

COMPRESSOR CONCEPTUAL DESIGN OPTIMIZATION

A Thesis
Presented to
The Academic Faculty

By

ANDREW SCOTT MILLER

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science in Aerospace Engineering

Georgia Institute of Technology

May 2015

COMPRESSOR CONCEPTUAL DESIGN OPTIMIZATION

Approved by:

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

Dr. Jimmy Tai
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Jeff Schutte
School of Aerospace Engineering
Georgia Institute of Technology

Mr. Russell Denney
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: April 17, 2015

DEDICATION

To my wife, Allison, and our children, most of whom we don't know yet.

ACKNOWLEDGEMENTS

First, I would like to thank my committee for their faith and instruction.

Thank you Doctor Mavris and Boeing Commercial Aircraft for funding my research.

Also thank you Allison for supporting my long hours at school.

TABLE OF CONTENTS

Dedication	iii
Acknowledgements.....	iv
List of Tables	ix
List of Figures	x
List of Symbols and Abbreviations.....	xii
Summary	xiv
Introduction	1
Background	3
Zooming	4
Cycle Design	5
Scaling	8
Compressor Design	11
Performance Prediction	13
Observations	14
Thesis Statement	16
Research Questions and Hypotheses	17
Experimental Resources	22

Task 1: Parametric Macro Geometry Definition	24
Procedure.....	24
Number of Stages	24
Compressor Annulus Geometry.....	25
Stage Calculations Note	27
Stage Pressure Ratios and Annulus Geometry	27
Velocity Triangles.....	28
Blade Geometry	30
Independent Variables.....	33
Experiment: Five-Stage Compressor Conceptual Design.....	37
Observations	45
Task 2: Performance Analysis	48
Loss Models.....	48
Profile Loss	48
Shock Loss	49
Endwall Loss	50
Off-Design Loss	51
Multiple-meanline Fidelity.....	52
Stall Prediction	55

Pressure versus Flow Characteristic	55
Boundary Layer Separation.....	56
Equivalent Relative Velocity Ratio	57
Experiment: NACA Five-Stage Transonic Compressor	60
Losses	62
Off-Design Analysis	63
Stall Prediction	65
Observations	70
Task 3: Opimization.....	72
Design Space	72
Optimization	79
Observations	86
Task 4: Map Generation.....	88
Conclusion.....	90
Appendix A: Five-Stage Compressor Design Space Lanscape.....	94
Appendix B: Parametric Compressor Conceptual Design Code	102
Appendix C: Loss Models Code	120
Appendix D: Map Generation Code	140
References	147

LIST OF TABLES

Table 1: Physical Compressor Scaling Parameters	10
Table 2: Assumptions for Conceptual Design	19
Table 3: NACA 5 Stage Compressor Design Point.....	38
Table 4: Inlet Annulus Parameters.....	39
Table 5: Discharge Parameters	41
Table 6: Stage Pressure Ratios	42
Table 7: NACA Design Values of Aspect Ratio and Solidity.....	44
Table 8: Blade Row Parameters	44
Table 9: Macro Geometry Parameters, Outputs and Assumptions	45
Table 10: NACA 5 Stage Compressor Design Point and Parameters	46
Table 11: NACA Compressor Design Point Overall Results.....	61
Table 12: Design Point Stage Total Pressure Ratios	62
Table 13: Location of Compressor Stall Indication	69
Table 14: Predicted Responses for Maximum Stall Margin Optimization.....	83
Table 15: Optimum Combination of Overall Parameters for Max Stall Margin	83
Table 16: Optimum Combination of Blade Row Parameters for Max Stall Margin	84
Table 17: Predicted Responses for Maximum Efficiency Optimization.....	85
Table 18: Optimum Combination of Overall Parameters for Max Efficiency	85
Table 19: Optimum Combination of Blade Row Parameters for Max Efficiency	86

LIST OF FIGURES

Figure 1: Notional Compressor Performance Map [2]	1
Figure 2: Standard Gas Turbine Engine Station Numbering [1].....	3
Figure 3: Velocity Triangle Parameters.....	29
Figure 4: Blade Geometry Parameters	31
Figure 5: Comparison of Study Compressor to NACA Compressor Annulus Geometry...	43
Figure 6: Streamline Design Calculation Stations	52
Figure 7: Radial Distribution of Loss, Rotor 1 Design Point [27].....	63
Figure 8: Compressor Map: Flow vs. Pressure Ratio	64
Figure 9: Compressor Map: Flow vs. Efficiency	64
Figure 10: Equivalent Relative Velocity Ratio Across Rotor 1.....	65
Figure 11: Equivalent Relative Velocity Ratio Along the Hub Contour, Design Point	66
Figure 12: Variation of Tuning Parameter for Best Performance.....	68
Figure 13: Stall Limits as Predicted by Criteria 1 and 3.....	70
Figure 14: Five Stage Compressor Design Space	74
Figure 15: Five Stage Compressor Feasible Space	76
Figure 16: Feasible Space reduced by Design Constraints.....	77
Figure 17: Design Space Overall Parameters with Feasible Space Highlighted.....	78
Figure 18: Design Space Aspect Ratio Combinations with Feasible Space Highlighted ...	79
Figure 19: Design Space Solidity Combinations with Feasible Space Highlighted	80
Figure 20: Compressor Performance Map for Optimal Six Stage Configuration.....	88

Figure 21: 5 Stage Design Space with Converged Points Highlighted 94

Figure 22: 5 Stage Design Space with Non-converged Points Highlighted..... 95

Figure 23: 5 Stage Design Space Aspect Ratios with Converged Points Highlighted 96

Figure 24: 5 Stage Design Space Aspect Ratios with Non-converged Points Highlighted 97

Figure 25: 5 Stage Design Space Solidity Values with Converged Points Highlighted..... 98

Figure 26: 5 Stage Design Space Solidity Values with Non-converged Points Highlighted
..... 99

Figure 27: 5 Stage Design Space Parameters with Converged Points Highlighted..... 100

Figure 28: 5 Stage Design Space Parameters with Converged Points Highlighted..... 101

LIST OF SYMBOLS AND ABBREVIATIONS

A	Area
B	Blockage
C	Flow absolute velocity
c_p	Working fluid specific heat
e_c	Polytropic efficiency
F_{net}	Thrust force
f	Fuel to air ratio
g	Acceleration of gravity
h	Specific enthalpy
K_l	Loading factor
k	Scaling factors
\dot{m}	Mass flow rate
N, ω	Shaft rotational speed
N_s	Stage Count
P, p	Pressure
PR, π	Pressure ratio
r	Radius
s	Specific entropy, Scaling factors
SM	Stall margin
T	Temperature
U	Rotor velocity
V	Flow velocity
W	Flow relative velocity, Mass Flow Rate
\dot{W}, \dot{w}	Power
α	Absolute flow angle to axis
β	Relative flow angle to axis

γ	Ratio of specific heats
δ	Deviation, Ratio of pressure to reference pressure
δ_c	Blade tip clearance
η	Adiabatic efficiency
θ	Camber angle, Ratio of temperature to reference temperature
ι	Incidence
κ	Blade metal angle
ρ	Density
τ	Shear force
ϕ	Radial flow angle from axis
ψ	Loading coefficient

subscripts

0 – 9	Gas turbine engine stations
1 – 2	Blade row inlet and exit
1 – 3	Stage calculation stations: rotor inlet, rotor exit and stator inlet, stator exit
<i>amb</i>	Ambient condition
<i>c</i>	Corrected parameter
<i>cmp</i>	Compressor overall parameter
<i>sh</i>	Shaft
<i>t</i>	Total or stagnation condition
<i>th</i>	Thermal
<i>x</i>	Axial direction
θ	Tangential direction

SUMMARY

Gas turbine engines are conceptually designed using performance maps that describe the compressor's effect on the cycle. During the traditional design process, the cycle designer selects a compressor design point based on criteria to meet cycle design point requirements, and performance maps are found or created for off-design analysis that meet this design point selection. Although the maps always have a pedigree to an existing compressor design, oftentimes these maps are scaled to account for design or technology changes. Scaling practices disconnect the maps from the geometry and flow associated with the reference compressor, or the design parameters which are needed for compressor preliminary design. A goal in gas turbine engine research is to bridge this disconnect in order to produce acceptable performance maps that are coupled with compressor design parameters.

A new compressor conceptual design and performance prediction method has been developed which will couple performance maps to conceptual design parameters. This method will adapt and combine the key elements of compressor conceptual design with multiple-meanline analysis, allowing for a map of optimal performance that is attached to reasonable design parameters to be defined for cycle design. This method is prompted by the development of zooming analysis capabilities, which allow compressor analysis to be incorporated into cycle analysis. Integrating compressor conceptual

design and map generation into cycle analysis will allow for more realistic decisions to be made sooner, which will reduce the time and cost used for design iterations.

In this research, a compressor conceptual design process is discussed which allows for low-fidelity compressor geometry to be defined by only $4 \times (\text{Number of Stages}) + 8$ parameters. The dimensionality of this process is reduced by the acceptance of six assumptions common to compressor conceptual design. The question is asked whether the resulting compressor macro geometry is consistent with expected trends. A hypothesis is validated that the assumptions made will allow for the resultant compressor geometry to follow expectant trends such as continuously reducing annulus area through the compressor.

It will be shown that although the assumptions made can result in configurations consistent with known geometry trends, more detailed constraints are required to allow the resulting geometries to converge when analyzed for performance. The landscape of the design space is visualized for future identification of constraints on parameter relationships such as the relationships between Inlet-to-Discharge Velocity ratio, Inlet Mach number, and Thickness-to-Chord Ratio.

A multiple-meanline performance prediction method is also presented which is based on the assumption of simple radial equilibrium. This method employs empirical loss models to predict the effect of profile, endwall and shock losses of each blade row on the performance of the machine. Another question is asked whether empirical loss models will follow expected trends throughout a compressor map as well as at the blade

row level. Another hypothesis is validated that the loss models compiled by Aungier will predict correct trends at the blade row level as well as at the machine level, resulting in simulated compressor maps that reflect measured data. Therefore the second hypothesis was also proven valid.

The design space of five-stage configurations spanned by ranges of the parameters discussed is visualized and discontinuities are identified. Optimization strategies are discussed and a surrogate model approach for the objective function is selected due to the discontinuous design space. An optimization method is demonstrated for two applications using the surrogate model generated. Finally, a compressor performance map generation method is demonstrated. The performance map and the related design parameters satisfy cycle requirements as well as compressor constraints.

In short, a new compressor conceptual design, analysis, and map generation method is developed and demonstrated that can bridge the disconnection between cycle design and compressor design. This method substantiates the thesis that a compressor conceptual design, analysis and optimization method can couple design parameters to performance maps, allowing the cycle designer to choose a suitable map that is attached to reasonable compressor design parameters.

INTRODUCTION

The first step in gas turbine engine design is to conceptually design the cycle. Conceptual cycle design is the process of selecting cycle design variables to meet thermodynamic performance requirements. These cycle design variables are used in equations that describe the component's effect on the working fluid [1]. The basic cycle design variables needed to describe a compressor's effect on the working fluid are the corrected mass flow rate, \dot{m}_c ; the corrected shaft speed, N_c ; pressure ratio, PR or π ; and efficiency, η . Given the compressor inlet conditions, values for these design variables define the exit conditions of the flow. The cycle design must also perform appropriately at off-design conditions. For a given compressor, PR and η are related to

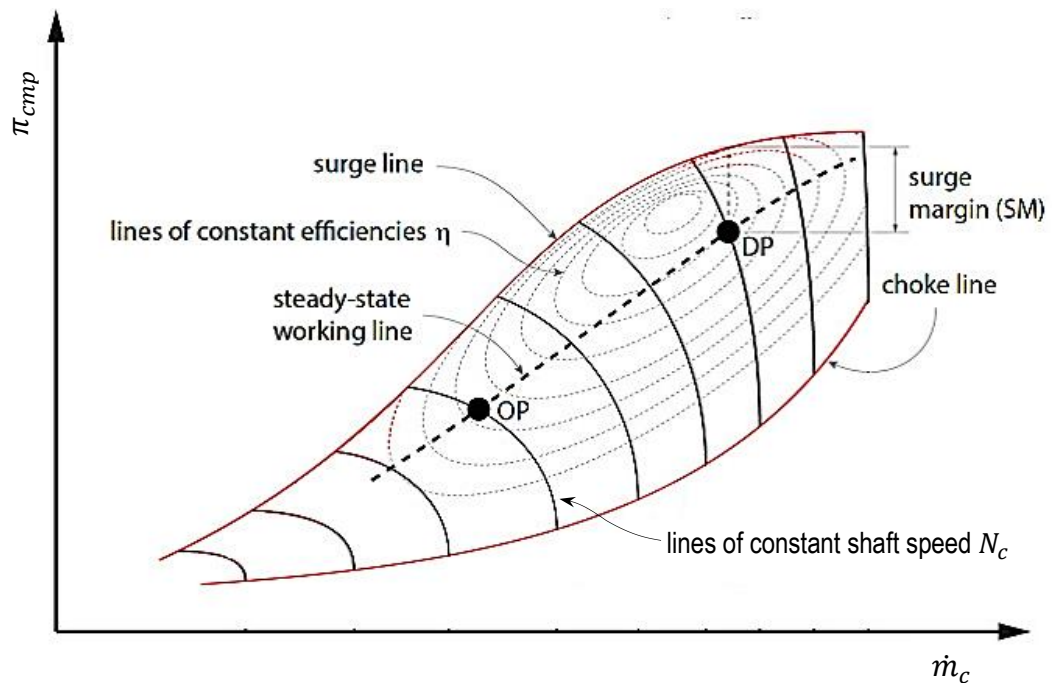


Figure 1: Notional Compressor Performance Map [2]

off-design \dot{m}_c and N_c conditions via performance maps (See Figure 1). These performance maps must be known in order for the cycle designer to successfully design a cycle. However, comprehensive libraries of compressor maps are not available for the cycle designer to find the optimal compressor for an application.

The library of compressor maps available to the cycle designer is populated by documented compressor design simulations or physical rig tests. These compressor designs can be organized into families for library classification and small scale extrapolation for design use [4]; however, these compressor families cannot characterize effects of unique design features, especially when scaled [5].

Because comprehensive libraries of compressor maps are not available, common practice is to scale a known compressor map to the desired design point. Although the scaled map has a pedigree to a known compressor design, the design features may not reasonably scale to the new application. The compressor designer is then tasked to find a design that performs adequately, although the resulting performance map will be unique.

BACKGROUND

A gas turbine engine core is comprised of four main components: a compressor, a burner, a turbine, and a shaft that delivers power from the turbine to the compressor, most generally in the configuration shown in Figure 2. The numbering convention is standardized as an Aerospace Recommended Practice by the Society of Automotive Engineers [6].

Design is typically categorized into three phases: Conceptual, Preliminary, and Detailed Design. For traditional design of a gas turbine engine, conceptual design refers to the cycle design and very low-fidelity knowledge of components. The design process is very iterative, requiring interaction between the design of the components and the design of the cycle through the preliminary and detailed design phases. Increasingly higher fidelity results are passed back from the compressor designer to the cycle designer for updated cycle analysis and optimization.

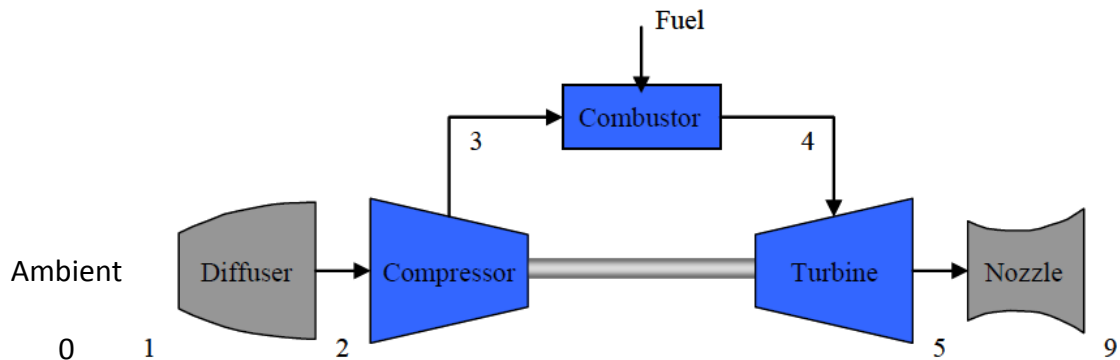


Figure 2: Standard Gas Turbine Engine Station Numbering [1]

During conceptual design, “very close interfacing between the cycle and component designers is necessary to ensure that [mechanical and aerothermodynamic constraints] are not exceeded” [9]. Because the cycle designer knows that constraints are likely violated by scaling a map, iterations will occur with the beginning phases of compressor design to determine an acceptable map for cycle design. The compressor designer will take the scaled map and its design point and determine design parameters of stage count, blade count, tip speed, size, etc. for a compressor that will perform close to the desired map. This interaction will be defined as Compressor Conceptual Design.

Zooming

A recent development in gas turbine engine design is multi-fidelity engine modeling. Also known as zooming, this development allows for higher fidelity models of engine components to be individually included in the zero-dimensional cycle analysis. These models can be individual component legacy codes of any resolution including computational fluid dynamics (CFD) analyses. This method allows for the compressor performance to be predicted for cycle analysis at engine conditions that may not be readily represented by a scaled map. Follen describes the benefits of zooming as fourfold [3]:

1. It allows for rapid evaluation of potential component designs in the context of an engine system, given the boundary conditions from the system and including the impact of the component on the system.

2. It enables rapid cycle analysis and optimization without having to update the zero-dimensional representation of the component.
3. It increases the fidelity of the engine system simulation because the component simulations can be physics-based, 1st principle analysis codes.
4. It allows for tailoring for application of computational power. The high resolution analysis that requires significant computing resources does not have to be applied to every component.

Zooming allows the designer to “zoom in” and investigate the physical flow occurring in an engine component [12]. The goal and purpose of the zooming process, however, is to include the high fidelity analysis results in the zero-dimensional system simulation in order to capture the system effects of the component [3]. This process is currently employed most successfully during engine development and certification, increasing the level of virtual testing conducted before committing the design to hardware [3] as well as verification of component stability throughout the flight envelope [2].

Follen promotes the ideas of evaluating potential component designs in context of the engine cycle as well as cycle optimization through the flight envelope using higher fidelity models of components [3]; however, component optimization within the context of engine cycle optimization should also be promoted as a potential benefit of zooming.

Cycle Design

The thermodynamic design point of the cycle is determined by either the cruise condition, where the engine will spend the most time and require the lowest fuel

consumption, or the top of climb condition, where the engine is required to produce the highest thrust and includes the operating conditions of pressure and temperature and Mach number [8]. The thermodynamic cycle of a gas turbine engine is based on the Brayton cycle.

The purpose of a gas turbine engine is to produce energy in terms of thrust or shaft work. The amount of thrust or shaft work an engine can produce is highly dependent upon the rate that mass flow is able to pass through all of the components. The thermodynamic efficiency of a gas turbine engine, or the ratio of shaft work to the heat addition in the combustor, is often defined in terms of pressure ratio across the compressor [1].

The other cycle-level parameters that define the compressor are the shaft speed, which must match the shaft speed of the turbine, and the adiabatic compressor efficiency, or the ratio of the real to ideal change in enthalpy across the compressor. Through iteration and optimization of the cycle design variables, the compressor design point, i.e. π , η , \dot{m} and N , is defined at the cycle design point.

After defining the cycle at design conditions, performance of the cycle at off-design conditions is analyzed. This analysis consists of evaluating the cycle performance at various atmospheric conditions, throttle settings and flight speeds. In order to simulate these conditions correctly, compressor performance maps are required which relate the compressor pressure ratio and efficiency to mass flow rate and shaft speed. These maps

are defined in terms of corrected mass flow rates and corrected shaft speeds through the following equations:

$$\dot{m}_c = \frac{\dot{m}\sqrt{\theta}}{\delta} \quad (1)$$

$$N_c = \frac{N}{\sqrt{\theta}} \quad (2)$$

Where $\theta = T_t/518.69^\circ R$ and $\delta = P_t/14.696 \text{ psia}$.

An example compressor performance map is shown in Figure 1. By convention, the abscissa is corrected mass flow rate and the ordinate is compressor pressure ratio with contours of constant efficiency. The solid, more vertical curves are lines of constant corrected speed and are usually identified by the percentage of design speed. The upper boundary of the map indicates stall or surge conditions and the lower boundary indicates choked conditions within the compressor. The dashed line through the map indicates the equilibrium operating line, or points of steady state mass flow and shaft speed as matched to the turbine [9]. This steady state operating line may or may not align with points of maximum efficiency at each speed, and its distance from the surge line is defined as the stall margin. The definition of the surge margin visualized in Figure 1 is defined by the distance between operation and surge at a constant mass flow rate as given in Equation (3) [2]. Surge margin can also be defined at a constant speed as the product of the ratios of pressure ratios and mass flow rates as given in Equation (56) [10].

$$SM_{const \dot{m}} = \left(\frac{PR_{surge} - PR_{operating}}{PR_{operating}} \right)_{\dot{m}_{c,op}} \quad (3)$$

$$SM_{const N} = \left(\frac{PR_{surge}}{PR_{operating}} \times \frac{\dot{m}_{c,op}}{\dot{m}_{c,surge}} \right)_{N_{op}} - 1 \quad (4)$$

Surge margin at constant mass flow is important for points of compressor acceleration due to shaft speed increasing before mass flow reaches steady state. Surge margin at constant speed is important for situations of unpredictable inlet conditions, in order to prevent a sudden drop in mass flow rate from stalling the engine.

Ideally, the design point would correspond with the peak efficiency of the compressor. However, the cycle designer selects the compressor design point based on criteria to meet overall engine or aircraft system requirements, such as thrust, fuel burn, or direct operating cost, instead of criteria to optimize the compressor itself, such as pressure ratio per stage or hub-to-tip radius ratio [10]. The end goal of the cycle designer is to guarantee that the cycle designed is realizable and buildable.

Scaling

Traditional practice is that, if the desired pressure ratio, efficiency, and mass flow rate combination are not available on any compressor maps within the cycle designer's library, any documented compressor map can become parametric by applying appropriate scaling laws. These scaling factors are defined by relating the design point of the reference map to the desired design point. The shaft speeds of the reference map are given as percent of design speed, and are therefore scaled by default to the new

application. The design point of the reference map is defined as the pressure ratio at maximum efficiency and 100% speed. These factors are then applied to the entire map by the following equations [11]:

$$s_m = \frac{\dot{m}_{c,desired}}{\dot{m}_{c,unscaled}} \quad (5)$$

$$s_{PR} = \frac{PR_{desired} - 1}{PR_{unscaled} - 1} \quad (6)$$

$$s_\eta = \frac{\eta_{desired}}{\eta_{unscaled}} \quad (7)$$

With concern for the compressor design, there are two possibilities that result from this map scaling approach: either the design scales with the map or it does not. If the compressor design can scale with the map and the resulting compressor design is an acceptable size, the performance map can be used and the design process can move on. If the compressor design either doesn't scale with the map or the map is scaled in a different manner than the compressor scaling allows, the reference design parameters are invalid. Because the goal of cycle design is to produce a buildable cycle, map scaling assumes that the design can scale with the map.

GE Oil & Gas has shown that a compressor design can in fact be aerodynamically scaled to achieve new performance [33]. The successful industrial gas turbines MS5001, MS6001, and MS9001 are aerodynamically scaled from the MS7001 machine. This scaling philosophy is based on the principle that inversely scaling the shaft speed to the physical scale will produce an aerodynamically and mechanically similar machine. In order to preserve the reference design features, velocity triangles must be fixed, and in

order to preserve the velocity triangles, blade linear speed must remain fixed. Therefore, shaft speed will scale as the inverse of the diameter scale factor, which will scale as the square root of the mass flow scale. Increasing the compressor diameter by 120%, which is the case of the MS9001 design, is not always acceptable, however. Also due to velocity triangles being preserved, pressure ratio and efficiency are required to be fixed. These scaling parameters are illustrated in Table 1.

Table 1: Physical Compressor Scaling Parameters

Scale Factor	0.5	1	2
Pressure Ratio	1	1	1
Efficiency	1	1	1
RPM	2	1	0.5
Velocities	1	1	1
Diameter	0.5	1	2
Flow	0.25	1	4
Weight	0.125	1	8
Tip Speed	1	1	1

The proven philosophy that aerodynamically scaling a machine is possible then leads to two restrictions: The scaled geometry must fit within size constraints of the application, or the required combination of pressure ratio, efficiency, and stall margin must be compatible on the existing performance map. In the cases where either of these limits is

violated, the reference design parameters are invalid and a new compressor will be designed with a unique performance map, i.e. the scaled map will also be invalid.

Compressor Design

The compressor is designed to achieve the design pressure ratio and mass flow requirements with adequate stall margin and high efficiency at the design point [10].

There are five steps of compressor design and build that are incorporated into industry practice of engine design. Each of these design phases introduces a higher order of fidelity, which increases the accuracy of the complex flow prediction for use in cycle analysis.

1. Meanline design
2. Multi-stream design
3. Throughflow optimization
4. CFD optimization
5. Build and rig test

Increasing fidelity through these phases is accomplished by increasing the number of calculations stations and employing higher-order calculations. Meanline design defines the annulus and performs calculations between blade rows along the mean radius of the compressor. At this phase, the flow is assumed to be inviscid and one-dimensional, and viscosity and other three-dimensional effects and losses are included through empirical models.

Multi-stream or multi-meanline design includes radial or spanwise variation of velocity triangles in order to determine blade twist along the meanlines of multiple concentric stream tubes that span the annulus. Multi-meanline analysis is based on radial equilibrium, meaning the radial pressure of each streamline must balance the centrifugal force of rotational flow, but still assumes airfoil profile and other empirical correlations for calculations between blade rows.

As mentioned, iterations with cycle design will occur throughout compressor design; however, compressor conceptual design is defined here as including meanline and multi-meanline design. Compressor conceptual design will provide the compressor designer the design parameters necessary for blade design, which is included in throughflow and three-dimensional optimization. Compressor conceptual design will also provide the compressor design features required for engine conceptual design calculations of weight, noise, and cost.

Throughflow optimization is based on streamline curvature analysis, meaning multiple fluid calculation stations are defined within each blade row along each streamline. This approach continues to assume radial equilibrium; however, blade profiles become design variables and loss models are no longer assumed.

