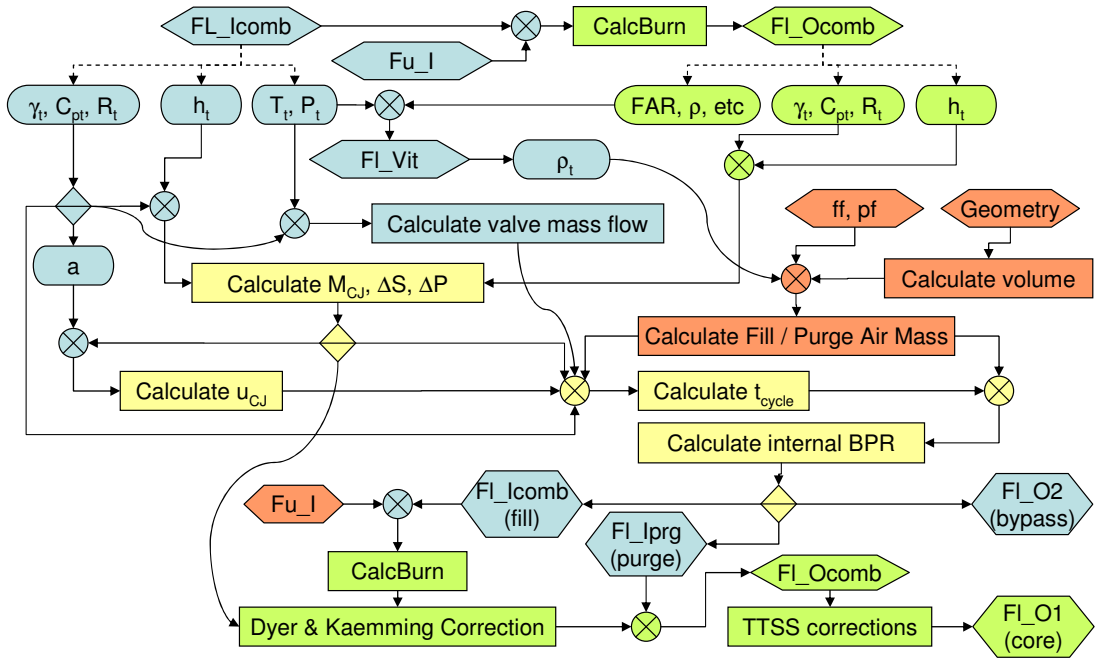


Figure 3.13: Configuration of the pulsed detonation combustor used for the hybrid engine.

- Detonation tubes
- Transition to steady state and bypass mixing device

Figure 3.12 shows how the fluid flows through the pulsed detonation combustor, but not how the information is passed through the code that calculates the fluid properties at entry and exit. The NPSS subroutine or "Element" for the PDC is included in Appendix C. Most of the code is borrowed from the NPSS "burner.int" file included in NPSS release 1.6.3. Major modifications to support the modeling pulsed detonation combustion occur after line 750.

A diagram of the information flow through the PDC_burner algorithm is shown in Figure 3.13. Although it appears complex, the diagram is meant to illustrate how the properties of the fuel an inlet flow are used to calculate the Chapman-Jouguet detonation solutions. This information is then combined with geometry inputs to determine mass flow and cycle times. Once mass flow is determined, the energy-conserved PDE solution is calculated and applied to the portion of the mass flow allocated to the detonation process. All fluids are mixed and corrected for the transition to steady state.

3.3.2.1 Detonation Properties. Central to the processes of the pulsed detonation combustor is detonation. The model used for this thesis calculates fluid properties behind the shock using the thermodynamic model described by Heiser and Pratt [4]. Implementation into NPSS required a few modifications to equations 2.7 to 2.9. This was accomplished by re-arranging the quantity $\frac{\tilde{q}}{\Psi}$ as follows:

$$\left(\frac{\tilde{q}}{\Psi}\right) = \frac{\left(\frac{q_{supplied}}{C_p T_0}\right)}{\left(\frac{T_3}{T_0}\right)} = \frac{h_4 - h_3}{\overline{C_p} \cdot T_3} \tag{3.27}$$

This allowed the PDC subroutine to utilize the component inlet temperature instead of trying to find the temperature at engine inlet. The modified equation uses specific heat and enthalpies $(\gamma, C_p, h_4 - h_3)$ from both the conditions that are given

and those that are solved for. Using the solved properties in the solution requires either iteration, or pre-calculation of the combustion.

Having calculated the $\left(\frac{\tilde{q}}{\Psi}\right)$ value, M_{CJ} is calculated using equation 2.7.

$$M_{CJ}^2 = (\gamma+1)\left(\frac{\tilde{q}}{\Psi}\right) + 1 + \sqrt{\left[(\gamma+1)\left(\frac{\tilde{q}}{\Psi}\right)+1\right]^2 - 1} \tag{3.28}$$

The Chapman-Jouguet Mach number is then used to calculate entropy gain and pressure rise across the shock using equations 2.8 and 2.9.

$$\frac{s_4 - s_3}{C_p} = -\ln\left[M_{CJ}^2\left(\frac{\gamma+1}{1+\gamma M_{CJ}^2}\right)^{\frac{\gamma+1}{\gamma}}\right] \tag{3.29}$$

$$\frac{p_4}{p_0} = \frac{1+\gamma M_{CJ}^2}{\gamma+1}\frac{p_3}{p_0} \geq 1 \tag{3.30}$$

The forms of the equations are fairly simple, and give rise to the question of their accuracy. Glassman [7, p 248] provides M_{CJ} data plotted as a function of equivalence ratio (ϕ) at standard pressure and temperature for a variety of fuels. Figure 3.14 shows the Chapman-Jouguet Mach number calculated using the closed form solution and NPSS thermodynamic routines at standard temperature and pressure overlaid on the data from Glassman. The calculated properties utilize fuel properties similar to those of Jet-A fuel. It will be noted that the GasTbl thermodynamic package is limited to results whose equivalence ratio is less than unity, so the allFuel package was used to generate results for and equivalence ratio greater than one. The M_{CJ} was shown in these charts to be marginally above values for a large hydro-carbon such as octane (C_8H_{18}). To determine whether this discrepancy is conservative or not requires an examination of what the M_{CJ} value is used for. The examination occurs through the calculation of entropy gain. Entropy gain is one of the two fluid properties input to solve for pressure and temperature. Figure 3.15 shows how entropy error has a large effect on pressure and a negligible effect on temperatures. Also, a larger Mach will result in a larger entropy gain, pushing us to the right on this figure. Lower

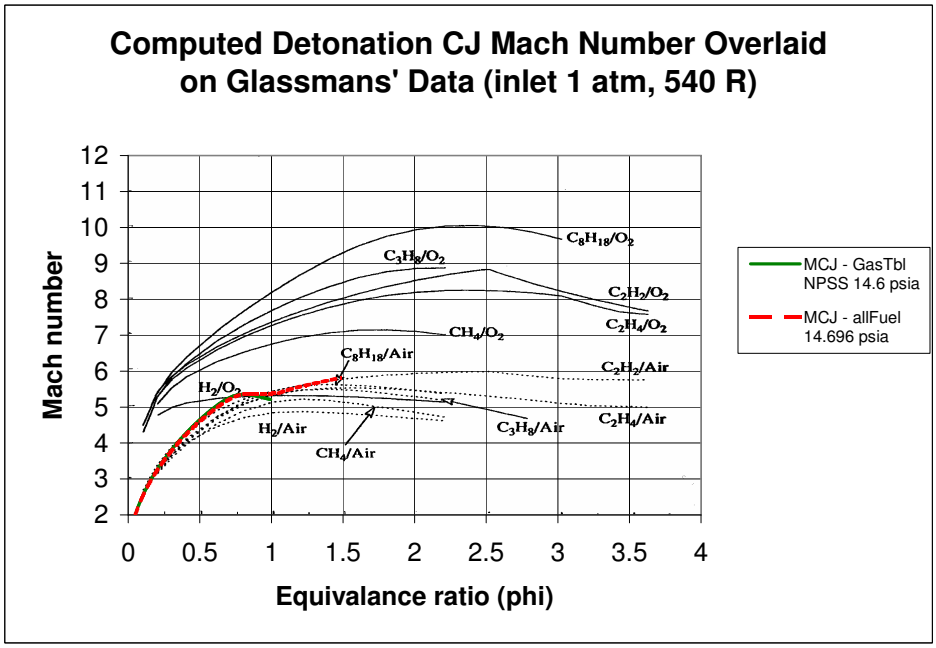


Figure 3.14: Chapman-Jouguet Mach number data overlaid on the chart from Glassman. [7, 248].

pressure decreases system performance, and the result is a conservative estimate of net thrust and TSFC.

Since the PDC is embedded in a turbofan, it experiences detonation at higher pressures than exist for the reviewed data [7, p248-250]. Figure 3.16 shows that M_{CJ} increases with increasing pressure. This is expected since increasing pressure decreases the mean free path between reacting particles, allowing for a faster transmission of energy. The calculated Mach number and the data from Glassman agree fairly well with experimental pulsed detonations, which see a detonation wave move at a Mach number slightly above 5 at standard conditions.

Having solved for pressure and entropy at detonation tube exit (station 4), it is possible to solve for all the all properties of the fluid at this point through thermodynamic relationships. However, Dyer and Kaemming [5] noted that this would be inaccurate since it ignores the eventual pressure loss that the gas will go through due to expansion waves. Figure 3.17 contains a T-S diagram explaining their proposed correction to the cycle. The expansion also results in the lowering of kinetic energy.

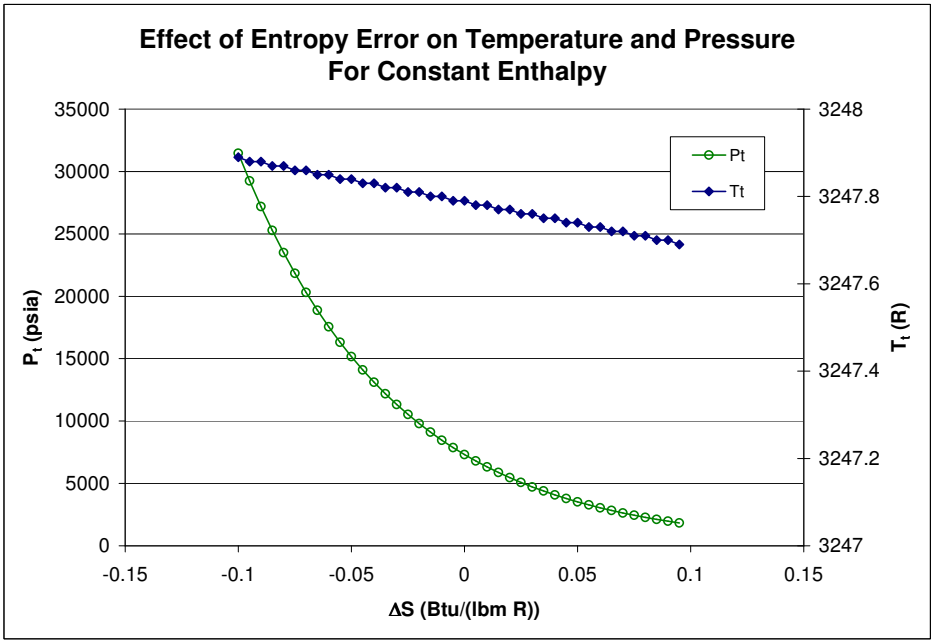


Figure 3.15: Small errors in calculation of the entropy strongly affect the pressure, but barely affect the temperature when all fluid properties are solved for using entropy and enthalpy.

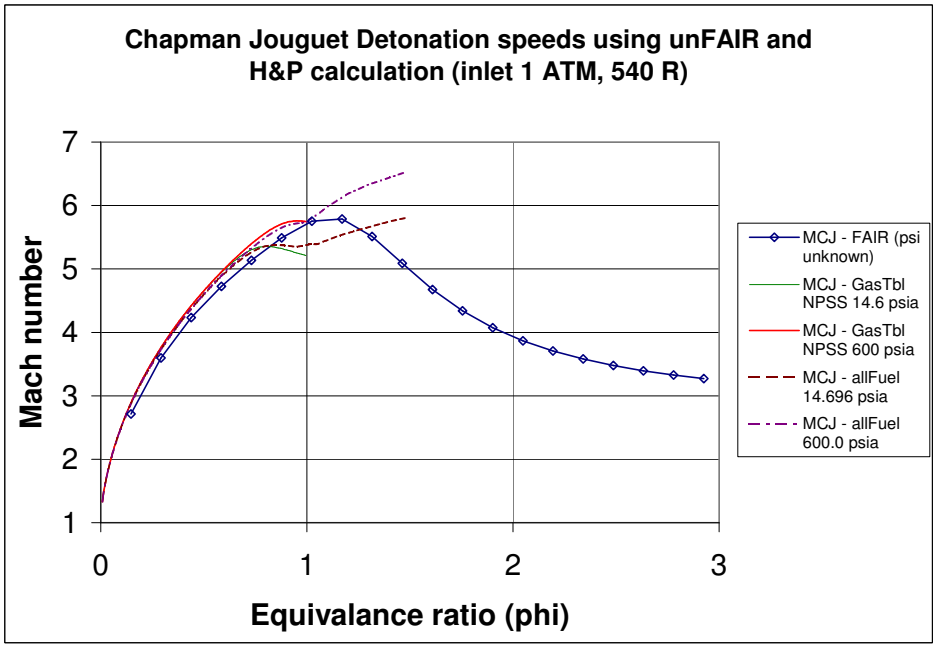


Figure 3.16: Chapman-Jouguet Mach number data calculated using several NPSS thermodynamics packages at standard and high pressures.

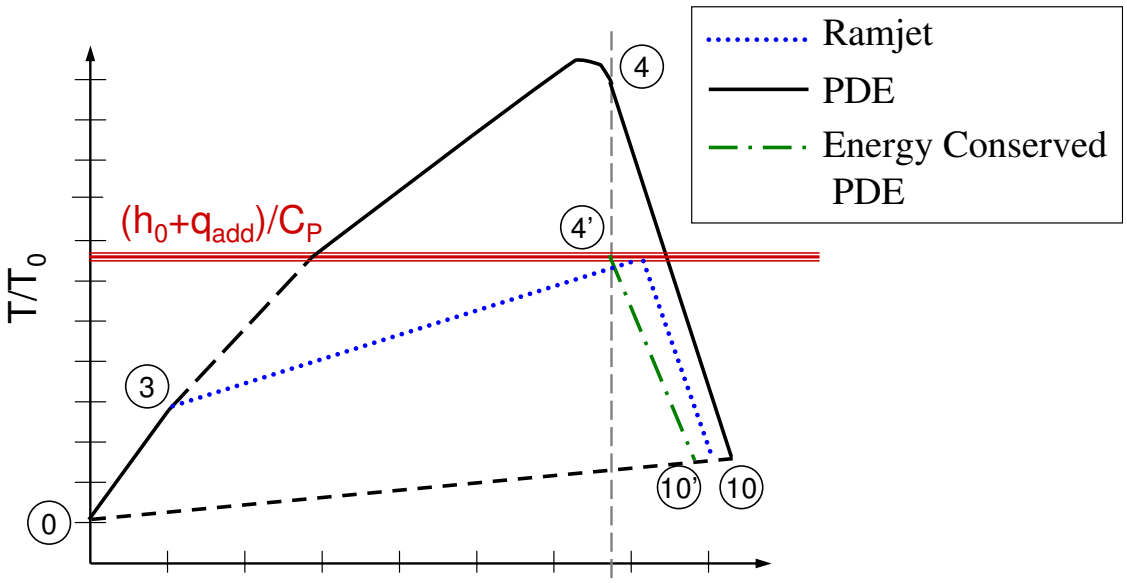


Figure 3.17: The T-S diagram shown in Dyer and Kaemming paper [5, fig 10] with the proposed correction to the PDE thermodynamic cycle. Conservation of energy will prevent incorrect calculation of entropy as the fluid exits the engine.

They proposed that the fluid properties should be solved for using the entropy gain as calculated above and the change in enthalpy liberated by the combustion process. Recall that this is the same change in enthalpy that was used to calculate $\frac{\tilde{q}}{\Psi}$. This would result in a different temperature and pressure profile for the gasses at station 4. Figures 3.18 and 3.19 show the deviation that the Dyer and Kaemming [5] formulation causes to the pressure and temperature of a detonated fluid at standard conditions. The underlying data are taken from Glassman [7, pp 249-250]. It should be noted that none of the P_2/P_1 data and Heiser and Pratt [4] calculations match the pressure ratio immediately behind the shock, but overestimate the time-averaged pressure ratio over the detonation time. Dyer and Kaemming [5] values appear to under-estimate the time-averaged pressure ratio over the detonation portion of the cycle.

The pressure ratio across the shock as computed by using enthalpy and entropy should give us the correct system performance, but may not accurately represent the actual conditions in the detonation tube. Note that the pressure ratio actually becomes inverted over portions of the equivalence ratio plot for atmospheric conditions.

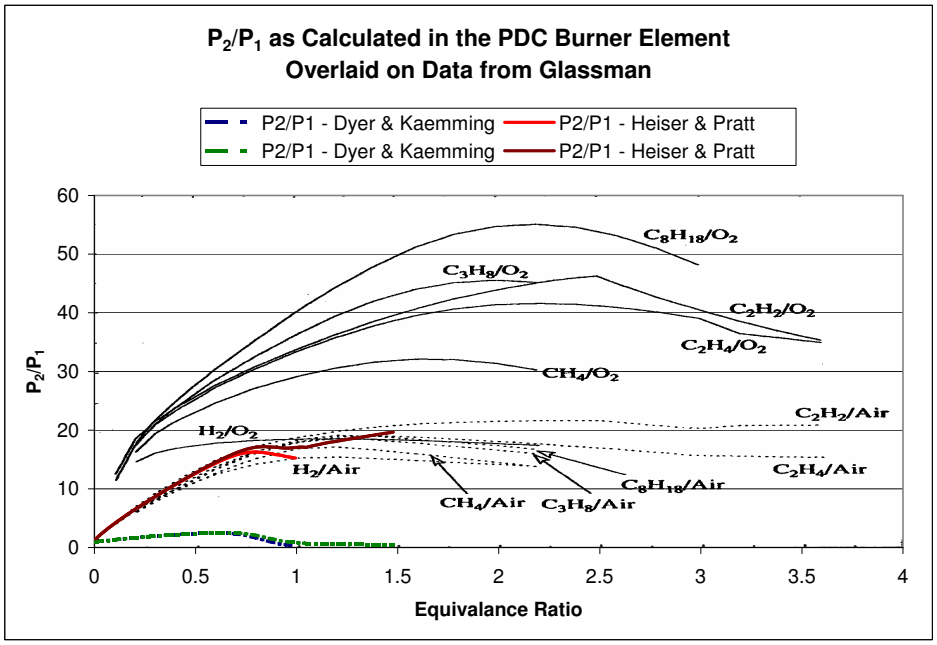


Figure 3.18: Ratio of the pressure behind to the pressure ahead of the detonation shock overlaid on data from Glassman [7, 250]. The upper line represents the pressure ratio as calculated by Heiser and Pratt, [4] the lower is the calculation using the Dyer and Kaemming [5] correction. .

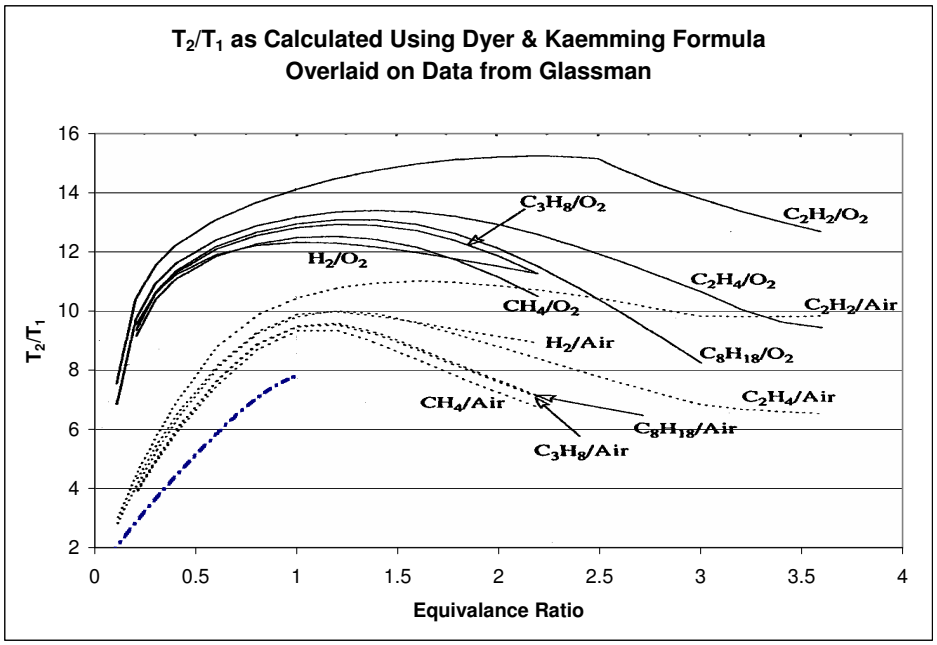


Figure 3.19: Ratio of the temperature behind to the temperature ahead of the detonation shock as calculated using the Dyer-Kaemming correction [5] . Overlaid on data from Glassman. [7, 249].

In section 3.3.2.3 we will see that we will use the pressure ratio based on the Heiser and Pratt [4] formulation to calculate the blowdown time of the system.

3.3.2.2 Determining Mass Flow Rate. Heat flow into the engine is controlled by the fuel flow and for a PDC fuel flow is defined by fill flow. PDC operation demands that the FAR be near unity for consistent detonation. Consequently the amount of air flowing into the tubes becomes the prime control of the engine. The air flowing through the detonation tubes is divided into two portions based on whether or not it is mixed with fuel. The mixed portion is termed the fill air, while the unmixed portion is called the purge air. Purge air is used for cooling of the detonation tubes while acting as a barrier between the hot combustion products of the previous and current cycle. The purge fraction pf and fill fraction ff are defined as fractions of the distance the fuel/air mixture or purge air flows relative to the tube length. Mathematically it looks like:

$$pf = \frac{l_{purge\,air}}{l_{tube}} \tag{3.31}$$

$$ff = \frac{l_{fuel-air\,mixture}}{l_{tube}} \tag{3.32}$$

where l denotes a length.

Purge and fill fractions can also be defined in terms of the volume (and subsequently mass) of the air for their respective portions of the tube filling process. Since the cross section of the fluid and the tube share the same area and the fractions are defined by the length that the respective fluids flow down the tube, the equations become:

$$pf = \frac{V_{purge\,air}}{A_{tube} \cdot l_{tube}} \tag{3.33}$$

$$ff = \frac{V_{fuel-air\,mixture}}{A_{tube} \cdot l_{tube}} \tag{3.34}$$

where A_{tube} is the cross sectional area of the tube, l_{tube} is tube length, $V_{purge\,air}$ is the purge air for one cycle, and $V_{fuel-air\,mixture}$ is the volume that the fuel-air mixture occupies in the detonation tube. Also, equation 3.34 can be re-arranged to define the volumes of purge and fill fluids in terms of tube volume and the purge/fill fractions:

$$V_{purge\,air} = pf \cdot A_{tube} \cdot l_{tube} \tag{3.35}$$

$$V_{fuel-air\,mixture} = ff \cdot A_{tube} \cdot l_{tube} \tag{3.36}$$

The purge air mass m_{purge} and fuel-air mass used for filling ($m_{fuel-air\,mixture}$) can be found using the total density (ρ_t), since the air is stopped in the tube once the valve closes:

$$m_{purge} = pf \cdot V_{tube} \cdot \rho_t \tag{3.37}$$

$$m_{fuel-air\,mixture} = ff \cdot V_{tube} \cdot \rho_t \tag{3.38}$$

These equations represent the amount of purge air and fuel-air mixture will flow into one tube during each cycle. The practice of allowing the purge and fill to sum to unity was observed when results were generated. This practice corresponds to completely filling a detonation tube each cycle without allowing any spillage or retention of hot gasses from the previous cycle.