Three-dimensional CFD optimization can most fully predict the three-dimensional flows at the endwall and boundary layer regions by making no assumptions of radial equilibrium or viscosity. This analysis is costly, however, due to the number of calculation stations and the calculations necessary at each station.

As mentioned, the end goal of the cycle designer is to guarantee that the cycle designed is realizable and buildable. A compressor conceptual design, analysis, and optimization method will provide a realizable performance map for cycle design. Meanline design combined with multi-meanline analysis methods are computationally inexpensive and can be incorporated into conceptual cycle design to allow performance maps to be coupled with their design parameters.

Performance Prediction

Compressor flow can be simply described as viscous, unsteady, and rotational, with very distinct pressure gradient. Because meanline and streamline analyses assume inviscid and two-dimensional flow, viscosity and three-dimensional effects are accounted for via loss models. These predict the pressure losses and under- or over-turning generated by specific secondary flows at various conditions. There are many loss models available throughout literature for compressor flow, including those compiled by Aungier [13] and Koch & Smith [14].

The history of compressor design has developed with a firm base in wind tunnel testing and airfoil and cascade analysis. These analyses have resulted in loss correlations that have defined the compressor design approach and have led to implementation of new design solutions. Although the empirical correlations derived for these loss models may lead to a better understanding of the physics of the flow, the loss models are still dependent on measured data [15].

Loss models are necessary not only to more accurately predict the optimal performance of the compressor, but also to predict the onset of unwanted performance within a compressor, such as surge or choke. Surge and choke must be predicted in order to delimit a useful performance map. Surge lines are particularly necessary for cycle design because compressors must be designed to avoid stall and surge at all operating conditions [17]. The characteristics and effects of stall and surge are very three-dimensional in nature. The assumptions required for axisymmetric streamline analysis methods do not allow for these 3D effects to be apparent to indicate if a simulated compressor is experiencing stall or surge. Other criteria have therefore been determined to indicate the onset of stall [18], which will provide limits or constraints on compressor operation to be included in design [17].

Observations

This brief summary of compressor design has revealed that:

- Compressor parameters can scale with performance maps; however, limitations to the scaling approach often require a new compressor design, rendering the scaled map invalid and requiring iteration with early phases of compressor design.
- Compressor multi-meanline design methods use comparatively very little computation time and are therefore applicable for use in conceptual design.
- Compressor flow and performance is difficult to predict, but loss models are available. These loss models can also predict boundaries of performance maps.

- Compressor optimization within cycle optimization is possible through zooming techniques.

In summary, a gap is observed between the traditional map scaling method and compressor design parameters which can be bridged by implementing compressor conceptual design, analysis and optimization into the cycle design process.

THESIS STATEMENT

A thesis has been developed to bridge this gap between compressor maps for cycle design and compressor design parameters:

A compressor conceptual design, analysis and optimization method can couple design parameters to performance maps, allowing the cycle designer to choose a suitable map that is attached to reasonable compressor design parameters.

In order to quickly create the compressor conceptual design that is most fit for an application and generate its performance map, four tasks must be accomplished.

1. Create a parametric compressor conceptual design method to define geometry
2. Implement loss models for performance analysis based on geometry
3. Find the design of optimal performance
4. Generate the performance maps for this design

Accomplishing these four tasks will provide a means for validating that this method will satisfy cycle requirements as well as compressor constraints at the cycle design point.

This method will adapt and combine the key elements of compressor multi-meanline design and performance prediction with decision-making methods, ensuring that requirements and constraints will be satisfied at the conceptual design phase. This method is first-principles based, augmented with empirical loss models.

RESEARCH QUESTIONS AND HYPOTHESES

Various methods for conceptually designing a compressor have been published in literature, including the processes used in designing the publicly available NACA compressors. Because of the interdependency of the many parameters which must be prescribed in order to achieve performance at a design point, there are multiple approaches to defining the compressor geometry. For example, the NACA five-stage transonic compressor stage count was determined by means of defining blade tip diffusion factors for acceptable stall-avoidance [22]. The NACA eight-stage subsonic compressor, using another approach, determined the stage count by means of lift coefficient of available airfoils. Due to the use of modern controlled diffusion and transonic airfoils, the use of diffusion factor or lift coefficient as a conceptual design parameter is no longer applicable.

Cohen, Rogers, and Saravanmutoo present a method of compressor design that involves enthalpy rise per stage as a design parameter [25]. Farokhi presents a similar method of design that uses a loading coefficient to determine the enthalpy rise per stage [24]. A conceptual compressor can be rather quickly designed through these processes. This design can be accomplished in four simple steps by assuming the flow to be adiabatic, steady state, and circumferentially uniform. All of these steps are derived from 1st principles and fully defined in literature except for defining blade geometries in Step 5. Empirical loss models are needed for this step as will be discussed.

1. Calculate the compressor inlet and discharge annulus geometry
2. Calculate the stage pressure ratios and annulus geometry
3. Sequentially solve the flow velocity triangles
4. Define the blade geometry parameters

The velocity triangles defined in Step 4 are necessary to achieve the stage pressure ratios; however, blade leading and trailing edge camber angles cannot be simply lined up with these flow angles and expected to achieve these flows. Incidence is the angle between the blade leading edge and the flow. After defining a design incidence angle, a model is required to determine appropriate deviation – the angle between the blade trailing edge and the flow – and blade metal angles.

Based on this design process and the loss models to be used, a number of low-fidelity design choices can be made which fully define the performance of a compressor. These design choices can be implemented as independent variables to define a parametric design space.

In order to fully define a compressor geometry from these four steps, more information is required than simply the cycle design point. Within these four steps to design a compressor, six assumptions are made to reduce the dimensionality of the problem.

The designer may need to be reminded that these assumptions are based on conceptual, low-fidelity design and are useful to reduce the design to $4N_s + 8$ design parameters, where N_s is the number of stages in the design. These assumptions will be

resolved as the design develops. These design parameters and their constraints will be discussed further in subsequent chapters.

Table 2: Assumptions for Conceptual Design

Assumptions for Conceptual Design
Constant hub, mean, or tip radius
Linear variation of axial velocity through the compressor
Constant work (enthalpy rise) per stage with the first stage lightly loaded
Repeating stage velocity triangles
Constant chord and thickness-to-chord ratio along the blade span
Fixed thickness-to-chord ratio and tip clearance through the machine

Given this procedure, the question follows regarding the trustworthiness of the approach:

Research Question 1: Will the parametric conceptual design method produce reasonable compressor geometries?

Compressor geometry has specific trends that are inherent to the nature of the component. These trends include a continuous decrease in annulus area through the flowpath and a smooth and almost constant pressure ratio distribution through the machine.

Hypothesis 1: The parametric design method enhanced with simplifying assumptions will allow the compressor design to follow known geometrical trends.

In order to demonstrate this conceptual design procedure, a well-documented compressor is needed that provides complete annulus and blade dimensions. The NACA five-stage transonic axial compressor meets this criteria, and the design parameters and measured performance data are found in a series of NACA reports [19][20][21][22][23]. After constructing the systems of equations necessary for compressor conceptual design, the design procedure can be followed using the requirements of the NACA five-stage compressor and evaluation of Hypothesis 1 will be possible as the conceptual design is validated against the measured data.

Following the trustworthiness of the compressor design method, the performance analysis must also be proven. Secondary flows introduce pressure losses based on flow location or geometrical factors. The presence of physical blades results in pressure losses due to profile shape, boundary layers on the blade surfaces, and shocks if the flow is transonic. Endwall flow through a blade tip clearance gap also has a significant effect. Once the geometries of the compressor and individual stages have been defined, these losses can be calculated and applied.

When these losses are calculated, it becomes necessary to reevaluate the compressor design to include their influences. As mentioned, blockage will influence the annulus area and therefore hub-to-tip radius ratios. The pressure losses are applied to stage

efficiencies as well as the overall efficiency and the original design assumptions are updated. This requires an iterative process to converge on the compressor efficiency.

Unless the three dimensional Navier-Stokes equations are applied, empirical correlations are necessary to account for physical geometries when determining compressor performance. The various models of these empirical correlations have individual effects on the prediction of compressor performance. The nature of these individual effects raises the following question:

Research Question 2: Will models of profile, endwall and shock losses and boundary layer predict correct trends of secondary flow effects to sufficiently predict compressor performance?

Compressor energy losses occur through many secondary flows. These flows can be described through associated loss models based on empirical regressions, and many loss models are available for use. The objective of this task is not to determine which loss model is most accurate, but to develop a compressor map generation method sufficient for conceptual cycle design that will satisfy engine requirements as well as compressor constraints.

An observation must be made that the developed methods are intended to ensure the feasibility of all compressor designs for a particular cycle application. These methods are not intended to improve the accuracy of the compressor performance prediction. The accuracy of the performance prediction is dependent upon the fidelity of the compressor model and the quality of the loss calculations.

This observation allows for use of any applicable loss models; however, as the research question implies, a choice must be made. It is seen that the loss models compiled by Aungier are based on theories and methods that have been validated by application to various compressor models.

Hypothesis 2: Streamline design enhanced by Aungier's empirical profile, endwall, shock and boundary layer loss models can be tuned to accurately predict correct trends of a compressor map.

In order to demonstrate loss prediction models, a well-documented compressor is again needed that provides complete inter-stage flow data as well as annulus and blade dimensions. The NACA five-stage transonic axial compressor again meets these criteria, for which the design parameters and measured performance data are found in a series of NACA reports [19][20][21][22][23]. After constructing the systems of equations necessary and including the loss models compiled by Aungier, the NACA five-stage compressor can be simulated and evaluation of Hypothesis 2 will be possible as it is validated against the measured data.

Experimental Resources

Conducting these experiments requires computational modeling and simulation capabilities. The following sections briefly describe the resources available to accomplish these experiments. While these specific tools have been selected for this research, the methodology developed in this proposal has been formulated

independently of the selected analysis codes. Therefore, the methodology should be applicable when other analysis codes are selected.

Numerical Propulsion System Simulation (NPSS)

The Numerical Propulsion System Simulation (NPSS) is a powerful software tool for engine thermodynamic cycle analysis. NPSS was originally developed by NASA Glenn Research Center in cooperation with the aerospace industry [28] and has since become the U.S. industry standard framework for propulsion system modeling and simulation. As an object-oriented code, NPSS supports the process of zooming for higher fidelity engine design [3]. NPSS includes a Newton-Raphson solver that is highly valuable for the proposed design method.

Object-Oriented Turbomachinery Analysis Code (OTAC)

The Object-Oriented Turbomachinery Analysis Code (OTAC) is a low-order analysis tool written in the NPSS framework and language and is also being developed by NASA Glenn Research Center [29]. OTAC was written to perform one-dimensional meanline analysis as well as streamline analysis of turbomachinery components including compressors and turbines of axial and centrifugal design. Models of turbomachinery components are structured to seamlessly integrate two-dimensional streamline analysis with zero-dimensional engine cycle analysis in NPSS through a combination of elements. These elements allow for the user to define the nature and geometry of the turbomachinery component, the fidelity of the analysis, as well as the mathematical approach based on the availability of various loss models [17].

TASK 1: PARAMETRIC MACRO GEOMETRY DEFINITION

Procedure

There are few “recipes” published for axial compressor macro geometry design. The design method developed for this research is based on the process laid out by Farokhi [24]. This design can be accomplished in four simple steps by assuming the flow to be adiabatic, steady state, and circumferentially uniform.

1. Calculate the compressor inlet and discharge annulus geometry
2. Calculate the stage pressure ratios and annulus geometry
3. Sequentially solve the flow velocity triangles
4. Define the blade geometry parameters

The goal of this parametric design process is to determine the geometry of a compressor design that is required for performance analysis. These steps are discussed in more detail in the following sections.

Number of Stages

As mentioned, the design will be reduced to $4N_s + 8$ design parameters, where N_s is the number of stages in the design. In order for this process to be used in an optimizer, which will facilitate the decision-making process, a fixed number of parameters are required. For this reason the number of stages will be defined initially. The stage count may still be varied by the designer and performance of optimal designs of varying stage

count is comparable. Varying stage loading by varying stage count will be reflected by the performance of the machine.

Compressor Annulus Geometry

Given the design point requirements of mass flow rate and inlet conditions, the first step is to calculate the annulus geometry at the compressor inlet. Other parameters that are needed are the inlet axial Mach number and a design decision of hub-to-tip radius ratio at the compressor face. Inlet Mach number may be governed by the speed of the aircraft or the discharge velocity of the upstream fan or low pressure compressor, and is used in determining the inlet flow area through equation (8).

$$\dot{m} = \rho V A_{flow} \quad (8)$$

Because of boundary layer growth in the inlet annulus, a value for blockage is applied through equation (9) to determine the physical compressor area. This physical area combined with the design choice of hub-to-tip radius ratio will determine the hub and tip radii.

$$A_{phys} = (1 + B)A_{flow} \quad (9)$$

$$r_{tip} = \sqrt{\frac{A_{phys}}{\pi \left(1 - \left(\frac{r_h}{r_t}\right)^2\right)}} \quad (10)$$

$$r_{hub} = r_{tip} \frac{r_h}{r_t} \quad (11)$$

$$r_{mean} = \frac{r_{hub} + r_{tip}}{2} \quad (12)$$

The compressor discharge annulus is defined in a similar manner, where the discharge flow conditions are based on the design pressure ratio and efficiency in equations (13) and (14).

$$P_{t,out} = \pi P_{t,in} \quad (13)$$

$$T_{t,out} = \tau T_{t,in} \quad (14)$$

$$\tau = \pi^{(\gamma-1)/\eta_c \gamma} \quad (15)$$

The axial velocity is often assumed constant through a conceptual compressor; however, a velocity or Mach number requirement is often given for appropriate combustor inlet conditions [19]. In order to facilitate a discharge velocity or Mach number limit, a velocity ratio is introduced, and the static conditions at the compressor discharge can be determined through use of equation (16).

$$T_{out} = T_{t,out} - \frac{C_x^2}{2c_p} \quad (16)$$

Given the flow area at the compressor exit through equations (8) and (9), a radius is needed for the overall annulus to be defined. At conceptual design, it is typical to define the tip, the hub, or the mean radius to be constant through the compressor, from which the other radii can be deduced [24]. If the mean radius is chosen to remain constant from the inlet, the hub and tip radii can be found through the following equation:

$$r_{tip,hub} = r_m \pm \frac{A_{phys}}{4\pi r_m} \quad (17)$$

Stage Calculations Note

There are two approaches to accomplishing Steps 2 through 4. One method is to carry out each step at each stage sequentially before moving to the next step. The other approach is to perform all four steps at each stage before moving to the next stage. These approaches are equivalent and the resulting design will be identical.

Stage Pressure Ratios and Annulus Geometry

Overall work is defined as the change in enthalpy across the system. This work is often distributed equally through the stages of a compressor; however, there are some texts that claim a necessary loading or unloading of the first stage [25][26]. In order to accommodate for this claim, a loading factor K_l is introduced that will load or unload the first stage in equation (18) and distribute the change to the other stages in equation (19).

$$\Delta h_{t,stage\ 1} = \frac{\Delta h_{t,cmp}}{N_s} (1 + K_l) \quad (18)$$

$$\Delta h_{t,stage} = \frac{\Delta h_{t,cmp}}{N_s} \left(1 - \frac{K_l}{StageCount - 1} \right) \quad (19)$$

Given the enthalpy rise at each stage, the next step is to determine the stage pressure ratios through the compressor. This will also check that the compressor will achieve the desired overall pressure ratio. This step is done by calculating the enthalpy at each stage entrance and then applying the following equation:

$$\pi = \left[\eta \frac{\Delta h_t}{h_{t1}} + 1 \right]^{\gamma/(\gamma-1)} \quad (20)$$

This calculation requires a polytropic stage efficiency, which can be calculated from the overall adiabatic efficiency of the machine through equation (56). At conceptual design, stage efficiency is typically assumed to be either constant or linearly varied through the compressor with an assumed mean value [23]. For this procedure, we will assume the polytropic efficiency as a fixed value for each stage adiabatic efficiency.

$$e_c = \left(\frac{\gamma - 1}{\gamma} \right) \frac{\ln PR}{\ln \left(1 + \frac{1}{\eta_c} \left(PR^{\gamma-1/\gamma} - 1 \right) \right)} + 0.05 \quad (21)$$

The stage annulus parameters, i.e. physical area and radii, are then defined through the same procedure as the compressor annulus parameters. Each stage inlet is based on the flow conditions of the upstream stage exit, and the stage exit flow conditions are based on the stage pressure ratio and assumption of stage efficiency. A simple linear estimation of blockage through the compressor can be given based on the NACA five-stage transonic compressor design values as:

$$B_{stage} = 0.00625 * BladeRow + 0.0375 \quad (22)$$

Velocity Triangles

The next step in designing a compressor is to determine inter-stage flow to be achieved by the blades. This is visualized through velocity triangles as shown in Figure 3. This can be done along one meanline or multiple streams at once, which will be discussed later.

As the reader can see from the figure, velocity triangles can be simply determined geometrically given a few parameter values. For the conceptual design at hand, the inlet axial velocity C_x and rotor linear speed U are known, and the design choice to not employ inlet guide vanes results in the flow being purely axial. The rotor inlet velocity triangle is fully defined from these attributes.

The rotor exit velocity triangle is defined by the Euler turbomachinery equation, equation (23), which is derived from first law of thermodynamics. The Euler equation is used to find the rotor exit tangential velocity, $C_{\theta 2}$. This, along with the known axial velocity and rotor speed, fully defines the rotor exit velocity triangle. For this design, all change in axial velocity is assumed to occur across rotor blade rows [23].

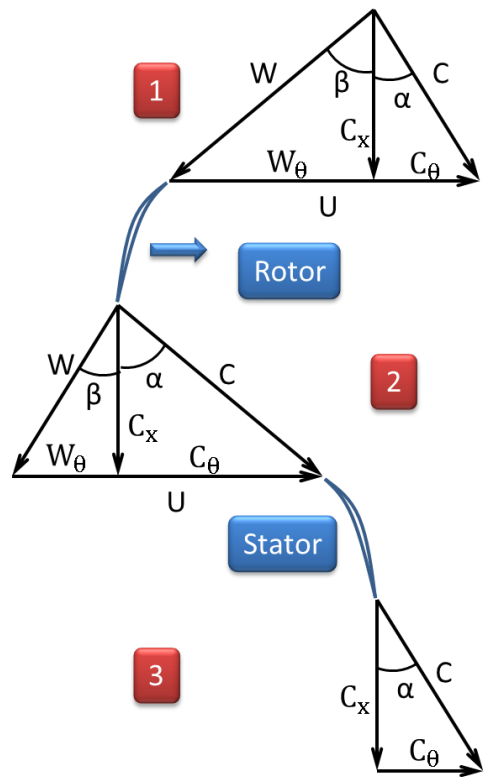


Figure 3: Velocity Triangle Parameters

$$\Delta h_{t,stage} = h_{t2} - h_{t1} = N(r_{m2}C_{\theta2} - r_{m1}C_{\theta1}) \quad (23)$$

The stator inlet velocity triangle is given by the absolute velocity components of the rotor exit flow.

In order to calculate the velocity triangle for the stator exit, an assumption or design decision must be made about the exit flow angle α_3 . This is typically done in conceptual design by assuming a repeating stage, meaning each stage entrance will have the same velocity triangles at the meanline [24]. In the design case where axial velocity varies through the compressor, a perfect repeating stage is not applicable; however, repeating rotor-inlet absolute flow angles are achievable. This assumption requires the stator exit absolute flow angle to be equal to the rotor inlet absolute flow angle to condition the flow for the downstream rotor. In the case of this conceptual design with no IGVs, the flow must be axial at the exit of each stage. Therefore, the velocity triangle for the stator exit is defined by a flow vector in the axial direction coupled with the blade linear speed.

Blade Geometry

The defined velocity triangle sequence is necessary to achieve the stage pressure ratio; however, blades cannot be simply lined up with these flow angles and expected to achieve these flows. Assuming a design incidence angle, the leading edge blade metal angle can be defined; however, iteration is required to determine deviation and the trailing edge blade metal angle. The blade metal angles κ_1 and κ_2 , as shown in Figure 4,

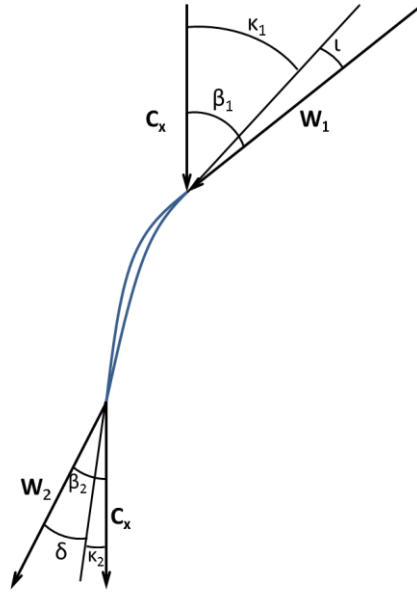


Figure 4: Blade Geometry Parameters

are the angles between the mean camber line and the rotational axis at the leading and trailing edges of the blade, respectively. These angles are determined based on

calculations of minimum-loss incidence and deviation angles. Incidence ι is the angle between the inlet flow velocity vector and the leading edge blade metal angle. Deviation δ is the angle between the exit relative flow velocity vector and the trailing edge blade metal angle.

The design incidence angle has been shown in practice to be desired close to zero [23]. For this procedure, we will assume it to be zero. Deviation angles must be determined by means of empirical models due to their dependence on and relationship to airfoil shape. Empirical models have been defined for various airfoil shapes, the most significant being NACA 65 series subsonic airfoils and double circular arc (DCA) transonic

airfoils. The airfoil family can be determined based on the blade tip inlet relative Mach number, which would be found in the velocity triangle calculations. Farokhi states that DCA airfoils are useful for relative Mach numbers above 0.8.

The deviation model compiled by Aungier is selected to for this research because it is based on theories and methods that have been validated by application to various compressor models. Aungier's deviation model is based on the NACA 65-series airfoil cascade data. The low-Mach-number deviation angle at min loss is given by:

$$\delta^* = K_{sh}K_{t,\delta}(\delta_0^*)_{10} + m\theta \quad (24)$$

Where $(\delta_0^*)_{10}$ is the deviation angle for a 10% thick airfoil at zero camber. The factor m corrects for different camber angles. The factors K_{sh} and $K_{t,\delta}$ are empirical corrections for airfoil shape and airfoil thickness, respectively.

This empirical deviation model is related to the blade camber angle, or the difference between the inlet and exit blade metal angles, as well as shape and thickness factors to account for variation from a standard shape and thickness. Deviation is highly dependent on the design choices of aspect ratio, thickness-to-chord ratio and solidity, as well.

This requires the designer to determine blade parameters of chord, thickness-to-chord ratio, tip clearance, number of blades, and airfoil before calculating deviation. Because the number of blades is a discrete variable, it will be determined by blade row solidity. Chord will also be determined by aspect ratio, in order to preserve the parametric value

of this procedure. By allowing the chord to be determined by aspect ratio, which is non-dimensional and exists within reasonable limits, the chord will not be limited to ranges specific to compressor size until constraints are defined.

The blade parameters that are needed to perform this calculation include blade chord, thickness-to-chord ratio, aspect ratio, solidity, and number of blades. Because the number of blades is a discrete variable, it will be determined by blade row solidity. Chord will also be determined by aspect ratio, in order to preserve the parametric value of this procedure. By allowing the chord to be determined by aspect ratio, which is non-dimensional and exists within reasonable limits, the chord will not be limited to ranges specific to compressor size until constraints are defined. These inputs are therefore reduced to thickness-to-chord ratio, aspect ratio, and solidity, all of which are independent and have recommended value ranges. The number of blades and chord are calculated from the following equations:

$$c = \frac{r_{tip} - r_{hub}}{AR} \quad (25)$$

$$N_{blades} = \text{ceil}\left(\frac{2\pi r_m \sigma}{c}\right) \quad (26)$$

Independent Variables

Based on this design process, a number of design choices are made in order to fully determine the geometry of the compressor. Following the four steps described, the parameters required for the design can be noted.

In order to define the compressor annulus, the cycle design variables of overall pressure ratio, efficiency, and mass flow rate will be provided from the cycle design. The compressor inlet axial Mach number, inlet hub-to-tip radius ratio, and inlet-to-discharge velocity ratio are design choices that can be determined from considerations of existing diffuser and combustor designs [23]. Because cycle analysis is typically zero-dimensional, these values could be based on ranges of existing diffuser and combustor characteristics. In order to reduce the dimensionality of this problem, a constant hub, mean, or tip radius can be assumed to define the exit radial dimensions. Inlet fluid conditions are also needed, but would be provided by the cycle analysis and would not be varied through iterations of compressor design.

According to this analysis, three parameters and a choice of constant radius are required to define the compressor annulus for Step 1.

The calculation for constant stage enthalpy rise per stage requires the number of stages and a loading factor, K_l . Each stage annulus can then be determined given a value for stage exit axial velocity, which is determined by the inlet-to-discharge velocity ratio. Step 2 can be completed from two parameters.

After defining the stage annulus parameters, the only parameter needed for the definition of velocity triangles through each stage is the stator exit flow angle. This parameter is determined by a design assumption of repeating stages, requiring the stator exit flow angle to be equal to the rotor inlet flow angle at each stage.

The blade geometry and loss prediction models require more detailed parameters, especially if the parametric compressor model will be used for streamline analysis. The parameters needed for these models are:

- Airfoil class
- Aspect Ratio
- Solidity
- Chord length
- Thickness-to-chord ratio
- Blade tip clearance
- Number of blades in the blade row

Chord and thickness-to-chord ratio can be varied with respect to blade span to increase fidelity, but assumptions of constant values along the blade are acceptable for conceptual design. Airfoil class, meaning NACA 65 or double circular arc series airfoils, is determined by the relative Mach number at tip of each blade row. Equations (25) and (26) also provide relationships between aspect ratio, chord, solidity, and blade count, allowing the dimensionality of the design to be reduced by two parameters per blade row. The dimensionality can be reduced even further by assuming a fixed thickness-to-chord ratio and a constant clearance or clearance-to-chord ratio for all blade rows. These would have to be carefully monitored, however, due to the optimizer exploiting

these assumptions. From this analysis, it is seen that as few as two parameters are needed at each blade row, and two parameters that are to be fixed for each blade row.

Overall, it is seen that a compressor design can be completed given only nine overall parameters and two per blade row parameters. The dimensionality of the design is reduced to these variables due to six assumptions. Of these variables, three overall parameters would be given by the cycle designer from the cycle analysis, and four would be determined based on acceptable ranges. These parameters can be rearranged into the following groups.

- Cycle performance parameters
 - Design Pressure Ratio
 - Design Efficiency
 - Design Actual Mass Flow Rate
 - Design Actual Shaft Speed
- Overall Compressor Inputs
 - Inlet Axial Mach number
 - Inlet hub-to-tip radius ratio
 - Stage Loading Coefficient
 - First Stage Loading Factor
 - Constant thickness-to-chord ratio
 - Constant blade tip clearance
- Inputs for Each Blade Row
 - Aspect Ratio
 - Solidity
- Assumptions
 - Constant hub, mean, or tip radius through the compressor
 - Constant axial velocity through the compressor

- Constant work per stage
- Repeating stage velocity triangles
- Constant chord along blade span
- Constant thickness-to-chord ratio along blade span

The blade row inputs are limited by mechanical or manufacturing constraints. Blade tip clearance is constrained by a minimum clearance value, determined by manufacturing ability or cost, and by a maximum clearance-to-chord or clearance-to-span ratio, determined by design experience.

Blade row solidity, a function of chord and number of blades, is bounded by a maximum value determined manufacturing ability as well as compressor weight and a minimum value determined by flow turning ability. Solidity is also limited to calculate whole numbers for blade count. Blade count must also be limited from being equal to the blade row immediately upstream due to the pressure oscillations that would occur. Minimum chord length will keep the leading edge radius realistic. Overall constraints of tip diameter and compressor length will limit hub-to-tip radius ratio and chord length.

These ranges and limits will result in a design space spanned by the compressor design parameters.

Experiment: Five-Stage Compressor Conceptual Design

Hypothesis 1: The parametric design method enhanced with simplifying assumptions will allow the compressor design to follow known geometrical trends.

In order to demonstrate this compressor conceptual design method, a well-documented compressor is needed to that provides complete inter-stage flow data as well as annulus and blade dimensions. The NACA five-stage transonic axial compressor meets this criteria, and the design parameters and measured performance data are found in a series of NACA reports [19][20][21][22][23]. For this study, a similar compressor will be designed via the method described and using parameter values chosen by the NACA designers.

Table 3: NACA 5 Stage Compressor Design Point

Design Point Parameter	Value
Overall Pressure Ratio	5
Mass Flow Rate	67.5 lbm/s
Shaft Speed	12605 RPM
Target Adiabatic Efficiency	80%

The goal of this experiment is to demonstrate the rapid design a five stage axial-flow compressor to the design point parameters given in Table 3.

The reader is reminded that the NACA five-stage transonic compressor is a complete and fabricated design, meaning some parameters were refined after completion of the conceptual design. The conceptual design method presented here does not include those refinements. Following the procedure laid out above, a compressor can be designed which is very similar to the compressor built in 1954. This process will use the

tools provided in OTAC to enhance the dexterity of the equations. The reader is directed to texts by Scott Jones at NASA Glenn or Russell Denney at Georgia Tech for details on this code[29][27].

The equations presented here become more versatile through OTAC, meaning this code contains more equations related to compressor flow than are presented in this text. For example, given the mass flow rate and inlet flow conditions of STP, an NPSS FlowStation element is created which determines the remaining stagnation properties. Enabling the OTAC functions and applying the shaft speed and design Mach number of 0.6 to this

Table 4: Inlet Annulus Parameters

Inlet Parameter	Study Compressor	NACA Compressor
Total Pressure	14.696 psia	14.7 psia
Total Temperature	518.7 R	518.7 R
Total Enthalpy	123.959 BTU/lbm	123.969 BTU/lbm
Axial Mach Number	0.60	0.60
Axial Velocity	647.183 ft/s	648 ft/s
Blockage	0.01	0.01
Radius Ratio rh/rt	0.50	0.50
Tip Radius	10.009 in	10.0 in
Hub Radius	5.004 in	5.0 in
Mean Radius	7.506 in	7.50 in

FlowStation defines the static flow properties and flow area. The NACA designers chose an inlet hub-to-tip radius ratio of 0.5 and assumed a blockage value of 1% based on previous experience, from which the physical area and then radii are calculated [23]. Thus the compressor face conditions and geometry are quickly defined and are shown in Table 4 compared to the NACA compressor.

Given the design pressure ratio and target adiabatic efficiency, the discharge FlowStation stagnation flow conditions can be set via equations (13) through (15). NACA designers chose the discharge velocity to be 520 ft/s due to combustor constraints [23], defining the velocity ratio to be $V_{discharge}/V_{inlet} = 0.8$. The static properties are then set through equation (16). The design decision for this compressor to be a constant tip machine and an assumed blockage value of 10% define the discharge conditions and annulus geometry as shown in

Table 5. The NACA compressor documentation does not give flow conditions or Mach number; however, they can be easily estimated from the design pressure ratio and efficiency using an assumption of constant c_p .

The stage count of the NACA compressor was determined through considering diffusion factor and stage pressure ratio simultaneously. Stagewise distributions of these parameters were chosen “from considerations of individual stage blade-loading limitation and off-design performance characteristics” [23]. Knowing that the study compressor will have five stages like the NACA compressor, this decision is made for us.

Table 5: Discharge Parameters

Discharge Parameter	Study Compressor	NACA Compressor
Total Pressure	73.48 psia	73.5 psia
Total Temperature	891.871 R	890.92 R
Total Enthalpy	214.271 BTU/lbm	213.63 BTU/lbm
Total Enthalpy Rise	90.312 BTU/lbm	89.665 BTU/lbm
Axial Velocity	520 ft/s	520 ft/s
Axial Mach Number	0.361	0.362
Blockage	0.10	0.10
Radius Ratio r_h/r_t	0.827	0.825
Tip Radius	10.009 in	10.0 in
Hub Radius	8.275 in	8.25 in

Furthermore, it was noted by the NACA designers that “improved off-design performance could be obtained by lightly loading the inlet stages, designing the intermediate stages close to their loading limit, and moderately loading exit stages” [23]. For the purpose of this study, only the first stage of the new design will be lightly loaded to demonstrate the capability of this assumption. It is found that a loading factor of $K_l = -0.23$ results in a pressure ratio distribution fairly well matched with the NACA design. This loading factor brings the first stage loading coefficient to $\psi = 0.47$: still

highly loaded, but within the suggested range suggested by Farokhi of $0.2 \leq \psi \leq 0.5$ [24].

The stage efficiencies required for equation (20) were assumed by the NACA designers based on previous experience. For the purpose of this study, the polytropic efficiencies will be calculated from the overall target efficiency and held constant through the stages. These efficiency values and their corresponding pressure ratios are shown in Table 6.

It is seen from the resulting overall pressure ratio that the study compressor should exceed the design pressure ratio requirement with losses present through the efficiency estimation.

Table 6: Stage Pressure Ratios

Stage	Study Compressor Stage Pressure Ratio	NACA Compressor Stage Pressure Ratio
1	1.374	1.390
2	1.462	1.480
3	1.405	1.425
4	1.361	1.350
5	1.326	1.265
Overall	5.094	5.006

Stage annulus geometries are then determined through equations (8) through (17), except that temperatures are known from the calculation of work and need not be recalculated. The results of the conceptual design are compared to the NACA design in Figure 5. The reader is reminded that this is a conceptual compressor design, and hub contours are not smoothed at this time. It is seen that the geometry of the study compressor matches very well to the NACA compressor and the hub contour is within 3% error.

The calculation procedure in OTAC uses the option of calculating Steps 3 and 4 simultaneously at each blade row before moving on to the blade row. In order to calculate the input variables for OTAC,

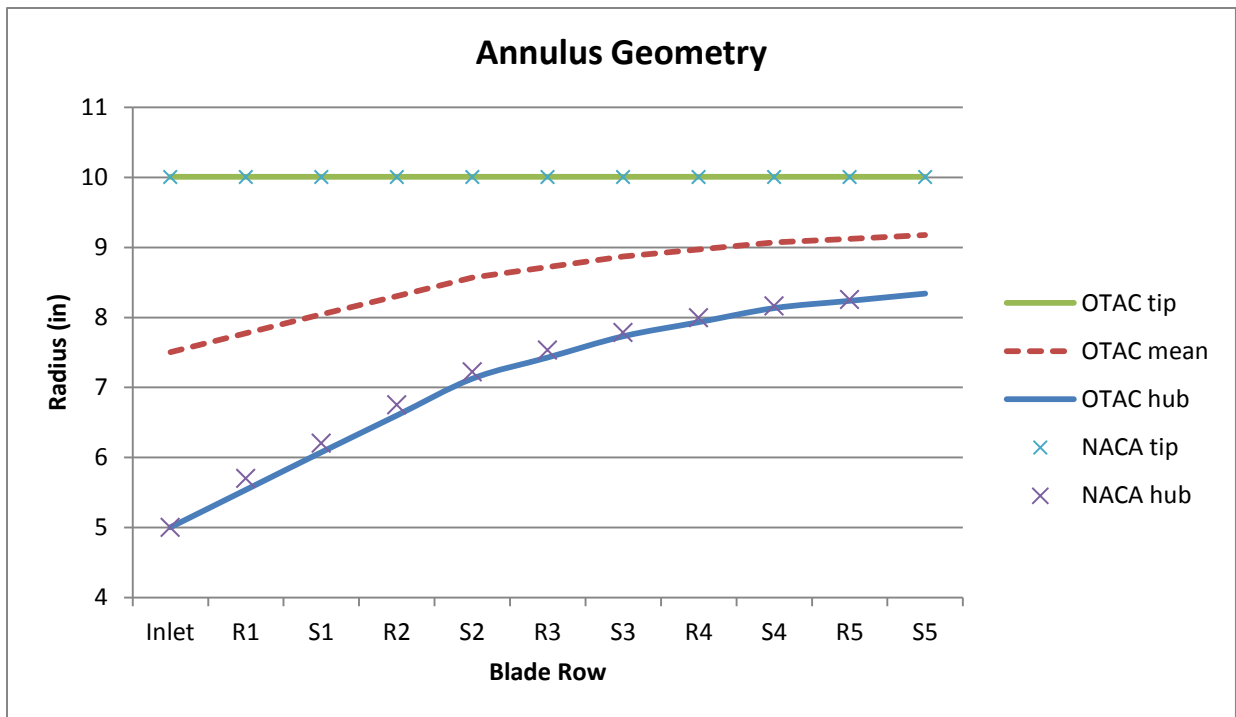


Figure 5: Comparison of Study Compressor to NACA Compressor Annulus Geometry

Table 7 gives the values of aspect ratio and solidity.

The airfoil for each blade row is determined based on the blade row inlet Mach number, which is calculated by a FlowStation. The parameters at the mean radius of each blade row are given in

Table 8.

Table 7: NACA Design Values of Aspect Ratio and Solidity

	R1	S1	R2	S2	R3	S3	R4	S4	R5	S5
AR	1.86	2.09	1.39	1.56	0.96	1.17	0.77	0.96	0.68	0.86
σ	1.15	1.23	1.30	1.35	1.40	1.33	1.25	1.25	1.05	1.15

Table 8: Blade Row Parameters

Blade Row	Chord	Thickness-to-Chord Ratio	Tip Clearance	Blade Count	Airfoil
R1	2.70	0.065	0.015	23	DCA
S1	2.025	0.0725	0.015	33	DCA
R2	2.73	0.065	0.015	27	DCA
S2	2.050	0.0725	0.015	39	DCA
R3	2.89	0.065	0.015	28	DCA
S3	2.044	0.0725	0.015	37	NACA65

R4	2.87	0.065	0.015	25	DCA
S4	2.042	0.0725	0.015	36	NACA65
R5	2.71	0.065	0.015	23	DCA
S5	2.042	0.0725	0.015	35	NACA65

Observations

A compressor conceptual design process is discussed which allows for low-fidelity compressor geometry to be defined by only $4 \times (\text{Number of Stages}) + 8$ parameters. The dimensionality of this process is reduced by the acceptance of six assumptions common to compressor conceptual design. These assumptions allow for the resultant compressor geometry to follow expectant trends such as continuously reducing annulus

Table 9: Macro Geometry Parameters, Outputs and Assumptions

Table 10: NACA 5 Stage Compressor Design Point and Parameters

Cycle Parameter Input		Value
Overall Pressure Ratio		5
Mass Flow Rate		67.5 lbm/s
Shaft Speed		12605 RPM
Target Adiabatic Efficiency		80%

Parameters	Value	Blade Row	AR	σ
Inlet Mach Number	0.6	R1	1.86	1.15
Inlet Hub/Tip Radius Ratio	0.5	S1	2.09	1.23
Inlet/Discharge Velocity Ratio	0.8	R2	1.39	1.30
Number of Stages	5	S2	1.56	1.35
1 st Stage Loading Factor	-0.23	R3	0.96	1.40
Constant Radius	Tip	S3	1.17	1.33
Blade Thickness/Chord Ratio	0.065	R4	0.77	1.25
Blade Tip Clearance	0.018	S4	0.96	1.25
		R5	0.68	1.05
		S5	0.86	1.15

area through the compressor. This process also allows for the blades at each blade row to be defined as double circular arc or NACA 65 series airfoils based on the relative Mach number at the blade tip. These output parameters are needed for performance prediction.

The geometry for a compressor was defined based on the NACA five-stage compressor design point and given some documented parameters. This geometry is shown to be very similar to the manufactured compressor, proving the parametric macro geometry definition process to be valid for the compressor conceptual design method.

TASK 2: PERFORMANCE ANALYSIS

Loss Models

Secondary flows introduce pressure losses based on flow location or geometrical factors. The presence of physical blades results in pressure losses due to profile shape, boundary layers on the blade surfaces, and shocks if the flow is transonic. Endwall flow through a blade tip clearance gap also has a significant effect. Once the geometries of the compressor and individual stages have been defined, these losses can be calculated and applied.

As observed earlier, there are many loss models available in literature, and the purpose of this research is not to improve accuracy or even to determine which loss models are most accurate. The profile, shock and endwall loss models compiled by Aungier are selected to for this research because they are based on theories and methods that have been validated by application to various compressor models. Denney, et. al, describes these loss models very succinctly [27].

Profile Loss

Aungier's method for profile loss is based on the NACA 65-series airfoil cascade data documented in NASA SP-36 [31]. This data was collected at low Mach numbers and adjustments are made for higher Mach numbers. The incidence angle of minimum loss is given by:

$$l^* = K_{sh}K_t(l_0^*)_{10} + n\theta \quad (27)$$

Where $(l_0^*)_{10}$ is the minimum-loss incidence angle for a 10% thick airfoil at zero camber. The factor n corrects for different camber angles. K_{sh} and K_t are empirical corrections for airfoil shape and airfoil thickness, respectively. The loss parameter at min-loss incidence is given in equation (28) as a function of min-loss equivalent diffusion factor (equation (29)). The parameters K_1 and K_2 are empirical factors that depend on airfoil geometry and Reynolds number.

$$\frac{\bar{\omega}^* \cos \beta_2^*}{2\sigma} \left(\frac{W_1^*}{W_2^*} \right) = K_1 \left[K_2 + 3.1(D_{eq}^* - 1)^2 + 0.4(D_{eq}^* - 1)^8 \right] \quad (28)$$

$$D_{eq}^* = \left(\frac{W_{max}}{W_1} \right)^* \frac{W_1^*}{W_2^*} \quad (29)$$

These equations are applicable for low Mach numbers. Mach number corrections are then applied for $M_{crit} < M_1' < 1$, M_1' being the fluid Mach number at the entrance to the blade passage. The minimum loss after Mach number corrections are applied is designated $\bar{\omega}_m$. Additional corrections are applied for operation off-design, i.e. at an incidence angle other than the min-loss incidence angle.

Shock Loss

The shock loss calculations are applied when $M_1' > 1$. The loss is assumed to be the relative stagnation pressure drop across a normal shock, the normal shock strength being determined by the average passage Mach number calculated in equation (30).

$$M_{pass} = \sqrt{M_1' M_2'} \quad (30)$$

M'_1 may be thought of as the Mach number in the center of the blade passage and M_2 may be thought of as the Mach number at the blade surface after Prandtl-Meyer expansion turning from M'_1 through an angle ϕ determined by:

$$\tan \phi = \frac{s \cdot \cos \psi}{s \cdot \sin \psi + R_u} \quad (31)$$

Where $\psi = 90 - \theta_u/2 - \gamma$ is the suction surface inlet angle, θ_u is the suction surface “camber angle” given by equation (32), and R_u is the suction surface radius of curvature given by equation (33). These equations assume double circular arc blade geometry.

$$\tan \frac{\theta_u}{4} = \tan \frac{\theta}{4} + \frac{t_b}{c} \quad (32)$$

$$R_u = \frac{c}{2} \sin \frac{\theta_u}{2} \quad (33)$$

Endwall Loss

The clearance gap total pressure loss is given in equation (34) as a function of leakage flow \dot{m}_c , equation (35), and the difference in pressure across the gap defined by equation (36).

$$\Delta P_t = \frac{\Delta P \dot{m}_c}{\dot{m}} \quad (34)$$

$$\dot{m}_c = \rho U_c N_{blades} \delta_c c \cdot \cos \gamma \quad (35)$$

$$\Delta P = \frac{\tau}{N_{blades} r_{tip} \delta_c c \cdot \cos \gamma} \quad (36)$$

The pressure difference across the gap must balance the blade torque:

$$\tau = \pi \delta_c [(r \rho C_m)_1 + (r \rho C_m)_2] [r_2 C_{\theta 2} - r_1 C_{\theta 1}] \quad (37)$$

The total clearance loss as calculated from the above equations is distributed linearly from the tip to the root of the blade, i.e. from the rotor tip or casing to the inner hub, and vice-versa for the stator.

Off-Design Loss

Away from the design point, the loss is assumed to vary depending on whether the incidence angle is on the stall or choke side of the minimum, according to the following three equations:

$$\bar{\omega} = \bar{\omega}_s + \bar{\omega}_m (1 + \xi^2) \text{ for } -2 \leq \xi \leq 1 \quad (38)$$

$$\bar{\omega} = \bar{\omega}_s + \bar{\omega}_m (5 - 4(\xi + 2)) \text{ for } \xi < -2 \quad (39)$$

$$\bar{\omega} = \bar{\omega}_s + \bar{\omega}_m (2 + 2(\xi - 1)) \text{ for } \xi > 1 \quad (40)$$

Where ξ is the measure of incidence angle away from min-loss.

$$\xi = \frac{l - l_m}{l_s - l_m} \text{ for } l \geq l_m \quad (41)$$

$$\xi = \frac{l - l_m}{l_m - l_c} \text{ for } l < l_m \quad (42)$$

When these losses are calculated, it becomes necessary to reevaluate the compressor design to include their influences. The pressure losses are applied to stage efficiencies as

well as the overall efficiency and the original design assumptions are updated. This requires an iterative process to converge on the compressor efficiency.

Multiple-meanline Fidelity

As mentioned earlier, multiple-meanline analysis introduces radial or spanwise variation of velocity triangles in order to determine blade twist through multiple stream tubes. The radial variations occur due to the blade speed increasing with radius. As a result of this radial change in blade speed, the blade airfoil geometry must vary to match the flow [5]. It should be noted that, for the convenience of this calculation, free vortex flow is assumed, which sets the rotor inlet absolute tangential velocity distribution as constant angular momentum profiles at all radii ($rC_\theta = \text{constant}$) [13]. This means that specific work is constant across the span of each blade row.

Streamline design introduces radial variation through a compressor by defining the compressor annulus as multiple concentric annuli, also known as stream tubes, each

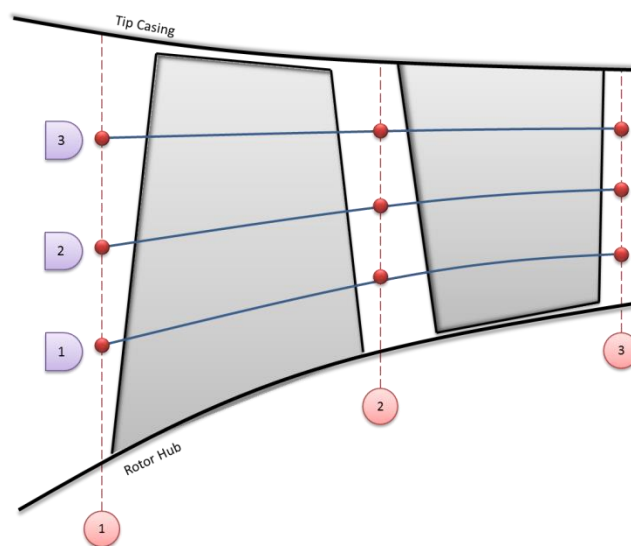


Figure 6: Streamline Design Calculation Stations

containing the same mass flow, as shown in Figure 6. In meanline analysis, the meanline location is fixed by the annulus geometry. In streamline analysis, the streamlines, or mean lines of each stream tube, are determined based on the stream tube annulus geometry. The assumption is made that no mass, momentum or energy transfer occurs between stream tubes. This assumption allows for simple radial equilibrium to be applied.

Introducing multiple streamlines results in a system of governing physics equations that must be solved at the entrance and exit of each blade row. These governing equations satisfy continuity, momentum and energy requirements by assuming the flow to be adiabatic, steady state, and circumferentially uniform. This system is comprised of $7n$ equations as follows, where n is the number of streamlines [17]:

$$n \quad \dot{m}_{1,i} = \dot{m}_{2,i} \quad (43)$$

$$n \quad h_{t2,i} - h_{t1,i} = \omega(r_{2,i}V_{\theta 2,i} - r_{1,i}V_{\theta 1,i}) \quad (44)$$

$$n - 1 \quad \frac{dP}{ds_{2,i}} = \rho_{2,i} \frac{V_{\theta 2,i}^2}{r_{2,i}} \cos \phi_{2,i} \quad (45)$$

$$n \quad P_{t2,i} = P_{t2ideal,i} - f_{loss} \quad (46)$$

$$1 \quad A_{2,mean,flow} = A_{2,mean,machine} \quad (47)$$

$$1 \quad r_{2,mean,flow} = r_{2,mean,machine} \quad (48)$$

$$n - 1 \quad r_{2inner,i+1} = r_{2outer,i} \quad (49)$$

$$n \quad \phi_{2,i,flow} = \phi_{2,i,machine} \quad (50)$$

$$n \quad \beta_{2,i} = \kappa_{2,i} + \delta_i \quad (51)$$

Equation (43) requires flow continuity through each blade segment. Equation (44) is the Euler turbomachinery equation that describes conservation of energy and momentum. Equation (45) is radial equilibrium, which balances the fluid pressure and centripetal acceleration for multiple radial segments. Equation (46) computes the fluid total pressure at the blade exit as the ideal total pressure minus a function of pressure losses due to secondary flows. Equations (47) and (48) represent the physical constraint of the machine annulus geometry on the fluid, assuming the flow to be unseparated but allowing for a boundary layer in the flow area. Equations (49) and (50) define the boundaries of the stream tubes. Equation (51) is then solved for the blade geometry at each streamline as described in the section above. This system of equations allows for each streamline to be analyzed along its own streamline through the meanline process described above. This facilitates the calculation of velocity triangles and deviation at multiple streams through the annulus.

The fidelity of the model is determined by the number of streams analyzed. It has been found that five to seven streams accounts for enough detail in the loss models to be a good balance between simplicity and reality.

Solver

These iterations have now become viable due to reliable solvers. While a number of methods exist for finding solutions to systems of many variables, the gradient-based Newton-Raphson approach is commonly used due to its quadratic convergence rate [5]. The Newton-Raphson approach is a gradient based solver that can be controlled by step size and convergence reference and is successfully integrated into the NPSS framework.

Stall Prediction

One objective of off-design analysis before selection of a compressor design is to determine stall margin. As mentioned, it is imperative that a compressor avoid stall at all operating conditions in order to protect the machine.

The first step in simulating compressor stall is to select a suitable stall indicator as characterized by the flow. There are three main flow characteristics described by Aungier that can be easily used as indicators of stall in axisymmetric throughflow analysis [13]. Two of these stall criteria have been selected for evaluation in the present study. Only one of these criteria must be satisfied to indicate the onset of stall. The following sections describe the major features of these stall criteria, and additional details may be found in Aungier's book.

Pressure versus Flow Characteristic

The first stall criterion is based on the machine pressure versus flow characteristic. It is known that a compressor is theoretically unstable when operating in the region where the gradient along a constant speed line is positive [16]. Therefore, an obvious stall criterion can be defined as the point at which the gradient of the constant speed line is equal to zero. In order to find this point, the machine pressure versus flow characteristic must be evaluated at multiple mass flow settings, and stall would be indicated during a post process analysis. The process for implementing this criterion in OTAC is as follows:

1. Determine the shaft speed at which to analyze the compressor. The shaft speed may be expressed in revolutions per minute or percent of the design shaft speed.
2. Determine a mass flow at which to begin analysis at the selected speed. This starting point must be where the gradient of the characteristic is known to be negative.
3. Decrease the mass flow rate in increments, the size of which depends on the fidelity required. At each flow rate, simulate the compressor and determine the overall pressure ratio (OPR).
4. When the OPR calculated is less than the OPR at the previous mass flow rate, indicate that the compressor is unstable and stop the analysis.

Boundary Layer Separation

The second stall criterion is a limit on a flow separation characteristic due to blade or endwall boundary layer. The flow separation characteristic requires boundary layer analysis. Endwall boundary layer analysis is known to be very complex and Aungier states that the current state of the art theory is insufficient to analyze the complete endwall flow problem. Nevertheless, Aungier's discussion of endwall boundary layer theory includes the influence of several important features identified through experimental and analytical studies, and will be sufficient for this work. Aungier's approach to identifying endwall boundary layer is coupled with the prediction of endwall blockage, and will be modeled in OTAC at a later date.

It is expected that the boundary layer separation criterion will become effective at low speeds. Research by Kulkarni shows that blockage, and therefore boundary layer, increases with decreasing mass flow[30]. Furthermore, Kulkarni describes increased boundary layer growth in the leading stages at low speeds and flows. This observation is supported by the knowledge that limiting stall progresses forward in the compressor with decreasing speed.