Knowing how much air to send through the valve at the opening of the detonation tube requires separating the fuel from the air in the fill portion of the pulse-detonation cycle. This can easily be done at this point because mixed fuel air mass can be written as:

$$m_{fuel-air\,mixture} = m_{fill\,air} + m_{fuel} \tag{3.39}$$

$$\frac{m_{fuel-air\,mixture}}{m_{fill\,air}} = \frac{m_{fill\,air}}{m_{fill\,air}} + \frac{m_{fuel}}{m_{fill\,air}} \tag{3.40}$$

$$\frac{m_{fuel-air\,mixture}}{m_{fill\,air}} = 1 + FAR \tag{3.41}$$

$$m_{fill\,air} = \frac{m_{fuel-air\,mixture}}{1 + FAR} \tag{3.42}$$

$$m_{fill\,air} = \frac{ff \cdot V_{tube} \cdot \rho_t}{1 + FAR} \tag{3.43}$$

where $m_{fill\,air}$ is the mass of the air that will be detonated and m_{fuel} is the fuel used during one cycle. Once the mass of the air flowing into the tubes during one cycle is known, we can get the time averaged steady state mass flow rate ($\dot{m}_{air\,total}$) by multiplying it with the cycle frequency (f).

$$\dot{m}_{air\,total} = (m_{fill\,air} + m_{purge}) \cdot f \tag{3.44}$$

However, since the valve is only open for a fraction of the detonation cycle, the actual mass flow rate through the valve may be much greater than the steady-state depending on synchronization of multiple tubes. The maximum mass flow rate through the inlet valves must be calculated to ensure that the mass flow does not exceed the capabilities of the system. Treating the valve as a nozzle, and assuming the geometry allows for choking, we follow Mattingly [3, 9-10] and Wilson and Korakianitis [26, 70-71] in writing the mass flow rate during the valve-open time ($\dot{m}_{vo}$) as:

$$\dot{m}_{vo} = \frac{A_{valve} \cdot P_t}{\sqrt{T_t}} \sqrt{\frac{\gamma}{R}} MFP \tag{3.45}$$

Where MFP is the mass flow parameter that can be written as a function of Mach number, or static-to-stagnation pressure ratio:

$$MFP = M \left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{-(\gamma+1)}{2(\gamma-1)}} \tag{3.46}$$

$$MFP = \sqrt{\left(\frac{2}{\gamma - 1}\right) \left[\left(\frac{P_t}{P}\right)^{\frac{\gamma-1}{\gamma}} - 1\right] \left(\frac{P_t}{P}\right)^{\frac{-(\gamma+1)}{\gamma}}} \tag{3.47}$$

This allows us to define the maximum mass flow rate for a given Mach number and valve inlet area. The flow rate during the purge and fill portions of the cycle will help

to define cycle time. More importantly, it will determine the valve-open time fraction (τ_{vo}). This parameter is important because it prescribes how many tubes need to be open at one time in order to maintain a constant flow into the PDC.

3.3.2.3 Cycle Time. Cycle time could have easily been an input for the program, but the work of Schauer et al. [21] showed that when a detonation tube is fired into a turbine the detonation cycle is changed. In order to get a more realistic feeling for what a minimum cycle time would be calculations were made based on some simple assumptions: choked flow at tube exit for blowdown, a detonation moving at M_{CJ}, and a valve flow at or below choked conditions. As described in Section 3.3.2.2, mass flow rate through the engine is calculated for the on-design condition as the mass of air to fill and purge multiplied by the frequency of filling. Frequency of filling is determined by the inverse of the cycle time (t_{cycle}):

$$f = \frac{1}{t_{cycle}} \tag{3.48}$$

Cycle time is the sum of several smaller periods of time: detonation time, blowdown time, purge time, and fill time. Detonation time is approximated as the time to initiate detonation and move the wave down the detonation tube at $M_2 = 1$, where the subscript 2 indicates the detonated fluid. Implementation requires the calculation of the velocity:

$$t_{detonation} = DDT + \frac{l_{tube}}{u_{CJ}} \tag{3.49}$$

Where DDT is the detonation-to-deflagration time, u_{CJ} is the Chapman-Jouguet detonation wave velocity and l_{tube} is the length of the detonation tube. Detonation-to-deflagration time is poorly understood, and is requested as a user input. Ideally, u_{CJ} would be calculated using the speed of sound in the combusted air. In the algorithm coded for this thesis, the properties of the detonated fluid were never completely solved for, thus, the detonation wave velocity was calculated using M_{CJ} multiplied by the speed of sound in the pre-detonation fuel-air mixture.

$$u_{CJ} = M_{CJ} \cdot \sqrt{\gamma_1 R_1 T_{t1}} \tag{3.50}$$

where the subscript 1 indicates that the fluid properties of the fuel-air mixture are taken before detonation. Again, since the pre-detonation fuel-air mixture is essentially at rest, the stagnation properties were used to calculate the speed of sound. It can easily be seen from Equation 3.49 that detonation time is directly proportional to the length of the tube.

Blowdown time ($t_{Blowdown}$) begins after the detonation shock has exited the tube. The shock leaves behind a hot compressed fluid expanding out of the end of the tube. If we assume the tube exit to be choked, then we will see an exponential decay of pressure and mass within the tube. As the mass flows through the choked throat at the end of the tube, the static properties of the gas change, ρ decreases, pressure decreases, and volume remains constant. Our basic steady-flow thermodynamic equation $\dot{m} = \rho A u$ will not allow us to calculate the amount of time it takes to discharge the gasses; this is not steady flow. Instead, we use the relationship identified in an experiment designed by Dutton and Coverdill [33] based off of the pressure ratio across the throat:

$$\Delta t = \frac{\ln\left(\frac{P_2}{P_1}\right)}{\left(\frac{\gamma+1}{2}\right)^{\frac{-(\gamma+1)}{2(\gamma-1)}}} \cdot \frac{Vol}{A_t a_1} \tag{3.51}$$

As seen in Figure 3.20, blowdown time is strongly influenced by the pressure ratio. What is not shown in this plot is how little γ influences blowdown time; its effect is inconsequential over the range of possible specific heat ratios found in an air breathing engine. Significantly, when the tube exit has a constant cross section, and the exit area is the same as that cross section, the trailing term will reduce to l_{tube}/a_1 forcing the blowdown time to be directly proportional to the tube length.

Blowdown time is the most complex piece of the cycle time to calculate. It utilizes a fraction of the pressure ratio calculated using the Heiser and Pratt formulation [4, eq 18]. Figure 3.18 indicates the Heiser and Pratt formula calculates the

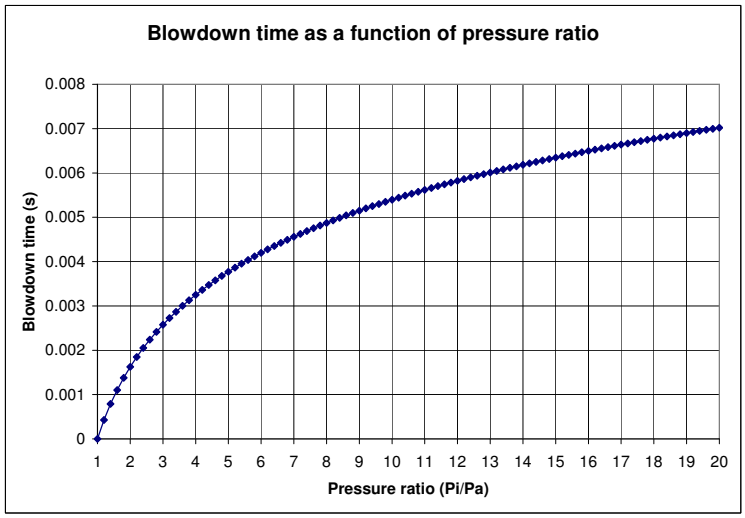


Figure 3.20: Calculated blowdown time plotted as a function of pressure ratio for a 36 inch tube and sonic velocity of 2200 ft/s.

pressure ratio corresponding to the characteristic pressure spike seen in PDE, while the Dyer and Kaemming [5] formulations generates a lower pressure ratio than is seen in the laboratory during blowdown. Attempting to err on the conservative side, we pick the over-estimate of the pressure ratio that yields a longer time. In order to keep the blowdown time reasonable, the pressure ratio is calculated as $\frac{4}{10}$ of the Chapman-Jouguet pressure rise calculated from the formulae from Heiser and Pratt [4].

Purge and fill times are closely related since they both deal with the flow of air into the detonation tube through a valve. As mentioned in subsection 3.3.2.2, we assume that the flow into the tube is constrained by flow through a valve whose throat has an area smaller than the tube cross section area. Once the mass flow rate for choked flow has been established (the maximum mass flow rate) based on a design valve inlet Mach number, purge and fill times (t_{purge} and t_{fill}) become a simple calculation:

$$t_{fill} = \frac{m_{fill\,air}}{\dot{m}_{vo}} \tag{3.52}$$

$$t_{purge} = \frac{m_{purge}}{\dot{m}_{vo}} \tag{3.53}$$

3.3.2.4 Internal Bypass Ratio. The configuration of the hybrid engine sets the main bypass ratio to a fixed value for all cases. This means that there is a constant mass flow rate into the core of the engine. Not all of this core flow will pass through the tubes however. Recall that only a small amount of purge air is needed to act as a buffer between the hot combusted gases of the previous cycle, and the volatile fuel-air mixture in the current cycle. If the baseline engine only required a fuel mass flow rate of 1 lbm/sec, then we expect the hybrid engine to require a similar flow rate. Since the detonation engine operates at a higher equivalence ratio than normal engines, it should require less air to mix with the fuel for a similar enthalpy generation. Balancing the mass flowing through the tubes with the fuel necessitates shunting some of the air around the detonation tubes. Ideally, this internal bypass (iBPR) air provides cooling of the tube walls, and will be mixed with the combustion products to lower the gas temperature before entering the turbines. This internal bypass might be utilized in the transition-to-steady-state device. This may not be desirable since such a connection provides a link to the compressor exit.

Mass flow for the iBPR is assumed to be steady state and, at design conditions, is defined by the formula:

$$iBPR = \frac{\dot{m}_{iBPR}}{\dot{m}_{tube\,air}} \tag{3.54}$$

where air mass flow rate through the internal bypass ($\dot{m}_{iBPR}$) is defined as:

$$\dot{m}_{iBPR} = \dot{m}_{core\,air} - \dot{m}_{tube\,air} \tag{3.55}$$

and where $\dot{m}_{core\,air}$ is the combined mass flow rate coming into the PDC before it is split.

No pressure loss term is applied to the iBPR because it is separated from the core flow after a dry-duct pressure loss term was applied. For the flow going into

the tubes, this dry duct pressure loss is intended to represent pressure loss through a valve.

Heat transfer from the detonation tubes to the iBPR could be accomplished through implementation of a simple heat transfer wall element in NPSS. This element is evoked after the detonation calculations are performed. Another option is to include a heat transfer element imbedded within the PDC element. Either choice will require several inputs:

- areas available to the two fluids
- film transfer coefficients
- fluid inlet properties
- wall mass
- wall specific heat

Attempts were made to include the wall heat transfer element in the model for this thesis, but NPSS solver difficulties caused this option to be pushed aside until more time can be devoted to improving the initial conditions for the solver.

3.3.2.5 Iterations. If the Chapman-Jouguet Mach number is calculated before we calculate mass flow and cycle time, we should be able to carefully craft a PDC routine that does not require iteration. Otherwise, iteration is required to balance the mass flow through the tubes for a given geometry since mass flow depends on cycle time, and cycle time depends on mass flow rate. Recall that total mass flow rate through the tubes is calculated using the air mass required for filling the tube to the designated purge and fill fractions, multiplied by the inverse of the cycle time (see equation 3.44). Fill and purge portions of the detonation cycle are calculated using the mass required to fill the tubes. Blowdown time is only a function of pressure ratio calculated using the Chapman-Jouguet Mach number, and is therefore independent of mass flow if the Mach number can be calculated at the outset. Detonation time is calculated using the DDT and the time it takes the detonation wave to traverse the

length of the tube, leaving it independent of mass flow but dependent on a detonation wave Mach number. As seen in Appendix C, the method of obtaining the shock properties for this thesis allows for a non-iterative solution.

3.3.2.6 Transition to Steady State Flow. Rasheed et al. [21] showed that exhausting a pulsed detonation combustor directly into a turbine lowers the turbine efficiency, and Rasheed et al. [18] discussed the structural ramifications affecting engine life. Since the pulsed flow into the turbine is unfavorable, the hybrid model is based on the assumption that the flow into the turbine sections is steady flow. No attempt was made to define the specific architecture that would transition the pulsed flow back to steady flow. Instead, a subelement to the PDC was created that allows for the application of a pressure drop and an enthalpy loss. Pressure loss was chosen because it translates to a diffusion process, whereas the enthalpy loss was chosen because it translates to work rates through equation 3.16.

3.3.2.7 Improvements. It has been said for centuries that "Rome was not built in a day," and the idea it communicates holds true for any engineering endeavor. Just as improvements were made to Rome with each passing year, improvements could be made with this. Some of the improvements that should be considered are:

- Replacing the detonation wave property calculations
- Implementation of a pulsed-detonation performance map built from CFD or experimental data similar to the method proposed by Paxson, [12]and implemented by Tangirala et al. [17]
- Improving blowdown time calculations
- Calculating heat transfer to bypass air
- Calculating heat transfer to purge air

- Calculating compression of purge air due to the shock initially generated by detonation
- Adaptation of the PDC element to allow for off-design performance

It had been hoped that many of these improvements could be included in the model used for the results section, but there came a point where work on the model had to stop so that deadlines for documentation could be met. Any of these changes have the potential to shift all results by several percent, which would then change any conclusions made in this thesis.

IV. Results

This chapter contains not only the data generated using the hybrid engine modeled in NPSS, it also contains data that demonstrate the close agreement of the baseline engine solutions generated using AEDsys and NPSS. The baseline engine comparison is significant because it provides an acceptable foundation for comparing the hybrid engine.

4.1 Comparison of Baseline Turbofan

Both the NPSS and AEDsys models of the baseline high-bypass turbofan were evaluated at on- and off-design conditions. For on-design conditions, reports for the component analysis were generated and engine parameters were carefully matched between the two programs. The close correlation in the results indicate a reasonable understanding of the NPSS program was achieved.

4.1.1 Baseline Engine On-Design Results. Tables 4.1 and 4.2 show the results and input parameters for baseline engine on-design. The off-design point included in these tables is not of any significance, but foreshadows the divergence of the off-design solutions. On-design thrust varies by approximately 1%, and thrust-specific fuel consumption (TSFC) varies by approximately 0.8%. The NPSS results using the janaf thermodynamic package (shown in tables 4.2 and 4.4) are marginally lower than results generated in NPSS by other thermodynamic packages such as GasTbl. It will be noted that the conditions at the inlet are nearly identical and slowly diverge at each station moving through the engine. This trend is even more apparent in the component interface reports in tables 4.3 and 4.4.

4.1.2 Baseline Turbofan Comparison Off-Design. Figure 4.1 shows throttle hooks generated by NPSS and AEDsys across several altitudes and range of thrust levels. The NPSS data was generated using the GasTbl thermodynamics package since it is the easiest to implement. An attempt was made to produce a similar chart using the janaf thermodynamics package. As seen in Figure 4.2, there was some

Table 4.1: AEDsys engine test data showing the design point. It is from this report that the calculated adiabatic efficiency may be taken.

```
 AEDsys (Ver. 4.010) Turbofan -
Separate Exhaust, Dual Spool Date:11/22/2006 9:40:50 PM
 Engine File: C:\AFIT\MENG733\AEDsys\HBtbfan\eng13 vsh.REF

   Input Constants
   Pidmax= 0.9950   Pi b  = 0.9600   Eta b = 0.9950   Pi n  = 0.9850
   Eta cL= 0.8827   Eta cH= 0.8573   Eta tH= 0.9057   Eta tL= 0.9084
   Eta mL= 0.9900   Eta mH= 0.9900   Eta PL= 0.9900   Eta PH= 0.9900
   Eta f = 0.8827   PTO L =    0.0KW PTO H =  105.7KW hPR   =  18400
   Bleed =   1.00%  Cool 1=   5.00%  Cool 2=   5.00%  Pi nf = 0.9800
   Control Limits:  Tt4   = 3200.0   Pi c  =  32.00   Tt3   = 1859.7
   Pt3   =  650.0   %N LP = 110.00   %N HP = 110.00
   ** Thrust Scale Factor =  1.0000

   Parameter                      Reference**            Test**
   Mach Number @ 0                   0.0100             0.0100
   Temperature @ 0                   518.67             518.67
   Pressure    @ 0                  14.6960            14.6960
   Altitude    @ 0                        0                  0
   Total Temp  @ 4                 2900.00            3118.30
   Pi r  / Tau r             1.0001/ 1.0000     1.0001/ 1.0000
   Pi d                              0.9950             0.9950
   Pi f  / Tau f             1.5600/ 1.1540     1.6943/ 1.1848
   Pi cL / Tau cL            1.5600/ 1.1540     1.6943/ 1.1848
   Pi cH / Tau cH           16.6667/ 2.4379    18.4128/ 2.5106
   Tau m1                            0.9711             0.9705
   Pi tH / Tau tH            0.2324/ 0.7163     0.2329/ 0.7199
   Tau m2                            0.9821             0.9811
   Pi tL / Tau tL            0.1991/ 0.6808     0.1737/ 0.6624
   Control Limit                                       PIC Max
   LP Spool RPM (% of Reference Pt)  100.00             109.57
   HP Spool RPM (% of Reference Pt)  100.00             103.86
   Alpha                              8.000              7.693
   Pt19/P19                          1.5213             1.6522
   P0/P19                            1.0000             1.0000
   Mach Number @ 19                  0.8011             0.8819
   Pt9/P9                            1.1323             1.1873
   P0/P9                             1.0000             1.0000
   Mach Number @ 9                   0.4351             0.5138
   Mass Flow Rate @ 0               1500.00            1667.02
   Corr Mass Flow @ 0               1499.88            1666.89
   Flow Area      @ 0               1761.143            1957.117
   Flow Area*     @ 0                30.428             33.814
   Flow Area  @ 8 + 18               27.442             27.442
   MB - Fuel/Air Ratio (f)          0.02523            0.02825
   Overall Fuel/Air Ratio (fo)      0.00249            0.00289
   Specific Thrust  (F/m0)            27.43              30.55
   Thrust Spec Fuel Consumption (S)  0.3274             0.3407
   Thrust (F)                         41145              50934
   Fuel Flow Rate                     13472              17355
   Propulsive Efficiency (%)           2.47               2.23
   Thermal Efficiency    (%)          34.76              37.04
   Overall Efficiency    (%)           0.86               0.83
```

Table 4.2: On design engine parameters for NPSS using janaf thermodynamics in a format similar to the AEDsys engine test format. Polytropic efficiencies displayed here (e_f, e_{cL}, e_{cH}, e_{tH}, e_{tL}) are different than those input into the AEDsys program.

```
 NCP         NPSS_1.6.3 - Rev: B     model:CmpareAEDsys.mdl with  mixers
 run by:        Ionio     solutionMode= STEADY_STATE
 converge=    1 case:    1    time:  0.000   timeStep:0.0500
 therm_package: Janaf iter/pas/Jac/Broy= 11/ 20/ 1/ 9 run: 11/22/06 21:51:14

Design Values
 Pidmax= 0.99500  Pi b  = 0.96000    Eta b = 0.99500  Pi n  = 0.98500
 Eta f = 0.88270 Eta cL= 0.88270 Eta cH= 0.85730 Eta tH= 0.90570  Eta tL= 0.90840
  e f  = 0.88982  e cL = 0.88982   e cH= 0.89984   e tH= 0.88414    e tL= 0.88912
  gam2 = 1.40023  gam25= 1.39843   gam3= 1.35088   gam4= 1.27704   gam45= 1.30092
  Eta mL= 0.99000  Eta mH= 0.9900                              gam5=1.32930
                 PTO L =    0.0kW  PTO H =  106.8kW  hPr   =   18400
 Bleed =   1.00%  Cool 1=   5.00%   Cool 2=   5.00%   Pi nf =  0.98000
  **  Thrust scale DOES NOT APPLY...

                              CASE 0: DESIGN    CASE 1: OFFDESIGN
 Converged? (0 - no, 1 - yes)              1                   1
 Mach Number at 0                     0.0100              0.0100
 Temperature at 0                    518.6700            518.6700
 Pressure    at 0                     14.6960             14.6960
 Altitude    at 0                      0.0000              0.0000
 Total Temp  at 4                      2900.0              3118.3
 Pi r / Tau r                  1.0001/ 1.0000      1.0001/ 1.0000
 Pi d                                 0.9950              0.9950
 Pi f/ Tau f                   1.5600/ 1.1534      1.6221/ 1.1867
 Pi cL/ Tau cL                 1.5600/ 1.1534      1.7073/ 1.1870
 Pi cH/ Tau cH                16.6667/ 2.3727     17.4925/ 2.4219
 Tau m1                               0.9760              0.9757
 Pi tH/ Tau tH                 0.2372/ 0.7466      0.2411/ 0.7534
 Tau m2                               0.9849              0.9841
 Pi tL/ Tau tL                 0.2030/ 0.7078      0.1821/ 0.6952
 LP Spool RPM (\% reference pt)       100.000             106.219
 HP Spool RPM (\% reference pt)       100.000              98.957
 Alpha                                 8.000               7.528
 Pt19/P19                              1.5213              1.5818
 P0/P19                                1.0000              1.0000
 Mach Number at 19                     0.7979              0.8367
 Pt9/P9                                1.1778              1.2340
 P0/P9                                 1.0000              1.0000
 Mach Number at 9                      0.4995              0.5690
 Mass Flow Rate at 0                1500.0000           1569.9409
 Corr Mass Flow at 0                1499.9050           1569.8415
 Flow Area at 0 (ft2)               1756.6819           1838.5912
 Flow Area* at 0 (ft2)                30.3544             31.7698
 Flow Area at 8 and 18 (ft2)          26.4735             26.4735
 MB - Fuel/Air Ratio (f)              0.02527             0.02845
 Overall Fuel/Air Ratio (fo)          0.00250             0.00297
 Specific Thrust  (F/m0)              28.0445             30.0347
 Thrust Spec Fuel Consumption (S)      0.3248              0.3601
 Thrust (Fn)                       41546.2151          46607.8897
 Fuel Flow Rate (lbm/hr)           13496.0711          16783.0420
 Propulsive Efficiency (%)             2.4480              2.2861
 Thermal Efficiency    (%)            35.3755             34.1603
 Overall Efficiency    (%)             0.8660              0.7809
```

Table 4.3: Component interface for AEDsys model on design conditions.

Design Interface Quantities (Version 4.010) Date:11/22/2006 9:29:25 PM
Filename: C:\AFIT\MENG733\AEDsys\HBtbfan\Test1h.AED
Engine File Name: C:\AFIT\MENG733\AEDsys\HBtbfan\eng13 vsh.REF
Thrust Scale Factor = 1.0000

Station	$\dot{m}$ (lbm/s)	γ	Pt (psia)	Tt (R)	P (psia)	T (R)	Mach	Velocity (ft/s)	Area (ft^2)	Area* (ft^2)	I (lbf)
0	1500.00	1.3986	14.697	518.68	14.696	518.67	0.0100	11.18	1761.14	30.428	3727487.9
1	1500.00	1.3986	14.697	518.68	9.644	460.00	0.8000	842.40	31.591	30.428	83144.5
2	1500.00	1.3986	14.624	518.68	12.330	494.06	0.5000	545.64	40.975	30.581	98186.8
13	1500.00	1.3981	22.813	598.56	19.708	573.95	0.4640	545.64	29.780	20.998	109950.3
core	166.67	1.3981	22.813	598.56	19.708	573.95	0.4640	545.64	3.309	2.333	12216.7
bypass	1333.33	1.3981	22.813	598.56	19.708	573.95	0.4640	545.64	26.471	18.665	97733.6
2.5	166.67	1.3981	22.813	598.56	19.708	573.95	0.4640	545.64	3.309	2.333	12216.7
3.0	166.67	1.3546	380.213	1425.49	357.841	1402.91	0.3015	545.64	0.445	0.218	25779.3
3.1	148.33	1.3546	380.213	1425.49	357.841	1402.91	0.3015	545.64	0.396	0.194	22943.6
3.2	148.33	1.3535	376.410	1425.49	378.031	1423.88	0.0800	145.80	1.438	0.196	78942.1
MB fuel	3.7421										
4	152.08	1.2952	365.004	2900.00	199.221	2529.99	1.0000	2376.71	0.302	0.302	19905.9
4.1	160.41	1.2952		2829.75							
4.4	160.41	1.2952	84.841	2099.67							
4.5	168.74	1.3201	84.841	2068.49	46.012	1786.21	1.0000	2016.10	1.209	1.209	18582.3
5	168.74	1.3410	16.894	1456.12	13.377	1372.29	0.6000	1068.65	6.026	5.047	17213.0
8	168.74	1.3388	16.640	1456.12	14.696	1410.99	0.4351	785.08	7.677	5.124	20364.4
18	1333.33	1.3535	22.357	598.56	14.696	530.66	0.8011	906.08	19.765	19.046	78161.6

Table 4.4: NPSS component interface data utilizing the janaf thermodynamics package.

NCP NPSS_1.6.3 - Rev: B model:CmpareAEDsys.mdl run by: Ionio
solutionMode= STEADY_STATE converge= 1 case: 0
time: 0.000 timeStep:0.0500 therm_package: Janaf Mode:DESIGN
$iter/pas/Jac/Broy = 5/7/1/3$ run: 11/22/06 21:56:58

Summary Output Data

MN	alt	dTs	W	Fg	Fn	TSFC	Wfuel	WAR	OPR	
0.010	0.0	0.00	1500.0	42066.8	41546.2	0.3248	13496.07	0.0000	25.870	

Station input	$\dot{m}$ (lbm/s)	γ	Pt (psia)	Tt (R)	P (psia)	T (R)	Mach	Velocity (ft/s)	Area (ft^2)	I (lbf)
FL 0	1500.00	1.40023	14.697	518.68	14.696	518.67	0.0100	11.17	1756.682	3718053.18
FL 1	1500.00	1.40023	14.624	518.68	12.327	493.96	0.5000	544.86	40.871	97952.09
FL 1.2	1333.33	1.40023	14.624	518.68	12.327	493.96	0.5000	544.86	36.330	87068.52
FL 1.3	1333.33	1.39884	22.813	598.23	19.233	569.79	0.5000	584.96	25.019	93531.76
FL 1.7	1333.33	1.39884	22.813	598.23	19.233	569.79	0.5000	584.96	25.019	93531.76
FL 1.8	1333.33	1.39884	22.356	598.23	14.696	530.71	0.7979	901.15	19.796	79237.46
FL 2	166.67	1.40023	14.624	518.68	12.327	493.96	0.5000	544.86	4.541	10883.57
FL 2.5	166.67	1.39884	22.813	598.23	19.233	569.79	0.5000	584.96	3.127	11691.47
FL 3	166.67	1.35485	380.212	1419.42	357.661	1396.84	0.3015	543.67	0.444	25661.88
FL 3.1	148.33	1.35485	380.212	1419.42	357.661	1396.84	0.3015	543.67	0.395	22839.07
FL 4	152.08	1.28191	365.004	2899.96	0.000	0.00	0.0000	0.00	0.000	0.00
FL 4.1	160.42	1.28465	365.004	2830.26	0.000	0.00	0.0000	0.00	0.000	0.00
FL 4.4	160.42	1.30687	86.565	2113.04	0.000	0.00	0.0000	0.00	0.000	0.00
FL 4.5	168.75	1.30880	86.565	2081.19	0.000	0.00	0.0000	0.00	0.000	0.00
FL 7	168.75	1.33625	17.573	1473.06	13.902	1388.26	0.6000	1072.49	5.820	17276.39
FL 8	168.75	1.33625	17.309	1473.12	14.696	1413.45	0.4995	900.35	6.677	18853.06

BURNERS	TtOut	eff	dPqP	LHV	Wfuel	FAR				
BRN36	2900.00	0.9950	0.0400	N/A	3.74891	0.02527				

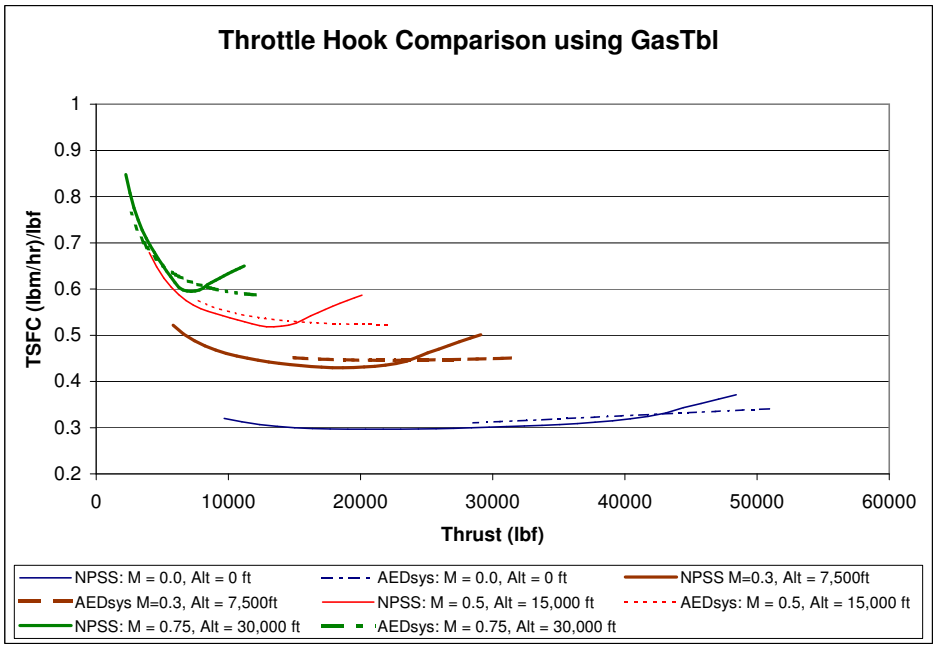


Figure 4.1: Throttle hook comparison of AEDsys and NPSS using the GasTbl thermodynamic package.

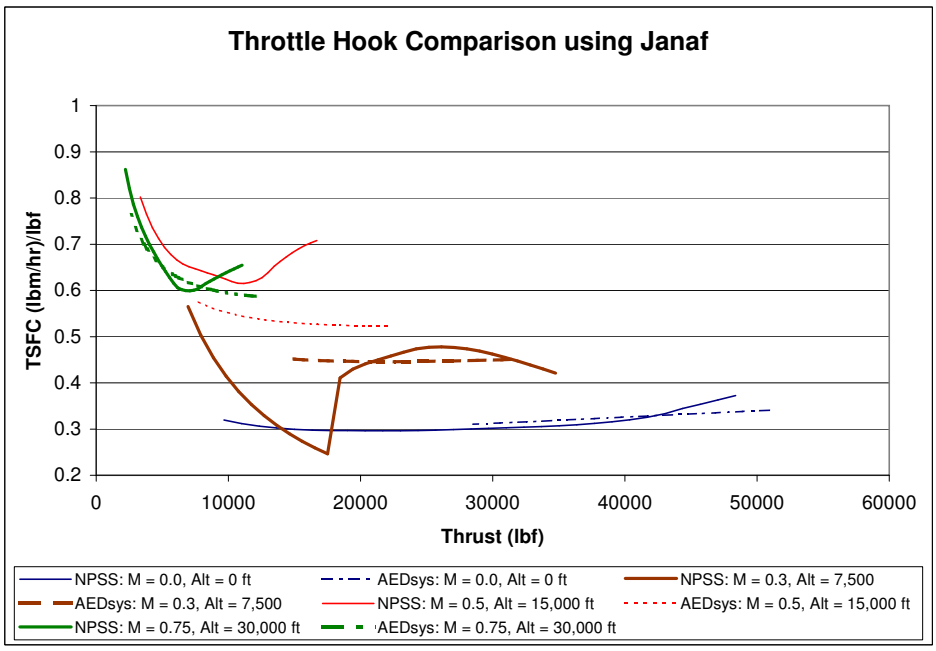


Figure 4.2: Throttle hook comparison of AEDsys and NPSS (using janaf thermodynamics routine) indicating model convergence problems.

difficulty in this implementation. It is unknown why the NPSS solver converged on the solutions at 7,500 and 15,000 feet that are so different than the data generated with an identical model using the GasTbl package. One explanation may be that there are subtle differences in the required inputs for the different thermodynamic packages that were not addressed. Further help from the NASA/GRC group should clear up any remaining difficulties and render the two charts nearly identical.

4.1.3 Explanation of Differences Between AEDsys and NPSS. The two programs display good agreement at the design point, and the off-design performance near the design rpm of the spool also matches closely. Differences for on- and off-design conditions are both noticeable and traceable. A detailed examination of the discrepancies between the two programs' solutions follows.

4.1.3.1 On-design. As seen in tables 4.1 and 4.2, there is an understandable 1% difference between the two net thrust calculations at design conditions. Part of the difference is due to a variation in temperature calculation across the compressor and turbine components as discussed in sections 3.2. The rest of the difference can be calculated by using the specific heat and enthalpy model variations as discussed in section 3.2.1.

The component reports displayed in tables 4.3 and 4.4 show the gradual divergence of total temperature as calculated by the two programs. In order to isolate the difference in power due to variations in total temperature solutions generated by each program we use the T_t and FAR values in the component analysis and calculate enthalpy using only one subroutine. As seen in Table 4.5, the NPSS engine cycle has more power being done across the components. The difference in engine thrust could be balanced by applying the average power difference to the respective spool: approximately 185 to 195 hp on the low pressure spool and 580 to 600 hp on the high pressure spool. The positive number indicates that the power should be applied to the NPSS generated solution, since AEDsys calculated the greater power. However,

Table 4.5: Difference in calculated power due to variations in temperature calculations across components in the NPSS and AEDsys engine cycle programs. For this table, the thermodynamic routine in column 2 was used to define the T_t and FAR while unFAIR was used to calculate the enthalpy.

Component	Routine for T_t4	$T_{t\ in}(^\circ R)$ $T_{t\ out}(^\circ R)$	FAR $\dot{m}_{fuel}(\frac{lbm}{s})$	$h_{in}(\frac{BTU}{lbm})$ $h_{out}(\frac{BTU}{lbm})$	$\dot{m}(\frac{lbm}{s})$	$-\dot{W}(hp)$	AEDsys −NPSS
LPC /FAN	FAIR	518.68 598.56	0.00 0.00	123.9208 143.0604	1500	40619.6	
	NPSS (GasTbl)	518.68 598.26	0.00 0.00	123.9208 142.9884	1500	40466.82	152.7812
	NPSS (janaf)	518.68 598.23	0.00 0.00	123.9208 142.9812	1500	40451.54	168.0592
HPC	FAIR	598.56 1425.49	0.00 0.00	143.0604 349.4501	166.67	48669.5	
	NPSS (GasTbl)	598.26 1419.66	0.00 0.00	142.9884 347.9242	166.67	48326.64	342.8601
	NPSS (janaf)	598.23 1419.42	0.00 0.00	142.9812 347.8614	166.67	48313.53	355.9703
HPT	FAIR	2829.75 2099.67	0.023886 3.7421	765.7417 547.8886	160.41	-49443.2	
	NPSS (GasTbl)	2830.35 2112.67	0.023921 3.74753	765.9601 551.6988	160.41	-48628	-815.193
	NPSS (janaf)	2830.26 2113.04	0.023929 3.74891	765.9409 551.8118	160.42	-48601	-842.18
LPT	FAIR	2068.49 1456.12	0.02268 3.742136	538.069 5.6976	168.74	-41152.4	
	NPSS (GasTbl)	2080.72 1472.54	0.022712 3.74753	541.6363 370.1795	168.75	-40936.4	-215.918
	NPSS (janaf)	2081.19 1473.12	0.022721 3.74891	541.7784 370.3408	168.75	-40931.8	-220.518

this will not resolve the differences because there is another variation between the programs causing the numbers to stand apart.

Following the same analysis described in section 3.2.1 we can calculate the difference in power computed across the compressors and turbines. Both the T_t and FAR values were held constant across all thermodynamic routines for this effort in order to isolate the error due only to variation in specific heat models. As seen in 4.6, the power difference generated by variations of the specific heat/enthalpy model is smaller and in the opposite direction of the temperature induced difference. The average power difference for the low and high pressure spools are -(42 to 52) hp and -(207 to 215) hp respectively. The negative sign indicates that AEDsys has under-

Table 4.6: Difference in calculated power based on variations between specific heat and enthalpy models used for AEDsys and NPSS. Temperatures and FAR values were generated using a rough mean of all solutions, while the enthalpies are calculated using the thermodynamic routine shown in column 2.

Component	Thermo routine	$T_{t\ in}(^\circ R)$ $T_{t\ out}(^\circ R)$	FAR $\dot{m}_{fuel}(\frac{lbm}{s})$	$h_{in}(\frac{BTU}{lbm})$ $h_{out}(\frac{BTU}{lbm})$	$\dot{m}(\frac{lbm}{s})$	$-\dot{W}(hp)$	FAIR −NPSS
LPC /FAN	FAIR	518.68	0.00	123.9208	1500	41352.94	
		600.00	0.00	143.4059			
	NPSS (GasTbl)	518.68	0.00	124.183	1500	41435.39	-82.4525
		600.00	0.00	143.707			
	NPSS (janaf)	518.68	0.00	-6.17665	1500	41444.84	-91.8966
		600.00	0.00	13.3518			
HPC	FAIR	600.00	0.00	143.4059	166.67	48557.77	
		1425.00	0.00	349.3218			
	NPSS (GasTbl)	600.00	0.00	143.707	166.67	48584.44	-26.6705
		1425.00	0.00	349.736			
	NPSS (janaf)	600.00	0.00	13.3518	166.67	48587.08	-29.3116
		1425.00	0.00	219.392			
HPT	FAIR	2830.00	0.023886	765.8178	160.41	-49438.7	
		2100.00	3.7421	547.9847			
	NPSS (GasTbl)	2830.00	0.023921	767.887	160.41	-49825.8	387.1551
		2100.00	3.74753	548.348			
	NPSS (janaf)	2830.00	0.023929	186.544	160.42	-49836.3	397.6378
		2100.00	3.74891	-33.0275			
LPT	FAIR	2075.00	0.02268	539.9567	168.74	-40375.7	
		1475.00	3.742136	370.8385			
	NPSS (GasTbl)	2075.00	0.022712	540.324	168.75	-40399.0	23.35082
		1475.00	3.74753	371.118			
	NPSS (janaf)	2075.00	0.022721	-18.749	168.75	-40369.6	-6.0401
		1475.00	3.74891	-187.832			

calculated power relative to NPSS based on the specific heat model. That is to say, if the power across the NPSS spools is decreased by the magnitude of these averages it will correct this difference.

When the two errors discussed above are summed, the results is a net power difference between the two programs of 131 to 151 hp on the low pressure spool and 372 to 385 hp on the high pressure spool. The positive number indicates that the AEDsys program has calculated a more power done on each spool, and this can be corrected by applying the power difference to the NPSS shaft. Tables 4.7 and 4.8 show the convergence upon a single solution when the two differences due to the variations in the programs are applied. This analysis only corrected for the differences across

Table 4.7: Comparison of thrust after application of extra shaft Power. Here the HPX is the difference between the two estimates described above.

Program	Thermo Routine	F_n (lbf) uncorrected	% Difference	F_n (lbf) w/ Corrective HPX	% Difference
AEDsys	FAIR	41145			
NPSS	GasTbl	41667	1.268684	41292	0.357
	Shaft Power Additions (hp)		LP: 131	HP: 372	

Table 4.8: Comparison of thrust after application of extra shaft power. Here the HPX is the difference between the two estimates described above.

Program	Thermo Routine	F_n (lbf) uncorrected	% Difference	F_n (lbf) w/ Corrective HPX	% Difference
AEDsys	FAIR	41145			
NPSS	janaf	41546	0.974602	41147	0.00486
	Shaft Power Additions (hp)		LP: 151	HP: 385	

the turbines and compressors, neglecting any deviations across ducts, mixers, burners, and nozzles. The results for the janaf package presumably are a closer fit because both janaf and FAIR are derived from the same NIST data, so the uncorrected components are not that different from each other. The GasTbl results do accumulate a 0.353% difference across the components.

4.1.3.2 Off-Design. In Figure 4.1 the throttle hooks generated by NPSS and AEDsys mimic each other, sharing a few common points near what would be considered 100% design shaft RPM. The two results diverge as the thrust requirements move the engine away from this central point. This is caused by differences in off-design assumptions. Mattingly [3, p 142] describes the assumptions implemented in AEDsys in detail. The third stated assumption is: off-design component efficiencies do not change from their on-design values. On the other hand, NPSS requires efficiency maps be used for off-design performance analysis. These maps allow the retrieval of an efficiency for the given conditions that may be different than the design condition. The solution for an off-design calculation might iterate several times before converging on a final efficiency number. Compressor and turbine efficiencies corresponding to the data displayed in Figure 4.1 is shown in Figure 4.3 and Figure

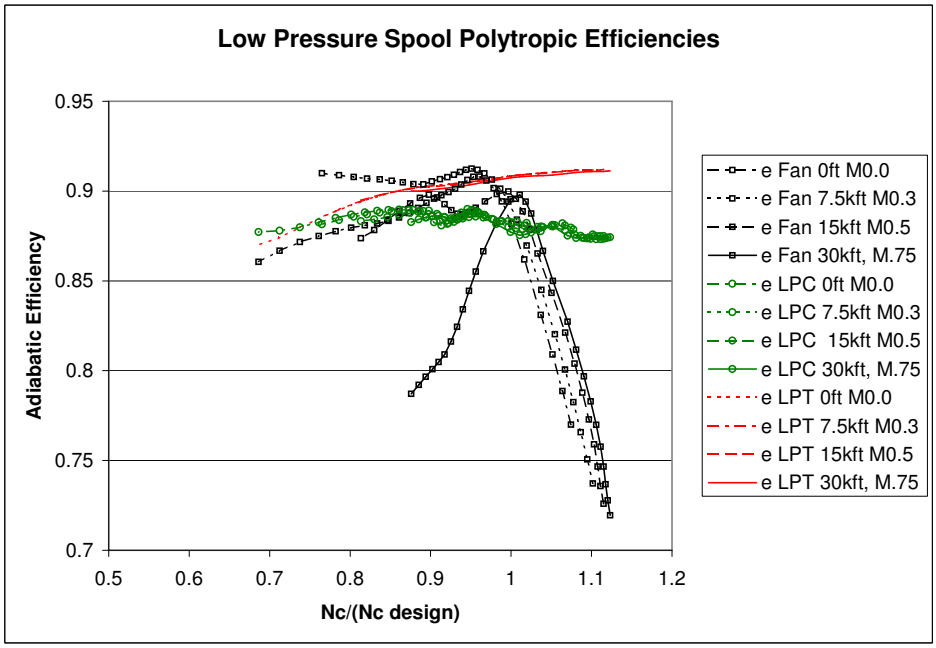


Figure 4.3: Low pressure spool component adiabatic efficiencies returned from the NPSS model using the GasTbl thermodynamic routine.

4.4. Note the severe drop in efficiency experienced by the fan above the 100% design speed. The dramatic change in efficiency is enough to account for the 10% divergence between the two sets of off-design data at high RPM, and the 4% difference at lower RPM.

Another potential source of difference in off-design performance is the implementation of control limits. Limits set in the controls section of AEDsys may be implemented in NPSS, but will affect the performance solution differently due to the differences in thermodynamic routines, off-design assumptions, and efficiency calculations. For the data contributing to Figure 4.1, no limits were imposed upon the NPSS model and the AEDsys model was only constrained by a maximum 110% of on-design rotational speed.

4.2 PDC Hybrid Results

Care was taken to balance the design of the hybrid engine to ensure a legitimate comparison with the baseline high bypass turbofan. Vigilance was required to verify

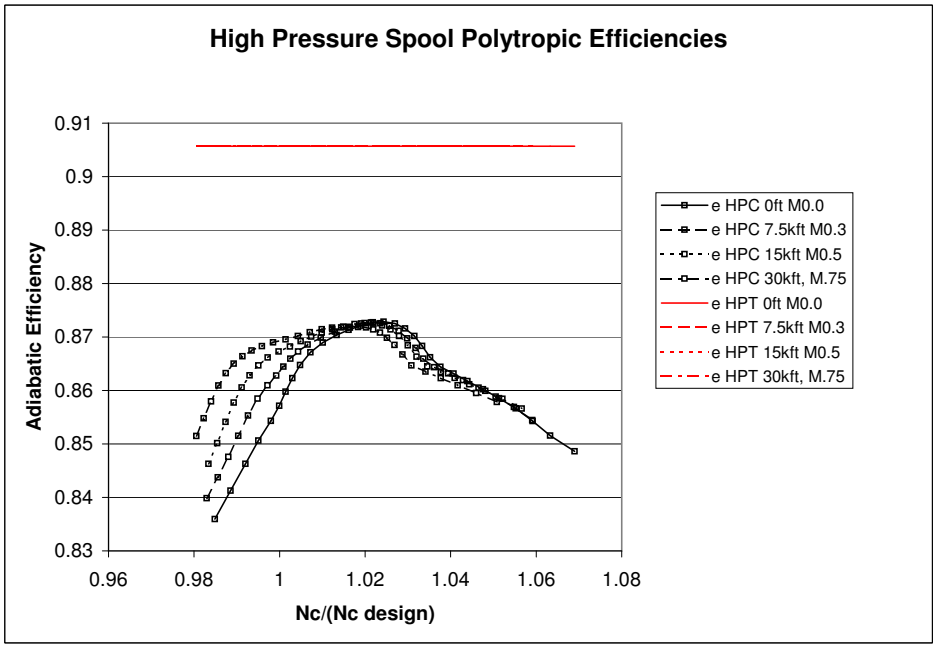


Figure 4.4: High pressure spool adiabatic efficiencies returned by NPSS model using the GasTbl thermodynamics.

the increased benefit apparent in the hybrid results do not stem from a violation of a design limit imposed only on the baseline engine. Most importantly, the total temperature at turbine inlet, and overall pressure ratio were monitored to prevent them from violating the limits for the baseline engine. This is important for on-design performance, but will be crucial in the future off design and mission performance analyses.

4.2.1 Sizing the PDC. Parametric studies were performed to understand the trends associated with the input variables. For the studies in this section we used an initial PDC configuration of 24 tubes as described in Table 4.9. The tubes were run at an $80 + 20$ 100% fill+purge cycle where the 100% indicates the fill, purge, and fuel masses completely fill the tube. When the results indicated that a 100% purge+fill cycle was not possible the solution was considered invalid and not included in the plots. The results had a few surprises.

Table 4.9: Initial configuration and calculated properties for the pulsed detonation combustor used to generate the trade studies in the results section.

Input Variables	Value	Units
Inner Diameter	2.0	in.
Length	36.0	in
Number of Tubes	24	
AR_{valve}	0.5	
M_{valve}	1.0	
ϕ	0.8992	
pf	0.20	
ff	0.80	
Output Parameter	**Value**	**Units**
iBPR	1.45684	
Frequency	59.608	Hz
Fn	27337.6	lbf
TSFC	0.395653	1/hr
T_{t4}	2635.88	R
OPR	38.214	

The parametric studies varied a single design input variable incrementally. The effect on the internal bypass ratio (iBPR), frequency, net thrust (Fn), thrust specific fuel consumption (TSFC), turbine inlet temperature (T_{t4}), and the overall pressure ratio (OPR) were recorded. Experience indicates that the mixer may return a negative Mach number for the secondary inlet flow without returning a program error, so that quantity was included in all output data in order to ensure that a solution was achieved. Also, since the cycle was run at 100%, fill fraction was calculated as $ff = 1.0 - pf$ instead of varying it independently.

4.2.1.1 Tube Diameter. The first thing that needed clarification was how the bypass ratio affected thrust, TSFC, T_{t4}, and overall pressure ratio. It was decided to control this by changing the detonation tube diameter. As seen in Figure 4.5, there is an optimal bypass ratio for the engine when all other parameters are fixed. At this optimal iBPR, the TSFC will hit a minimum. At greater bypass ratio

both thrust and TSFC make unfavorable turns, and below this bypass ratio both the net thrust and the TSFC climb gradually.

There were some issues varying the diameter, and the NPSS code only converged on solutions over a surprisingly small range of diameters. Smaller diameters were limited because the flow through the tube choked. At higher diameters, all the engine core flow was being sucked into the tubes, and none was going into the bypass air, resulting in a hotter turbine inlet temperature and preventing a 100% purge+fill cycle. The range of tube diameters should increase if the tube length is decreased from 36 inches to something shorter.

4.2.1.2 Tube Length. It had been hypothesized that a shorter tube length would allow for increased frequency and greater thrust. However, when we varied tube length and plotted the results seen in Figure 4.6 it was apparent that there would be a benefit to utilizing longer tubes. Not only did thrust increase despite the falling frequency, but there was minimal impact to the total temperature at turbine inlet. If tube length were varied in conjunction with tube diameter to maintain a constant volume, this chart would look significantly different. Perhaps that is something that can be done in the near future.

4.2.1.3 Number of Tubes. Future pulsed detonation combustors will have to deal with the issue of how many tubes to utilize. This has greater ramifications than simply determining iBPR. If sequential firing of some sort is desired, and an effort is made to keep the detonation pulses symmetric about the shafts, it will be advantageous to have a number of tubes easily divisible by $1/\tau_{vo}$ because that term determines how many tubes are required to have a quasi-constant feed from the inlet plenum into the tubes. Also, if firing more than one tube to balance the PDC component, it will help to have an array of tubes divisible by 2, 3, 4, 6, etc.

Number of tubes has the same affect on each of the propulsive properties that the tube diameter did. However, the steps between points are not continuous, since

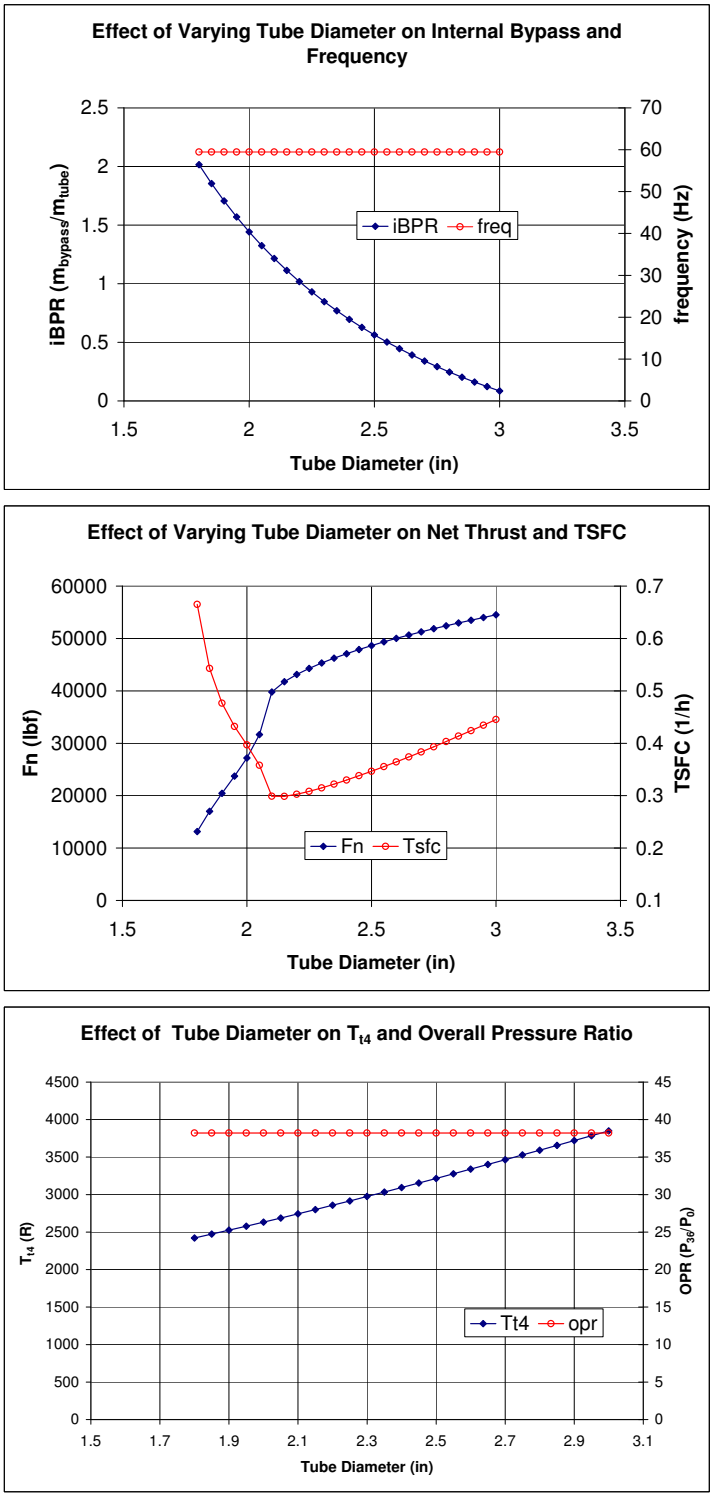


Figure 4.5: Tube diameter is one method of controlling mass flow through the detonation tubes. There is an internal bypass ratio that optimizes the TSFC. Increasing mass flow through the tubes also increases the fuel flow, and raises the temperature of the core flow.