The indication of stall due to boundary layer separation will be employed in the near future, but was not included in the present study. The method for implementing this theory into OTAC is documented in a later section of this report.

Equivalent Relative Velocity Ratio

Aungier's third stall criterion is based on a relative velocity ratio characteristic through each blade row. Various stall criteria due to relative velocity ratio characteristics have been proposed. A well-known stall criterion was proposed by de Haller in 1953 as indicating stall when:

$$\frac{W_2}{W_1} < 0.72 \quad (52)$$

This "de Haller limit" is generally accepted as applicable to stall at the endwalls, but not necessarily along the span of a blade. This is due to the various blade geometries commonly used in modern compressors, such as the controlled diffusion blade profile, whose purpose is to reduce separation while achieving lower velocity ratios.

Aungier developed a stall criterion related to an equivalent velocity ratio characteristic. The equivalent relative velocity ratio is based on the fact that one employs pressures

rather than velocities when applying an incompressible flow model to compressible flow.

$$W_{RE} = \sqrt{\frac{P_{t2rel} - P_{s2}}{P_{t1rel} - P_{s1}}} \quad (53)$$

Assuming that blade rows in a compressor act as simple two-dimensional diffusers, a limit on the equivalent velocity ratio can be found that will indicate stall.

$$W_{RE \text{ limit}} = \frac{\left(0.15 + 11 \frac{t_b}{c}\right) / \left(0.25 + 10 \frac{t_b}{c}\right)}{1 + 0.4 \left(\frac{\theta\sigma}{2 \sin(\theta/2) \cos \gamma}\right)^{0.65}} \quad (54)$$

This limit is a correlation identified by Aungier between the velocity ratios and geometry of simple two-dimensional diffusers operating at their peak static pressure recovery. Peak static pressure recovery in an axial compressor is recognized as an indicator of stall. The numerator in this limit equation represents a weak correlation to the thickness-to-chord ratio of the blade row, and $\theta\sigma / (2 \sin(\theta / 2) \cos \gamma)$ in the denominator assumes the geometry of the two-dimensional diffuser as having a length approximated by a circular-arc camberline and breadth approximated by the staggered spacing. This geometry correlation value must be bounded from excessive extrapolation from the regressed data, and Aungier imposed that:

$$\frac{\theta\sigma}{2 \sin(\theta/2) \cos \gamma} \geq 1.1 \quad (55)$$

An adaptation of $W_{RE\ limit}$ was found for highly diffusing blades, where wake blockage can be expected to affect stall. This adaptation is correlated to the equivalent diffusion factor. The blade segment can now be considered in a state of stall when:

$$W_{RE} < \begin{cases} W_{RE\ limit} & D_{eq} \leq 2.2 \\ (2.2/D_{eq})^{0.6} \times W_{RE\ limit} & D_{eq} > 2.2 \end{cases} \quad (56)$$

The process for implementing this criterion in OTAC is as follows:

1. At a number of radii along each blade row, the equivalent relative velocity ratio, Equation (16), is calculated from simulation data. These radii correspond to the blade segments in OTAC.
2. The equivalent relative velocity ratio limit, Equation (17), is also calculated at the same radii.
3. The stall criterion, Equation (19), is applied. If the equivalent relative velocity ratio exceeds its limit, stall is “flagged” at that radius on the blade row.
4. If more than a certain percentage of a blade row are flagged as meeting the stall criterion, the entire blade row is considered stalled. Following this, if more than a certain percentage of blade rows are considered stalled, the compressor is presumed to be surged. OR
5. If more than a certain percentage of the calculation stations through the compressor are flagged as meeting the stall criterion, the compressor is presumed to be surged.

For the purposes of this study, it was found that surge could be indicated when either 40 percent of one blade row or 10 percent of the machine was flagged as stalled.

Aungier also made a disclaimer to this stall criterion. At speeds lower than about 85 percent of design, flow profile predictions become questionable and this equivalent relative velocity ratio limit becomes unreliable. Therefore, this stall criterion is applied only to speeds greater than 85 percent of the design speed.

Experiment: NACA Five-Stage Transonic Compressor

Hypothesis 2: Streamline design enhanced by Aungier's empirical profile, endwall, shock and boundary layer loss models can be tuned to accurately predict correct trends of a compressor map.

In order to demonstrate loss prediction models, a well-documented compressor is again needed that provides complete inter-stage flow data as well as annulus and blade dimensions. The NACA five-stage transonic axial compressor again meets these criteria, for which the design parameters and measured performance data are found in a series of NACA reports [19][20][21][22][23].

For this study, the documented design values of the NACA compressor is modeled and simulated, which allows for comparison with the measured performance values. This input should reproduce the design intent velocity triangles at the design point, within the assumption of simple radial equilibrium. NACA's design intent incidence and deviations angles are also input in order to reproduce the NACA design blade metal angles.

This process will again use the tools provided in OTAC to enable a system of equations ready for this analysis. The reader is directed to texts by Scott Jones at NASA Glenn or Russell Denney at Georgia Tech for details on this code[29][27].

The resulting model matched the NACA design point performance, as shown in Table 11 below. The table summarizes the results of the five-stage compressor design point in terms of pressure ratio and adiabatic efficiency for the OTAC model run in both single-streamline (meanline) mode and with seven streamlines. The efficiency for the OTAC runs matched the estimated efficiency of the NACA compressor. There is a slight underestimation of pressure ratio in both cases. Table 12 compares the total pressure rise per stage from the NACA report and from the single-stream OTAC model. The match between the two is good with small discrepancies at the first two stages driving the overall lower pressure rise.

Table 11: NACA Compressor Design Point Overall Results

	NACA Compressor	Simulated Compressor Meanline	Simulated Compressor 7-stream
Pressure Ratio	5	4.92	4.87
Efficiency	85	85.7	85

Table 12: Design Point Stage Total Pressure Ratios

	NACA Compressor	Simulated Compressor
Stage 1	1.39	1.36
Stage 2	1.48	1.46
Stage 3	1.43	1.43
Stage 4	1.35	1.35
Stage 5	1.27	1.28
Overall PR	5.01	4.92

Losses

The loss models described are implemented in the model through means of sockets. This allows the calculations to be applied to each stream of each blade row individually. The radial distribution of loss at the design point for rotor one (R1) is presented in Figure 7. This figure shows that the profile loss varies in a nearly parabolic manner from hub to tip, reaching a maximum value near mid-span. The endwall loss is only a small contribution to the total loss at the tip. The shock loss takes effect ear mid-span where the blade relative Mach numbers become supersonic.

It may be appreciated that the loss seen by the model will vary depending upon the number of blade segments chose, since a single blade segment located at the meanline will not see the shock loss at all. Since the blade segments are distributed in equal area annuli, the best resolution is achieved when there is a sufficient number of segments

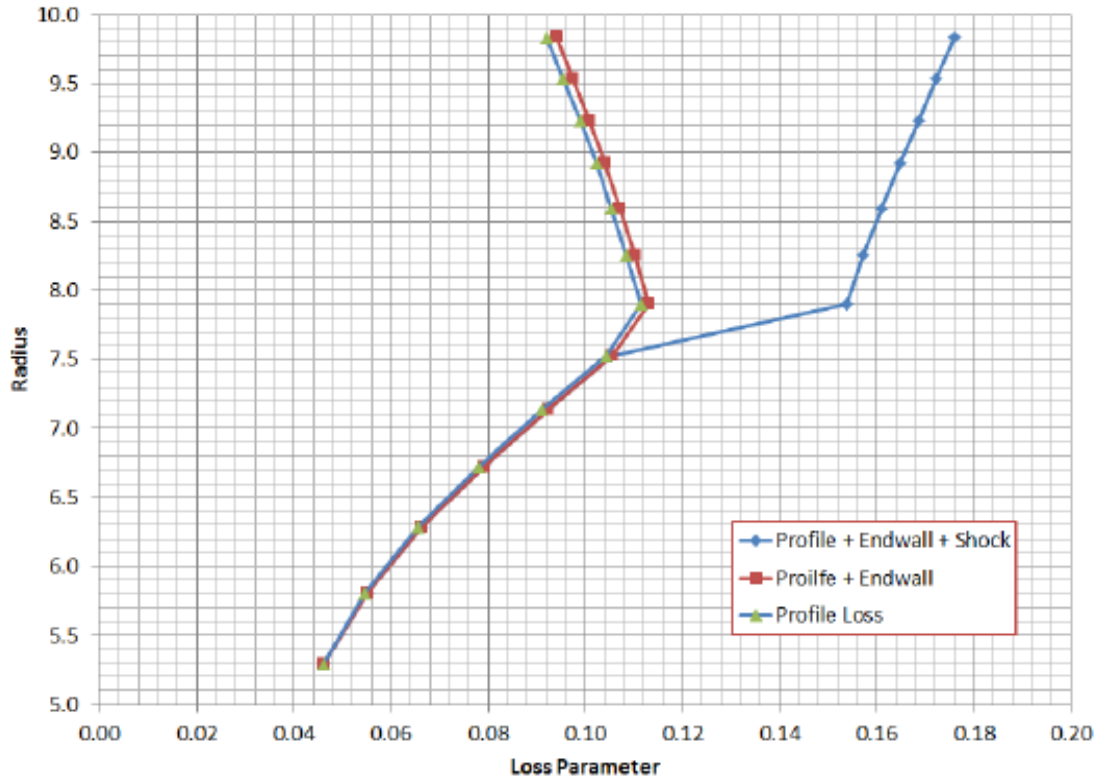


Figure 7: Radial Distribution of Loss, Rotor 1 Design Point [27]

such that the mean radius of the first and last segments are sufficiently close to the hub and tip, respectively. The required number of segments depends upon the blade row hub-to-tip radius ratio, and is most significant for the first rotor, which has the smallest hub-to-tip ratio or the longest blades. For the NACA five-stage compressor model, five to seven segments appeared to be sufficient for resolving the loss.

Off-Design Analysis

The objective of off-design analysis is to generate a complete compressor performance map. This is accomplished by running the OTAC model between the limits of stall and choke over a range of rotor speeds.

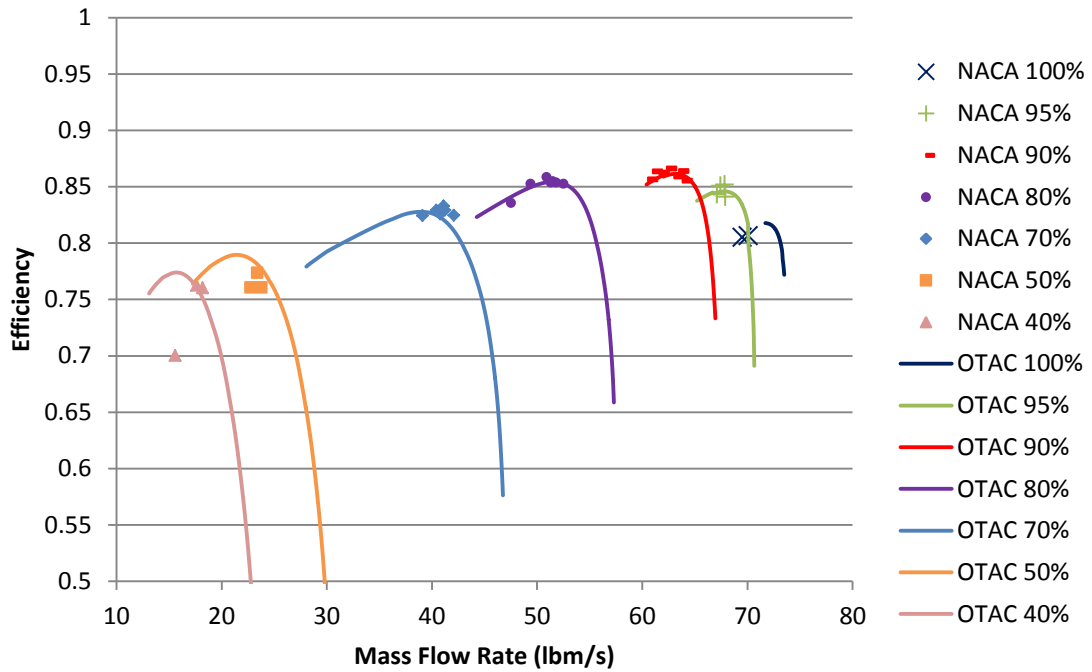


Figure 9: Compressor Map: Flow vs. Efficiency

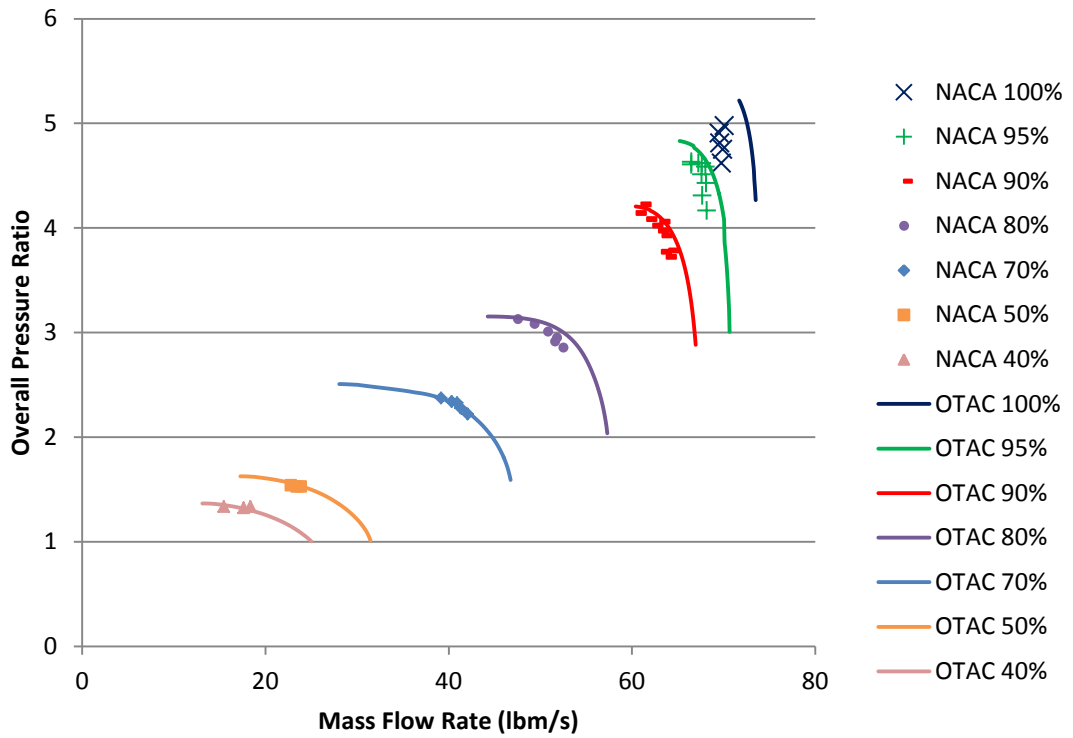


Figure 8: Compressor Map: Flow vs. Pressure Ratio

Off-design points may be run “manually” by changing the rotor speed and mass flow. However, it is desirable to automate this process and step through a range of speeds and mass flow rates within a series of calculation loops. The chief difficulty in doing this is determining the appropriate mass flows to set.

The model was defined to run at five streamlines to most efficiently match the measured data. The resulting overall compressor pressure ratio and efficiency maps are presented in Figure 9 and Figure 8. Also included in these figures are the measured data from the NACA rig test. It can be seen that the simulated compressor map matches very well to the measured data.

Stall Prediction

Because multiple simulated flow values are required to indicate stall by Aungier’s 1st stall criterion, only the 3rd criterion can be analyzed at the design point. The radial distributions of W_{RE} and as its limit at the design point for rotor one (R1) are presented

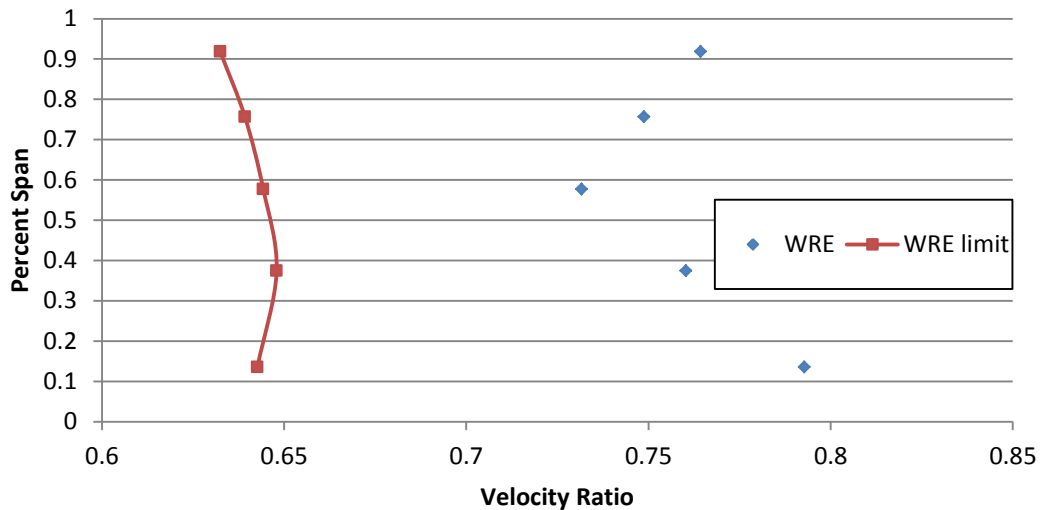


Figure 10: Equivalent Relative Velocity Ratio Across Rotor 1

in Figure 10. This graph shows that the relative velocity ratio is lower at mid-span than at either the tip or the hub, or that the static-to-stagnation pressure difference is greater at the hub and tip exits.

In order to compare with Aungier’s results, Figure 11 displays the equivalent relative velocity ratio and its limit at the hub of each blade row at the design point. This data conforms to Aungier’s results very well.

It should be noted that the NACA compressor was not run to surge at all speeds. The maximum pressure ratio obtained was limited by turbine inlet temperature as opposed to stall. Audible compressor surge was encountered only at speeds of 50, 70, and 80 percent of design [23].

A tuning factor was needed to evaluate the stall line, as a result of no equivalent velocity ratio violating Aungier’s third criterion before the pressure ratio gradient of the speed lines changed sign. This tuning factor will be discussed later.

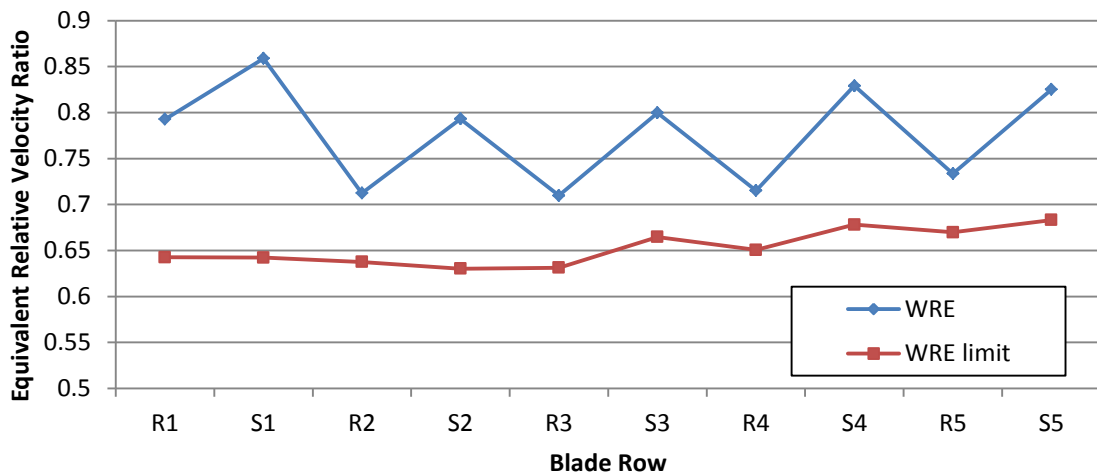


Figure 11: Equivalent Relative Velocity Ratio Along the Hub Contour, Design Point

Stall Criterion 1: Pressure Versus Flow Characteristic

This criterion is a simple and effective limit to the speed lines. The results, however, allow the compressor much lower mass flow rates than expected. The common trend of a surge line on a compressor pressure ratio map is to have an increasing slope with respect to mass flow rate, which is not apparent in this simulation. This trend is especially breached at and below the 70 percent speed line. The measured data at this shaft speed do not indicate a pressure ratio gradient close to zero; however, audible surge was documented. It is not known how close the measurements were to the audible surge condition, but the common trend of increasing slope is seen in the data. These results suggest that another stall criterion is necessary at lower speeds.

Stall Criterion 3: Equivalent Relative Velocity Ratio

The equivalent relative velocity ratio was not predicted as simply as Aungier described it. Because the limiting value is dependent on a regressed fit to simple two-dimensional diffuser data, a tuning factor, designated as k_{stall} , was introduced to allow the criterion to be met. The value of this tuning factor increases $W_{RE\ limit}$ by 5% in order for stall to be indicated at the lowest measured values of mass flow rate at the highest speed lines, as seen in Figure 12.

A detailed analysis of the equivalent relative velocity ratio criterion in a seven streamline simulation indicates that stall is flagged in the rear stages at high speeds and moves towards the front of the compressor as speed decreases. Table 13 presents the location of the compressor where stall is indicated by this criterion for 10% of the machine. The value after the decimal point is the blade segment, where 1 is at the hub and 7 is at the tip. It can be seen that stall is indicated at the rear stages at high speeds and moves forward with decreasing speed as expected. Stall occurs first in the hub region, and rotor 2 appears to be the most limiting blade row. It can be seen that stall over more than 40 percent of one blade row occurs at the 100, 95, and 90 percent speeds, and the remaining speeds indicate stall over ten percent of the machine before any one blade row stalls. It is not known, however, whether these predictions are consistent with the behavior of the physical compressor.

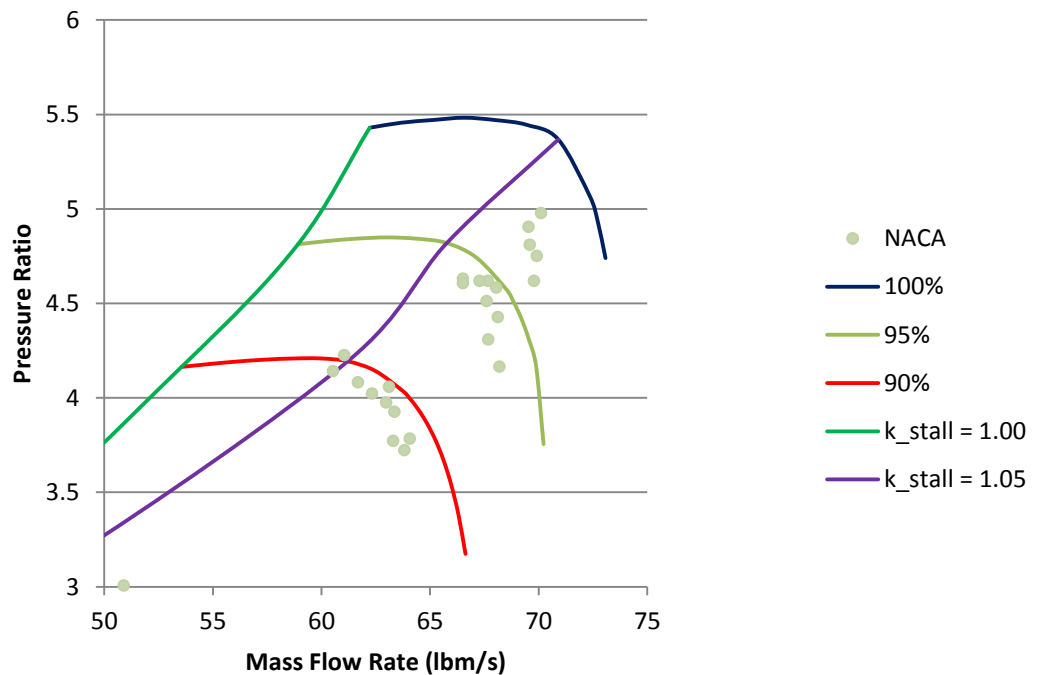


Figure 12: Variation of Tuning Parameter for Best Performance

Table 13: Location of Compressor Stall Indication

Speed	Stalled Blade Segments						
100%	R2.2	R3.1	R4.1	R4.2	R5.1	R5.2	R5.3
95%	R2.1	R2.2	R2.3	R2.4	R3.1	R4.1	R5.1
90%	R1.7	R2.1	R2.2	R2.3	R3.1	R4.1	R5.1
80%	R1.7	S1.1	R2.1	S2.1	R3.1	R4.1	R5.1
70%	R1.1	S1.1	R2.1	S2.1	R3.1	R4.1	R5.1
60%	R1.1	S1.1	R2.1	S2.1	R3.1	R4.1	R5.1
50%	R1.2	S1.1	R2.1	S2.1	R3.1	R4.1	R5.1

Transition Between Criteria

A few additional speed lines were included in the OTAC simulation in order to recognize stall trends with higher fidelity. Two speed lines were added at 85 percent and 88 percent of design speed to test Aungier’s claim that the third criterion is only applicable above 85 percent. As seen in Figure 13, this transition between stall criteria is actually very smooth with no steps in the surge line, and the third criterion does in fact become superseded by the first criterion below this speed.