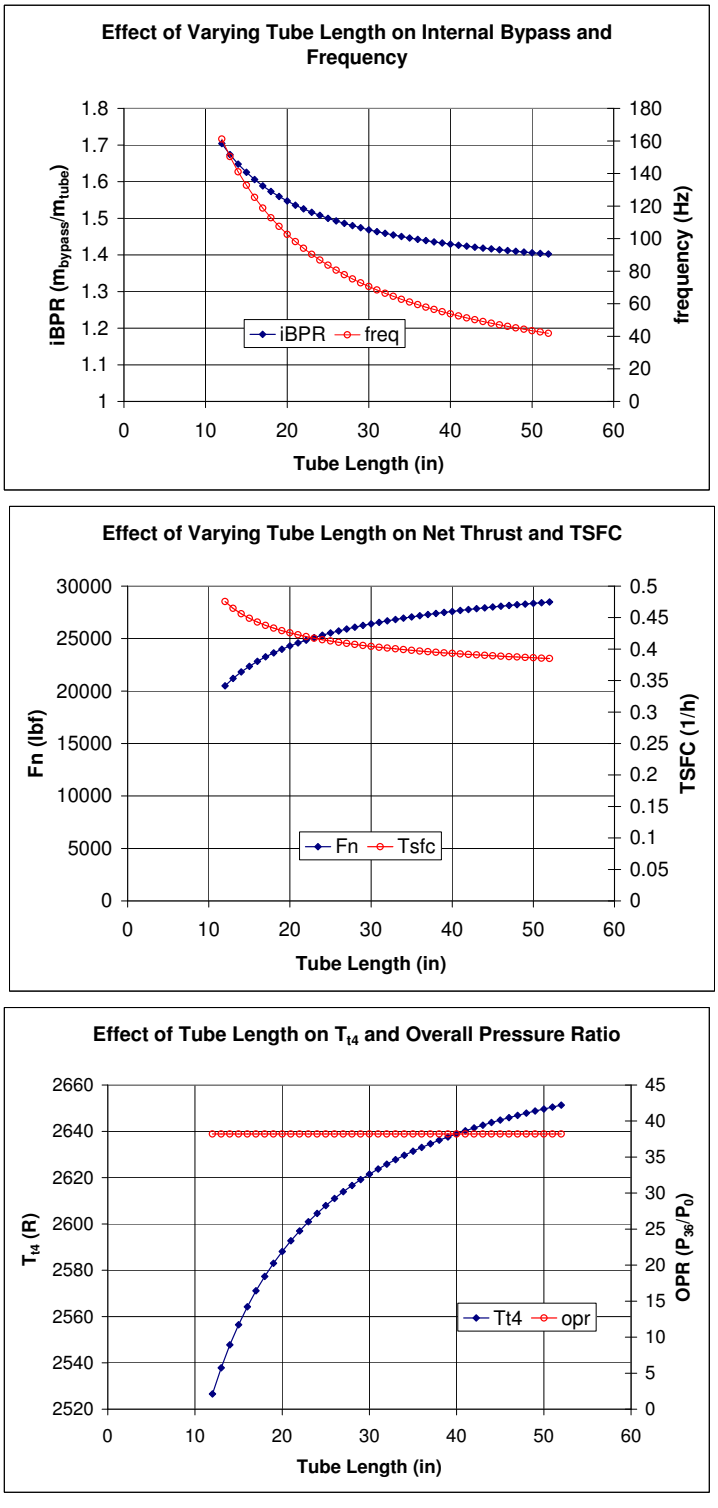


Figure 4.6: Increasing tube length slows the cycle time and allows for greater thrust without a drastic impact on total temperature at turbine inlet.

it is not possible to have $4\frac{1}{2}$ or 13.33333 tubes. It will be necessary to balance the iBPR with a combination of tube diameter and number of tubes.

4.2.1.4 Valve Inlet Area Ratio. One of the most important design choices that could be made is the size of the valves. Mass flow into the tubes is governed by the area of the valve throat. As seen in Figure 4.8, when the area is significantly smaller than the detonation tube diameter, the net thrust and thrust specific fuel consumption will suffer. It should not be assumed that an area ratio of 0.6 is magical simply because the break point in Figure 4.8 is at 0.6; the problem is more complex, and that number might shift if another parameter (such as fill fraction) is modified.

It should be noted that Figure 4.8 indicates that there is a significant benefit to be obtained in this design option, because the turbine inlet temperature does not react drastically, even though the net thrust and TSFC in Figure 4.8 do.

4.2.1.5 Valve Inlet Mach Number. Not surprisingly, the detonation tube inlet valve throat allows better Fn and TSFC for higher Mach numbers. The Mach number and area at the valve throat define the fill time required to fill the tube, which impacts frequency. If geometry is held constant, frequency will define thrust. It appears that most of the benefit of a high filling rate can be obtained by a valve throat Mach number of 0.8 for this configuration. It is anticipated that this trend would continue for other configurations.

4.2.1.6 Compressor Pressure Ratio. Since overall pressure ratio was not changed by any of the previous parametric studies, it was decided to see what effect varying the high pressure compressor had. Figure 4.10 shows the plot. This is the only set of plots that shows a varying OPR, which was expected. It was unexpected to find the minima in TSFC. The TSFC plot of Figure 4.10 shows similar trends to those of Figures 4.5, 4.7, and 4.8, which indicates that the odd shape is due to the changing internal bypass ratio. The baseline engine's high pressure compressor on-

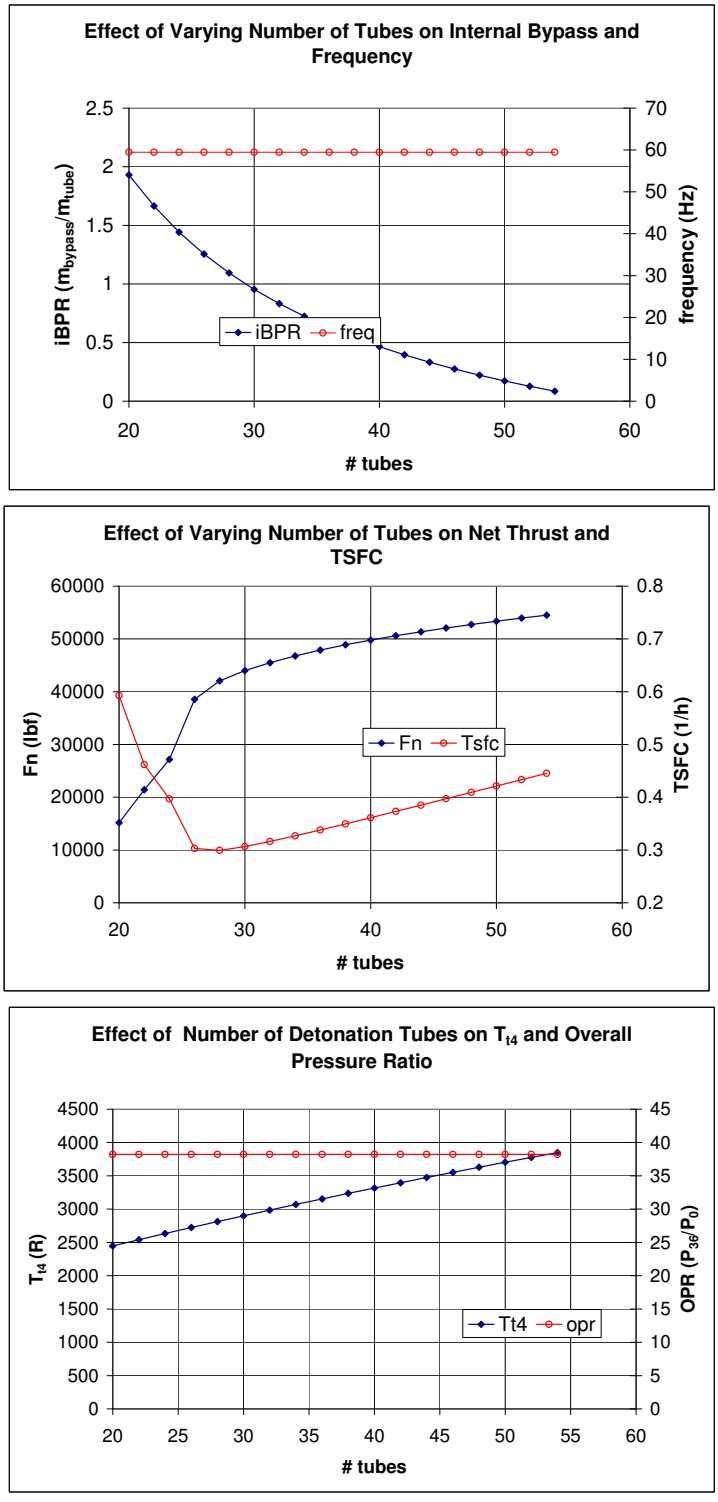


Figure 4.7: Increasing the number of tubes affects the hybrid engine in much the same way as increasing tube diameter. It is the resulting internal bypass that causes the change in performance.

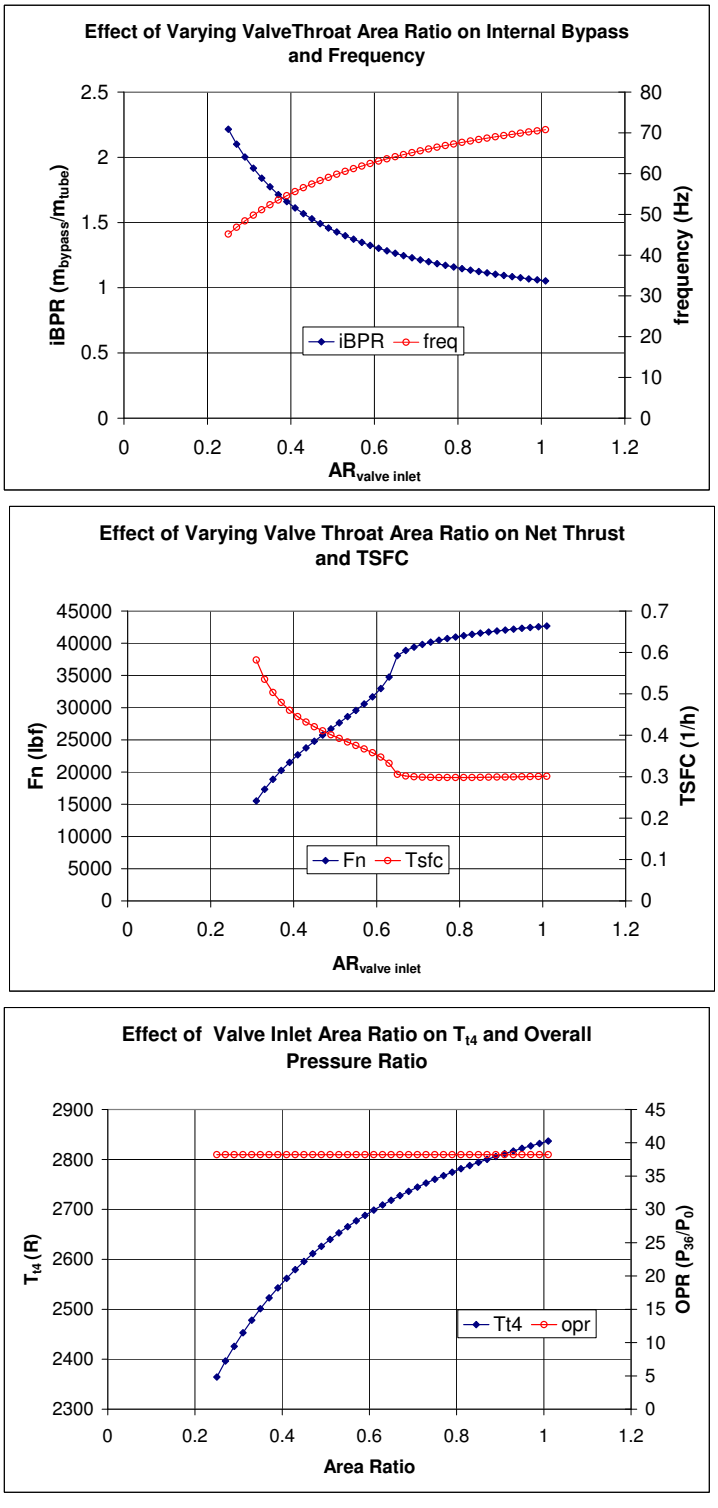


Figure 4.8: Increasing valve inlet area significantly benefits TSFC and F_n while leaving a minimal impact upon T_{t4}.

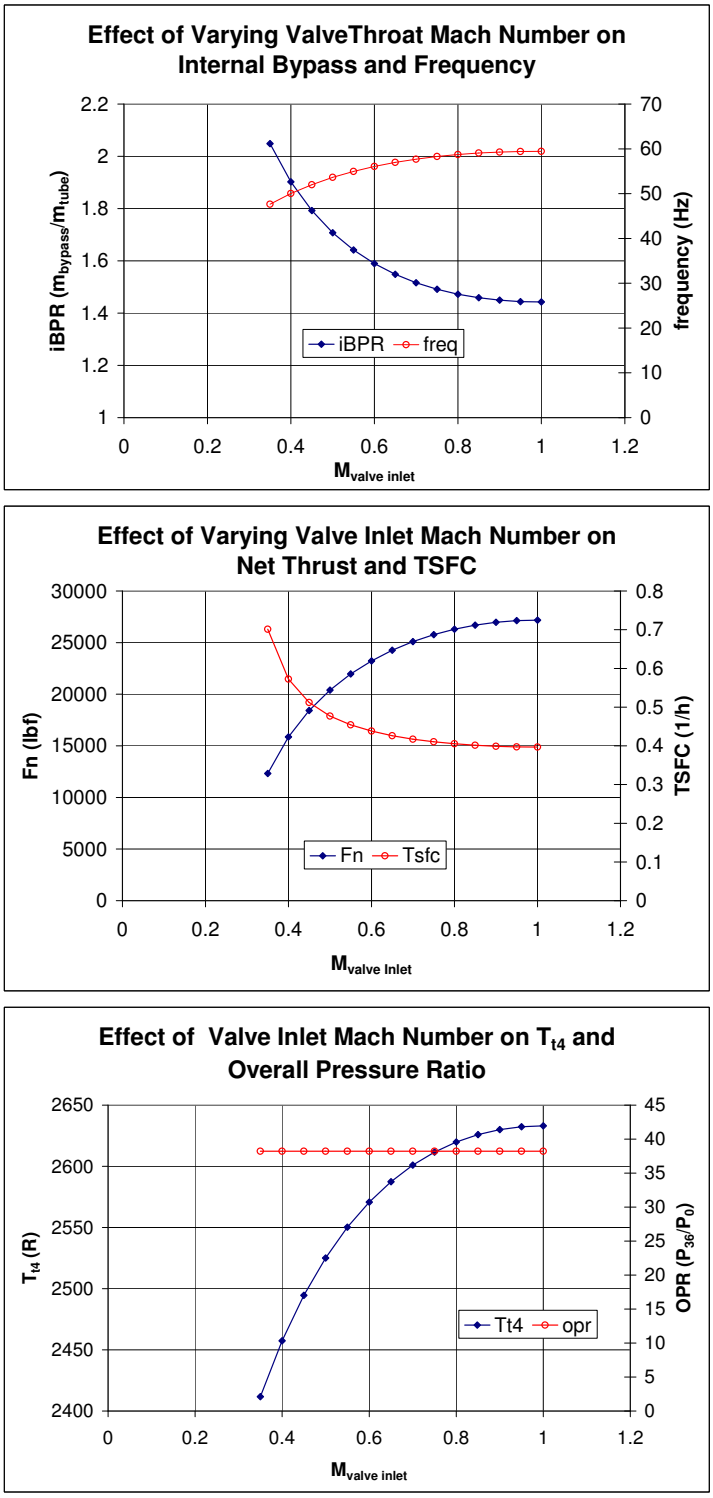


Figure 4.9: Choked flow at valve throat does allow more fill mixture to flow through the tubes (lower iBPR), which causes a rise in Tt4. TSFC however, hits a minimum at Mach 0.6 for the selected configuration.

design pressure ratio is 16.66667, which is still on the left-hand portion of the TSFC curve. Increasing the pressure ratio would result in a lower SFC, but the overall pressure ratio is already rather high. The increase in turbine inlet temperature that accompanies the drop in TSFC should not be ignored. It is possible to drop the fuel consumption in other ways without increasing the pressure ratio. For the problem in this thesis, the compressor pressure ratio in the hybrid engine was matched to the ratio in the baseline engine to keep the power across shafts identical.

4.2.1.7 Mixer Primary Inlet Mach Number. Since the NPSS mixer element has proven so difficult, it was decided to see how varying the inlet mach number to the mixer affected the hybrid engine. Varying this parameter had no effect on iBPR, frequency, OPR, or Tt4. However, there was some difference seen in the TSFC and Fn. Figure 4.11 shows the effect that the inlet Mach number has on the engine. Solutions were sought between a Mach number of 0.15 and 1.0, but only achieved above 0.85. At Mach numbers below this value, the solver returned negative values for the secondary inlet flow velocity. Since the mixer uses an energy balance to find the mixed total temperature there is no reason that changing the Mach number at inlet should affect a change in T_{t4}. However, since the mixer routine performs an impulse balance, we do expect to see a change in impulse as inlet Mach numbers vary. This effect should cascade down to the net thrust calculation.

4.2.1.8 Varying Fill, Purge, and Equivalence Ratio in a 100% Cycle. Figures 4.12 and 4.13 show how varying the purge/fill fraction line and equivalence ratio affect the net thrust and TSFC. This has important ramifications for the leap to off-design studies, since this is the anticipated method of thrust control. It should also be noted that the optimal TSFC does not occur at an equivalence ratio of unity.

4.2.1.9 Component Interface Report. A quick search was made based on the results of the parametric studies for a possible hybrid configuration. It was found that a bank of 24 tubes with the characteristics shown in Table 4.12 provided

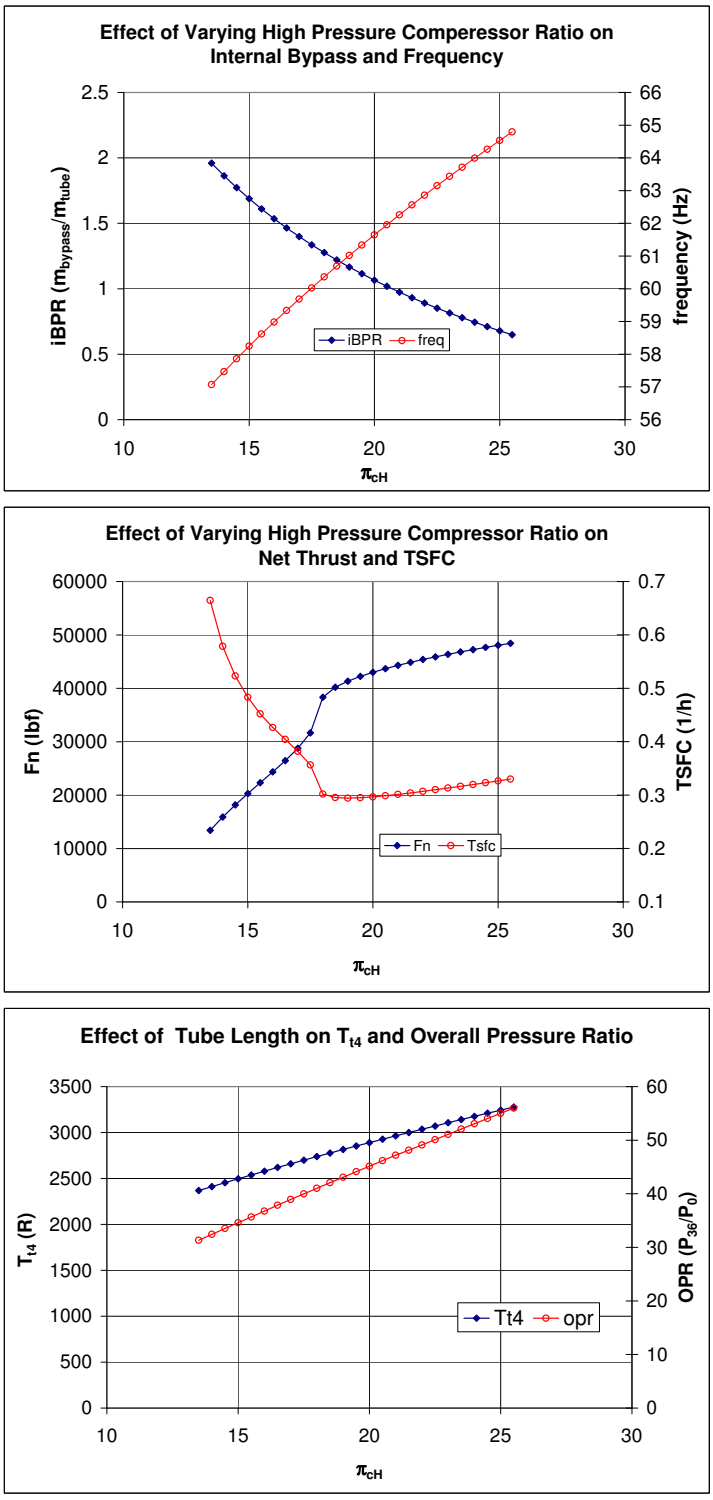


Figure 4.10: Varying the compressor pressure ratio increases T_{t4} while also increasing the frequency and thrust. Internal bypass decreases with increasing compressor ratio.

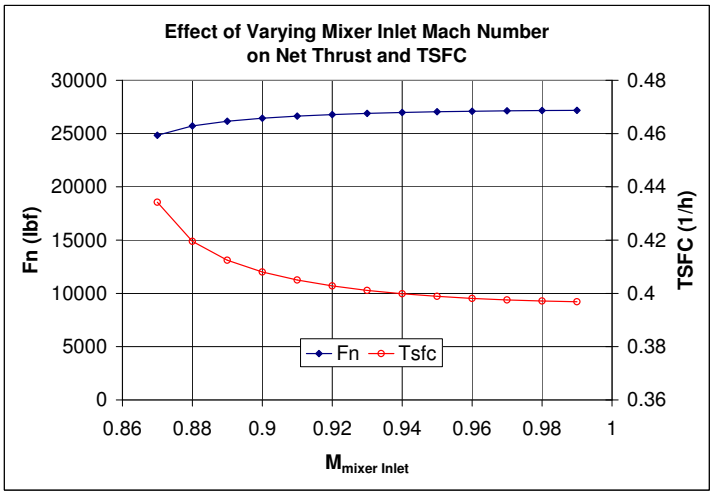


Figure 4.11: Mixer primary inlet Mach number affects only the thrust and TSFC. Presumably this is due to the change in impulse terms generated.

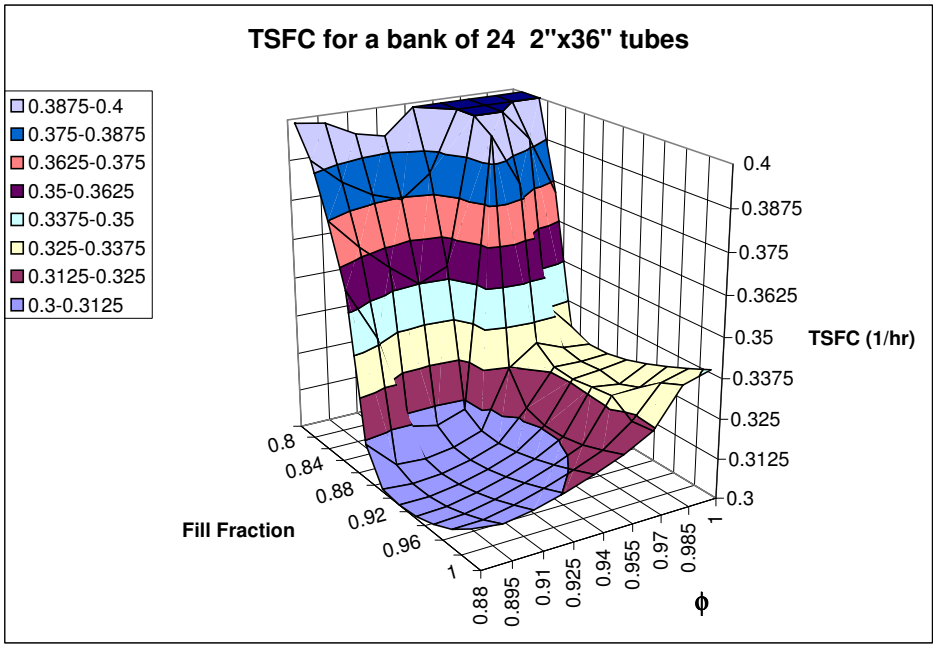


Figure 4.12: Thrust specific fuel consumption is plotted as a function of possible control vectors. For on-design performance, there may be a benefit of utilizing equivalence ratios less than unity.