Also seen in this figure is that the relative velocity ratio criterion supersedes the pressure ratio gradient again at and below the 70 percent speed line. Although it is

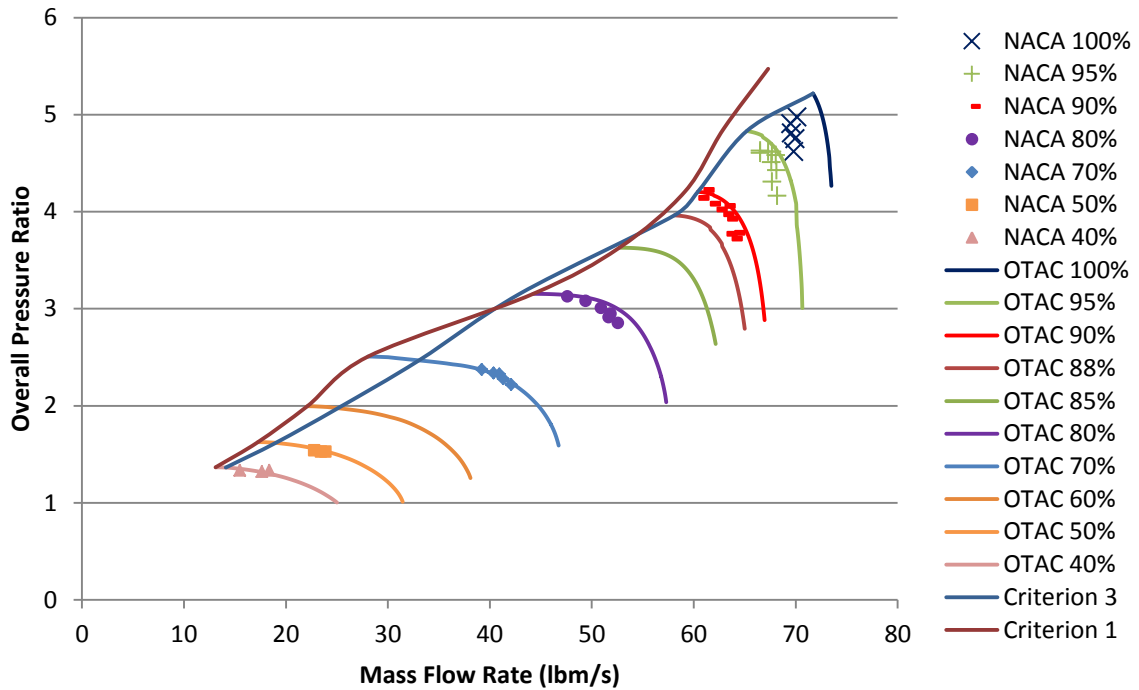


Figure 13: Stall Limits as Predicted by Criteria 1 and 3

closer to the measured data than the pressure ratio gradient, this criterion still does not indicate stall at an accurate enough flow rate to be useful at low speed.

Observations

The goal of this experiment was to demonstrate that Aungier's empirical profile, endwall, and shock loss models and stall prediction methods can be tuned to accurately predict correct trends of a compressor map. It was shown that the calculated losses do follow expected trends at the blade level and are then distributed throughout a map. No loss model calibration was deemed necessary beyond definition of certain parameters within the guidelines provided by Aungier. The effect of increasing the number of streamlines was observed generally to improve the accuracy of the loss prediction.

Of the three stall prediction criteria proposed by Aungier, the two stall criteria applied in this study are shown to be successful with certain limitations.

The equivalent relative velocity ratio criterion was very successful at indicating stall near the measured limits of the NACA rig tests at high speeds. This criterion did require some calibration for use, and a process for identifying and adjusting tuning factors was demonstrated. It was observed that the hub regions indicated stall first, and that indications of stall moved from the rear stages to the front stages with decreasing shaft speed as expected. Furthermore, it was observed that this criterion applies most credibly above 85 percent of design speed.

The pressure ratio gradient criterion is an obvious limit to compressor operation. This criterion, however, is inadequate at low speeds and another stall criterion must be found for compressor design use. The boundary layer separation criterion is expected to become effective at these low speeds and may be the solution to this dilemma.

While more complete means of predicting compressor performance can be found and modeled, this preliminary investigation has demonstrated that the method developed is suitable for analysis of conceptual designs. These models will be applied to the conceptual compressor design geometry to complete the compressor conceptual design and analysis procedure.

TASK 3: OPIMIZATION

Decision making is a very involved subject, especially when concerned with a complex system such as an axial compressor. There are many factors to consider, such as engine-level effects, weight, size, cost, and lifespan, naming only a few.

The four main components of decision are alternatives, criteria, preferences, and models [32]. Alternatives are the combinations of parameter values that lead to unique compressor designs. Criteria are the factors of considerations, such as design point and off-design performance, which are used in cycle design. Models are what link the criteria to the alternatives, in our case the design and analysis procedure. Preferences must come from the application for which the design is intended. While all applications will desire peak performance, there are many applications for a compressor within a gas turbine system that will be governed by various preferences. For example, a military application may have preferences of more robust performance and constraints based on size and weight while regarding cost as a lower priority. On the other hand, an industrial application will prefer higher efficiencies at lower cost, without as much regard for size or weight.

Design Space

The design and analysis process developed here greatly expands the capability of the designer to make informed decisions. The design space of alternatives for a specific combination of cycle parameters of pressure ratio, efficiency, mass flow rate and shaft

speed can be relatively quickly populated and explored. The feasible alternatives in the design space can be identified and evaluated against the preferences of the application.

Beside the cycle performance parameters of pressure ratio, efficiency, speed, and mass flow rate, ranges of each of the parameters have been published in literature or otherwise identified [24][23].

- Overall Compressor Inputs
 - Stage Count
 - Constant Radius = Tip, Mean, or Hub
 - Inlet Axial Mach number = [0.4 – 0.6]
 - Inlet hub-to-tip radius ratio = [0.4 – 0.7]
 - First Stage Loading Factor = [-0.25 – 0.05]
 - Inlet-to-discharge velocity ratio = [0.8 – 1.1]
 - Constant thickness-to-chord ratio through machine = [0.05 – 0.10]
 - Constant tip clearance through machine = [0.015 – 0.030]

- Inputs for Each Blade Row
 - Aspect Ratio = [0.6 – 4.0]
 - Solidity = [0.85 – 2.00]

For a five stage machine, there will be 28 parametric inputs, 8 overall inputs and 2×10 blade row inputs. For a six stage machine, however, there will be 32 parametric inputs, 8 overall inputs and 2×12 blade row inputs. Applying a design of experiments to these

parameters and inputting each combination into the compressor conceptual design process can quickly populate the design space.

It is noted that pressure ratio and efficiency are included both as design parameters as

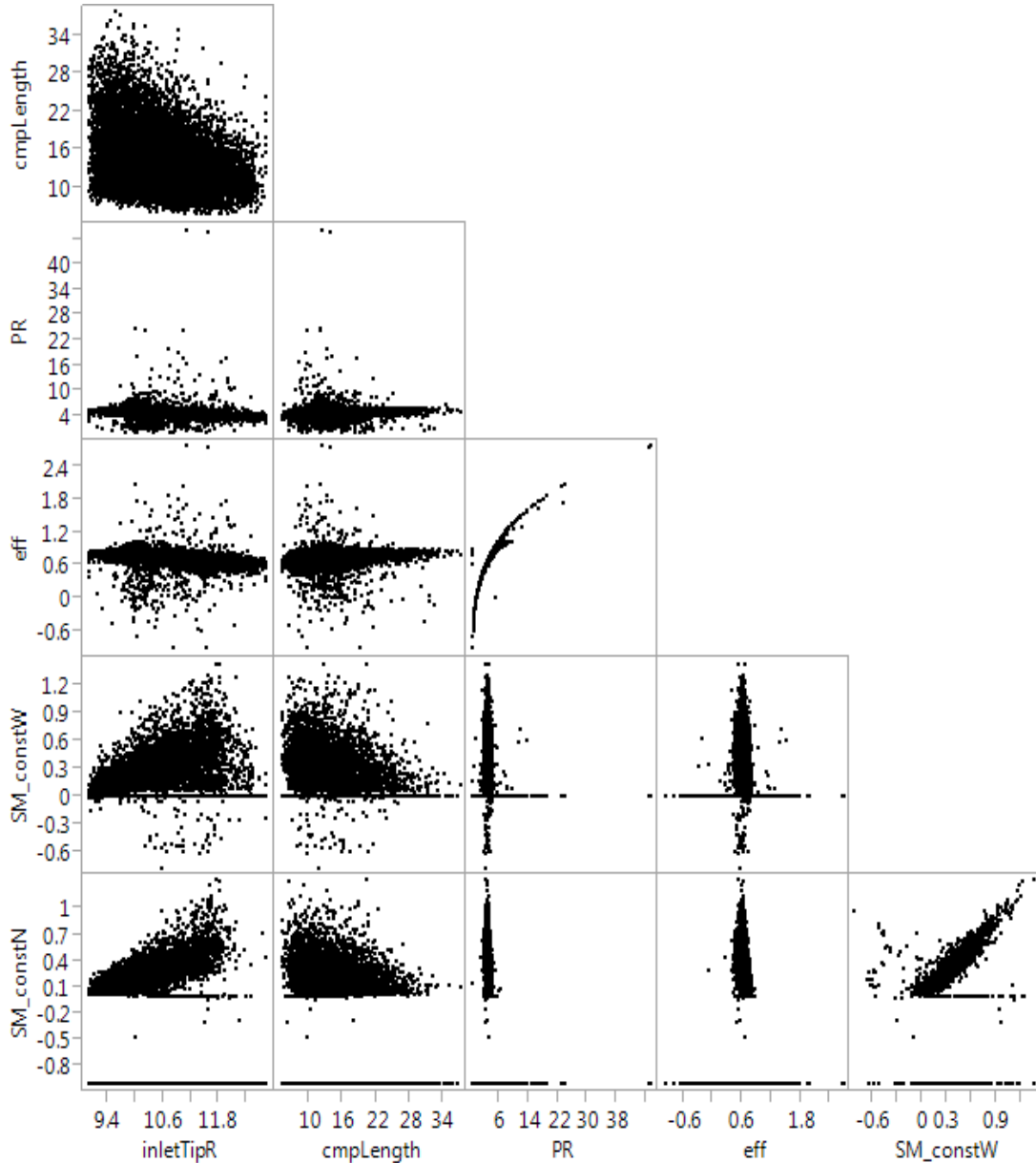


Figure 14: Five Stage Compressor Design Space

$$\begin{aligned}
 PR_{des} &= 5, \\
 \eta_{des} &= 85\%, \\
 \dot{m}_{des} &= 67.5 \text{ lbm/s}, \\
 N_c &= 12605 \text{ RPM}
 \end{aligned}$$

well as responses, due to the design parameters determining the overall temperature ratio and stage velocity triangles, following which the loss models determine the responses. Because discussion of weight, cost and other engine-level responses is beyond the scope of this research, the set of responses to be included in this discussion are the performance parameters – pressure ratio, efficiency, constant speed stall margin and constant flow stall margin – and size parameters of length and maximum tip diameter. Other responses can be included such as chord lengths to determine constraint activity.

Based on the previous discussions of a compressor design to increase the pressure of 67.5 lbm/s by 5x at an efficiency of 85% using a 5 stage machine, a design space is initially populated by over 12,000 configurations using a Latin-hypercube (space filling) design and endpoints. The responses of this design space are shown in Figure 14. The application for this compressor would be for aviation engines, so a constant tip design was chosen.

Whenever a configuration NPSS solver did not converge, the responses will show that constant speed stall margin is zero as well as the efficiency is either greater than one or less than zero. Over 8,000 of the 12,000 cases resulted in an unstable solver, which will be designated as infeasible results. When these infeasible results are filtered, the design space constricts and the solution space is much more reasonable. It is seen in Figure 15 that the response efficiency will fall within the range of zero to one, and the majority of the stall margin values will fall within the typical range of 15-25%.

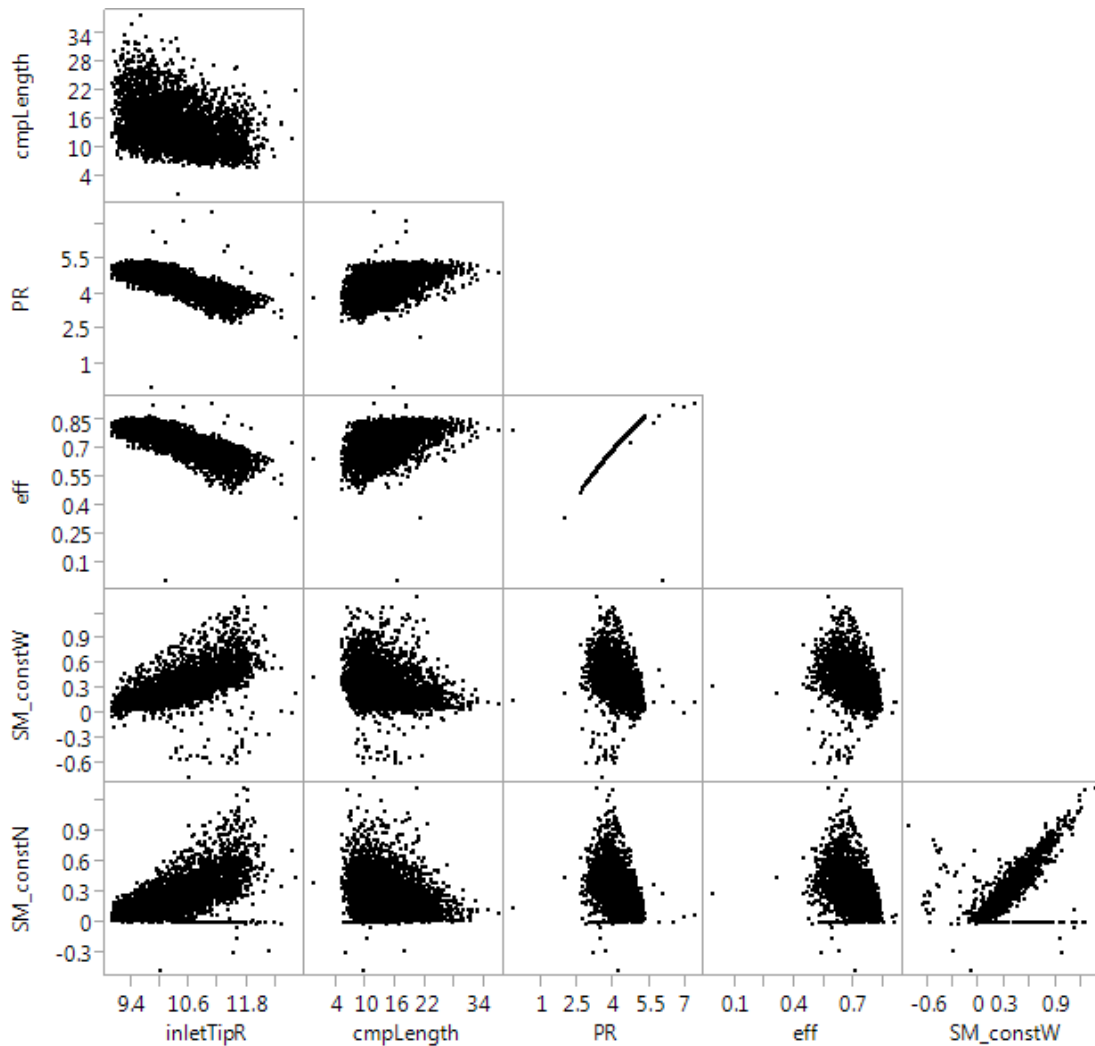


Figure 15: Five Stage Compressor Feasible Space

Although the infeasible configurations have been excluded, some configurations are still seen with a constant speed stall margin of less than one or a constant flow stall margin of less than zero. A constant speed stall margin of less than zero is explained by the pressure ratio decreasing as flow rate is decreased, meaning the apex of the speed line is at or to the right of the design point. A constant flow stall margin of less than zero is explained by the pressure ratio decreasing as speed is increased, meaning the apex of the increased speed line is to the right of the design point and the increased speed line has dipped below the design pressure ratio. These configurations can be found as

feasible by Aungier’s third stall criterion at the design point before being found as unstable as stall margin is calculated.

Because the ranges of response efficiency and pressure ratio extend below the desired results, the design space can be further narrowed to designs with pressure ratio above 5 and efficiency above 85%, as shown in Figure 16. Although the length of the NACA five-stage compressor previously discussed is not documented, the majority of the designs

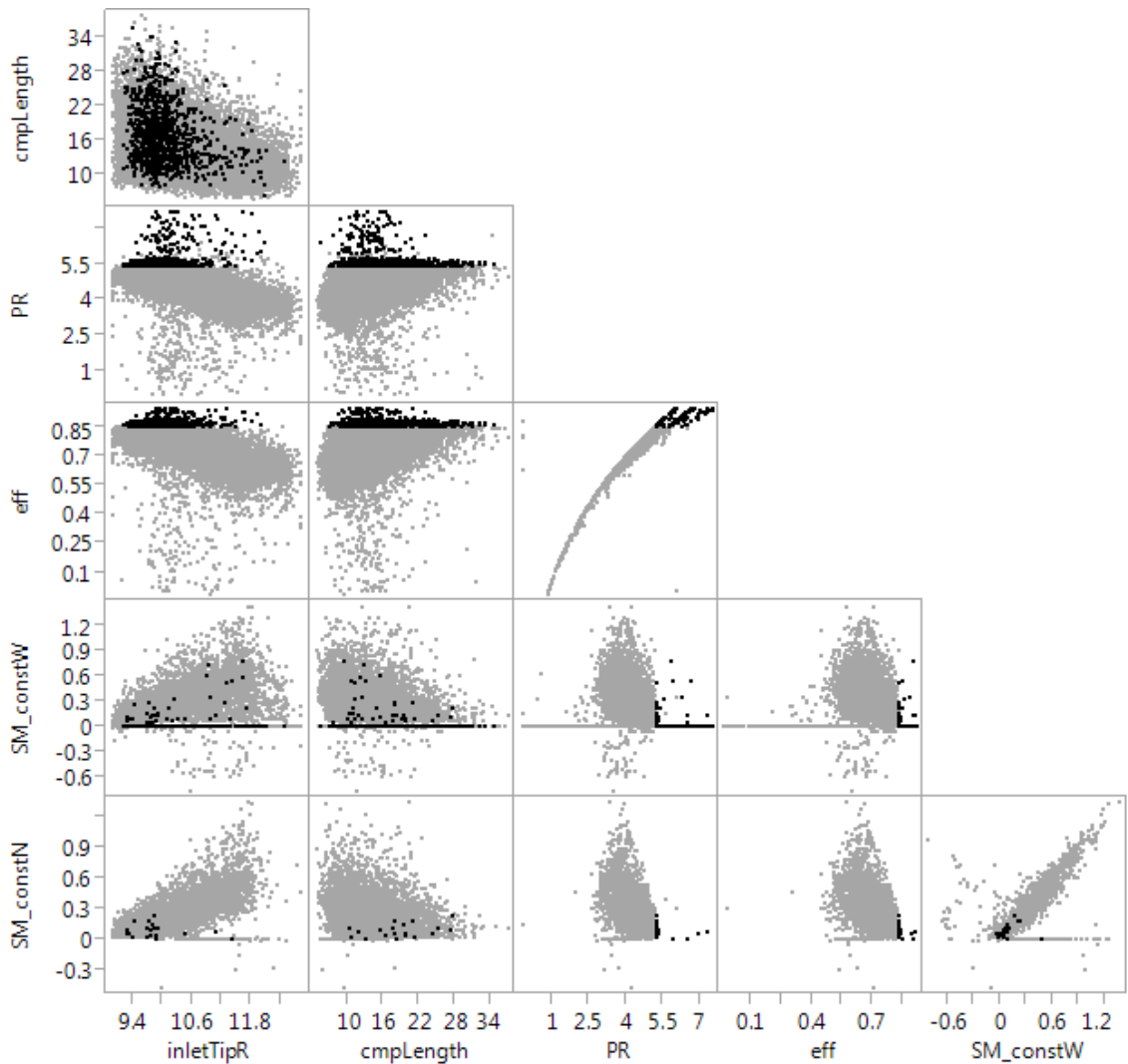


Figure 16: Feasible Space reduced by Design Constraints

$\eta > 85\%$
 $PR > 5.0$

shown here fall within a 10 inch maximum radius constraint or standard set by the NACA compressor. It is also seen that efficiency increases with decreasing tip radius.

When the overall compressor parameters for these designs are investigated in Figure 17, it can be seen that both stall margins decrease with inlet hub-to-tip radius ratio as well as inlet Mach number, identifying a lower bound for inlet Mach number at about 0.5 and a lower bound for radius ratio at about 0.4. Although the set of feasible

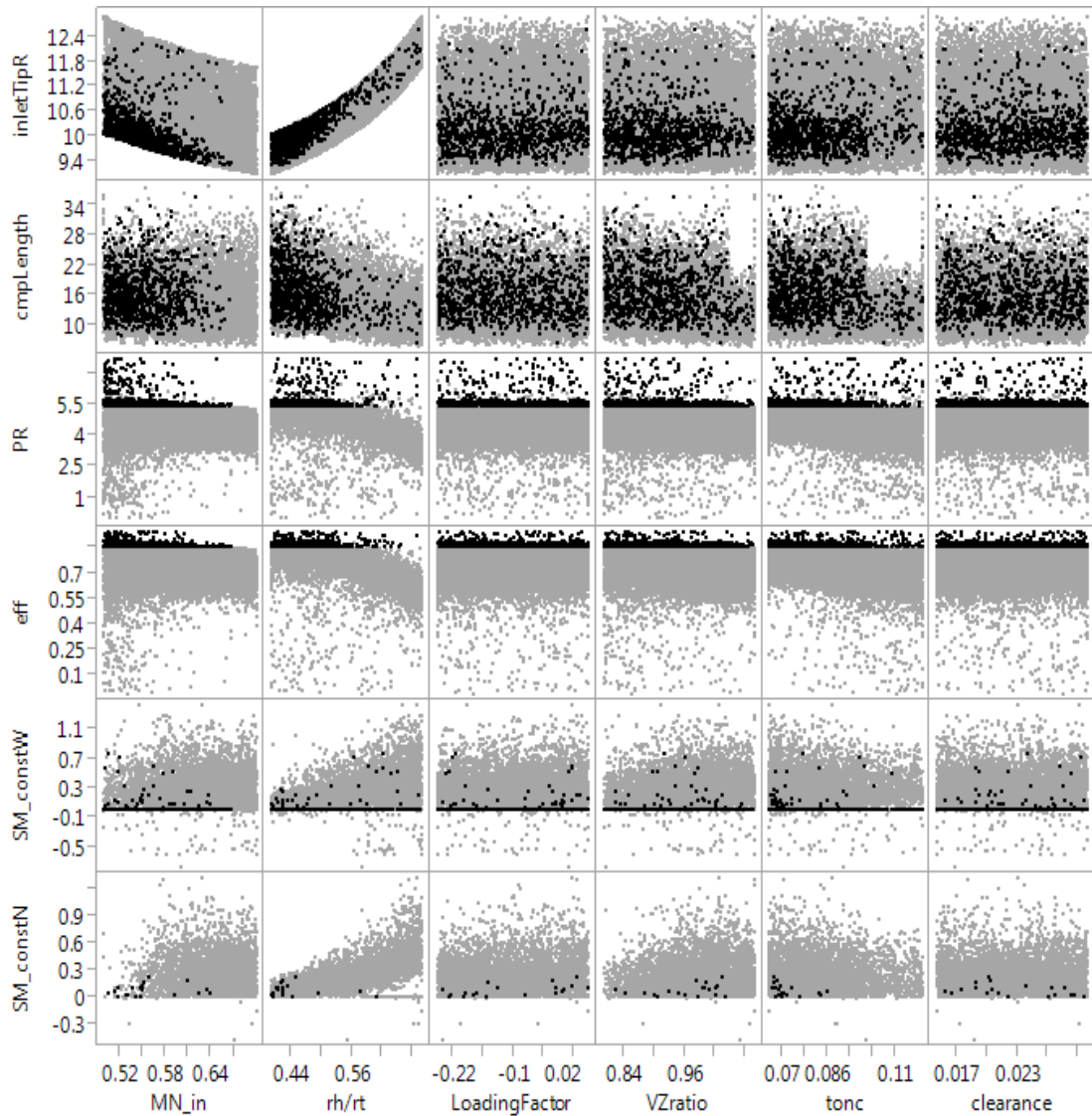


Figure 17: Design Space Overall Parameters with Feasible Space Highlighted

configurations includes a rather even spread of most of these parameters within their respective ranges, the range of thickness-to-chord ratio (tonc) is seen to converge to the lower values, between 0.065 and 0.086.

Figure 18 and Figure 19 show that the design space responses do not limit the design space based on aspect ratio or solidity of any blade row.

Optimization

Decision making strategies can be numerical in order to find the optimal result. Many numerical optimizers have been demonstrated for similar complex systems, each of which has its strengths and weaknesses.