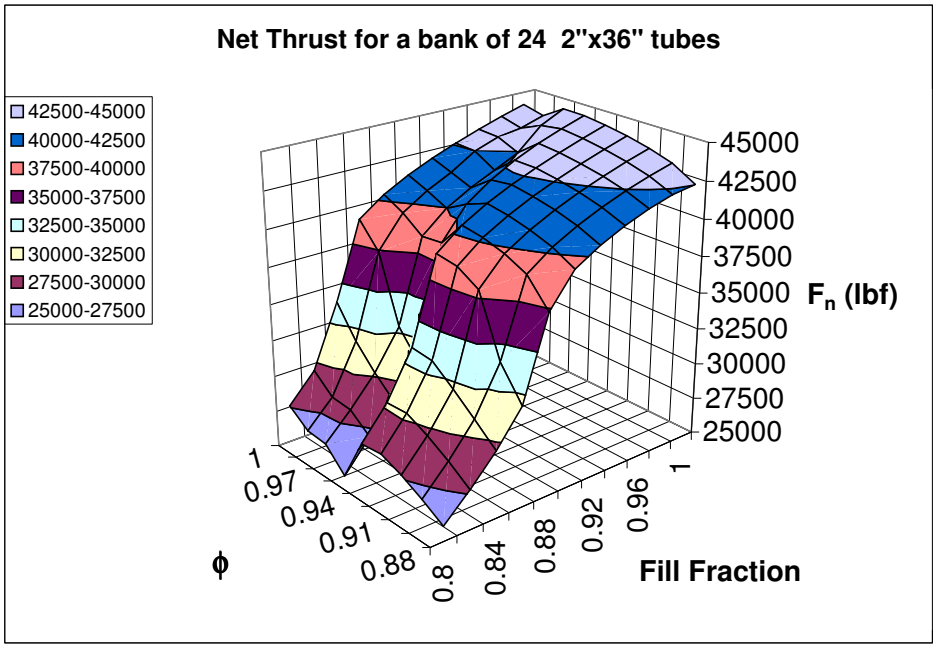


Figure 4.13: Net Thrust plotted as a function of possible control vectors. For on-design performance, Thrust is increased by running with no purge and an equivalence ratio of 1.

an improvement over the baseline engine shown in Table 4.11. This should not be mistaken as an optimized hybrid engine: optimization is a multi-variable non-linear problem requiring more time than is available for this thesis. However, optimization should be done prior to using data to make decisions. The net thrust for the hybrid engine was carefully matched to allow a correct comparison between the two engines. The component interface data is shown in Table 4.12.

4.2.2 Effects of the Transition to Steady State. Shown here is the effect of applying losses through the transition to steady state. For figures 4.14 and 4.15 an array of 24 tubes, with 2.1 inch diameter and 36 inch lengths operating at an 80+20 100% cycle. Area ratio of the detonation tube inlet valves was set at 0.7, with a Mach number of 0.8 was used. Equivalence ratio was set at approximately 90%. It should be compared with the baseline turbofan whose on-design TSFC = 0.3268 1/h and Fn = 41667 lbf.

Table 4.10: Improved configuration of the PDC for the hybrid engine based off of the parametric studies.

Parameter	Value	units
Tube Inner Diameter	2.10	inches
Tube Length	36.0	inches
Number of Tubes	24	
AR_{valve}	0.70	
M_{valve}	0.80	
ϕ	0.9012	
pf	0.2349	
ff	0.7651	
Output Parameter	**Value**	**Units**
iBPR	1.05101	
Frequency	64.906	Hz
Fn	41677.6	lbf
TSFC	0.298583	1/hr
T_{t4}	2798.11	R
OPR	38.02	

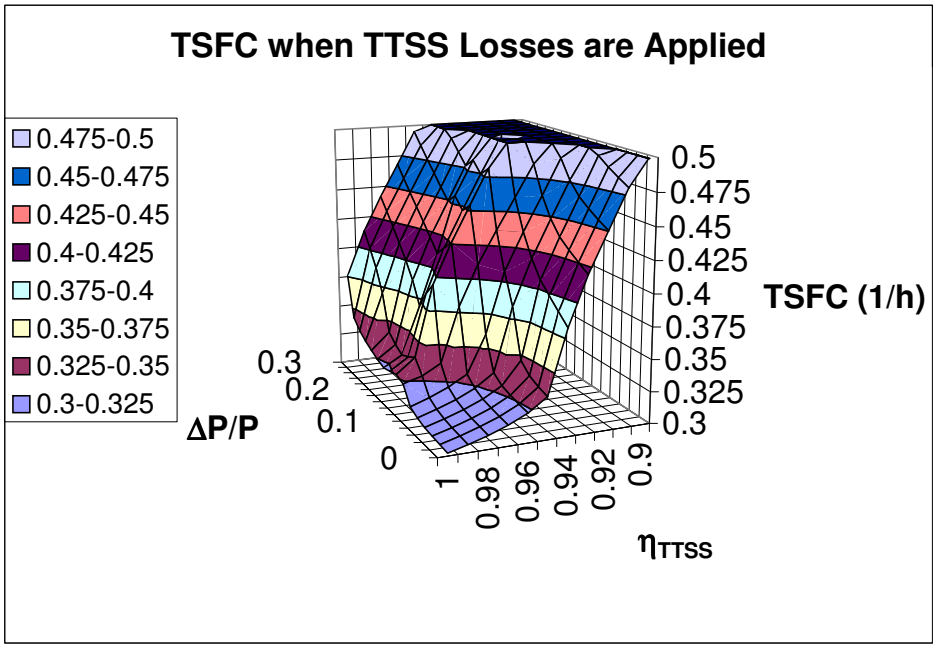


Figure 4.14: Thrust specific fuel consumption is plotted as a function of combustion temperature retention (efficiency) and pressure loss as the flow exiting the PDC transitions from pulsed to steady state.

Table 4.11: NPSS component interface data using the GasTbl thermodynamic package.

NCP NPSS_1.6.3 - Rev: B model:Baseline HBTF bl run by: Ionio
solutionMode= STEADY_STATE converge= 1 case: 0
time: 0.000 timeStep:0.0500 therm_package:GasTbl Mode: DESIGN
$iter/pas/Jac/Broy = 5/7/1/3$ run: 02/26/0718 : 01 : 01

Summary Output Data

MN	alt	dTs	W	Fg	Fn	TSFC	Wfuel	WAR	OPR	
0.010	0.0	0.00	1500.0	42187.4	41667.0	0.3238	13491.12	0.0000	25.870	

Station input	$\dot{m}$ (lbm/s)	γ	Pt (psia)	Tt (R)	P (psia)	T (R)	Mach	Velocity (ft/s)	Area (ft^2)	I (lbf)
FL 0	1500.00	1.39978	14.697	518.68	14.696	518.67	0.0100	11.16	1757.394	3719559.18
FL 1	1500.00	1.39978	14.623	518.68	12.329	493.97	0.5000	544.91	40.864	97950.38
FL 12	1333.33	1.39978	14.623	518.68	12.329	493.97	0.5000	544.91	36.324	87067.00
FL 13	1333.33	1.39963	22.812	598.26	19.286	569.80	0.5000	585.02	24.947	93528.29
FL 17	1333.33	1.39963	22.812	598.26	19.286	569.80	0.5000	585.02	24.947	93528.29
FL 18	1333.33	1.39963	22.356	598.26	14.696	530.38	0.8003	903.41	19.734	79200.91
FL 2	166.67	1.39978	14.623	518.68	12.329	493.97	0.5000	544.91	4.540	10883.38
FL 2.5	166.67	1.39963	22.812	598.26	19.286	569.80	0.5000	585.02	3.118	11691.04
FL 3	166.67	1.35513	380.206	1419.66	357.634	1397.08	0.3015	543.78	0.444	25662.10
FL 3.1	148.33	1.35513	380.206	1419.66	357.634	1397.08	0.3015	543.78	0.395	22839.27
FL 4	152.08	1.28210	364.998	2900.00	0.000	0.00	0.0000	0.00	0.000	0.00
FL 4.1	160.41	1.28490	364.998	2830.35	0.000	0.00	0.0000	0.00	0.000	0.00
FL 4.4	160.41	1.30753	86.667	2112.67	0.000	0.00	0.0000	0.00	0.000	0.00
FL 4.5	168.75	1.30942	86.667	2080.72	0.000	0.00	0.0000	0.00	0.000	0.00
FL 7	168.75	1.33651	17.606	1472.54	13.927	1387.77	0.6000	1072.94	5.810	17280.40
BL 3	1.67	1.35513	380.206	1419.66	0.000	0.00	0.0000	0.00	0.000	0.00
FL 8	168.75	1.33651	17.342	1472.54	14.696	1412.26	0.5022	905.45	6.640	18801.27

BURNERS	TtOut	eff	dPqP	LHV	Wfuel	FAR				
BRN36	2900.00	0.9950	0.0400	18400	3.74753	0.02526				

Table 4.12: NPSS component interface data, for the hybrid engine at design conditions using GasTbl.

NCP NPSS_1.6.3 - Rev: B model: PDC HBTF i4 run by: Ionio
solutionMode= STEADY_STATE converge= 1 case: 0
time: 0.000 timeStep:0.0500 therm_package:GasTbl Mode: DESIGN
$iter/pas/Jac/Broy = 6/8/1/4$ run: 02/26/0718 : 11 : 14

Summary Output Data

MN	alt	dTs	W	Fg	Fn	TSFC	Wfuel	WAR	OPR	
0.010	0.0	0.00	1500.0	42189.4	41668.9	0.2978	12406.98	0.0000	38.020	

Station input	$\dot{m}$ (lbm/s)	γ	Pt (psia)	Tt (R)	P (psia)	T (R)	Mach	Velocity (ft/s)	Area (ft^2)	I (lbf)
FL 0	1500.00	1.39978	14.697	518.68	14.696	518.67	0.0100	11.16	1757.394	3719559.18
FL 1	1500.00	1.39978	14.623	518.68	12.329	493.97	0.5000	544.91	40.864	97950.38
FL 12	1333.33	1.39978	14.623	518.68	12.329	493.97	0.5000	544.91	36.324	87067.00
FL 13	1333.33	1.39963	22.812	598.26	19.286	569.80	0.5000	585.02	24.947	93528.29
FL 17	1333.33	1.39963	22.812	598.26	19.286	569.80	0.5000	585.02	24.947	93528.29
FL 18	1333.33	1.39963	22.356	598.26	14.696	530.38	0.8003	903.41	19.734	79200.91
FL 2	166.67	1.39978	14.623	518.68	12.329	493.97	0.5000	544.91	4.540	10883.38
FL 2.5	166.67	1.39963	22.812	598.26	19.286	569.80	0.5000	585.02	3.118	11691.04
FL 3	166.67	1.35513	380.206	1419.66	322.038	1359.00	0.5000	890.15	0.293	18186.40
FL 3.1	148.33	1.35513	380.206	1419.66	322.038	1359.00	0.5000	890.15	0.261	16185.89
Fl 3.9	75.78	1.23755	558.776	3948.84	331.327	3552.86	0.9500	2626.45	0.115	11663.06
Fl 3.92	76.01	1.35513	364.998	1419.66	331.338	1384.05	0.3801	682.50	0.172	9836.24
FL 4	151.79	1.28625	443.527	2798.11	350.626	2654.71	0.6117	1483.89	0.287	21499.31
FL 4.1	160.12	1.28892	443.527	2732.72	255.877	2411.89	0.9500	2203.67	0.254	20323.79
FL 4.4	160.12	1.31257	98.409	2005.42	53.265	1729.44	1.0000	1983.06	0.972	17324.77
FL 4.5	168.46	1.31430	98.409	1978.18	53.229	1704.62	1.0000	1970.10	1.015	18096.57
FL 7	168.46	1.34422	17.883	1360.18	14.130	1280.20	0.6000	1033.46	5.475	16551.60
BL 3	1.67	1.35513	380.206	1419.66	0.000	0.00	0.0000	0.00	0.000	0.00
FL 8	168.46	1.34422	17.615	1360.18	14.696	1298.37	0.5243	909.04	6.070	17604.71

BURNERS	TtOut	eff	dPqP	LHV	Wfuel	FAR				
BRN36	4443.75	0.9950	0.0400	18400	3.44638	0.06161				

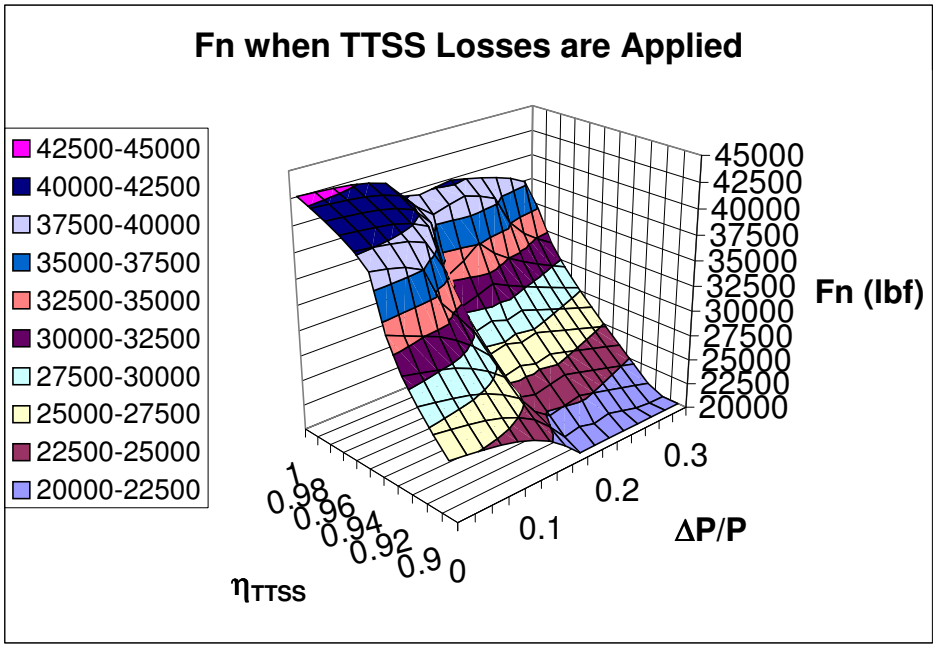


Figure 4.15: Net thrust is plotted as a function of combustion temperature retention (efficiency) and pressure loss as the flow exiting the PDC transitions from pulsed to steady state.

4.2.2.1 Effects on Thrust. Thrust is shown in Figure 4.15 to aid the understanding of the TSFC plot. Turbofan engines do not enjoy a constant TSFC throughout their operating range, and often the TSFC decreases with increasing thrust. This said, the peak of the Fn carpet is slightly higher than our baseline engine. All of the thrust between an efficiency of 0.96 and $\frac{\Delta P}{P}$ of 0.3 are above the baseline engine on-design thrust of 41667 lbf.

4.2.2.2 Effects on TSFC. Thrust specific fuel consumption (TSFC) for the hybrid engine did see an improvement over the baseline engine. Figure 4.14 shows that the transition-to-steady-state can go through a normalized pressure drop of 30% and still match the baseline engine performance for the chosen configuration. Conversely, if no pressure were lost, it would need to retain 96% of its total temperature rise in order to meet that same level of performance. The plot is marginally optimistic since the Fn values corresponding to the TSFC below 0.325 are above the baseline engine's thrust. By adjusting the $ff + pf$ it is possible to balance the hybrid

engine's thrust with the baseline engine's thrust. This would not significantly change the region of improved TSFC over what is shown here, however.

V. Conclusions

Hybrid turbofan engines utilizing a pulsed detonation combustor continue to hold the promise of improving propulsive efficiency. In order to achieve an improvement in efficiency, total fluid losses before passing through the turbine must be minimized.

5.1 Baseline Engine Comparison

In order to make a correct comparison between the hybrid and baseline engine, the baseline engine first had to be shown as acceptable. Comparison efforts demonstrated the agreement of the NPSS model with an industry accepted model in AEDsys. The discrepancies exist due to differences in solution approach, thermodynamic models, and underlying efficiency assumptions.

5.1.1 On-Design Comparison. On-design net thrust and TSFC for the baseline high bypass turbofan engine cycle was shown to agree to within 1.2% between AEDsys and NPSS. When differences in temperature calculations and thermodynamic models are translated into shaft power, the model agreement between the two engines drops to under 0.35%.

5.1.2 Off-Design Comparison. For off-design conditions both programs produce the same trends for Fn and TSFC, but the solutions diverge as the conditions move the shaft RPM from the design point. Close evaluation shows that this divergence is due to Mattingly's [3] assumption of constant efficiency for compressors and turbines. Great care must be taken when utilizing the NPSS package to ensure that the solutions generated are valid.

5.2 Hybrid Engine Performance

Only a well-designed PDC will produce a more efficient engine. If minimal losses are experienced in the transition to steady state flow the hybrid engine will outperform the baseline engine.

5.2.1 Geometric Constraints. Parametric studies for the initial configuration indicated that it is important to keep valve openings relatively large. It is also important to optimize the internal bypass ratio by carefully selecting the number and diameter of detonation tubes. Our optimized hybrid engine held 24 tubes that were each 36 inches long and had an inner diameter of 2.1 inches. The tubes were run on an 80% fill + 20% purge 100% total air cycle using an equivalence ratio of approximately 0.9. Valve throat area was set at 70% of the tube cross section, and mass flowed through the throat at Mach number 0.8.

5.2.2 Promise of a More Efficient Engine. Once the design of the PDC has been optimized, the hybrid engine can be evaluated for potential improvements. The detonation combustor describe in section 5.2 showed that an ideal transition to steady state flow would allow a 8.0% decrease in TSFC while maintaining thrust at design conditions. An optimization exercise should be able to increase that improvement, and enlarge the region in which the engine can experience losses while transitioning the pulsed flow to steady state.

5.3 Recommendations

Hybrid turbofan engines utilizing pulsed detonation combustors hold the promise of improved fuel efficiency and greater thrust. The work documented in this thesis should be continued and improved. A greater range of engine parameters needs to be evaluated in order to achieve the best engine configuration. Measurements of how promising the technology can be will only be realized once an optimized PDC is identified.

There are things that should be done to improve the model. Cycle time calculations need to be improved to reflect experimental data. This may require the implementation of CFD studies or look-up tables. Effects of heat transfer through the tube walls might be improved, and the effect of the detonation wave on the purge fluid sitting in the tube should also be considered. A specific architecture for a device

that transitions the flow to steady state could be introduced and programmed. These are things that cannot be completed before this work must be evaluated, but which might be performed in the coming months.

Appendix A. NPSS Model Code

This is the code that defines the models in NPSS. It shows which thermodynamic package was being used, and which elements are called. These files would be called through use of a run file that tells NPSS what to do with the calculations, and then changes variables as needed to generate data.

Listing A.1: Model file for the baseline high-bypass turbofan engine.

```
//
//-------------------------------------------------------------
//   T U R B O J E T   E N G I N E   B U I L D                |
//                                                            |
//   B U I L D   A N D   V E R I F Y   T U R B O J E T        |
//                                                            |
//   D E S I G N    P O I N T    O N L Y                      |
//                                                            |
//-------------------------------------------------------------
//   T U R B O J E T   C O N F I G U R A T I O N
//-------------------------------------------------------------

cout<<"\t-------------------------------------------------------\n"
    <<"\t Baseline High Bypass Turbfan Built to Match AEDsys \n"
    <<"\t-------------------------------------------------------\n\n"...
       ;

// Set model name
MODELNAME = "Baseline HBTF CmpareAEDsys.mdl with mixers";
//-------------------------------------------------------
// set the thermo package
//-------------------------------------------------------
//  setThermoPackage("GasTbl");
   setThermoPackage("Janaf");

//-------------------------------------------------------
// include the standard intepretted things
//-------------------------------------------------------
#include <InterpIncludes.ncp>
#include "ncp.view"
//#include "bleed_macros.fnc"
//#include "NewDuct.int"

//-------------------------------------------------------------
// #include the definition file for the user defined engine
// performance component
//-------------------------------------------------------------
#include "EngPerf.cmp" ;


//-------------------------------------------------------
// MODEL DEFINITION
```

```
//-------------------------------------------------------
// ##################### FLIGHT CONDITIONS ####################
Element FlightConditions AMBIENT {
            // Specify Design conditions
  alt = 0.0;         // design altitude (ft)
  MN = 0.01;         // design Mach number
  //  Ps = 14.696;   // ambient pressure (psia)
  //  Ts = 59.0;         // ambient temperature (F)
  W = 1500.00;       // design mass flow (lbm/s)
}

//########################## Inlet ############################
Element Inlet INLET {

  eRamBase = 0.995; //Ram Recovery Factor?
}

// ##################### Splitter ##############################
Element Splitter SPLIT {
    BPR =  8.0;  // Bypass Ratio

}

// ######################### FAN ############################
// here the fan represents the outer portion of the
// Low pressure compressor spool
Element Compressor Fan21 {
//  // use these lines if no compressor map is imlemented
//  effDes = 0.88042; //0.882886;
//  PRdes = 1.56;

    // use these lines if compressor map is used...
    #include "fan.map" ; //Compressor sub-element map
    S_map.effDes = 0.8827; //0.88289;
    S_map.PRdes = 1.56;

}

// ################ Bypass Duct/ Nozzle/ Sink #################
Element Duct Bypass13 {
    // AEDsys assumes flow in bypass duct is
    // isentropic (p109, #9)
}

Element Nozzle Noz18 {
    dPqP = 1.0-0.98;
    // pressure loss from nozzle inlet to throat

    PsExhName = "AMBIENT.Fl_O.Ps";

    // AEDsys uses a fixed convergent nozzle
    switchType = "CONIC";
```

```
}

Element FlowEnd NozSink19 {

}

// ############### Low Pressure Compressor ###################
Element Compressor LPC20 {
//  // use these lines if no compressor map is imlemented
//  effDes = 0.88042; // set on-x isentropic efficiency
//  PRdes = 1.56;

    // use these lines if compressor map is used...
    #include "lpc.map";
    S_map.effDes = 0.8827;// set on-x isentropic efficiency
    S_map.PRdes = 1.56;

}

// ################ High Pressure Compressor ###################
Element Compressor HPC25 {
//  // use these lines if no compressor map is imlemented
//  effDes = 0.85755; // set on-x isentropic efficiency
//  PRdes = 16.66667;

    // use these lines if compressor map is used...
    #include "hpc.map" ; // Compressor sub element map
    S_map.effDes = 0.8573 ; // on-x isentropic efficiency
    S_map.PRdes = 16.66667 ; // Set on-x pressure ratio

}

// ################   Bleed starting point   ###################
Element Bleed BLD3 {
    // ========================= BLEEDS ====================
    // Three Bleeds are taken off of the back side of the
    //  High pressure Compressor
    BleedOutPort BL_Cool_301 {
        fracW = 0.05;   // mass flow (5% for cooling HPT)
    }
    BleedOutPort BL_Cool_302 {
        fracW = 0.05;   // mass flow (5% for cooling LPT)
    }
    BleedOutPort BL_Env_303 {
        fracW = 0.01;   // mass flow fraction (1% bleed)
    }

}

// ########################## Fuel ############################
Element FuelStart FUEL32{
    LHV = 18400;    // BTU/lbm - Lower Heating Value of fuel -
```

```
                        // default is 18400 BTU/lbm

    }

// ########################## Burner ########################
    Element Burner BRN36{
      effBase = 0.995;  // component efficiency
      dPqPBase = 1.0 - 0.96;
      //pi b = 1.0-(dP/P) pressure drop acrss burner

      // Change from burner default of FAR to TEMPERATURE
      switchBurn = TEMPERATURE;

      // Total temp. at exit (degrees Rankine)
      //***** not to be used with FAR
      TtCombOut = 2900.0;

    }

// ################### Bleed Mixer/IGV ######################
    Element Bleed MIX40 {
        BleedInPort BlIn40{
            Pscale = 0.88;
        }

    }

    // ###################### HP Turbine ########################
    Element Turbine HPT41 {
        #include "hpt.map";  //High Pressure Turbine Map
        S_map.effDes = 0.9057;//0.90555;0.91075;

    }

// ###################### Bleed Mixer ########################
    Element Bleed MIX44 {
        BleedInPort BlIn44{
            Pscale = 0.68;
        }

    }

    // ###################### LP Turbine ########################
    Element Turbine LPT45 {
        #include "lpt.map"  //Low Pressure Turbine Map
        S_map.effDes = 0.9084;//0.90836;0.90906;

    }

// ########################## Nozzle ########################
    Element Nozzle Noz8 {
        dPqP = 1.0-0.985;
```

```
        PsExhName = "AMBIENT.Fl_O.Ps";
        // AEDsys uses a fixed convergent nozzle for core exit
200     switchType = "CONIC";

    }

    // ######################## Terminate Flow ################
205 Element FlowEnd Sink39 {
        // sink for the environmental bleed...
    }

    Element FlowEnd NozSink9 {
210     // sink for the core airflow
    }

    // %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    // Put shafts in the model
215 // %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    //###################### Low-Pressure Shaft ################
    Element Shaft LPShf {
        ShaftInputPort LPC, FAN, LPT ;
220     Nmech = 2000.0; // Shaft RPM at design point
        inertia = 1.0;
        // inertia is only needed for transient analysis

        HPX = 0.0  +131.;
225     // Horsepower extracted from the shaft hp ( = 325.7 kW)

        fracLoss = 1.0 - 0.99;
        // Fractional loss on positive port torque (1.0 - eta_m)

230 }

    //###################### High Pressure Shaft ###################
    Element Shaft HPShf {
        ShaftInputPort HPT, HPC ;
235     Nmech = 11000.0;
        inertia = 1.0;

        HPX = 143.178 +372;
        // Horsepower extracted from the shaft hp
240     // ( = 105.7 kW)/ eta m ( = 0.99)

        fracLoss = 1.0 - 0.99;
        // Fractional loss on positive port torque (1.0 - eta_m)

245 }

    //###################### Engine Performance ######################
    Element EngPerf PERF{
        // defined in another file to do
```

```
250     //post-processing for viewers
    }

    //-------------------------------------------------------------
    //                   Flow Connections                          //
255 //                                                             //
    //        This is where the flow is defined for the engine     //
    //-------------------------------------------------------------//
    //

260 //############## Ambient to Splitter ##########################
    linkPorts( "AMBIENT.Fl_O",  "INLET.Fl_I",   "FL0" );
    linkPorts( "INLET.Fl_O",    "SPLIT.Fl_I",   "FL1" );

    //##############  Bypass air     ###############################
265 linkPorts( "SPLIT.Fl_02",   "Fan21.Fl_I",       "FLb2" );
    linkPorts( "Fan21.Fl_O",    "Bypass13.Fl_I",    "FLb3" );
    linkPorts( "Bypass13.Fl_O",     "Noz18.Fl_I",       "FLb7" );
    linkPorts( "Noz18.Fl_O",    "NozSink19.Fl_I",   "FLb8" );

270 //############## Core Air Flow   ###############################
    linkPorts( "SPLIT.Fl_01",   "LPC20.Fl_I",       "FL2" );
    linkPorts( "LPC20.Fl_O",    "HPC25.Fl_I",       "FL25" );
    linkPorts( "HPC25.Fl_O",    "BLD3.Fl_I",            "FL3" );
    linkPorts( "BLD3.Fl_O",         "BRN36.Fl_I",       "FL31");
275 //###############   Fuel Flow   ###############################
    linkPorts( "FUEL32.Fu_O",   "BRN36.Fu_I",       "Fu3" );
    linkPorts( "BRN36.Fl_O",    "MIX40.Fl_I",       "FL4" );
    linkPorts( "MIX40.Fl_O",    "HPT41.Fl_I",       "FL41");
    linkPorts( "HPT41.Fl_O",    "MIX44.Fl_I",       "FL44");
280 linkPorts( "MIX44.Fl_O",    "LPT45.Fl_I",       "FL45");
    linkPorts( "LPT45.Fl_O",    "Noz8.Fl_I",        "FL7" );
    linkPorts( "Noz8.Fl_O",     "NozSink9.Fl_I",    "FL8" );

    //############### Bleed port linkage ##########################
285 //linkBleedCB("BLD3", "MIX40", 0.05, 1.0, 1.0, "BL 1");
    //linkBleedCB("BLD3", "MIX44", 0.05, 1.0, 1.0, "BL 2");
    //linkBleedCB("BLD3", "Sink39", 0.01, 1.0, 1.0, "BL 3");
     linkPorts( "BLD3.BL_Cool_301", "MIX40.BlIn40", "BL 1");
     linkPorts( "BLD3.BL_Cool_302", "MIX44.BlIn44", "BL 2");
290  linkPorts( "BLD3.BL_Env_303",  "Sink39.Fl_I",  "BL 3");

    //$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
    //          Mechanical (Shaft) connections
    // $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
295
    //################ Low-Pressure Spool ########################
    linkPorts("LPC20.Sh_O", "LPShf.LPC",    "LP1");
    linkPorts("LPT45.Sh_O", "LPShf.LPT",    "LP2");
    linkPorts("Fan21.Sh_O", "LPShf.FAN",    "LP3");
300
    //############### High-Pressure Spool ########################
```

```
linkPorts("HPC25.Sh_O", "HPShf.HPC",    "HP1");
linkPorts("HPT41.Sh_O", "HPShf.HPT",    "HP2");

// ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
//   Begin Run Definition
// vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv

cout << "^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n"
     << "               Begin Run Input definitions     \n "
     << "vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv\n\n";


    ==========================================
```

Listing A.2: Model file for the hybrid high-bypass turbofan engine.

```
//
//-------------------------------------------------------------
//   T U R B O J E T   E N G I N E   B U I L D                |
//                                                            |
//   B U I L D   A N D   V E R I F Y   T U R B O J E T        |
//                                                            |
//   D E S I G N    P O I N T    O N L Y                      |
//                                                            |
//-------------------------------------------------------------
//   T U R B O J E T   C O N F I G U R A T I O N
//-------------------------------------------------------------

cout <<"\t-----------------------------------------------------\n"
     <<"\t Hybrid Pulsed Detonation Combustor High Bypass Turbofan...
        \n"
     <<"\t-----------------------------------------------------\n\n";

// Set model name
MODELNAME = "PDC HBTF"; // Pulsed Detonation Combustor
            //  High Bypass Turbofan";
//-------------------------------------------------------
// set the thermo package
//-------------------------------------------------------
  setThermoPackage("GasTbl");
//   setThermoPackage("Janaf");

//-------------------------------------------------------
// include the standard intepretted things
//-------------------------------------------------------
#include <InterpIncludes.ncp>
#include "ncp.view"

//-------------------------------------------------------
// #include the definition file for the user defined engine
// performance component
//-------------------------------------------------------
#include "EngPerf.cmp" ;
```

```
//-------------------------------------------------------
// MODEL DEFINITION
//-------------------------------------------------------
// ################# FLIGHT CONDITIONS ####################
Element FlightConditions AMBIENT {
            // Specify Design conditions
  alt = 0.0;         // design altitude (ft)
  MN = 0.01;         // design Mach number
  W = 1500.00;       // design mass flow (lbm/s)
}

//######################## Inlet ############################
Element Inlet INLET {
  eRamBase = 0.995; //Ram Recovery Factor
}

// ##################### Splitter ##########################
Element Splitter SPLIT {
    BPR =  8.0;  // Bypass Ratio

}

// ####################  FAN  ############################
// here the fan represents the outer portion of the Low
//  pressure compressor spool
Element Compressor Fan21 {
//  // use these lines if no compressor map is used
//  effDes = 0.88042; //0.882886;
//  PRdes = 1.56;

    // use these lines if compressor map is used...
    #include "fan.map" ; //Compressor sub-element map
    S_map.effDes = 0.8827; //0.88289;
    S_map.PRdes = 1.56;

}


// ############## Bypass Duct/ Nozzle/ Sink ################
Element Duct Bypass13 {
    // AEDsys assumes flow in bypass duct is
    //  isentropic (p109, #9)
}

Element Nozzle Noz18 {
    // pressure loss from nozzle inlet to throat
    dPqP = 1.0-0.98;

    PsExhName = "AMBIENT.Fl_O.Ps";
```

```
        // AEDsys uses a fixed convergent nozzle for bypass exit
        switchType = "CONIC";
    }

    Element FlowEnd NozSink19 {

    }

    // ################ Low Pressure Compressor ##################
    Element Compressor LPC20 {
    //  // use these lines if no compressor map is used
    //  effDes = 0.88042; // set on-X isentropic efficiency
    //  PRdes = 1.56;

        // use these lines if compressor map is used...
        #include "lpc.map";
        S_map.effDes = 0.8827;  //set on-X isentropic efficiency
        S_map.PRdes = 1.56;

    }

    // ################## High Pressure Compressor ##################
    Element Compressor HPC25 {
    //  // use these lines if no compressor map is imlemented
    //  effDes = 0.85755; // set on-X isentropic efficiency
    //  PRdes = 16.66667;

        // use these lines if compressor map is used...
        #include "hpc.map" ; // Compressor sub element map
        S_map.effDes = 0.8573; // set on-X isentropic efficiency
        S_map.PRdes = 16.66667 ; // Set the pressure ratio on-X

    }

    // ###############   Bleed starting point   ###################

    Element Bleed BLD3 {
        // ===================== BLEEDS ======================
        // Three Bleeds are taken off of the back side of the
        //  High pressure Compressor
        BleedOutPort BL_Cool_301 {
            fracW = 0.05;
            // mass flow (5% for cooling turbine)
        }
        BleedOutPort BL_Cool_302 {
            fracW = 0.05;
            // mass flow (5% for cooling turbine)
        }
        BleedOutPort BL_Env_303 {
            fracW = 0.01;
            // mass flow fraction (1% environmental bleed)
        }
```

```
}

// ########################## Fuel ############################

Element FuelStart FUEL32{

    LHV = 18400; // BTU/lbm - Lower Heating Value of the fuel
                 // default is 18400 BTU/lbm

}

// ########################## Burner ##########################
#include "PDC_burner.int"
Element PulseDetonationCombustor BRN36{
    effBase = 0.995;
    dPqPBase = 1.0-0.96;
    switchBurn = FAR;
    FAR = (0.0683 * 1.00); //approx. stoichiometric conditions
    purgeFrac = 0.2;
    fillFrac = 0.8;
    lTube = 36;
    n_tubes = 24;
    dTube = 2.0;
}
// ######################## Wall heat exchange #################
//Element Wall WALL38{
// Ahx1 = PI*36; // area of wall inside PDT
// Ahx2 = PI*36*1.02; // area that bypass flow sees
// ChxDes1 = 0.7;//heat transfer film coefficient - blind guess...
// ChxDes2 = 0.7;//
// CpMat = 0.1481;//specific heat of material (titanium @ 2160 R)
// //     # tubes  pi/4  length     oD    iD(in)        rho(lbm/ft...
   ^3) Titanium
// massMat = 36.*(PI/4.*(36./12.)*(2.25**2-2.**2)/144.)*280.93;//...
   mass of material in lbm
//
//}

// ############## PDC bypass mixer/ ############
Element Mixer MIX39{
    Fl_I1.MN = .15;
    Fl_I2.MN = .15;

}

// ########################## Bleed Mixer/IGV ###################
Element Bleed MIX40 {
    BleedInPort BlIn40{
        Pscale = 0.88;
    }

}
```

```
// ########################## HP Turbine ######################
Element Turbine HPT41 {
    #include "hpt.map";  //High Pressure Turbine Map
    S_map.effDes = 0.9057;//0.90555;0.91075;

}
// ########################## Bleed Mixer ####################
Element Bleed MIX44 {
    BleedInPort BlIn44{
        Pscale = 0.68;
    }

}

// ########################## LP Turbine ####################
Element Turbine LPT45 {
    #include "lpt.map"  //Low Pressure Turbine Map
    S_map.effDes = 0.9084;//0.90836;0.90906;

}

// ########################## Nozzle ########################
Element Nozzle Noz8 {
    dPqP = 1.0-0.985;
    PsExhName = "AMBIENT.Fl_O.Ps";
    switchType = "CONIC";
    // AEDsys uses a fixed convergent nozzle for core exit

}

// ########################## Terminate Flow ################
Element FlowEnd Sink39 {
//  BleedInPort BlIn44{
//      Pscale = 0.96;
//  }
    // sink for the environmental bleed...
}

Element FlowEnd NozSink9 {
    // sink for the core airflow
}

// %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
// Put shafts in the model
// %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

//#################### Low-Pressure Shaft ################
Element Shaft LPShf {
    ShaftInputPort LPC, FAN, LPT ;
    Nmech = 2000.0;
```

```
        inertia = 1.0; // inertia only for transient analysis

        // Horsepower extracted from the shaft hp  ( = 325.7 kW)
        HPX = 0.0;

        // Fractional loss on positive port torque (1.0 - eta_m)
        fracLoss = 1.0 - 0.99;
}

//###################### High Pressure Shaft ###################
Element Shaft HPShf {
        ShaftInputPort HPT, HPC ;
        Nmech = 11000.0;
        inertia = 1.0;

        // Horsepower extracted from the shaft hp
        //  ( = 105.7 kW)/ eta m ( = 0.99)
        HPX = 143.178;

        // Fractional loss on positive port torque (1.0 - eta_m)
        fracLoss = 1.0 - 0.99;
}

//################### Engine Performance ####################
Element EngPerf PERF{
}

//_________________________________________________________//
//                  Flow Connections                       //
//                                                         //
//        This is where the flow is defined for the engine //
//_________________________________________________________//
//

//############## Ambient to Splitter ########################
linkPorts( "AMBIENT.Fl_O",  "INLET.Fl_I",   "FL0" );
linkPorts( "INLET.Fl_O",    "SPLIT.Fl_I",   "FL1" );

//#############  Bypass air     ############################
linkPorts( "SPLIT.Fl_02",   "Fan21.Fl_I",       "FLb2" );
linkPorts( "Fan21.Fl_O",    "Bypass13.Fl_I",    "FLb3" );
linkPorts( "Bypass13.Fl_O",     "Noz18.Fl_I",       "FLb7" );
linkPorts( "Noz18.Fl_O",    "NozSink19.Fl_I",   "FLb8" );

//############## Core Air Flow   ############################
linkPorts( "SPLIT.Fl_01",   "LPC20.Fl_I",       "FL2" );
linkPorts( "LPC20.Fl_O",    "HPC25.Fl_I",       "FL25");
linkPorts( "HPC25.Fl_O",    "BLD3.Fl_I",            "FL3" );
linkPorts( "BLD3.Fl_O",         "BRN36.Fl_I",       "FL31");
//##############   Fuel Flow   ############################
linkPorts( "FUEL32.Fu_O",   "BRN36.Fu_I",       "Fu3" );
//linkPorts( "BRN36.Fl_O1",     "WALL38.Fl_I1",     "Wa1" );
```

```
//linkPorts( "BRN36.Fl_O2",     "WALL38.Fl_I2",     "Wa2" );
//linkPorts( "WALL38.Fl_O1",    "MIX39.Fl_I1",      "Fl39");
//linkPorts( "WALL38.Fl_O2",    "MIX39.Fl_I2",      "Fl392");
linkPorts( "BRN36.Fl_O1",   "MIX39.Fl_I1",      "Fl39");
linkPorts( "BRN36.Fl_O2",   "MIX39.Fl_I2",      "Fl392");
linkPorts( "MIX39.Fl_O",    "MIX40.Fl_I",       "FL4");
linkPorts( "MIX40.Fl_O",    "HPT41.Fl_I",       "FL41" );
linkPorts( "HPT41.Fl_O",    "MIX44.Fl_I",       "FL44");
linkPorts( "MIX44.Fl_O",    "LPT45.Fl_I",       "FL45" );
linkPorts( "LPT45.Fl_O",    "Noz8.Fl_I",        "FL7");
linkPorts( "Noz8.Fl_O",     "NozSink9.Fl_I",    "FL8" );

//############### Bleed port linkage ##########################
//linkBleedCB("BLD3", "MIX40", 0.05, 1.0, 1.0, "BL 1");
//linkBleedCB("BLD3", "MIX44", 0.05, 1.0, 1.0, "BL 2");
//linkBleedCB("BLD3", "Sink39", 0.01, 1.0, 1.0, "BL 3");
 linkPorts( "BLD3.BL_Cool_301", "MIX40.BlIn40", "BL 1");
 linkPorts( "BLD3.BL_Cool_302", "MIX44.BlIn44", "BL 2");
 linkPorts( "BLD3.BL_Env_303",  "Sink39.Fl_I",  "BL 3");

//$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
//          Mechanical (Shaft) connections
// $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

//############### Low-Pressure Spool ########################
linkPorts("LPC20.Sh_O", "LPShf.LPC",    "LP1");
linkPorts("LPT45.Sh_O", "LPShf.LPT",    "LP2");
linkPorts("Fan21.Sh_O", "LPShf.FAN",    "LP3");

//############### High-Pressure Spool ########################
linkPorts("HPC25.Sh_O", "HPShf.HPC",    "HP1");
linkPorts("HPT41.Sh_O", "HPShf.HPT",    "HP2");

// ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
//   Begin Run Definition
// vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv

cout << "^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n"
     << "               Begin Run Input definitions     \n "
     << "vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv\n\n";
```

In order to better understand the differences between AEDsys and NPSS, the underlying thermodynamic routines were evaluated. Since the FAIR routine from AEDsys was not available as a callable routine or stand-alone package, it was coded in FORTRAN based on the formulas included in Mattingly [28]. Output was checked with the tabulated results included in that same text.

Listing B.1: Constants module used for the unFAIR subroutine.

```
module constant_module

! Programmer: 1Lt Ionio Andrus
! AFIT/ENY MENG 732
! 16 Apr 05

integer, public, parameter :: RKIND = selected_real_kind(13)

real(kind = RKIND), public, parameter :: zero = 0.0_RKIND
real(kind = RKIND), public, parameter :: one  = 1.0_RKIND
real(kind = RKIND), public, parameter :: two  = 2.0_RKIND
real(kind = RKIND), public, parameter :: three= 3.0_RKIND
real(kind = RKIND), public, parameter :: four = 4.0_RKIND
real(kind = RKIND), public, parameter :: five = 5.0_RKIND
real(kind = RKIND), public, parameter :: six  = 6.0_RKIND
real(kind = RKIND), public, parameter :: seven= 7.0_RKIND
real(kind = RKIND), public, parameter :: eight= 8.0_RKIND
real(kind = RKIND), public, parameter :: nine = 9.0_RKIND
real(kind = RKIND), public, parameter :: ten  =10.0_RKIND
real(kind = RKIND), public, parameter :: half = 0.5_RKIND
real(kind = RKIND), public, parameter :: quarter = 0.25_RKIND
real(kind = RKIND), public, parameter :: tenth  = 0.1_RKIND
real(kind = RKIND), public, parameter :: one_n_half= 1.5_RKIND
real(kind = RKIND), public, parameter :: thousand = 1000.0_RKIND
!real(kind = RKIND), public, parameter :: zero = 0.0_RKIND
!real(kind = RKIND), public, parameter :: zero = 0.0_RKIND
!
real(kind = RKIND), public, parameter :: R_Air =286.96_RKIND !J/...
   kgK

!define some conversion multipliers
real(kind=RKIND),public,parameter::K_to_R = 1.800
real(kind=RKIND),public,parameter::kJ_kg_to_BTU_lbm=0.4299226139
real(kind=RKIND),public,parameter::BTU_lbm_to_ft2_s2 = 25037.00
real(kind=RKIND),public,parameter::BTU_s_to_hp = 1.41485320412

end module constant_module
```

Listing B.2: Main callable unFAIR subroutine.

```
subroutine unFAIR(T, h, Pr, phi, Cp, R, gam, a, FAR)
use constant_module
implicit none

REAL (kind = RKIND) :: A0, A1, A2, A3, A4, A5, A6, A7
REAL (kind = RKIND) :: h_ref, phi_ref, phi_r1, phi_r2
REAL (kind = RKIND) :: T, FAR                    !intent in
REAL (kind = RKIND) :: Pr, R, Cp, h, phi, gam, a  !intent out
REAL (kind = RKIND) :: cp_a, h_a, phi_a, cp_p, h_p, phi_p
character *3 :: flow

!=============================================================
!Check to see if FAR affects
if (FAR .lt. 1.0E-9_RKIND) flow = "no"

!=====  Define coeficients from Table 2.2 for air alone  ======
  A0     =  2.5020051E-01_RKIND
  A1     = -5.1536879E-05_RKIND
  A2     =  6.5519486E-08_RKIND
  A3     = -6.7178376E-12_RKIND
  A4     = -1.5128259E-14_RKIND
  A5     =  7.6215767E-18_RKIND
  A6     = -1.4526770E-21_RKIND
  A7     =  1.0115540E-25_RKIND
  h_ref  = -1.7558886_RKIND !BTU/lbm
  phi_r1  = 0.0454323_RKIND !BTU/(lbm R)

!======   Equations 2.60,2.61, 2.62 for air alone   ===========
cp_a = A0 + A1*T + A2*T**2 + A3*T**3 +  &
     A4*T**4 + A5*T**5 + A6*T**6 + A7*T**7

h_a =  h_ref +A0*T +A1/two*T**2 +A2/three*T**3 +A3/four*T**4 +&
     A4/five*T**5 + A5/six*T**6 + A6/seven*T**7 + A7/eight*T**8

phi_a = phi_r1 +A0*log(T) + A1*T +A2/two*T**2 +A3/three*T**3 +&
     A4/four*T**4 + A5/five*T**5 + A6/six*T**6 + A7/seven*T**7
!=============================================================

!==== Now change coefficients for the products of combustion. =
  A0     =  7.3816638E-02_RKIND
  A1     =  1.2258630E-03_RKIND
  A2     = -1.3771901E-06_RKIND
  A3     =  9.9686793E-10_RKIND
  A4     = -4.2051104E-13_RKIND
  A5     =  1.0212913E-16_RKIND
  A6     = -1.3335668E-20_RKIND
  A7     =  7.2678710E-25_RKIND
  h_ref  = 30.58153_RKIND !BTU/lbm
  phi_r2  = 0.6483398_RKIND !BTU/(lbm R)

!===== Equations 2.60, 2.61, 2.62 for products of combustion =
```

```
   cp_p = A0 + A1*T + A2*T**2 + A3*T**3 +  &
          A4*T**4 + A5*T**5 + A6*T**6 + A7*T**7

55 h_p = h_ref +A0*T +A1/two*T**2 +A2/three*T**3 +A3/four*T**4 +&
         A4/five*T**5 +A5/six*T**6 +A6/seven*T**7 +A7/eight*T**8

   phi_p = phi_r2 +A0*log(T) +A1*T +A2/two*T**2 +A3/three*T**3 +&
          A4/four*T**4 +A5/five*T**5 +A6/six*T**6 +A7/seven*T**7
60 !=============================================================


   !============     Equation 4.26 a,b,c,d    ===================
   R = 1.9857117_RKIND /(28.97_RKIND - FAR*0.946186_RKIND)
65                                                  !BTU/(lbm R)
   Cp = (Cp_a + FAR*Cp_p)/(one+FAR)
   h = (h_a + FAR*h_p)/(one + FAR)
   phi = (phi_a + FAR*phi_p)/(one + FAR)

70 !============     Equation 2.55 - "reduced pressure "  =======
   phi_ref = 1.578420959_RKIND     ! BTU/(lbm R)  phi@492.00 R
                      ! (a hair above freezin')
   Pr = exp((phi - phi_ref)/R)

75 !============    do the additional calculations  =============
   gam = Cp/ (Cp-R)
   a = sqrt(gam*R*BTU_lbm_to_ft2_s2*T)
   !=============================================================

80 !=============================================================
   end subroutine
```

Appendix C. Pulsed Detonation Combustor Code

This appendix contains the NPSS code for the pulsed detonation combustor element used by the NPSS model. The code is based on the work burner element generated by NASA/GRC that was included with the NPSS release 1.6.3.