The responses of pressure ratio, efficiency, size and stall margin are comparable between the five stage and the six stage machines, although optimization must be

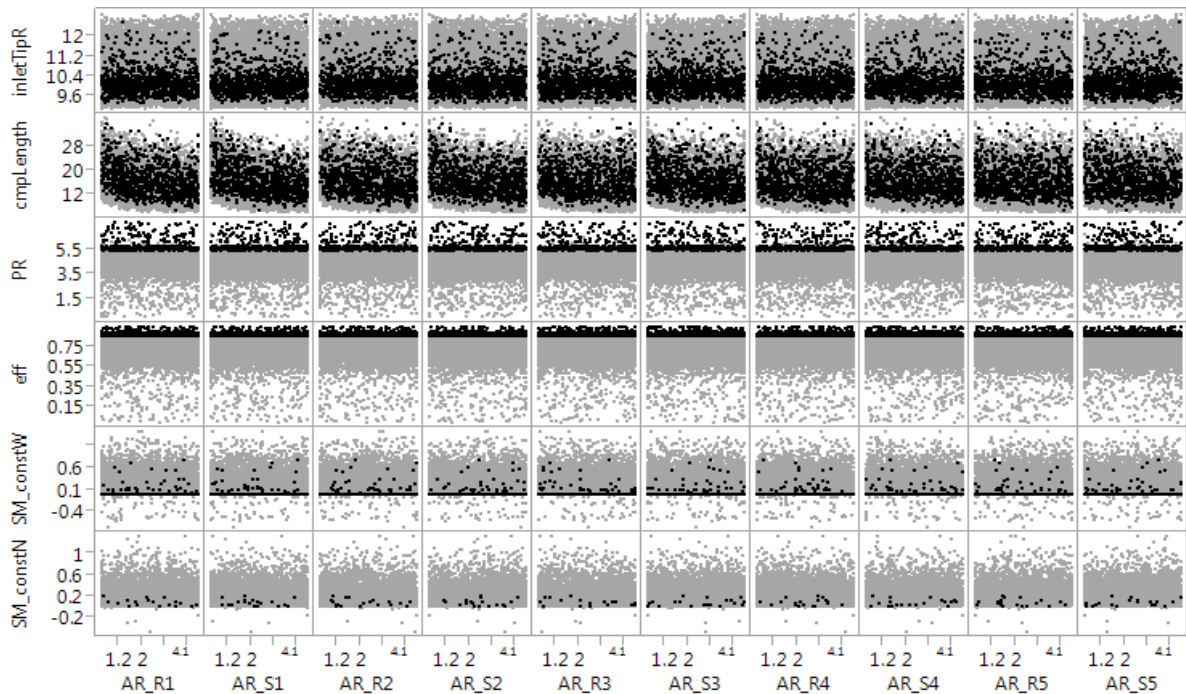


Figure 18: Design Space Aspect Ratio Combinations with Feasible Space Highlighted

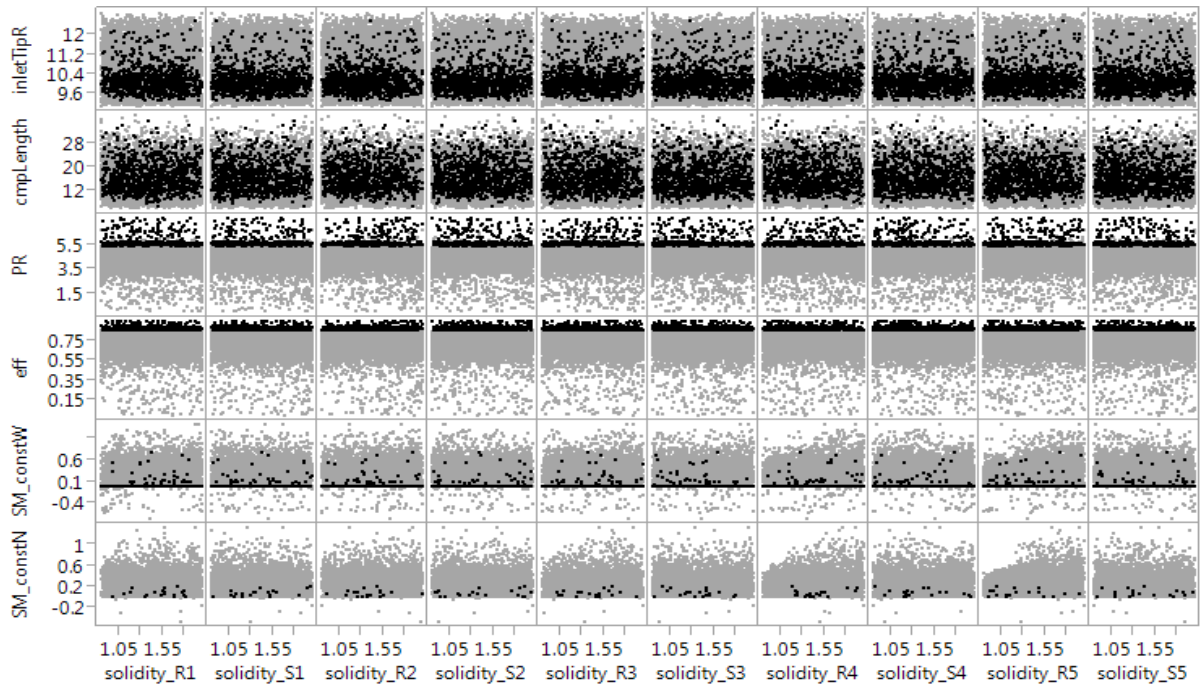


Figure 19: Design Space Solidity Combinations with Feasible Space Highlighted

performed for the five stage configurations and the six stage configurations separately due to the varying number of parameters.

Given a specific application and its requirements, an optimizer can be used to find the compressor design of optimal performance. In order to develop an algorithm to find the optimal compressor design, the parameters to be optimized and the variables to be perturbed must be organized into an optimization objective statement. These parameters and variables are those discussed above that allow the compressor model to become parametric.

Objective Function

- Maximize Design Point Constant Speed Stall Margin
- Given:
 - Design Pressure Ratio
 - Design Efficiency
 - Design Corrected Mass Flow Rate
 - Design Corrected Shaft Speed
- With respect to:
 - Overall Compressor Inputs
 - Stage Count
 - Constant Radius
 - Inlet Axial Mach number
 - Inlet hub-to-tip radius ratio
 - First Stage Loading Factor
 - Inlet-to-discharge velocity ratio
 - Constant thickness-to-chord ratio through machine
 - Constant tip clearance through machine
 - Inputs for Each Blade Row
 - Aspect Ratio
 - Solidity
- Subject to:
 - Compressor Constraints:
 - Minimum Pressure Ratio of Design Requirement
 - Minimum Efficiency of Design Requirement
 - Maximum Overall Diameter
 - Maximum Discharge Hub-to-Tip Radius Ratio
 - Blade Row Constraints
 - Radius ratio_n > Radius Ratio_{n-1}
 - Blade Count_n ≠ Blade Count_{n-1}
 - Minimum and Maximum Thickness-to-Chord Ratio
 - Minimum and Maximum Clearance
 - Minimum and Maximum Aspect Ratio
 - Minimum and Maximum Solidity

Although continuous non-dimensional variables were selected as parameters in order to apply documented ranges, the design space is seen to include many failed cases, indicating that a gradient-based optimizer would not be reasonable. Another option is to implement this objective function into a genetic algorithm or particle swarm optimizer in order to sample the design space en route to finding the configuration of optimal performance. However, due to the inconsistency of the design space seen in Figure 14, a particle swarm algorithm was not able to locate enough feasible points to be effective.

In order to still demonstrate the ability of this conceptual design method to be used in an optimizer, enough cases are run to create response surfaces and prediction profiles. JMP, a powerful data analysis and visualization tool, is used to efficiently fit functions to the response data. Desirability profiles can then be set to evaluate the prediction profiles in order to determine the optimal combination of parameters.

Two optimization objective functions are defined to demonstrate this ability. The first is to maximize constant speed stall margin given a minimum efficiency constraint. The second is to maximize efficiency with a minimum pressure ratio constraint. These optimizations are carried out on five- and six-stage configurations and their responses are given here.

Table 14: Predicted Responses for Maximum Stall Margin Optimization

Response	5 Stage Optimum	6 Stage Optimum
Pressure Ratio	5.46	5.12
Efficiency	85%	85%
Constant Speed Stall Margin	87.5%	168%
Maximum Tip Radius (in)	10.75	10.32

Based on this method, the maximum constant speed stall margin in the design space of five- and six-stage compressors can be up to 168%, which means the ratio of pressure ratio to mass flow at the surge line of the design speed is almost 2.7x that ratio at the design point. This would be possible on a speed line characterized by a low speed, where the pressure ratio will decrease gradually as mass flow is increased. The vector of parameter values for this six-stage configuration is given in Table 15 and

Table 16.

Table 15: Optimum Combination of Overall Parameters for Max Stall Margin

Parameter	6 Stage Optimum
Inlet Mach Number	0.530
Inlet Hub-to-Tip Radius Ratio	0.488
1st Stage Loading Factor	0.05
Inlet-to-Discharge Velocity Ratio	1.1
Thickness-to-Chord Ratio	0.065

Tip Clearance	0.015
----------------------	-------

Table 16: Optimum Combination of Blade Row Parameters for Max Stall Margin

Parameter	6 Stage Optimum					
Stage	1	2	3	4	5	6
Rotor Aspect Ratio	4	0.77	0.6	0.6	0.6	0.6
Stator Aspect Ratio	0.6	2.25	0.6	4	0.6	4
Rotor Solidity	1.86	2	1.45	1.77	1.99	1.8
Stator Solidity	1.13	1.38	2	2	2	0.85

It is seen that the optimizer will drive some of these parameters to the edge of their ranges, such as the thickness-to-chord ratio and blade tip clearance. Also of note is that the combinations or distributions of aspect ratio and solidity through the configuration are often driven to the edges of their ranges. Unfortunately, when this vector of input parameters is run through the macro geometry definition and performance analysis procedure, the result does not converge.

The second optimization demonstration is to maximize efficiency given a minimum pressure ratio constraint. The responses and parameters of this optimization are given in Table 17 through Table 19.

Table 17: Predicted Responses for Maximum Efficiency Optimization

Response	5 Stage Optimum	6 Stage Optimum
Pressure Ratio	5.19	5.19
Efficiency	90.6%	93.8%
Constant Speed Stall Margin	1%	7%
Maximum Tip Radius (in)	10.07	10.02

Table 18: Optimum Combination of Overall Parameters for Max Efficiency

Parameter	6 Stage Optimum
Inlet Mach Number	0.5
Inlet Hub-to-Tip Radius Ratio	0.4
1st Stage Loading Factor	-0.25
Inlet-to-Discharge Velocity Ratio	1.1
Thickness-to-Chord Ratio	0.065
Tip Clearance	0.015

Again, the optimizer will drive some of these parameters to the edge of their ranges, including many of the aspect ratio values. It is seen in this case that many of the solidity values are not driven to the edges of the range. Unfortunately, when this vector of input parameters is run through the macro geometry definition and performance analysis procedure, the result again does not converge.

Table 19: Optimum Combination of Blade Row Parameters for Max Efficiency

Parameter	6 Stage Optimum					
Stage	1	2	3	4	5	6
Rotor Aspect Ratio	4	0.6	4	0.75	2.59	4
Stator Aspect Ratio	0.6	0.6	0.6	0.6	4	0.6
Rotor Solidity	1.78	1.34	1.45	1.48	1.02	0.85
Stator Solidity	1.97	1.29	2	0.85	2	2

Observations

The design space of five-stage compressor configurations designed to achieve a pressure ratio of 5 at an efficiency of 85% is visualized. It is seen that this design space is not continuous and many configurations in the design space do not converge to a solution in the solver. It is demonstrated that a surrogate model of the valid area of the design space can be created and used as an optimization objective function. Unless more detailed constraints are identified, the optimizer will drive many of the parameters to the edges of their ranges, resulting in configurations that do not belong to the valid design space. The author believes a rule for the distributions of aspect ratio and solidity through the compressor is required to constrain a compressor conceptual design to a feasible solution space.

The figures in Appendix A are made available for five-stage configurations to characterize the design space and the effects of each parameter on other parameters. The blue highlighted points are those which do converge in the solver, and the black points are those which do not.

TASK 4: MAP GENERATION

The objective of this research is to generate a complete compressor performance map for use in cycle design. This can be done by running the OTAC model over a range of rotor speeds, and up and down each speed line within the limits of stall and choke. Off-design points may be run “manually” by changing the rotor speed and mass flow rate. However, it is desirable to automate this process and step through a range of speeds and mass flows within a series of calculation loops. The chief difficulty in doing this is determining the appropriate mass flows to set. It has been found most effective to define a starting flow for each speed line, near the min-loss or peak efficiency point, then to step downward in flow to move up the speed line until stall is encountered in

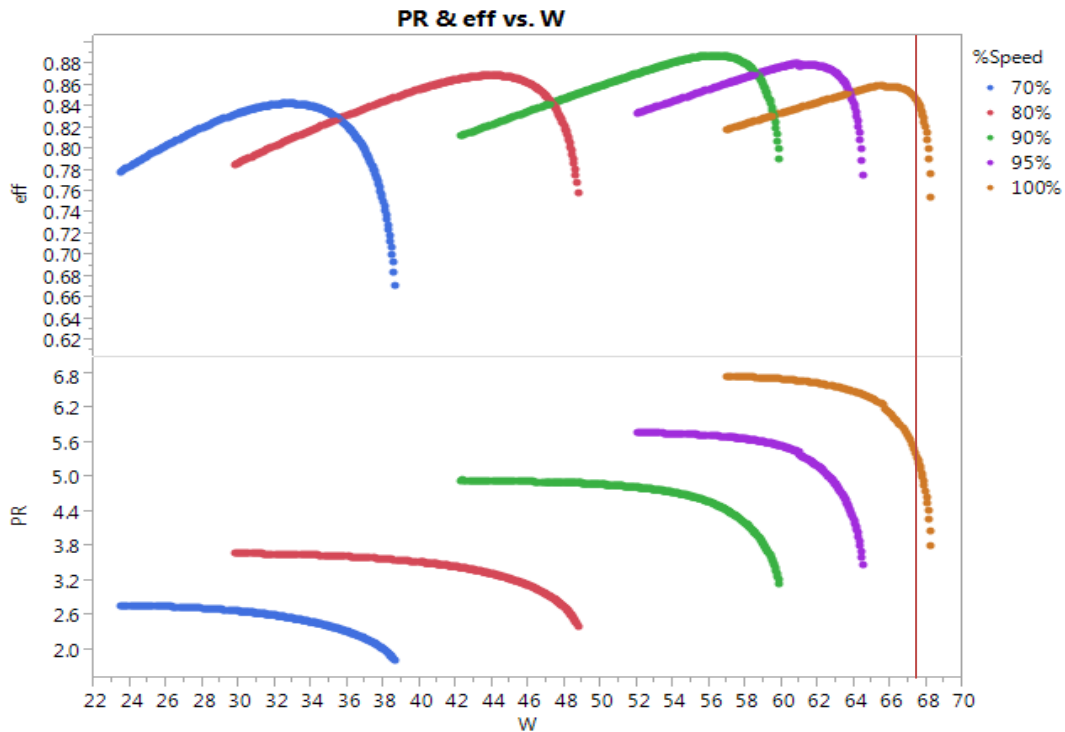


Figure 20: Compressor Performance Map for Optimal Six Stage Configuration

one or more blade rows, and then to return to the starting flow and step upward in flow to move down the speed line until choke is encountered in one or more blade rows. The first problem is to determine a suitable starting flow for each speed line. This can be accomplished by a trial and error process at each speed line. Generally it is found that if the mass flow is too large, then the solver fails to converge.

It can be seen that this compressor performs in an expected manner, with pressure ratio characteristic slope becoming steeper for increasing speed and the efficiency map showing the characteristic ridge sometimes known as the “backbone”. These maps are suitable for use in cycle design and off-design cycle analysis and by nature satisfy both cycle design requirements and compressor constraints.

In order to demonstrate this map generation procedure, a point in the design space is identified which will converge. This point is selected for its large stall margin at the design point. The point selected will achieve a pressure ratio of 5.34 at an efficiency of 84.5% and a constant speed stall margin of 46.7%. The performance map generated for this design is shown in Figure 20.

CONCLUSION

During the traditional cycle design process, existing compressor maps are often scaled in order to meet overall engine or cycle design point requirements. These scaling factors account for design changes or technology improvements, but also disconnect the compressor map from the physical flow characteristics of a compressor. The necessity of creating a connection to physical compressor flow at conceptual cycle design is recognized as a goal in gas turbine engine research.

A compressor conceptual design process is discussed which allows for low-fidelity compressor geometry to be defined by only $4 \times (\text{Number of Stages}) + 8$ parameters. The dimensionality of this process is reduced by the acceptance of six assumptions common to compressor conceptual design. The question was asked whether the resulting compressor macro geometry is consistent with expected trends. The assumptions made also allow for the resultant compressor geometry to follow expectant trends such as continuously reducing annulus area through the compressor. This process also allows for the blades at each blade row to be defined as double circular arc or NACA 65 series airfoils based on the relative Mach number at the blade tip. Therefore the first hypothesis is proven valid.

It has been shown that although the assumptions made can result in configurations consistent with known geometry trends, more detailed constraints are required to allow the resulting geometries to converge when analyzed for performance. The author believes a rule for the distributions of aspect ratio and solidity through the compressor

is required to constrain a compressor conceptual design to a feasible solution space. The landscape of the design space is given and it is seen that more obvious constraints can be defined for the relationships between Inlet-to-Discharge Velocity ratio, Inlet Mach number, and Thickness-to-Chord Ratio.

A multiple-meanline performance prediction method is also presented which is based on the assumption of simple radial equilibrium. This method employs empirical loss models to predict the effect of profile, endwall and shock losses of each blade row on the performance of the machine. Another question was asked whether empirical loss models will follow expected trends throughout a compressor map as well as at the blade row level. The loss models compiled by Aungier were selected for this study because they have been tested and proven. It was demonstrated that these loss models will predict correct trends at the blade row level as well as at the machine level, resulting in simulated compressor maps that reflect measured data. Therefore the second hypothesis was also proven valid.

The design space of five-stage configurations spanned by ranges of the parameters discussed is visualized and discontinuities are identified. Optimization strategies are discussed and a surrogate model approach for the objective function is selected due to the discontinuous design space. An optimization method is demonstrated for two applications using the surrogate model generated. Decision-making strategies are highly influenced by the application of the design, and it is left to the designer to decide and develop a more robust decision-making method. The author recommends a genetic

algorithm such as a particle-swarm method because of the discontinuous landscape of the design space.

Finally, this compressor conceptual design, analysis and decision-making process culminates in the demonstration of a compressor performance map generation that is coupled with conceptual design parameters. The performance map and the related design parameters satisfy cycle requirements as well as compressor constraints. This map generation is simple and straight forward, resulting in pressure ratio and efficiency maps that follow expected trends.

In short, a new compressor conceptual design, analysis, and map generation method is developed that can bridge the disconnection between cycle design and compressor design. This method incorporated identification of compressor design parameters and prediction of blade row ad overall performance with the objective to satisfy cycle requirements as well as compressor constraints at the cycle design point. Through development of this method and validation of the presented hypotheses, the objective of this research has been met. Therefore the overall thesis is also valid.

A compressor conceptual design, analysis and optimization method can couple design parameters to performance maps, allowing the cycle designer to choose a suitable map that is attached to reasonable compressor design parameters.

This research will provide valuable contributions to the field of aerospace engineering. Implementation of this method will allow for more informed decisions to be made sooner. This research demonstrates that the method developed is suitable for serious

conceptual design applications within an integrated multidisciplinary design environment.

APPENDIX A: FIVE-STAGE COMPRESSOR DESIGN SPACE LANDSCAPE

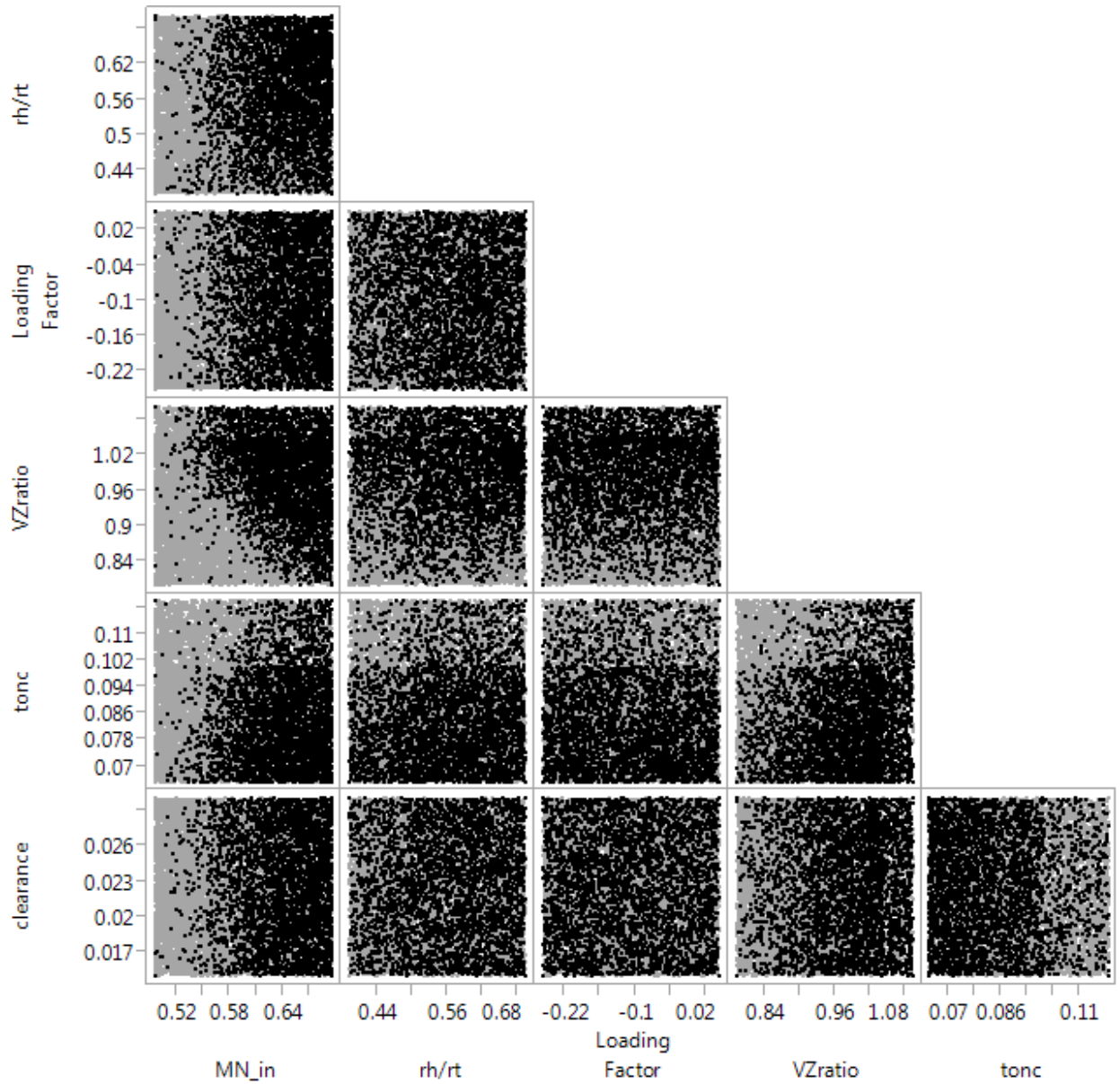


Figure 21: 5 Stage Design Space with Converged Points Highlighted

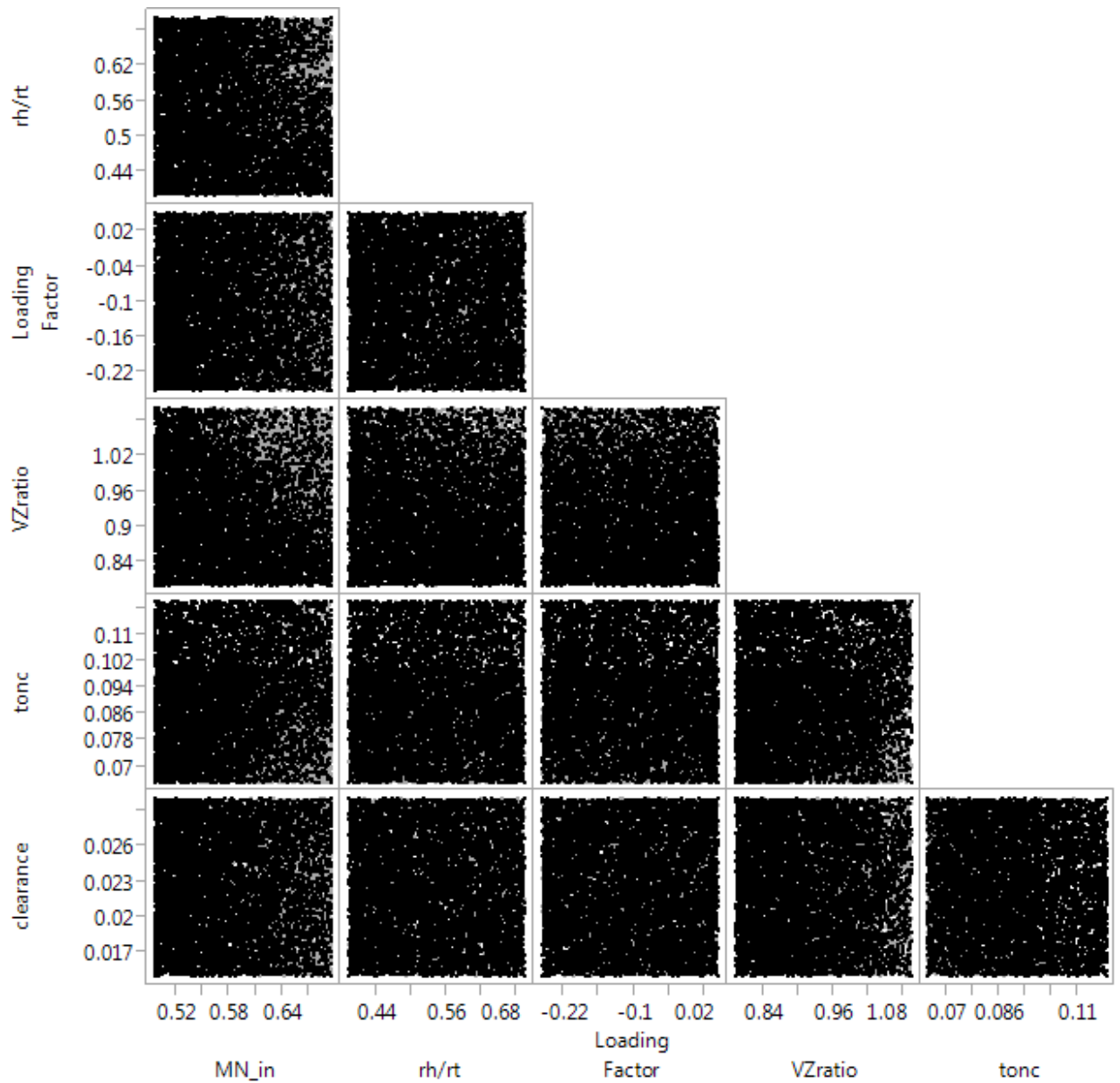


Figure 22: 5 Stage Design Space with Non-converged Points Highlighted

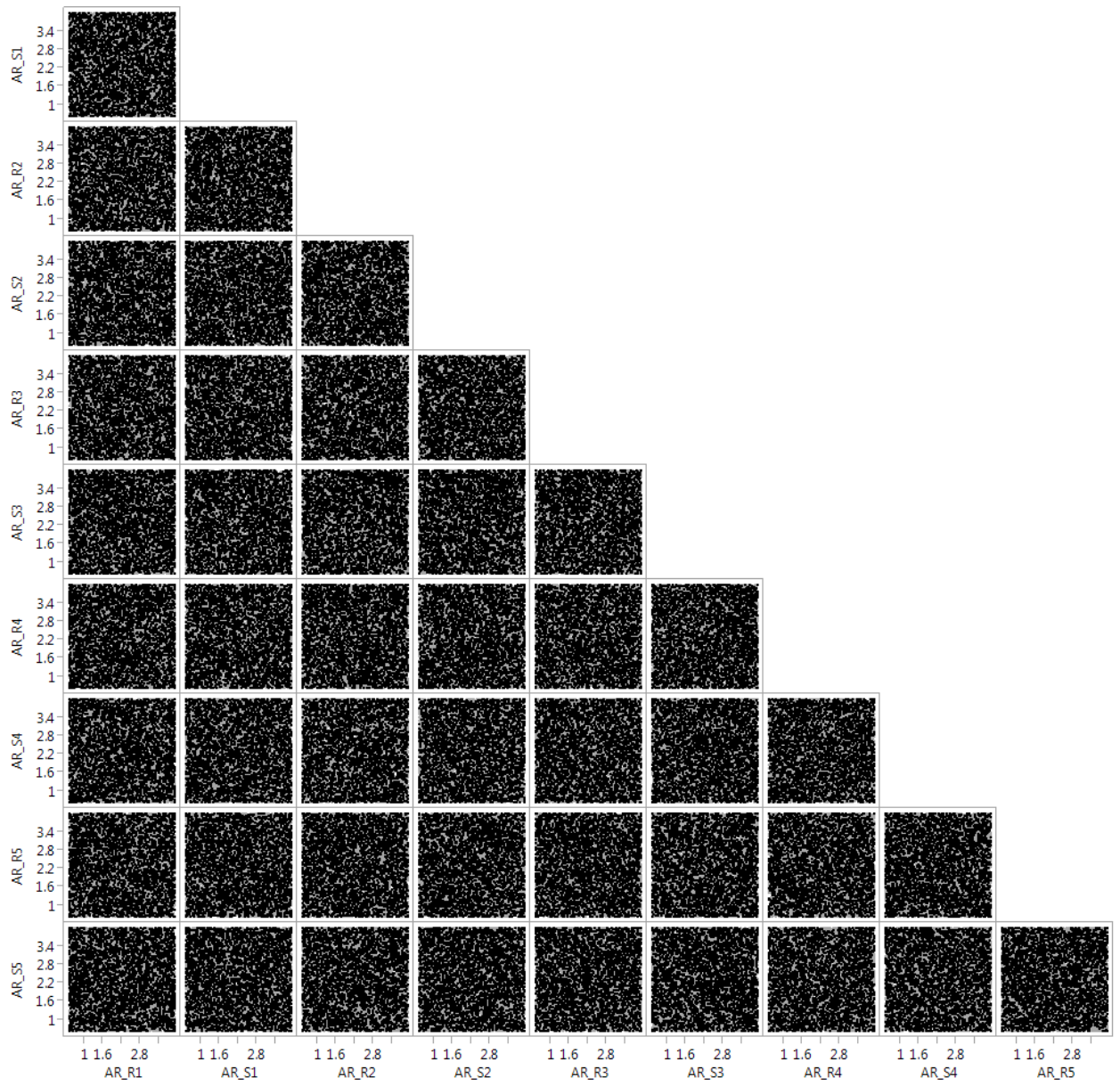


Figure 23: 5 Stage Design Space Aspect Ratios with Converged Points Highlighted

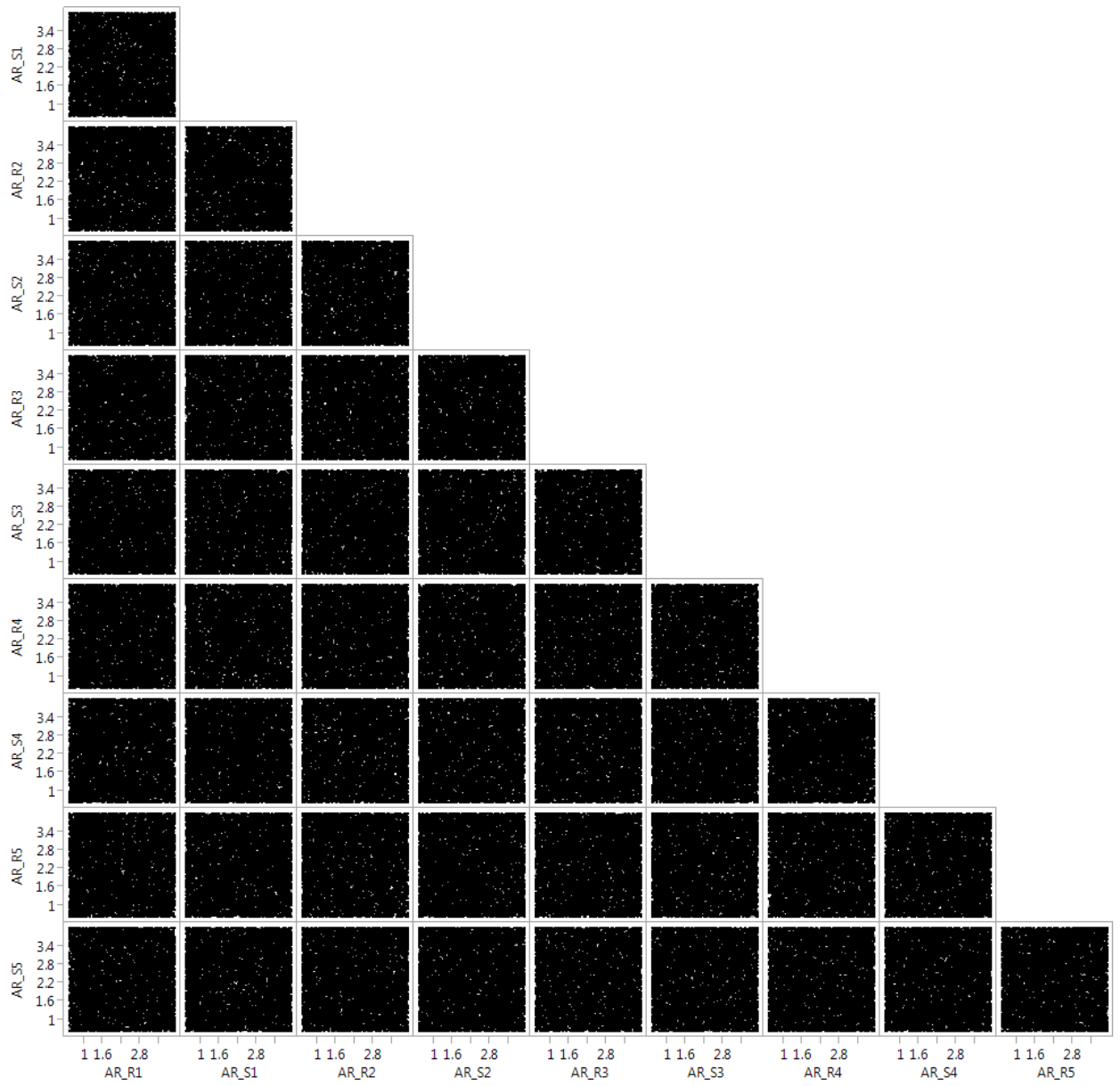


Figure 24: 5 Stage Design Space Aspect Ratios with Non-converged Points Highlighted

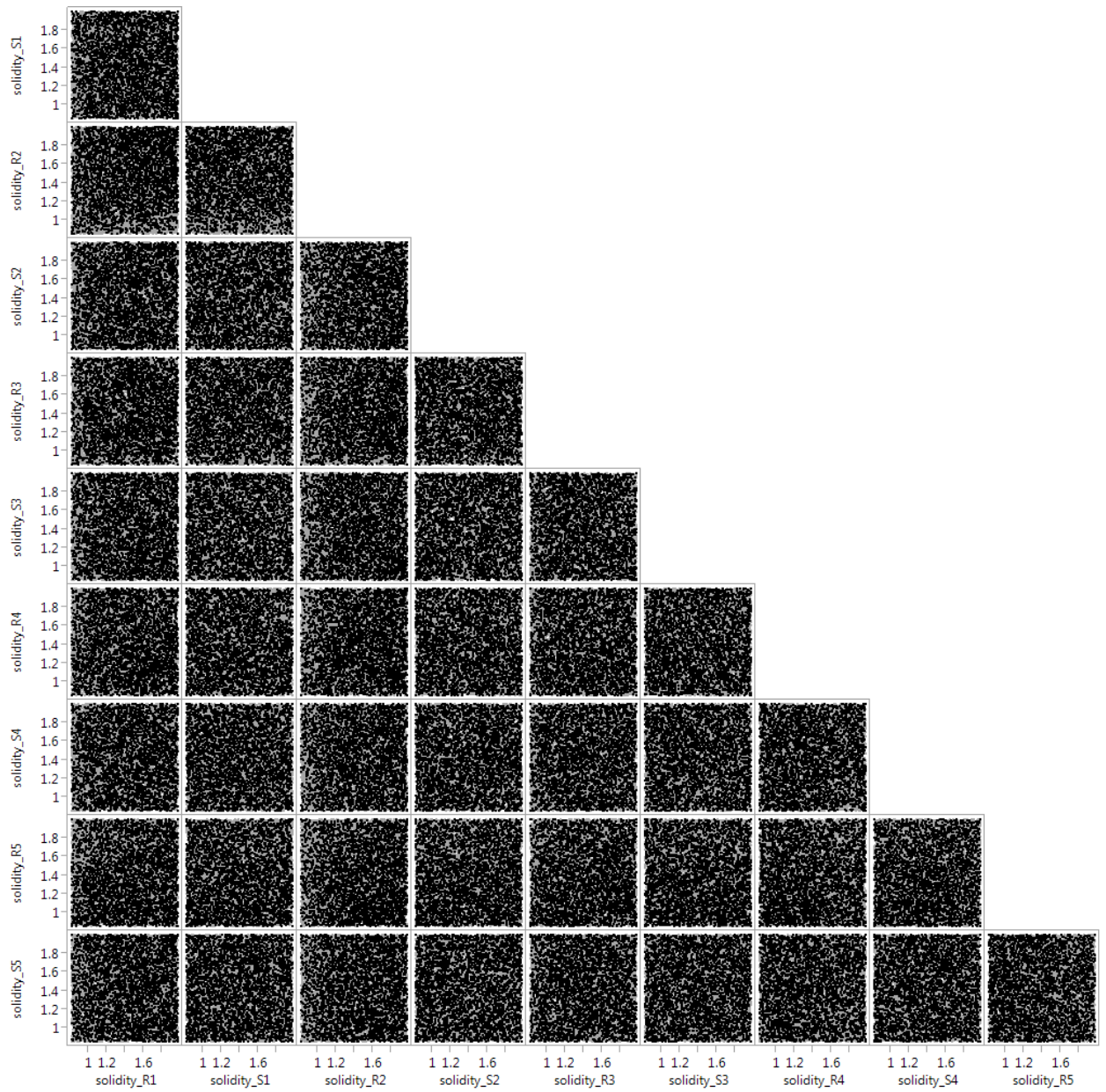


Figure 25: 5 Stage Design Space Solidity Values with Converged Points Highlighted

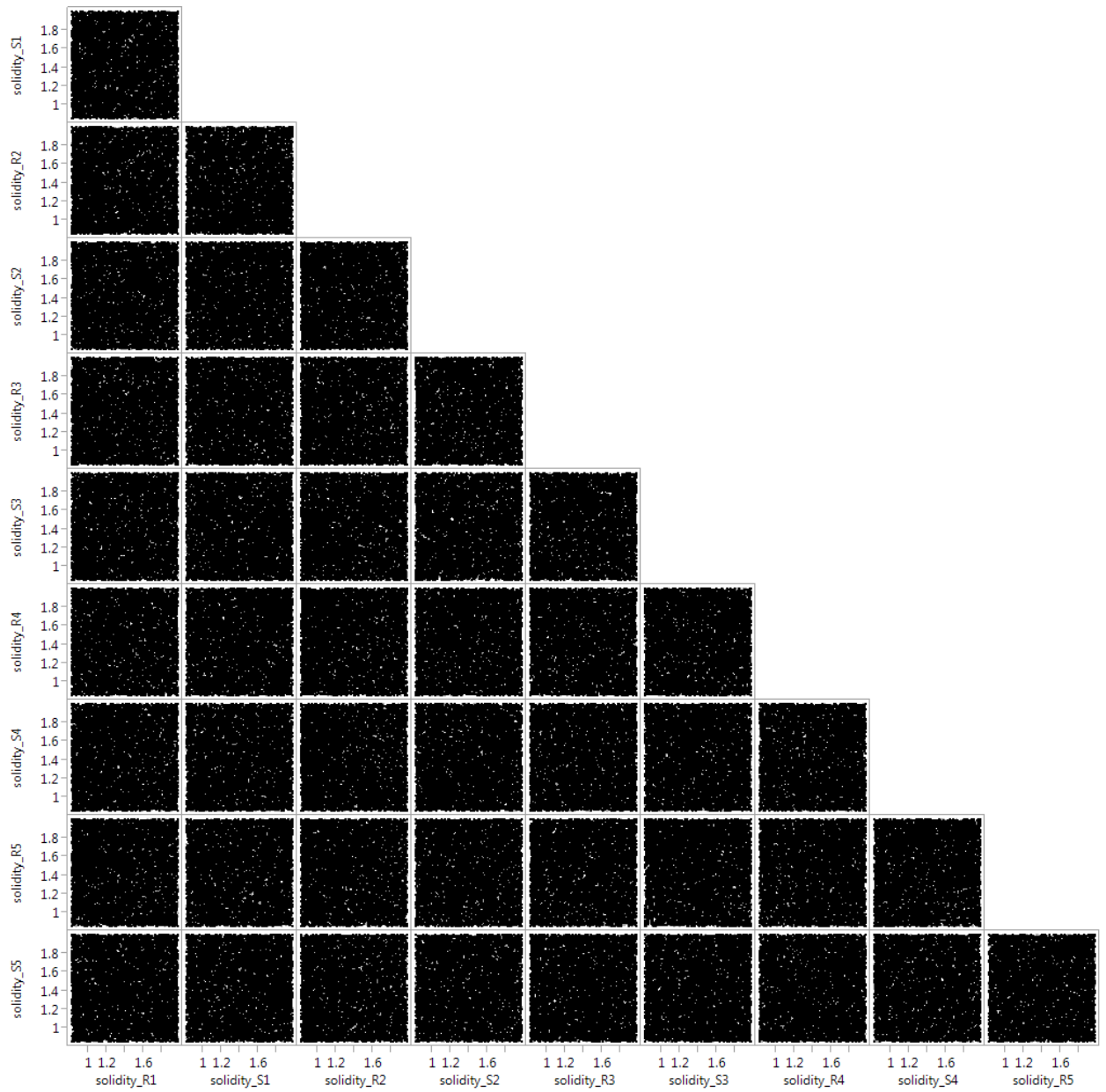


Figure 26: 5 Stage Design Space Solidity Values with Non-converged Points Highlighted

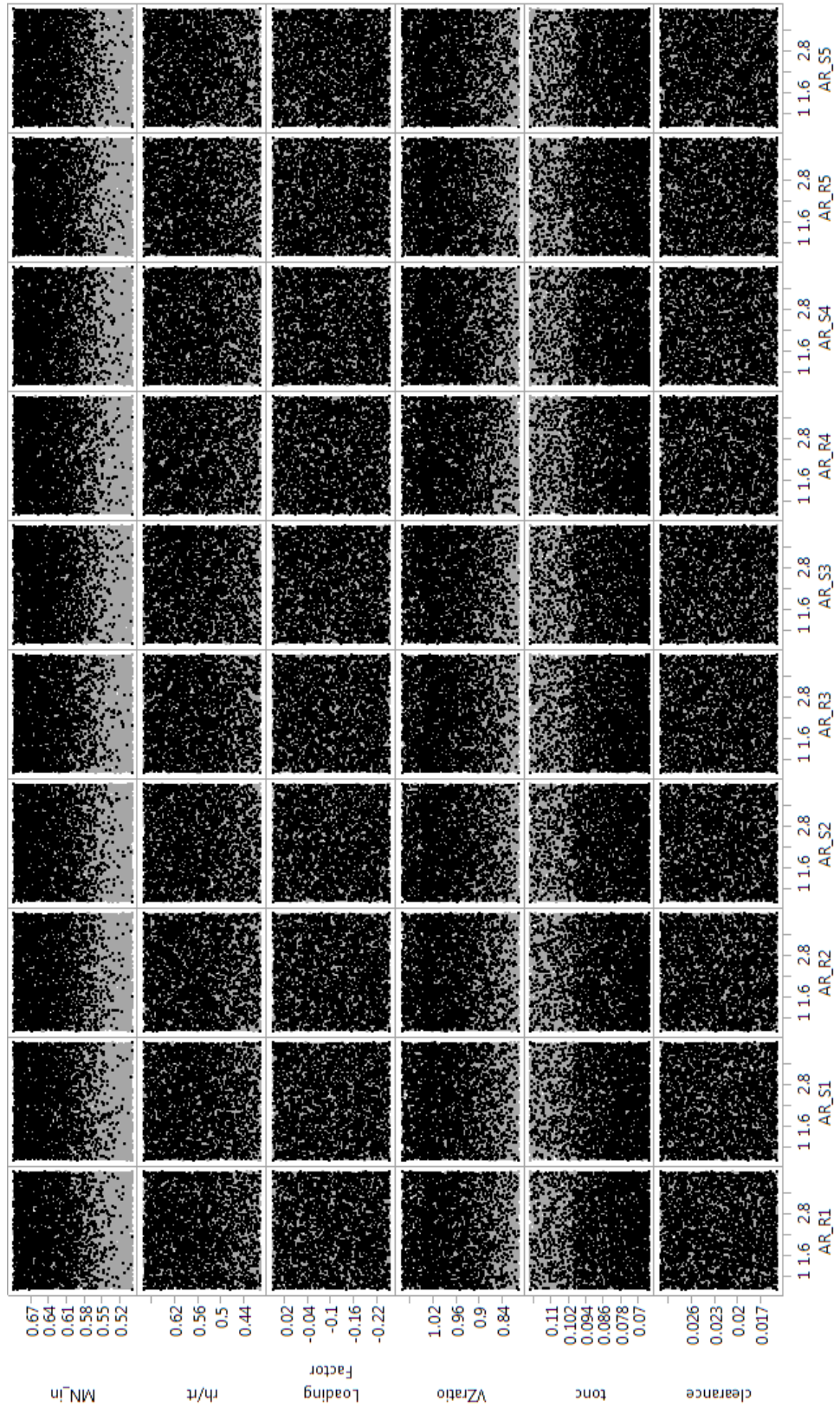


Figure 27: 5 Stage Design Space Parameters with Converged Points Highlighted

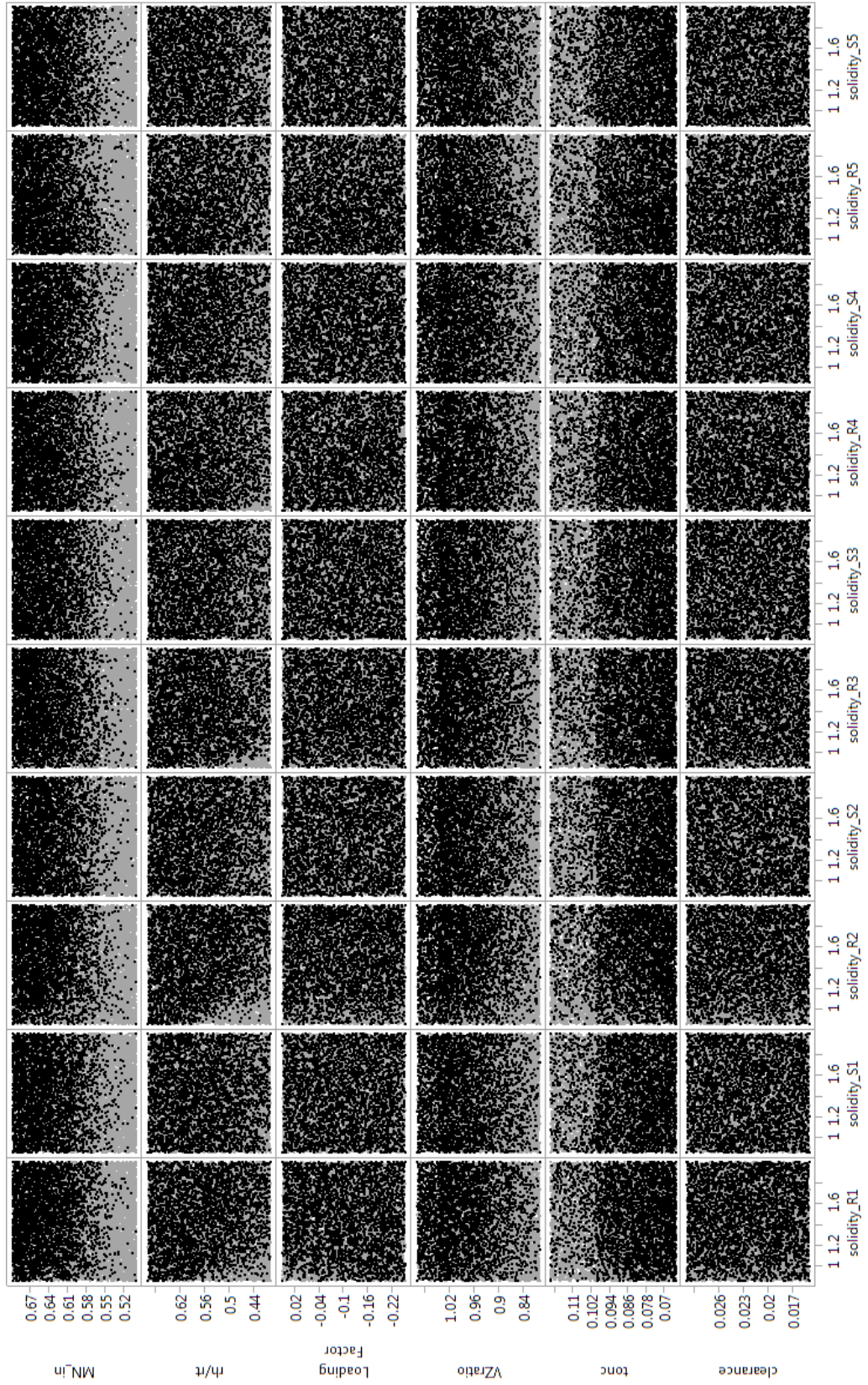


Figure 28: 5 Stage Design Space Parameters with Converged Points Highlighted

APPENDIX B: PARAMETRIC COMPRESSOR CONCEPTUAL DESIGN

CODE

```
//-----  
// runDOE.npss  
// Written by: Andrew Miller  
// Date: 4/7/2015  
// Aerospace Systems Design Laboratory  
// Georgia Institute of Technology  
// Atlanta, GA 30332  
//-----  
  
//-----  
//      DESIGNER INPUTS  
//-----  
real mdot = 67.5;      // lbm/s  
real ShaftSpeed = 12605.;      // RPM  
real PR = 5.;  
real efficiency = 0.85;  
real StageCount = 4;  
string constRadius = "Tip";      // "Hub"; "Mean"; "Tip";      // What radius is constant?  
int isLoaded = TRUE;      // TRUE; FALSE;      // Is the first stage loaded or unloaded?  
real const_tonc;// = 0.07;      // NACA 5 Stage average  
real const_clearance;// = 0.018; // NACA 5 Stage average  
  
cout << endl;  
cout << " DESIGN PARAMETERS " << endl;  
cout << " W = " << mdot << " lb/s \n";  
cout << " N = " << ShaftSpeed << " RPM \n";  
cout << " PRdes = " << PR << " \n";  
cout << " effDes = " << efficiency << " \n\n";  
cout << " Stage Count = " << StageCount << endl;  
  
real inletTipR;  
real PR_des;  
real eff_des;  
real SpeedStallMargin;  
real FlowStallMargin;  
real MN_in;  
real rhqrt;  
real LoadingFactor;  
real VZratio;  
real TB_solidity[StageCount-1][2];
```

```

real TB_AR[StageCount-1][2];
real place[1][2];
real cmpLength;
real rhqrt_out;

//Choose beginning and ending case numbers and number of design variables in doe_table.txt
int start_case_num = 1;
int end_case_num = 1;
int case_num;
string junk;
int varval;

for (StageCount = 4; StageCount <= 7; StageCount++){
    TB_solidity.append(place);
    TB_AR.append(place);
#include <view\DOE.view>
DOEoutHeadings();

//Read design variables from DoE.txt
InFileStream desvar;
//desvar.open("input\doe_table.txt");
desvar.open("input\DOEmaster20050.txt");

//Read design variable names
junk = desvar.getline(); //Eat header

//Eat cases if the DoE is not starting at case 1
if (start_case_num != 1) {
    for (case_num = 1; case_num < start_case_num; case_num++) {
        cout << "deleting case number " << case_num << endl;
        junk = desvar.getline();
    }
}

for (case_num = start_case_num; case_num <= end_case_num; case_num++) {
    cout << endl + "-----INITIATING CASE " + toString(desvar.getReal()) + "-----" + endl + endl;

    //-----
    //          OVERALL DESIGN VARIABLES
    //-----
    MN_in = desvar.getReal();
    rhqrt = desvar.getReal();
    LoadingFactor = desvar.getReal();
    VZratio = desvar.getReal();

    //-----
    //          BLADE ROW DESIGN VARIABLES
    //-----

```

```

const_tonc = desvar.getReal();
const_clearance = desvar.getReal();

for (varval = 0; varval < StageCount; varval++) {
    TB_AR[varval][0] = desvar.getReal();
    TB_solidity[varval][0] = desvar.getReal();
    TB_AR[varval][1] = desvar.getReal();
    TB_solidity[varval][1] = desvar.getReal();
}

junk = desvar.getline();

// write doe inputs to a file so OTAC can read them in
OutputStream inputs;
inputs.open("input\doeinputs.int");

inputs << "//-----\n";
inputs << "//          DESIGNER INPUTS\n";
inputs << "//-----\n";
inputs << "real mdot = " + toStr(mdot) + ";    // lbm/s\n";
inputs << "real ShaftSpeed = " + toStr(ShaftSpeed) + ";  // RPM\n";
inputs << "real PR = " + toStr(PR) + ";\n";
inputs << "real efficiency = " + toStr(efficiency) + ";\n";
inputs << "real StageCount = " + toStr(StageCount) + ";\n";
inputs << "string constRadius = \"\" + constRadius + "\";\n";
inputs << "int isLoaded = " + toStr(isLoaded) + "; // TRUE or FALSE\n\n";

inputs << "//-----\n";
inputs << "//          OVERALL DESIGN VARIABLES\n";
inputs << "//-----\n";
inputs << "real MN_in = " + toStr(MN_in) + ";\n";
inputs << "real rhqrt = " + toStr(rhqrt) + ";\n";
inputs << "real LoadingFactor = " + toStr>LoadingFactor) + ";\n";
inputs << "real VZratio = " + toStr>VZratio) + ";\n\n";

inputs << "//-----\n";
inputs << "//          OVERALL DESIGN VARIABLES\n";
inputs << "//-----\n";
inputs << "real const_tonc = " + toStr(const_tonc) + ";\n";
inputs << "real const_clearance = " + toStr(const_clearance) + ";\n";
inputs << "real TB_AR[][] = " << TB_AR << ";\n";
inputs << "real TB_solidity[][] = " << TB_solidity << ";\n";

inputs.close();

system("run DesignSpace.run");

InFileStream response;