Listing C.1: NPSS element file for the pulsed detonation combustor.

```
#ifndef __PDC__
#define __PDC__

//************************************************************
// * Air Force Institute of Technology
// * 2950 Hobson Way, Bldg 641
// * Wright Patterson AFB, OH 45433
// *
// * Ionio Q. Andrus, Capt., USAF
// *
// BASED ON "Burner.int" included in NPSS, written by~~
// * NASA Glenn Research Center
// * 21000 Brookpark Rd
// * Cleveland, OH 44135
// *
//************************************************************

#include <InterpIncludes.ncp>

class PulseDetonationCombustor extends Element {

//-------------------------------------------------------------
//     ******* DOCUMENTATION *******
//-------------------------------------------------------------

 title = "";

  description = isA() + " will calculate performance for
  pulsed detonation combustor.";

  usageNotes = "

The burner element performs high level burner performance
calculations.  This element works with an entrance fluid and
fuel stream.  It mixes the two flows together and then
performs the burn calculations.  Please note that the burner
has no control over the actual fuel stream conditions--fuel type,
LHV, etc.  These values are properties of the fuel flow itself
and are usually set in the FuelStart element.

There are two ways to specify the burner exit conditions.  The
first way is specify the burner fuel-to-air ratio.  The second
way is to set equivalence ratio.  The type of input used is
controlled by an option switch.
```

```
The burner tracks several different pressure losses.  The first,
dPqP, accounts for duct friction pressure drops and approximates
the pressure loss through valves.  The second, dPqPRayleigh,
accounts for the Rayleigh pressure drop. dPRayleigh is input or
calculated - see switchHotLoss, an iteration is necessary since
the pressure loss itself is a function of the exit conditions.

The burner also allow two efficiencies to be input.  The first
efficiency, eff, refers to the efficiency based on enthalpy
change.  The second efficiency, effChem, refers to the efficiency
based on temperature change.  Both terms can be input.  However,
the enthalpy efficiency is always applied first.

Additionally,

The user can request a pre burner pressure loss dPqP.  The
pressure loss calculations are performed before all the other
calculations are done.  This means that the combustion entrance
pressure will not match the value indicated by the burner entrnce.

The user can request a heat transfer Qhx.  The heat transfer
calculations are performed after all the other calculations are
done.  This means that if heat transfer is being used, the exit
temperature will not match the value indicated by the burner
calculations.

";

background = "";

  //-------------------------------------------------------------
  //     ******* SETUP VARIABLES ********
  //-------------------------------------------------------------

  real a_dPqP {
    value = 0.0;  IOstatus = "input";  units = "none";
    description = "Duct friction pressure drop adder";
  }
  real a_dPqPAud {
    value = 0.0;  IOstatus = "unset";  units = "psia";
    description = "Audit factor adder applied to pressure ratio";
  }
  real a_eff {
    value = 0.0;  IOstatus = "input";  units = "none";
    description = "Adiabatic efficiency adder";
  }

  real a_effChem {
   value = 0.0;  IOstatus = "input";  units = "none";
   description = "Chemical efficiency adder";
  }
```

```
real ARvalve {    // Added 15Feb2007 - IA
 value = 0.5;  IOstatus = "input";  units = "none";
 description = "Ratio of valve throat area to tube cross section...
     area";
}
real deltaS {     //Added 17Jan2007 - IA
    value = 0.0; IOstatus = "output"; units = "none";
    description = "Change in entropy due to detonation";
}
real DDT {        //Added 17Jan2007 - IA
    value = 0.0005; IOstatus = "input"; units = "none"; //...
       seconds
    description = "Detonation to deflaration time in seconds";
}
real dPqP {
  value = 0.0;  IOstatus = "output";  units = "none";
  description = "Adjusted duct friction pressure drop";
}
real dPqPBase {
  value = 0.0;  IOstatus = "input";  units = "none";
  description = "Duct friction pressure drop ";
}
real dPqPRayleigh {
  value = 0.0;  IOstatus = "input";  units = "none";
  description = "Adjusted Rayleigh pressure drop";
}
real dTube {      //Added 17Jan2007 - IA
    value = 2.0; IOstatus = "input"; units = "none"; // inches...
       ...
    description = "Inside diameter of the detonation tube";
}
real eff {
  value = 1.0;  IOstatus = "output";  units = "none";
  description = "Adjusted adiabatic burner efficiency";
}
real effBase {
  value = 1.0;  IOstatus = "input";  units = "none";
  description = "Adiabatic burner efficiency, from socket ";
}
real effChem {
  value = 1.0;  IOstatus = "input";  units = "none";
  description = "Adjusted chemical efficiency";
}
real effChemBase {
  value = 1.0;  IOstatus = "input";  units = "none";
  description = "Chemical efficiency, from socket";
}
real eqRatio {
  value = 1.0;  IOstatus = "input";  units = "none";
  description = "Equivalence ratio for fuel-air mixture";
}
real FAR {
```

```
    value = 0.0;  IOstatus = "output";  units = "none";
    description = "Fuel-to-air ratio";
  }
  real FARDes {
    value = 0.0;  IOstatus = "output";  units = "none";
    description = "Fuel-to-air ratio at design";
  }
  real fillFrac {  //Added 17Jan2007 - IA
      value = 1.0; IOstatus = "input"; units = "none";
      description = "Fill fraction ";
  }
  real fuelFractV {
    value = 0.0;  IOstatus = "input";   units = "none";
    description = "Fraction of the incoming flow velocity fuel
    enters the burner";
  }
  real iBPR {   //added 17Jan2007 - IA
      value = 1.0; IOstatus = "output"; units = "none";
      description = "Bypass ratio internal to the PDC";
  }
  real iBPRdes {    //added 1Feb2007 - IA
      value = 1.0; IOstatus = "output"; units = "none";
      description = "Bypass ratio internal to the PDC at
      design conditions";
  }
  real lTube {  //added 17Jan2007 - IA
      value = 36; IOstatus = "input"; units = "none"; //inches??
      description = "length of the individual detonation tubes";
  }
  real n_tubes{ //added 17Jan2007 - IA
      value = 36; IOstatus = "input"; units = "none";
      description = "Total number of detonation tubes used
      in the PDC";
  }
  real MCJ {    //added 17Jan2007 - IA
      value = 3.0; IOstatus = "output"; units = "none";
      description = "Chapman-Jouguet Mach number of the
      detonation wave.";
  }
  real Mvalve {     //added 15Feb2007 - IA
      value = 1.0; IOstatus = "input"; units = "none";
      description = "Mach number of flow passing through
      the valve throat.";
  }
  real qadd{    //added 17Jan 2007- IA
      value = 0.0; IOstatus = "output"; units = "none";
      description = "Heat addition due to fuel combustion";
  }
  real Qhx {
    value = 0.0;  IOstatus = "input";  units = "Btu/sec";
    description = "Heat loss to thermal mass storage";
  }
```

```
real PqPRayleigh {
  value = 1.0;  IOstatus = "output";  units = "none";
  description = "Adjusted Rayleigh pressure drop";
}
real PqPRayleighDelta {
  value = 0.0;  IOstatus = "output";  units = "none";
  description = "Bounded Rayleigh pressure drop - for loop only"...
     ;
}
real PqPRayleighError {
  value = 1.0;  IOstatus = "output";  units = "none";
  description = "Adjusted Rayleigh pressure drop error";
}
real PqPRayleighMin {
  value = 0.05;  IOstatus = "input";  units = "none";
  description = "Rayleigh pressure drop lower limit - for loop ...
     only";
}
real PqPRayleighStep {
  value = 0.05;  IOstatus = "input";  units = "none";
  description = "Maximum step for Rayleigh pressure drop
  - for loop only";
}
real PqPRayleighNew {
  value = 1.0;  IOstatus = "output";  units = "none";
  description = "Previous adjusted Rayleigh pressure drop
  - for loop only";
}
real purgeFrac {  //Added 17Jan2007 - IA
    value = 0.25; IOstatus = "input"; units = "none";
    description = "Purge fraction coefficient for flow";
}
real s_dPqP {
  value = 1.0;  IOstatus = "input";  units = "none";
  description = "Duct friction pressure drop scalar";
}
real s_dPqPAud {
  value = 1.0;  IOstatus = "unset";  units = "none";
  description = "Audit factor scalar applied to pressure ratio";
}
real s_eff {
  value = 1.0;  IOstatus = "input";  units = "none";
  description = "Adiabatic efficiency scalar";
}
real s_effChem {
  value = 1.0;  IOstatus = "input";  units = "none";
  description = "Chemical efficiency scalar";
}
real tauBlDn {    // Added 17Jan2007 - IA
    value = 5.; IOstatus = "input"; units="none";
    description = "Blowdown time constant";
}
```

```
real tauValveOpen {   // Added 18Jan2007 - IA
    value = 0.33333; IOstatus = "output"; units="none";
    description = "time valve open/ time cycle - from 0 to 1";
}
real tCycle{      // Added 17Jan2007 - IA
    value = 0.01; IOstatus = "output"; units = "none"; //seconds
    description = "Detonation engine cycle time (= 1/frequency)"...
       ;
}
real tolRayleigh {
  value = 4e-05;  IOstatus = "input";  units = "none";
  description = "Iteration tolerance on momentum pressure drop";
}
real tolWfuel {
  value = 1e-05;  IOstatus = "input";  units = "none";
  description = "Iteration tolerance on temperature burn";
}
real TtCombOut {
  value = 0.0;  IOstatus = "input";  units = "R";
  description = "Exit temperature";
}
real TtLast {
  value = 0.0;  IOstatus = "input";  units = "R";
  description = "Previous exit temperature - for loop only";
}
real TTSSeff{     // Added 17Jan2007 - IA
    value = 1.0; IOstatus = "input"; units = "none";
    description = "Efficiency factor for the transition device."...
       ;
}
real TTSSdPqP{    // Added 17Jan2007 - IA
    value = 0.0; IOstatus = "input"; units = "none";
    description = "Change in Pressure divided by Pressure
    for transistion to steady state calculation.";
}
real tValve{      // Added 17Jan2007 - IA
    value = 0.0002; IOstatus = "input"; units = "none"; //...
       seconds
    description = "Time for valves to open/close";
}
real Wfuel {
  value = 0.0;  IOstatus = "input";  units = "lbm/sec";
  description = "Combustor fuel flow";
}
real WfuelError {
  value = 0.0;  IOstatus = "input";  units = "lbm/sec";
  description = "Combustor fuel flow error";
}
real WfuelLast {
  value = 0.0;  IOstatus = "input";  units = "lbm/sec";
  description = "Previous combustor fuel flow - for loop only";
}
```

```
  real WfuelNew {
    value = 0.0;  IOstatus = "input";  units = "lbm/sec";
    description = "Next combustor fuel flow - for loop only";
300 }
  int countFuel {
    value = 0;  IOstatus = "output";
    description = "Fuel loop counter";
  }
305 int countFuelMax {
    value = 50;  IOstatus = "input";
    description = "Fuel loop maximum counter";
  }
  int countRayleigh {
310   value = 0;  IOstatus = "output";
    description = "Rayleigh loop counter";
  }
  int countRayleighMax {
    value = 25;  IOstatus = "input";
315   description = "Rayleigh loop maximum counter";
  }
  int flagRayleighLossTooMuch {
    value = 0;  IOstatus = "output";
    description = "If true, Rayleigh loop results in too much loss...
       ";
320 }
  int flagRayleighChoked {
    value = 0;  IOstatus = "output";
    description = "If true, Rayleigh loop results in supersonic ...
       flow";
  }
325
  // for backward compatibilty with old "aud"
  FunctVariable a_dPqPaud {
    units = "none"; IOstatus = "input";
    getFunction = "get_aAud"; setFunction = "set_aAud";
330 }
  real get_aAud() { return a_dPqPAud; }
  void set_aAud(real userValue) { a_dPqPAud = userValue; }

  FunctVariable s_dPqPaud {
335   units = "none"; IOstatus = "input";
    getFunction = "get_sAud"; setFunction = "set_sAud";
  }
  real get_sAud() { return s_dPqPAud; }
  void set_sAud(real userValue) { s_dPqPAud = userValue; }
340
  //-------------------------------------------------------------
  //   ******* OPTION VARIABLE SETUP *******
  //-------------------------------------------------------------

345 Option switchAud {
    allowedValues = { "BASE", "AUDIT" }
```

```
    description = "Determines if the audit factors are used";
    IOstatus = "input";
    trigger=TRUE;
}


Option switchBurn {
    allowedValues = { "FAR",  "EQRATIO" }; //"FUEL", "WFUEL", "...
        TEMPERATURE", __ mod 18 Dec 2006 - IA - added "FILLFRACTION...
        "
    description = "Switch determines if burner is running to fuel ...
        flow, FAR, or T4.  Setting option to FUEL will burn using ...
        the burner value as an input.  Setting the option to WFUEL ...
        will burn using the value coming in from the fuel station."...
        ;
    trigger=TRUE;
}

Option switchDes {
    allowedValues = { "DESIGN", "OFFDESIGN" };
    description = "Design switch";
    trigger=FALSE;
}
// input kept in for backward compatible (remove later)
Option switchHotLoss {
    allowedValues = { "INPUT", "CALCULATE","input" };
    description = "Switch determines if the hot pressure loss is ...
        input or iterated on";
    trigger=TRUE;
}

//-------------------------------------------------------
// **** SETUP PORTS, FLOW STATIONS, SOCKETS, TABLES ******
//-------------------------------------------------------

// FLUID PORTS
FluidInputPort Fl_I {
    description = "Incoming flow";
}

FluidOutputPort Fl_O1 {
    description = "Exiting combustion flow";
}

FluidOutputPort Fl_O2 {
    description = "Exiting bypass flow";
}

// FUEL PORTS

FuelInputPort Fu_I {
    description = "Incoming fuel flow";
```

```
}

// BLEED PORTS

// THERMAL PORTS

// MECHANICAL PORTS

// FLOW STATIONS

//__________flow stations modified 18 Dec 2006- IA
FlowStation Fl_Icomb {
      description = "Inlet station to detonation tube section
      of burner (after the initial pressure loss is applied)";
}

FlowStation Fl_IcombAir {
      description = "Copy of the inlet station to detonation ...
         tube
      section of burner(after the initial pressure loss is ...
         applied,
      before flow is split and partitioned)";
}

FlowStation Fl_Iprg {
      description = "Station containing detonation tube purge ...
         fluid";
}

FlowStation Fl_Ocomb {
   description = "Exit station to combustion section of burner
(before thermal storage heat transfer is calculated)";
}

FlowStation Fl_Vit {
   description = "Vitiated Fluid flow station before detonation (...
      cold)";
}
 // ____________________----end flow station modifications

// SOCKETS

Socket S_dPqP {
  allowedValues = { "dPqPBase" };
  description = "Dry duct and valve pressure loss";  //__ mod -...
     IA- 18 Dec 2006
  socketType = "dPqP";
}

Socket S_eff {
  allowedValues = { "effBase", "effChemBase" };
  description = "PulseDetonationCombustor adiabatic efficiency";
```

```
    socketType = "BURN_EFFICIENCY";
}

Socket S_Qhx {
  allowedValues = { "Qhx" };
  description = "Thermal storage socket";
  socketType = "HEATTRANSFER";
}


// TABLES

//-------------------------------------------------------------
// ******* INTERNAL SOLVER SETUP *******
//-------------------------------------------------------------

//-------------------------------------------------------------
//  ******  ADD SOLVER INDEPENDENTS & DEPENDENTS  ******
//-------------------------------------------------------------

//-------------------------------------------------------------
// ******* VARIABLE CHANGED METHODOLOGY *******
//-------------------------------------------------------------
void variableChanged( string name, any oldVal ) {
 // Check to see what variables were changed....
 // Change input/output status as necessary  - IA- 18 Dec 06

  if( name == "switchBurn" ) {
    if ( switchBurn == "FAR" ) {
  FAR.IOstatus = "input";
  Wfuel.IOstatus = "output";
  TtCombOut.IOstatus = "output";
  eqRatio.IOstatus = "output";
    }
//      else if ( switchBurn == "FUEL" ) {
//  FAR.IOstatus = "output";
//  Wfuel.IOstatus = "input";
//  TtCombOut.IOstatus = "output";
  //    }
//      else if ( switchBurn == "WFUEL" ) {
//  FAR.IOstatus = "output";
//  Wfuel.IOstatus = "output";
//  TtCombOut.IOstatus = "output";
  //    }
   //_________ added 5 Feb 2007 -IA-
     else if ( switchBurn == "EQRATIO" ) {
  FAR.IOstatus = "output";
  Wfuel.IOstatus = "output";
  TtCombOut.IOstatus = "output";
  eqRatio.IOstatus = "input";
    }
    //___________ end of additions -IA-
```

```
    }

    else if( name == "switchHotLoss" ) {
      if ( switchHotLoss == "INPUT" ) {
495   dPqPRayleigh.IOstatus = "input";
      }
      else if ( switchHotLoss == "input"  ){ switchHotLoss = "...
         INPUT"; }
      else {
    dPqPRayleigh.IOstatus = "output";
500   }
    }

    else if( name == "switchAud" ) {
      a_dPqPAud.IOstatus  = "inactive";
505   s_dPqPAud.IOstatus  = "inactive";
      if( switchAud == "AUDIT" ) {
    a_dPqPAud.IOstatus  = "input";
    s_dPqPAud.IOstatus  = "input";
      }
510 }
  }


  //-------------------------------------------------------
515 //   ******* PERFORM ENGINEERING CALCULATIONS *******
  //-------------------------------------------------------

  void calcPreLoss() {

520

    //-------------------------------------------------------------
    // Check to see if the pressure sockets are empty, if not ...
       thenexecute
    //-------------------------------------------------------------
    if ( !S_dPqP.isEmpty() ) {
525   S_dPqP.execute();
    }
    dPqP = dPqPBase * s_dPqP + a_dPqP;  // calculate pressure ...
       losses (dry duct and Valve)
    if( switchDes == "OFFDESIGN" ) {
      if( switchAud == "AUDIT" ) {
530   dPqP = dPqP * s_dPqPAud + a_dPqPAud;
      }
    }
       //comment -IA- Collect total enthalpy at inlet
    real hin = Fl_I.ht;
535 real Pin = ( 1 - dPqP ) * Fl_I.Pt;  //coment -IA- apply ...
       pressure losses as calculated above
       //comment -IA- copy flow to combustor flow
    Fl_Icomb.copyFlowStatic( "Fl_I" );
    Fl_Icomb.setTotal_hP( hin, Pin );
```

```
}

void calcBurn() {

  real TtCombOutTemp;
  real htStoich;
  real WFuelLimit;
  real WFuelHeat;

  Fl_Ocomb.copyFlow( "Fl_Icomb" );

  //-------------------------------------------------------
  // Efficiency
  //-------------------------------------------------------
  if ( !S_eff.isEmpty() ) {
    S_eff.execute();
  }
  eff = effBase * s_eff + a_eff;
  effChem = effChemBase * s_effChem + a_effChem;

  //-------------------------------------------------------
  // Burn
  //-------------------------------------------------------
  Fl_Ocomb.burn( "Fu_I", eff );

  //-------------------------------------------------------
  // if inputting a PW type of efficiency adjust the temperature
  //-------------------------------------------------------
  if ( effChem < 1.0 ) {
    TtCombOutTemp = effChem *( Fl_Ocomb.Tt - Fl_Icomb.Tt ) +
  Fl_Icomb.Tt;
    Fl_Ocomb.setTotalTP( TtCombOutTemp, Fl_Icomb.Pt ); // use ...
       Pin
  }

}

void calcRayleighLoss() {

  flagRayleighChoked = 0;
  flagRayleighLossTooMuch = 0;

  PqPRayleigh = 1.0;
  PqPRayleighError = 0.0;


  //-------------------------------------------------------
  // self-convergent iteration loop for internal momentum ...
     pressure drop calc
  //-------------------------------------------------------
  for( countRayleigh=0; countRayleigh<=countRayleighMax;
```

```
 countRayleigh++) {

  //-------------------------------------------------------
  // input or output dPqPRayleigh
  //-------------------------------------------------------
  if( switchHotLoss == "INPUT" ) {
PqPRayleigh = 1.0 - dPqPRayleigh;
  }
  else if( switchHotLoss == "CALCULATE" ) {
dPqPRayleigh = 1.0 - PqPRayleigh;
  }


  //-------------------------------------------------------
  // calculate momentum pressure drop
  //-------------------------------------------------------
  real PtCombOut = PqPRayleigh * Fl_Icomb.Pt;

  Fl_Ocomb.setTotal_hP( Fl_Ocomb.ht, PtCombOut );


  //-------------------------------------------------------
  // Check momentum pressure drop
  //-------------------------------------------------------
  PqPRayleighNew = PqPRayleigh;

  if ( switchHotLoss == "CALCULATE" ) {


//-------------------------------------------------------
// make this thing a constant area burner
//-------------------------------------------------------
Fl_Ocomb.A = Fl_Icomb.A;
flagRayleighChoked = 0;
if( Fl_Ocomb.MN > 1.0 ) {
  // when MN > 1.0 FlowStation static calc is
  //                        not consistent with Area
  //        Fl_Ocomb.MN = 1.0;
  // do not do this - creates major iteration problems
  flagRayleighChoked = 1;
}


//-------------------------------------------------------
// Calculate the exit static pressure from the momentum eqn
// assume the fuel has the same velocity as the entrance flow
//-------------------------------------------------------
real PsMomMeth1;
PsMomMeth1 = Fl_Icomb.W*Fl_Icomb.V - Fl_Ocomb.W*Fl_Ocomb.V;
PsMomMeth1 = PsMomMeth1/C_GRAVITY;
PsMomMeth1 = PsMomMeth1 + Fl_Icomb.Ps * Fl_Icomb.A;
```

```
        PsMomMeth1 = PsMomMeth1/Fl_Ocomb.A;
        real PsMomMeth2;
        //PsMomMeth2 = Fl_Ocomb.W*Fl_Icomb.V;
        PsMomMeth2 = Fl_Icomb.W*Fl_Icomb.V + Wfuel*Fl_Icomb.V*...
           fuelFractV;
645     PsMomMeth2 = PsMomMeth2/C_GRAVITY;
        PsMomMeth2 = PsMomMeth2 + Fl_Icomb.Ps * Fl_Icomb.A;
        PsMomMeth2 = PsMomMeth2/Fl_Ocomb.A;
        PsMomMeth2 =
          PsMomMeth2/(1.0+Fl_Ocomb.gams*Fl_Ocomb.MN*Fl_Ocomb.MN);
650     //PsMomMeth1 = PsMonMeth2;

        //---------------------------------------------------
        // Note Meth1 = Meth2 when MN <= 1.0
        // Use Meth2 - seems more stable the Meth1 when MN > 1.0
655     //---------------------------------------------------
        PqPRayleighNew = (PsMomMeth2/Fl_Ocomb.Ps) * PqPRayleigh;
          }

          // Check against tolerance
660       PqPRayleighError = PqPRayleighNew - PqPRayleigh;
          if( abs(PqPRayleighError) < tolRayleigh ) { break; }

          // Bounding of PqPRayleigh movement to PqPRayleighStep
          real sign;
665       sign = PqPRayleighError/abs(PqPRayleighError);
          PqPRayleighDelta = sign *
        min(abs(PqPRayleighError),PqPRayleighStep);
          PqPRayleighNew = PqPRayleigh + PqPRayleighDelta;

670       // Lower limit of PqPRayleigh - limit too much loss to ...
             PqPRayleighMin
          if( PqPRayleighNew < PqPRayleighMin ) {
        if( flagRayleighLossTooMuch == 1 ) {
          ESOreport( 1023901,"Rayleigh pressure loss limited, too much...
              loss", FALSE );
          break;
675     }
        PqPRayleighNew = PqPRayleighMin;
        flagRayleighLossTooMuch = 1;
          }
          else {
680     flagRayleighLossTooMuch = 0;
          }

          /*
          // debug info
685       cout << Fl_Ocomb.A << " ";
          cout << Fl_Ocomb.MN << " ";
          cout << Fl_Ocomb.Ps << " ";
          cout << PsMomMeth1 << " ";
          cout << PsMomMeth2 << " ";
```

```
690       cout << PqPRayleigh << " ";
          cout << PqPRayleighNew << " ";
          cout << endl;
          */

695
          //-------------------------------------------------------
          // check for convergence
          //-------------------------------------------------------
          if( countRayleigh >= countRayleighMax ) {
700     ESOreport( 1023901,"Rayleigh iteration failed to converge, ...
           counter exceed max", FALSE );
        break;
          }

          PqPRayleigh = PqPRayleighNew;
705
        }

        if( flagRayleighChoked == 1 ) {
          ESOreport( 1023901,"Rayleigh Fl_Ocomb.MN exceed choked
710     condition", FALSE );
        }
      }


715   void calculate() {


        //-------------------------------------------------------
        // Preburning pressure loss
720     //-------------------------------------------------------
        calcPreLoss();  // creates Fl_Icomb, applies pre-losses

        real FARin = Fl_Icomb.FAR;
        real WARin = Fl_Icomb.WAR;
725
        if (Fl_I.MN == 0. && Fl_I.Aphy == 0.){
            Fl_Icomb.MN = 0.4;
            Fl_Icomb.setTotal_hP(Fl_Icomb.ht, Fl_Icomb.Pt);
        }
730
        //-------------------------------------------------------
        // Pre-calculate Burning to obtain enthalpy,etc.
        //-------------------------------------------------------
        if ( switchBurn == "FAR" ) {
735       //-------------------------------------------------------
          // determine the fuel weight flow from the input FAR
          //-------------------------------------------------------
          Wfuel = ( Fl_Icomb.W/(1. + FARin +WARin))*(FAR -FARin);
          Fu_I.Wfuel =  Wfuel;
740       eqRatio = FAR/Fu_I.FARst; // Added 5 Feb 2007 - IA
```

```
        calcBurn();
        calcRayleighLoss();

745     TtCombOut = Fl_Ocomb.Tt;

      }
      //#########################################################
      // Added 5 February 2007 - IA
750   // do an equivalence ratio calculation
      else if (switchBurn == "EQRATIO") {
          FAR = eqRatio*Fu_I.FARst;
          Wfuel = (Fl_Icomb.W/(1.+ FARin+ WARin))*(FAR - FARin);
          Fu_I.Wfuel =  Wfuel;
755
          calcBurn();
          calcRayleighLoss();

          TtCombOut = Fl_Ocomb.Tt;
760   }

      //---------------------------------------------------------
      //make a flow station that has props of cold vitiated air
      //---------------------------------------------------------
765   Fl_Vit.copyFlowStatic("Fl_Ocomb");
      Fl_Vit.setTotalTP(Fl_Icomb.Tt, Fl_Icomb.Pt);

      //---------------------------------------------------------
      //copy inlet flow for pure air reference to be used later
770   //---------------------------------------------------------
      //Take a snapshot of air after it has entered detn tubes
      Fl_IcombAir.copyFlowStatic("Fl_Icomb");

      // Copy input flow properties for  internal bypass flow
775   // - W set later
      Fl_O2.copyFlow("Fl_IcombAir");

      //---------------------------------------------------
      // On-design loop
780   //---------------------------------------------------
      if (switchDes == "DESIGN"){

        //-------------------------------------------------
      //  Initialize local variables
785   //-------------------------------------------------
      real uCJ, a_1, rhoVit, freq, PcqPi, errors;
      real gamt, Cpt, beta, MCJ2, PcqPi2;
      real Atube, Vtube;//, mCycle, Wtube;
      real MFP, Wvalve, gma_I;
790   real mFillAir, mPurgeAir, mPureAir;
      real tDetonation, tDetProp, tBlowdown, tPurge, tFill, iVel;
      real gam_s, gmm_fc;
```

```
        real WtotAir,  Wbypass;

795     int count;

        //---- initiated but not iterated -----------------------
        //static density of cool vitiated fluid
        rhoVit = Fl_Vit.rhot;   //(lbm/ft^3)
800
        // speed of sound in pure air, stagnated in detonation
        // tube that the detonation wave propogates in to
        a_1 = sqrt(Fl_Icomb.gamt*Fl_Icomb.Rt*Fl_Icomb.Tt*25037.);

805
        // ====================================================
        // Calculate Chapman-Jouguet Mach number for wave
        //  as described in Heiser and Pratt
        //====================================================
810     //*** input variables:  //
        //*** output variables: //MCJ, deltaS, qadd //
        //*** Flow Stations:    //Fl_Ocomb, Fl_Icomb //
        // local variables:     //gamt, Cpt, qadd, beta, MCJ2 //

815     //------ Arithmetically average specific heats -----------
        // arithmetic mean of gamma for stopped fluid
        gamt = (Fl_Ocomb.gamt + Fl_Icomb.gamt)/2.0;

        // arithmetic mean of Cp for a stopped fluid
820     Cpt = (Fl_Ocomb.Cpt + Fl_Icomb.Cpt)/2.0;

        //----- Calculate heat addition per Heiser-Pratt cycle ---
        // calculate non-dimensional heat addition
        qadd = (Fl_Ocomb.ht - Fl_Icomb.ht)/(Cpt*Fl_Icomb.Tt);
825
        //------- Calculate Chapman-Jouget Mach number --------
        beta = (gamt + 1.0)*qadd+1.0;
        MCJ2 = beta + sqrt( beta**2 - 1.0 );
        MCJ = sqrt(MCJ2);
830
        //--------- Calculate Entropy gain based on CJ detonation -
        deltaS = Cpt*(-log(MCJ2*((gamt+1.0)/
            (1.0+gamt*MCJ2))**((gamt+1.0)/gamt)) );

835     //---- calculate the pressure rise using the H &P method --
        PcqPi = (1.0+ gamt*MCJ2)/(gamt+1.0);
        uCJ = a_1*MCJ;


840     //------ Calculate tube volume and Area -------------------
        Atube = (PI/4.)*dTube**2/144.; // ft^2
        Vtube = Atube*(lTube/12); // ft^3

        //------ calculate the valve inlet mass flow rate --------
```

```
845     gma_I=Fl_IcombAir.gamt;
        MFP = Mvalve*sqrt( (gma_I*32.174)/(Fl_IcombAir.Rt*778.16) )
            *(1.+(gma_I-1.)/2.*Mvalve**2)**( (gma_I+1.)/(2.*(1.-gma_I)...
               ));
        Wvalve = (Fl_IcombAir.Pt/sqrt(Fl_IcombAir.Tt))
                    *(Atube*144.*ARvalve)*MFP;
850
        //-------------------------------------------------------
        // On-Design: Calculate bypass ratio
        //-------------------------------------------------------
        //*** input Variables:  //dTube, lTube, n_tubes, fillFrac
855     //                      // purgeFrac,
        //*** iterated Variables // freq
        //*** output Variables: // iBPR
        //*** local variables:  //WfillAir, WpurgeAir, WpureAir, ...
           WtotAir
        //                        Wbypass, WpurgeAir, Wvit, //
860     //*** Flow Stations:    // Fl_IcombAir, Fl_Icomb, Fl_Iprg, ...
           Fl_Vit, //

        //---- Calculate the split and partition of flow ----------
        // amount of air that will be mixed with fuel - 1 tube
        mFillAir = Vtube*(rhoVit*fillFrac)/(1.+FAR);
865
        // amount of air that will purge during each cycle - 1 tube
        mPurgeAir = Vtube*(Fl_IcombAir.rhos*purgeFrac);

        // total air per cycle flowing though one tube
870     mPureAir =  mFillAir + mPurgeAir;

        //-------------------------------------------------------
        // Timing - calculate frequency
        //-------------------------------------------------------
875     //*** input Variables: // DDT, tValve, Ltube, ff, pf, tCycle
        //*** iterated Variables: // uCJ, PcqPi
        //*** output Variables // tCycle, tauValveOpen, freq
        //*** local variables: // tDetonation, tDetProp, tBlowdown, ...
           tPurge,
        //                     // tFill
880     //-------------------------------------------------------


        //---------------- Detonation time ------------------
        // DetProp time is relatively independant of fill fractn
885     tDetProp= lTube/(uCJ*12);

        //DDT is input, tDetonationPropogatio calcd
        // (may need to iterate)
        tDetonation = DDT + tDetProp;
890
        //-------------- Blowdown time ----------------------
        // assume choked flow at tube exit and calculateon
```

```
        //  blowdown based draw-down time of a pressurized
        //  tank calculated on pressure differential
895     gam_s = Fl_IcombAir.gams; // larger gamma is more conservative
        gmm_fc =  ((gam_s + 1.)/2.)**(-(gam_s+1.)/( 2.*(gam_s-1.)) ); ...
           //

    //####  tBlowdown: Use ~1/2 calcd pressure (to match experimental ...
       data)
        // we'll use CJ det wave velocity as the speed of sound in the...
            gas
900     //   since a cannot be directly calc'd
        // note tBlowdown is proportional to tube length
        // tauBlDn is proportional to tube length...
        tBlowdown = (log(0.4*PcqPi)/gmm_fc)*(lTube/uCJ);

905     //------------- Fill and Purge time -----------------
        // Use the choked flow at valve inlet and the mass flow rate ...
           as
        // calculated outside the loop to calculate fill time (m/ mdot...
           )
        tPurge = tValve + mPurgeAir/Wvalve; //(s)
        tFill = tValve + mFillAir/Wvalve; //(s)
910     //Improvement could be made by calculating vitiated air ...
           velocity...

        //------------ Cycle Time output calculation ------------
        tCycle = tDetonation + tBlowdown + tPurge + tFill;
        tauValveOpen = (tPurge+tFill)/tCycle;
915     freq = 1./tCycle;
    //cout << "\n \n tDetonation, tBlowdown, tPurge, tFill PcqPi...
       "<<" "<< tDetonation <<" "<< tBlowdown<<" "<< tPurge<<" "<< ...
       tFill<<"  freq" << 1/tCycle << " " << PcqPi << endl;

        //---------- Set total mass flow through tubes ----------
        WtotAir = mPureAir*n_tubes*freq;
920     // steady-state flow rate into tubes

        // conservation of mass check
        if (WtotAir > Fl_I.W) {

925         fillFrac = fillFrac*(Fl_I.W/WtotAir);
            purgeFrac = purgeFrac*(Fl_I.W/WtotAir);

            mFillAir = Vtube*(rhoVit*fillFrac)/(1.+FAR);
            // amount of air that will be mixed with fuel - 1 tube
930
            mPurgeAir = Vtube*(Fl_IcombAir.rhos*purgeFrac);
            // amount of air that will purge during each cycle -1 tube

            mPureAir =  mFillAir + mPurgeAir;
935         // total air per cycle flowing though one tube
```

```
        WtotAir = Fl_I.W;
        cout << " !pf & ff changed to: " << purgeFrac << "  " << ...
           fillFrac << endl;
    }
940
    //--------------- Set iBPR -----------------------------
        Wbypass = Fl_I.W - WtotAir;
        // steady-state flow rate sent to bypass

945     iBPR = Wbypass/WtotAir;
        // steady-state internal PDC bypass ratio

        iBPRdes = iBPR;

950     //--------- Set bypass exit flow  SPLIT --------------
        Fl_O2.W = Wbypass;

        //----- Set purge and fill stations  PARTITION -------
        Fl_Iprg.copyFlowStatic("Fl_IcombAir");
955     // copy flow for purge function

        // -------------  PURGE AIR -------------------
        Fl_Iprg.AphyDes  = (Atube*144)*n_tubes;//Set phys area
        Fl_Iprg.W = mPurgeAir*freq*n_tubes;    //set m dot
960
       // --------------- FILL AIR --------------------
        Fl_Icomb.copyFlow("Fl_IcombAir");
        Fl_Icomb.AphyDes  = Atube*144.*n_tubes*tauValveOpen;
        // Actual area is  multiplied by tauVO to get equivalent
965     //  area. - Fluid flows steadily through this area

        Fl_Icomb.W = mFillAir*n_tubes*freq; //
        Fl_Icomb.setTotal_hP(Fl_IcombAir.ht, Fl_IcombAir.Pt);
        //sets time-averaged static conditions
970
        //-----------------------------------------------------
        // Burning
        //-----------------------------------------------------
        // FAR was calculated prior to enteringh this
975   // point - so we just need to modify
      // Wfuel based on changed Fl_Icomb.W
        Wfuel = ( Fl_Icomb.W /( 1. + FARin + WARin))*( FAR - FARin...
            );
        Fu_I.Wfuel =  Wfuel;

980     calcBurn();
        calcRayleighLoss();

        TtCombOut = Fl_Ocomb.Tt;

985     //=====================================================
        // Apply Dyer-Kaemming correction to obtain tube flow
```

```
        // at exit(ignores the kinetic energy of  shock wave.)
        //=====================================================
        Fl_Ocomb.setTotal_hS(Fl_Ocomb.ht, Fl_Icomb.S+deltaS);
990
    }

    // OFF-DESIGN CODE GOES HERE

995 //-----------------------------------------------------------
    // Add split flows back to combusted flow
    //-----------------------------------------------------------
    Fl_Ocomb.add("Fl_Iprg"); //add purge flow in (uncorrected)

1000 //=============================================================
    // Apply corrections to the flow for transition to ...
    // steady state (TTSS)
    //=============================================================
    //*** local Variables: // Snew, Pnew
1005 //*** Input Variables: // deltaS, TTSSeff, TTSSdPqP
    //*** Flwo stations:   // Fl_Ocomb, Fl_Vit
    real hnew, Pnew; //

    //------ Calculate new Entropy and Pressure --------------
1010 //            eff = (dht)TTSF/(dht)comb + 1.
    //         current h    - ( h gained)*(1.-eff)
    hnew = Fl_Ocomb.ht - (Fl_Ocomb.ht -
                    Fl_Icomb.ht)*(1.0-TTSSeff);
    Pnew = Fl_Ocomb.Pt*(1.0-TTSSdPqP);
1015
    //End of 12Jan2007 additinos - IA
    //#########################################################
    Fl_O1.copyFlow( "Fl_Ocomb" );

1020 //------- update fluid properties based on new S & P
    Fl_O1.setTotal_hP(hnew, Pnew); //added 12Jan2007 - IA


    //-----------------------------------------------------------
1025 // Thermal storage calculations
    //-----------------------------------------------------------
    if ( !S_Qhx.isEmpty() ) {
      S_Qhx.execute();
    }
1030 real hout = Fl_O1.ht  - Qhx / Fl_O1.W;
    Fl_O1.setTotal_hP( hout, Fl_O1.Pt );


    //-----------------------------------------------------------
1035 // store the design value of FAR for use in guessing
    //-----------------------------------------------------------
    if ( switchDes == "DESIGN" ) {
      FARDes = FAR;
```

```
        }
1040
    }

    //------------------------------------------------------------
    // register the appropriate errors at build time
1045 //------------------------------------------------------------
    void VCinit()
    {
      ESOregCreate( 1023901, 8, "", TRUE, FALSE, TRUE ); // ...
         provisional
      ESOregCreate( 1093901, 8, "", TRUE, FALSE, TRUE ); // ...
         provisional
1050 }

    }

#endif
```

Bibliography

1. Petters, Dean P. and Felder, James L., "Engine System Performance of Pulse Detonation Concepts Using the NPSS Program". 38th AIAA/ASME/SAE/SAE/ASEE Joint Propulsion Conference & Exhibit, AIAA-2002-3910. American Institute of Aeronautics and Astronautics (AIAA), Indianapolis, Indiana, 7-10 July 2002.

2. Smith, C.F., Snyder, P.H., Emmerson, C.W., and Nalim, M.R., "Impact of the Constant Volume Combustor on a Supersonic Turbofan Engine". 38th AIAA/ASME/SAE/SAE/ASEE Joint Propulsion Conference & Exhibit, AIAA-2002-3916. American Institute of Aeronautics and Astronautics (AIAA), Indianapolis, Indiana, 7-10 July 2002.

3. Mattingly, Jack D., Heiser, William H., and Pratt, David T., *Aircraft Engine Design*. American Institute of Aeronautics and Astronautics (AIAA), Reston, Virginia, second edition, 2002. ISBN 1563475383.

4. Heiser, William H. and Pratt, David T., "Thermodynamic Cycle Analysis of Pulse Detonation Engines". *AIAA Journal of Propulsion and Power*, 18(1):68–76, January-February 2002.

5. Dyer, R.S. and Kaemming, T.A., "The Thermodynamic Basis of Pulsed Detonation Engine Thrust Production". 38th AIAA/ASME/SAE/SAE/ASEE Joint Propulsion Conference & Exhibit, AIAA-2002-4072. American Institute of Aeronautics and Astronautics (AIAA), Indianapolis, Indiana, 7-10 July 2002.

6. Kuo, Kenneth Kuan-yun., *Principles of Combustion*. John Wiley & Sons, New York, NY, 1986. ISBN 0-471-09852-3.

7. Glassman, Irvin., *Combustion*. Academic Press, New York, NY, third edition, 1996. ISBN 0-12-285582-2.

8. Strehlow, Roger A., *Combustion Fundamentals*. McGraw Hill, New York, NY, 1984. ISBN 0-07-062221-3.

9. Shapiro, A.H., *The Dynamics and Thermodynamics of Compressible Fluid Flow*. Ronald, New York, NY, 1953. Pp 193-211.

10. Pratt, D.T., Humphrey, J.W., and Glenn, D.E., "Morphology of Standing Oblique Detonation Waves". AIAA Journal of Propulsion and Power, 7(5):837–845, September-October 1997.

11. Harris, P.G., Guzik, S.M., and Stowe, R.A., "Design Methedology for a Pulse Detonation Engine as a Ramjet Replacement". 40th AIAA/ASME/SAE/SAE/ASEE Joint Propulsion Conference & Exhibit, AIAA-2004-3400. American Institute of Aeronautics and Astronautics (AIAA), Fort Lauderdale, Florida, 11-14 July 2004.

12. Paxson, Daniel E., *A General Numerical Model for Wave Rotor Analysis*. NASA Technical Memorandum NASA-TM-105740, National Air and Space Administration, Lewis Research Center, Cleveland, Ohio, July 1992.

13. Paxson, Daniel E., "A Performance Map for Ideal Air Breathing Pulse Detonation Engines". 37th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, AIAA-2001-3465. American Institute of Aeronautics and Astronautics (AIAA), Salt Lake City, Utah, 8-11 July 2001.

14. Rasheed, A., Tangirala, V.E., Vandevort, C.L., Dean, A.J., and Haubert, C., "Interactions of a Pulsed Detonation Engine with a 2D Blade Cascade". 42nd AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2004-1207. American Institute of Aeronautics and Astronautics (AIAA), Reno, Nevada, 5-9 January 2004.

15. McManus, Keith R. and Dean, Anthony J., "Experimental of a Two-Stage Pulsed Detonation Combustor". 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, AIAA-2005-3773. American Institute of Aeronautics and Astronautics (AIAA), Tucson, Arizona, 10-13 July 2005.

16. Rasheed, Adam, Furman, Anthony, and Dean, Anthony J., "Wave Experimental Investigations of an Axial Turbine Driven by a Multi-tube Pulsed Detonation Combustor System". 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, AIAA-2005-4209. American Institute of Aeronautics and Astronautics (AIAA), Tucson, Arizona, 10-13 July 2005.

17. Tangirala, V.E., Murrow, K., Fakunle, O., and Dean, A.J., "Thermodynamic and Unsteady Flow Considerations in Performance Estimation for Pulse Detonation Applications". 43rd AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2005-226. American Institute of Aeronautics and Astronautics (AIAA), Reno Nevada, 10-13 January 2005.

18. Rasheed, Adam, Furman, Anthony, and Dean, Anthony J., "Wave Attenuation and Interactions in a Pulsed Detonation Combustor-Turbine Hybrid System". 44th AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2006-1235. American Institute of Aeronautics and Astronautics (AIAA), Reno Nevada, 9-12 January 2006.

19. Baptista, M., Rasheed, A., Badding, B., Velagandula, O., and Dean, A.J., "Mechanical Response in a Multi-tube Pulsed Detonation Combustor-Turbine Hybrid System". 44th AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2006-1234. American Institute of Aeronautics and Astronautics (AIAA), Reno Nevada, 9-12 January 2006.

20. Rasheed, Adam, Furman, Anthony, and Dean, Anthony J., "Wave Interactions in a Multi-tube Pulsed Detonation Combustor-Turbine Hybrid System". 42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, AIAA-2006-4447. American Institute of Aeronautics and Astronautics (AIAA), Sacramento, California, 9-12 July 2006.

21. Schauer, Fred, Bradley, Royce, and Hoke, John., "Interactions of a Pulsed Detonation Engine with a Turbine". 41st Aerospace Sciences Meeting and Exhibit, AIAA-2003-891. American Institute of Aeronautics and Astronautics (AIAA), Reno, Nevada, 6-9 January 2003.

22. Akbari, P., Nalim, M.R., and Snyder, P.H., "Numerical Simulation and Design of a Combustion Wave Rotor Deflagrative and Detonative Propagation". 42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, AIAA-2006-5134. American Institute of Aeronautics and Astronautics (AIAA), Sacramento, California, 9-12 July 2006.

23. Paxson, Daniel E., "Performance Evaluation Method for Ideal Airbreathing Pulse Detonation Engines". *AIAA Journal of Propulsion and Power*, 20(5):945–947, September-October 2004. AIAA-11426.

24. NASA. *NPSS User Guide.*, National Aeronautics and Space Administration, NASA John H. Glenn Research Center at Lewis Field, 21000 Brookpark Rd., Cleveland, Ohio 44135-3191, revision u edition, April 4, 2005. Software Release NPSS_1.6.3.

25. Oates, Gordon C., *Aerothermodynamics of Gas Turbine and Rocket Propulsion.* American Institute of Aeronautics and Astronautics (AIAA), Reston, Virginia, third edition, 1997. ISBN 1563472414.

26. Wilson, David Gordon and Korakianitis, Theodosios., *The Design of High-Efficiency Turbomachinery and Gas Turbines.* Prentice-Hall, Upper Saddle River, New Jersey, second edition, 1998. ISBN 0-13-312000-7.

27. Çengal, Yunus A. and Boles, Michael A., *Thermodynamics: An Engineering Approach.* WCB/McGraw-Hill, New York, New York, third edition, 1988. ISBN 0-07-011927-9.

28. Mattingly, Jack D. and von Ohain, Hans., *Elements of Propulsion: Gas Turbines and Rockets.* American Institute of Aeronautics and Astronautics (AIAA), Reston, Virginia, 2006. ISBN 1563477793.

29. McKinney, John S., *Simulation of Turbofan Engine: Part I. Description of Method and Balancing Technique.* Technical Report AFAPL-TR-S7-125, United States Air Force, Aero Propulsion Laboratory, Dayton, Ohio, November 1967.

30. McKinney, John S., *Simulation of Turbofan Engine: Part II. User's Manual and Computer Program Listing.* Technical Report AFAPL-TR-S7-125, United States Air Force, Aero Propulsion Laboratory, Dayton, Ohio, November 1967.

31. NASA. *NPSS Reference Sheets.*, National Aeronautics and Space Administration, NASA John H. Glenn Research Center at Lewis Field, 21000 Brookpark Rd., Cleveland, Ohio 44135-3191, revision u edition, April 4, 2005. Software Release NPSS_1.6.3.

32. Keenan, Joseph H., Chao, Jing, and Kaye, Joseph., *Gas Tables*. John Wiley and Sons, New York, NY.

33. Dutton, J. Craig and Coverdill, Robert E., "Experiments to Study the Gaseous Discharge and Filling of Vessels". International Journal of Engineering Education, 13(2):123–134, February 1997.

Vita

Captain Ionio Q. Andrus graduated with the class of 1994 from Hillcrest High School in Ammon, ID. After graduation he spent two years proselyting in Brazil for his church. Among other things, this allowed him to gain a unique perspective on international relations.

He studied at Idaho State University and Embry-Riddle Aeronautical University before transferring to Utah State University where he completed the rigorous Mechanical Engineering curriculum, receiving his Bachelor of Science degree in December of 2002. While earning his undergraduate degree he worked as a student researcher for the Space Dynamics Laboratory in the science group. As part of his work duties he was able to co-author several conference papers submitted to SPIE. He also authored a paper on orbital dynamics that was selected by AIAA to be presented at the 2002 World Space Congress student paper competition.

Captain Andrus was commissioned through Detachment 860 AFROTC at Utah State University. His first assignment was to the National Air and Space Intelligence Center located at Wright-Patterson AFB. He provided engineering analysis on several systems of high interest to the United States Air Force. In August of 2005 he entered the Graduate School of Engineering and Management, Air Force Institute of Technology. Upon graduation, he will be assigned to the 412th Test Wing at Edwards AFB, CA.

Permanent address: 2950 Hobson Way
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433

Index

The index is conceptual and does not designate every occurrence of a keyword. Page numbers in bold represent concept definition or introduction.

AEDsys, 2

C-5, 14
Chapman-Jouget
    Mach number, 36
Chapman-Jouguet, 4
    Mach number, 35–38, 48
    detonation, 35
    Mach Number, 8
    Mach number, 8, 45
    points, 5, 6
chapter, 14
chapter 2, 4
compressor map, 16
computational thermodynamics, 9, 21, 22, 24
Constant Volume Combustion Cycle Engine, 10
cycle
    energy conserved PDE, 9
    PDE, 9
    ramjet, 9

DDT, 44, 48
detonate, 39
detonation, 7, 40
detonation-to-deflagration, 44

efficiency
    adiabatic, 23
    adiabatic, 18, 20, 61, 62
    polytropic, 20, 23, 53
energy
    conservation of, 9, 39
    conservation of, 9
enthalpy, 10, 21–25, 39
    differences, 27
entropy, 8–10, 36–39
entropy function, 25
equivalence ratio, 39
error, 24
expansion waves, 37

f90 Code Listings
    appendix2/constantModule.f90, 96
    appendix2/unFAIR.f90, 96
FAIR, 25
FAR, 8, 30
fill, 9, 41
fill fraction, 41
fill time, 3
frequency, 11, 43

GasTbl, 26, 56, 57

heat flux, 4
heat transfer, 48, 81
Hugoniot, 4

inputs, 18, 19
internal bypass, 33
internal bypass ratio, 46, 47
iterate, 6
iteration, 11, 48

janaf, 26, 27, 51, 55, 56

maintenance, 1
mass flow
    choked, 45
    parameter, 43
    valve, 43
mass flow
    parameter, 43
    valve-open, 43
mixer, 16, 33, 63, 71
model improvements, 49

noise, 1
nozzle, 10
    conical, 16
NPSS, 2, 26
NPSS Code Listings
    appendix1/PDCburner.int, 99
    appendix3/CmpareAEDsys.mdl, 83
    appendix3/CmparePDC.mdl, 89

octane, 36
optimized, 74

PDC, 14, 48
    experimental, 12
    NPSS element, 35
PDE, 8, 11
performance map, 10
power, 20, 23, 29
pressure ratio, 11, 46
pulsed detonation, 10
pulsed detonation combustor, 1, 34
purge, 41
purge fraction, 41

Rankine, 4

shock, 35, 36, 39, 40, 45
solver, 20
specific heat, 11, 25, 26
    differences, 27
stator, 12

TF-39-GE-1C, 14
the end, 82
time
    blowdown, 45, 46
    averaged, 9, 39
    blowdown, 11, 45, 46
    cycle, 11, 44, 48
    detonation, 44
    fill, 46
    purge, 46
transition to steady state, 13, 33, 49
translation, 18
turbine, 12
turbofan
    hybrid, 31
    high bypass, 16
    split stream, 16

unFAIR, 25
unsteady, 33

Von-Neumann Spike, 7
VSH, 20

ZND detonation, 6

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704–0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
22–03–2007	Master's Thesis	Sep 2005 – Mar 2007

4. TITLE AND SUBTITLE

Comparative Analysis of a High Bypass Turbofan Using a Pulsed Detonation Combustor

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Andrus, Ionio Q., Captain, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management
2950 Hobson Way, Building 640
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GAE/ENY/07-M02

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory Propulsion Directorate
Turbine Engine Division
Engine Integration and Assessment Branch
1950 Fifth Street
WPAFB OH 45433-7251

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/PRTA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

It has been proposed that the implementation of a pulsed detonation combustor in a high-bypass turbofan engine would result in an engine that is both more efficient and more reliable. The validity of the performance claims are evaluated based on a comparison between the baseline and hybrid turbofans. The hybrid pulsed detonation engine was modeled in the Numerical Propulsion Simulation System (NPSS) and shares a common architecture with the baseline turbofan model, except that the combustor of the baseline engine is replaced with a pulsed detonation combustor. Detonation effects are calculated using a closed form solution of the Chapman-Jouguet Mach number with a total energy correction applied. Cycle time is calculated to provide a reasonable estimate of frequency for the user input geometry, and the losses due unsteady flow are accounted for by applying pressure and temperature losses to the fluid. A parametric study was performed to evaluate the effects of these losses on net thrust and TSFC. There is a definite level of acceptable loss that if surpassed makes pulsed detonation combustion a good candidate for inclusion into a hybrid turbofan engine.

15. SUBJECT TERMS

High Bypass Turbofans, Turbofan Engines, Detonation Waves, Combustion, Hybrid Propulsion, Hybrid Simulation, Aircraft Engines, Air Breathing Engines, Air Breathing Engines (Unconventional), Thermodynamic Cycles

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Paul I. King, (ENY)
U	U	U	UU	143	19b. TELEPHONE NUMBER *(include area code)* (937) 255–3636, ext 4628