```



```

        response.open("out\DOEtempout.int");
        inletTipR = response.getReal();
        cmpLength = response.getReal();
        rhqrt_out = response.getReal();
        PR_des = response.getReal();
        eff_des = response.getReal();
        SpeedStallMargin = response.getReal();
        FlowStallMargin = response.getReal();
        response.close();

        DOEdata();
        //system("PAUSE");
    }    // end of DOE FOR loop
}

// *****
// * DesignSpace.run
// * Modeled by: Andrew Miller
// * Aerospace Systems Design Laboratory
// * Georgia Institute of Technology
// * Atlanta, GA 30332
// * April 7, 2015
// *****

//-----
//      SET THERMO PACKAGE
//-----

setThermoPackage( "GasTbl" );

//-----
//      INCLUDE OTAC INTERPRETED ELEMENTS AND FILES
//-----
#include <elements\OTAC.fnc>
#include <elements\OTACstart.int>
#include <elements\Expander.int>
#include <elements\Reducer.int>
#include <elements\BladeRowASDL.int>
#include <elements\BladeSegmentASDL.int>
#include <elements\OTACPerfStationAvg.int>
#include <elements\OTACPerfStageASDL.int>
#include <elements\OTACPerfMachine.int>
//-----
//      INCLUDE LOSS MODEL ELEMENTS AND FILES
//-----

```

```

#include <elements\AirfoilASDL.int>
#include <elements\DeviationASDL.int>
#include <elements\OTACLossModel_ProfileNew_Aungier.int>
#include <elements\OTACLossModel_EndwallAungier.int>
#include <elements\OTACLossModel_ShockAungier.int>
#include <elements\StallSocket_Aungier.int>
//-----
//      INCLUDE OTHER FILES
//-----
#include <elements\print_macros.fnc>
#include <elements\solver_macros.fnc>
#include <view\OTACpage4.view>
pvStream.filename = "out/Param_4stg_Map.out";
#include <view\rows.view>

OTACdefaults {
    numberOfStreams = 5;
}
OTACdefaults.calculate();

//-----
// Model of Compressor
//-----
#include <models\ParametricModel.int>
cout << constraint_counter << " constraints violated" << endl;
if (constraint_counter > 0){
    break;
}

//-----
// Add Performance Element
//-----
PERF {}

#include <view\MapData.view>
#include <elements\checkStall.int>
#include <elements\Triangle.int>
#include <elements\BigOutput2.int>
#include <elements\saveGeometry.int>
real step_size = 0.1;
int step = 0;
int stall_flag = 0;
real minloss_out = 1000;
real PR_old = 0;
int solverCount;

```

```

        OutFileStream DOEout;
        DOEout.open("out\\DOEtempout.int");

//-----
// Isentropic FlowStation for Efficiency Calculation
//-----
FlowStation OutIsen {
    description = "Fictitious isentropic flowstation";
}

//-----
// Turbomachinery Analysis, not Cycle Analysis
//-----
OTACenable();

//-----
//          RUN THE MODEL DESIGN POINT
//-----
setOption( "switchDes", "DESIGN" );
autoSolverSetup();
solver.removeDependent( "rotorShaft.integrate_Nmech" );
solver.solutionMode = "ONE_PASS";

run();

real PR_out = (reducer.Fl_O.Pt/expander.Fl_I.Pt);
checkStall();
if(stall_flag == 0){
    cout << "point works!" << endl;
    solver.solutionMode = "STEADY_STATE";
    run();
}

//printPride();
OTACpage.display();
//PERF.print_perf();
Triangles();

PR_out = (reducer.Fl_O.Pt/expander.Fl_I.Pt);
OutIsen.setTotalSP(expander.Fl_I.entropy,reducer.Fl_O.Pt);
real eff_out = (OutIsen.ht-expander.Fl_I.ht)/(reducer.Fl_O.ht-expander.Fl_I.ht);
real loss_out = (reducer.Fl_O.ht-expander.Fl_I.ht)*(1-eff_out);

cout<<"---- Design Point ----"<<endl;
    cout << "    W = " << start.W << " lb/s \n";
    cout << "    N = " <<rotorShaft.Nmech<< " RPM \n";
    cout << "    PR = " << PR_out << " \n";
    cout << "    eff = " << eff_out << " \n";

```

```

        cout<<endl;
        cout<<endl;

BigOutHeadings();
BigOutData();
PR_old = 0;
checkStall();
cout << "stalled? " << stall_flag << endl;
    DOEout << rtip_in << " " << cmpLength << " " << rhqrt_out << " " << PR_out << " " <<
eff_out << " ";
if (stall_flag == 1){
    DOEout << "0 0 ";
}

if(stall_flag == 0 && solver.converged == 1) {
    //-----
    //      SAVE GEOMETRY FOR OFF-DESIGN
    //-----
    saveGeometry();

    //-----
    //      REPEAT DESIGN POINT IN OFF-DESIGN MODE
    //-----

    setOption( "switchDes", "OFFDESIGN" );
    autoSolverSetup();
    solver.removeIndependent( "rotorShaft.ind_Nmech" );
    solver.removeDependent( "rotorShaft.integrate_Nmech" );
    solver.solutionMode = "STEADY_STATE";

    run();

    OTACpage.display();
    //printPride();
    //printPrideOTAC();
    //saveIndependents();

    real PR_des = (reducer.Fl_O.Pt/expander.Fl_I.Pt);
    OutIsen.setTotalSP(expander.Fl_I.entropy,reducer.Fl_O.Pt);
    real eff_des = (OutIsen.ht-expander.Fl_I.ht)/(reducer.Fl_O.ht-expander.Fl_I.ht);
    loss_out = (reducer.Fl_O.ht-expander.Fl_I.ht)*(1-eff_out);

    cout<<"---- Off-Design Point ----"<<endl;
        cout << "    W = " << start.W << " lb/s \n";
        cout << "    N = " << rotorShaft.Nmech<< " RPM \n";
        cout << "    PR = " << PR_des << " \n";
        cout << "    eff = " << eff_des << " \n";

```

```

        cout<<endl;
        cout<<endl;

BigOutData();
checkStall();
cout << "stalled? " << stall_flag << endl;

//break;

//-----
//      RUN TO STALL FOR STALL MARGIN
//-----

// Constant Speed Stall Margin
while (stall_flag == 0) {
    cout << "----Decreasing flow----" << endl;
    start.W = mdot - step * step_size;

    run();

    PR_out = (reducer.Fl_O.Pt/expander.Fl_I.Pt);
    OutIsen.setTotalSP(expander.Fl_I.entropy,reducer.Fl_O.Pt);
    eff_out = (OutIsen.ht-expander.Fl_I.ht)/(reducer.Fl_O.ht-expander.Fl_I.ht);
    loss_out = (reducer.Fl_O.ht-expander.Fl_I.ht)*(1-eff_out);

    //-----
    // Check Aungier Stall Criteria
    //-----
    checkStall();

    //-----
    // Outputs
    //-----
    cout << "    N = " << rotorShaft.Nmech << " RPM \n";
    cout << "    W = " << start.W << " lb/s \n";
    cout << "    PR = " << PR_out << " \n";
    cout << "    eff = " << eff_out << " \n";
    cout << stall_segments3 << stall_segments1 << endl;

    // for next iteration
    stall_segments1 = "";
    stall_segments3 = "";
    step++;
}

real SpeedStallMargin = (PR_out / PR_des) * (mdot / start.W);
cout << "Constant Speed Stall Margin = " << SpeedStallMargin << endl;

```

```

DOEout << SpeedStallMargin << " ";

// Step back to design point
while (step > 0) {
    start.W = mdot - step * step_size;
    run();
    step--;
}
start.W = mdot;
stall_flag = 0;
PR_old = 0;

// Constant Flow Stall Margin
while (stall_flag == 0) {

    cout << "----Increasing speed----" << endl;
    rotorShaft.Nmech = ShaftSpeed * (1.0 + 0.1 * step * step_size);

    run();

    PR_out = (reducer.Fl_O.Pt/expander.Fl_I.Pt);
    OutIsen.setTotalSP(expander.Fl_I.entropy,reducer.Fl_O.Pt);
    eff_out = (OutIsen.ht-expander.Fl_I.ht)/(reducer.Fl_O.ht-expander.Fl_I.ht);
    loss_out = (reducer.Fl_O.ht-expander.Fl_I.ht)*(1-eff_out);

    //-----
    // Check Aungier Stall Criteria
    //-----
    checkStall();

    //-----
    // Outputs
    //-----
    cout << "    N = " << rotorShaft.Nmech << " RPM = " << rotorShaft.Nmech /
ShaftSpeed * 100 << "%\n";
    cout << "    W = " << start.W << " lb/s \n";
    cout << "    PR = " << PR_out << " \n";
    cout << "    eff = " << eff_out << " \n";
    cout << stall_segments3 << stall_segments1 << endl;

    // for next iteration
    stall_segments1 = "";
    stall_segments3 = "";
    step++;
}

real FlowStallMargin = (PR_out - PR_des) / PR_des;
cout << "Constant Flow Stall Margin = " << FlowStallMargin << endl;

```

```

        DOEout << FlowStallMargin << " ";
    }

DOEout.close();
//-----
// ParametricModel.int
// Written by: Andrew Miller
// Date: 4/7/2015
// Aerospace Systems Design Laboratory
// Georgia Institute of Technology
// Atlanta, GA 30332
//-----

//-----
// function to create a parametric compressor model
//-----
    #include <input\doeinputs.int>

// Initialize variables
real Tt_in, Pt_in;
real Tt_out, Pt_out;
real B_in, B_out;
real tau;
real effPoly;
real rtip_in, rmean_in, rhub_in;
real rtip_out, rmean_out, rhub_out;
real Aphys;
real TipMN;
string R_fs, S_fs, Stage_1;
string Rname, Sname;
int iStg;
int iBR;
real delh_cmp, delh_stg, delh, ht_out;
real PR_stage;
real beta_out;
string Airfoil;
real Rheight, Sheight, height;
string Station;
real Rsolidity, Ssolidity;
real constraint_counter;
real rhqrt_out;

//-----
//          INITIALIZE ELEMENTS AND LINK PORTS
//-----

    OutFileStream model;

```

```

model.open("models\\model.mdl");

model << "OTACstart start { }\n";
model << "Expander expander { }\n\n";
for (iStg = 1; iStg <= StageCount; iStg++){
    model << "// STAGE " << iStg << endl;
    Rname = "R" + toStr(iStg);
    Sname = "S" + toStr(iStg);
    model << "BladeRow " + Rname + " { }\n";
    model << "BladeRow " + Sname + " { }\n\n";
}
model << "Reducer reducer { }\n";
model << "FlowEnd end { }\n\n";

model << "Shaft rotorShaft {\n ShaftInputPort";
for ( iStg = 1; iStg <= StageCount; iStg++ ){
    model << " Sh_ROT" + toStr(iStg);
    if (iStg < StageCount){
        model << ",";
    }
}
model << ";\n Nmech = " << ShaftSpeed << ";\n}\n\n";

model << "linkPorts( \"start.Fl_O\", \"expander.Fl_I\", \"station0\");\n";
for (iStg = 1; iStg <= StageCount; iStg++){
    Rname = "R" + toStr(iStg);
    Sname = "S" + toStr(iStg);
    Stage_1 = "S" + toStr(iStg-1);
    if (iStg == 1){
        Stage_1 = "expander";
    }
    Station = "station" + toStr(2*iStg-1);
    model << "linkSegmentPorts( \"\" << Stage_1 << "\", \"\" << Rname << "\", \"\" <<
Station << "\" );\n";
    Station = "station" + toStr(2*iStg);
    model << "linkSegmentPorts( \"\" << Rname << "\", \"\" << Sname << "\", \"\" <<
Station << "\" );\n";
}
    Station = "station" + toStr(2*StageCount+1);
    model << "linkSegmentPorts( \"\" << Sname << "\", \"reducer\", \"\" << Station << "\"
);\n";
    Station = "station" + toStr(2*StageCount+2);
    model << "linkPorts( \"reducer.Fl_O\", \"end.Fl_I\", \"\" << Station << "\" );\n";

for ( iStg = 1; iStg <= StageCount; iStg++ ){
    Rname = "R" + toStr(iStg) + ".Sh_O";
    Sname = "rotorShaft.Sh_ROT" + toStr(iStg);
    Station = "mlink_r" + toStr(iStg);

```



```

model << "linkPorts( \"\" << Rname << "\", \"\" << Sname << "\", \"\" << Station <<
"\");\n";
}

model << "\nOTACPerf PERF{\n machineType = \"COMPRESSOR\";\n stageDef = {";
for (iStg = 1; iStg <= StageCount; iStg++){
    Rname = "R" + toStr(iStg);
    Sname = "S" + toStr(iStg);
    model << "\n        { \"\" +Rname+ "\", \"\" +Sname+ "\" }";
    if (iStg < StageCount){
        model << ",";
    }
}
model << "\n    }\n}\n\n";

model << "\n\n//        BLADE ROW INPUTS\n\n";
model << "OTACdefaults.calculate();\n";
model << "string meanSegVm;\n";
model << "meanSegVm = \"FI_O\" + toStr(OTACdefaults.midStream) + \".Vm\";\n\n";

//-----
//        CALCULATE INLET CONDITIONS & GEOMETRY
//-----
Tt_in = 518.67;
Pt_in = 14.696;
B_in = 0.01;

FlowStation Inlet;
Inlet.setTotalITP(Tt_in, Pt_in);
Inlet.startOTAC();
Inlet.W = mdot;
Inlet.omega = ShaftSpeed * 2 * PI / 60;           // rad/sec
Inlet.setVs_alphaPhiMN(0.00, 0.00, MN_in);
Aphys = Inlet.Aphy / (1-B_in);

rtip_in = sqrt( Aphys / PI / (1 - rhqrt**2) );
rhub_in = rhqrt * rtip_in;
rmean_in = (rhub_in + rtip_in) / 2;

Inlet.radius = rmean_in;
Inlet.setVs_alphaPhiMN(0.00, 0.00, MN_in);
//printUnits(Inlet.list("Variable", FALSE));

model << "start{\n";
model << "    Tt = \" << Inlet.Tt << \";\n";
model << "    Pt = \" << Inlet.Pt << \";\n";
model << "    W = \" << Inlet.W << \";\n";
model << "    MNdes = \" << Inlet.MN << \";\n";

```

```

model << "    radiusMean_in = " << Inlet.radius << ";\n";
model << "    alpha = " << Inlet.alpha << ";\n";
model << "}\n\n";

//-----
//      DISCHARGE CONDITIONS & GEOMETRY
//-----
Pt_out = Pt_in * PR;
tau = PR**((Inlet.gams-1)/Inlet.gams/efficiency);
Tt_out = Tt_in * tau;
B_out = 0.0125 * StageCount + 0.0375;

FlowStation Discharge;
Discharge.setTotalTP(Tt_out, Pt_out);
Discharge.startOTAC();
Discharge.Vflow = VZratio * Inlet.Vz;
Discharge.W = mdot;
Discharge.omega = ShaftSpeed * 2 * PI / 60;           // rad/sec
Aphys = Discharge.Aphy / (1-B_out);
if (constRadius == "Tip"){
    rtip_out = rtip_in;
    rhub_out = sqrt(rtip_out**2 - Aphys / PI);
    rmean_out = (rhub_out + rtip_out) / 2;
    cout << " Constant " + constRadius + " Radius (in) = " << rtip_out << endl;
}
else if (constRadius == "Hub"){
    rhub_out = rhub_in;
    rtip_out = sqrt(Aphys / PI + rhub_out**2);
    rmean_out = (rhub_out + rtip_out) / 2;
    cout << " Constant " + constRadius + " Radius (in) = " << rhub_out << endl;
}
else if (constRadius == "Mean"){
    rmean_out = rmean_in;
    rtip_out = Aphys / 4 / PI / rmean_out + rmean_out;
    rhub_out = rmean_out - Aphys / 4 / PI / rmean_out;
    cout << " Constant " + constRadius + " Radius (in) = " << rmean_out << endl;
}
rhqrt_out = rhub_out / rtip_out;
Discharge.radius = rmean_out;
Discharge.setVs_alphaPhiMN(0.00, 0.00, Discharge.MN);
//printUnits(Discharge.list("Variable", FALSE));

delh_cmp = Discharge.ht - Inlet.ht;
delh_stg = delh_cmp / StageCount;
effPoly = ((Inlet.gams - 1) / Inlet.gams) * log(PR) / log(1 + (PR ** ((Inlet.gams - 1) /
Inlet.gams) - 1) / efficiency);

for (iStg = 1; iStg <= StageCount; iStg++){

```

```

B_out = 0.0125 * iStg + 0.0375;

// Stator Exit Flow Station
S_fs = "Stator" + toStr(iStg);
create( "", "FlowStation", S_fs );

if (isLoaded == TRUE){
    if (iStg == 1){
        delh = delh_stg * (1 + LoadingFactor);
    } else {
        delh = delh_stg * (1 - LoadingFactor / (StageCount - 1) );
    }
} // Create Constraint: Stage Loading < 0.6 ?

Station
Stage_1 = "Stator" + toStr(iStg-1); // Previous Stage Exit Flow

if (iStg == 1){
    Stage_1 = "Inlet";
}
ht_out = Stage_1->ht + delh;
PR_stage = (effPoly * delh / Stage_1->ht + 1)**(Stage_1->gamt/(Stage_1->gamt-
1));

Pt_out = Stage_1->Pt * PR_stage;

S_fs->setTotal_hP(ht_out, Pt_out);
//cout << "Stage " + toStr(iStg) + " PR = " << S_fs->Pt / Stage_1->Pt << endl;

S_fs->startOTAC();
S_fs->Vflow = iStg * (Discharge.Vflow - Inlet.Vflow) / StageCount + Inlet.Vflow;
S_fs->W = mdot;
S_fs->omega = ShaftSpeed * 2 * PI / 60; // rad/sec
Aphys = S_fs->Aphy / (1-B_out);
if (constRadius == "Tip"){
    rtip_out = rtip_in;
    rhub_out = sqrt(rtip_out**2 - Aphys / PI);
    rmean_out = (rhub_out + rtip_out) / 2;
}
else if (constRadius == "Hub"){
    rhub_out = rhub_in;
    rtip_out = sqrt(Aphys / PI + rhub_out**2);
    rmean_out = (rhub_out + rtip_out) / 2;
}
else if (constRadius == "Mean"){
    rmean_out = rmean_in;
    rtip_out = Aphys / 4 / PI / rmean_out + rmean_out;
    rhub_out = rmean_out - Aphys / 4 / PI / rmean_out;
}
S_fs->radius = rmean_out;

```

```

beta_out = atan( S_fs->U / S_fs->Vflow );
S_fs->setVs_betaVz(beta_out, S_fs->Vflow);
Sheight = rtip_out - rhub_out;

// Rotor Exit Flow Station
R_fs = "Rotor" + toStr(iStg);
create( "", "FlowStation", R_fs );
R_fs->setTotal_hP(ht_out, Pt_out);
R_fs->startOTAC();
R_fs->W = mdot;
R_fs->omega = ShaftSpeed * 2 * PI / 60; // rad/sec
R_fs->radius = ( S_fs->radius + Stage_1->radius ) / 2; // Is this a good
assumption?
beta_out = atan( ( R_fs->U - C_GRAVITY * C_BTUtoFT_LBF * delh / R_fs->U ) /
S_fs->Vflow );
R_fs->setVs_betaVz(beta_out, S_fs->Vflow);
Rheight = R_fs->radiusOuter - R_fs->radiusInner;
//printUnits(R_fs->list("Variable", FALSE));

// Write Blade Row Inputs to Model File
for (iBR = 0; iBR <= 1; iBR++){
    if (iBR == 0){
        model << "R" << iStg << " {\n";
        Rname = "Rotor" + toStr(iStg);
        height = Rheight;

        // Define Rotor Airfoil
        TipMN = sqrt( Stage_1->Vz**2 + (Stage_1->radiusOuter / 12 *
Stage_1->omega)**2 ) / Stage_1->Vsonic;
        if(TipMN > 0.8){
            Airfoil = "DCA";
        } else {
            Airfoil = "NACA65";
        }
        //cout << Rname + " TipMN = " << TipMN << "; Airfoil = " +
Airfoil << endl;

        // Set Options and Dependents
        model << "    setOption( \"switchRotate\", \"ROTATING\"
);\n";
        model << "    setOption(          \"switchBladeAngleSign\",
\"POSITIVE\");\n";
        model << "    setOption( \"switchSpanwiseInput\", \"WORK\"
);\n";
        model << "    setOption( \"switchAirfoilType\", \"\" << Airfoil
<< "\");\n";
        model << "    switchLossBasis = \"PTOTREL\";\n\n";

```

```

        model << "      Table S_INPUTvR( real pctSpan ) {\n";
        model << "          pctSpan = { 0.0, 1.0 }\n";
        model << "          WORK = { " << -delh << " , " << -delh << "

}\n      }\n";

        model << "      Table S_INCIDENCEvR( real pctSpan ) {\n";
        model << "          pctSpan = { 0.0, 1.0 }\n";
        model << "          incDes = { 0.0, 0.0 }\n      }\n\n";

    } else {
        model << "S" << iStg << " {\n";
        Rname = "Stator" + toStr(iStg);
        height = Sheight;

        // Define Stator Airfoil
        TipMN = R_fs->MN;
        if(TipMN > 0.6){
            Airfoil = "DCA";
        } else {
            Airfoil = "NACA65";
        }
        //cout << Rname + " TipMN = " << TipMN << "; Airfoil = " +
Airfoil << endl;

        // Set Options and Dependents
        model << "      setOption(          \"switchRotate\",
\n\"NON_ROTATING\" );\n";
        model << "      setOption(          \"switchBladeAngleSign\",
\n\"NEGATIVE\" );\n";
        model << "      setOption( \"switchSpanwiseInput\", \"ALPHA\"
);\n";
        model << "      setOption( \"switchAirfoilType\", \"\" << Airfoil
<< "\");\n";
        model << "      switchLossBasis = \"PTOT\";\n\n";

        model << "      Table S_INPUTvR( real pctSpan ) {\n";
        model << "          pctSpan = { 0.0, 1.0 }\n";
        model << "          AlphaEx = { 0.0, 0.0 }\n      }\n";
        model << "      Table S_INCIDENCEvR( real pctSpan ) {\n";
        model << "          pctSpan = { 0.0, 1.0 }\n";
        model << "          incDes = { 0.0, 0.0 }\n      }\n\n";

    }

// BLADE ROW INDEPENDENT VARIABLES
model << "      tonc_in = " << const_tonc << ";\n";
model << "      clearance = " << const_clearance << ";\n";
model << "      aspectRatio = " << TB_AR[iStg-1][iBR] << ";\n";

```

```

// Other blade row variables
model << "    radiusMean_in = " << Rname->radius << ";\n";
model << "    blockage_in = " << B_out << ";\n";
model << "    chord_in = " << height / TB_AR[iStg-1][iBR] << ";\n";
model << "    numberOfBlades = " << ceil(2 * PI * Rname->radius *
TB_solidity[iStg-1][iBR] * TB_AR[iStg-1][iBR] / height) << ";\n";
model << "    N_row = " << iStg << ";\n\n";

model << "    dep_DESIGN_MeanVelocity.eq_lhs = meanSegVm;\n";
model << "    dep_DESIGN_MeanVelocity.eq_rhs = \"\" << Rname->Vz
<< "\";\n\n";

// Include loss models
model << "    addSubelementToStack( \"Airfoil\", \"S_Airfoil\" );\n";
model << "    addSubelementToStack( \"Deviation\", \"S_Deviation\"
);\n";
model << "    addSubelementToStack(          \"ProfileLossModel\",
\"profileloss\" );\n";
model << "    addSubelementToStack(          \"EndwallLossModel\",
\"endwallloss\" );\n";
model << "    addSubelementToStack(          \"ShockLossModel\",
\"shockloss\" );\n";
model << "    addSubelementToStack( \"Stall\", \"S_Stall\" );\n\n";

    }

}

model.close();
#include <models/model.mdl>

// *****
// * saveGeometry.int
// *****

// for each blade row, save the leading edge and trailing edge blade angles and radii
OutputStream geo {filename="models\CmpGeom.map";}
int rowNum;
string BladeRows3[];
void saveGeometry() {
    BladeRows3 = list("BladeRow",TRUE);
for (rowNum = 0; rowNum < BladeRows3.entries(); rowNum++){

```

```
        BladeRows3[rowNum]->saveDesignBladeAngles(           BladeRows3[rowNum],
"TB_BladeInletAngle",    BladeRows3[rowNum]->LEradiusValues, BladeRows3[rowNum]-
>LEangleValues );
        BladeRows3[rowNum]->saveDesignBladeAngles(           BladeRows3[rowNum],
"TB_BladeExitAngle",    BladeRows3[rowNum]->TERadiusValues, BladeRows3[rowNum]-
>TEangleValues );
    }
}
```

APPENDIX C: LOSS MODELS CODE

```
//=====
//          OTAC
//      Object-Oriented Turbomachinery Analysis Code
//
//  written by: Justin S. Gray, Scott M. Jones, and Thomas M. Lavelle
//
//=====
```

```
#include <InterpIncludes.ncp>
```

```
class Deviation extends Subelement {
```

```
//-----
//      ++++++++ DOCUMENTATION ++++++++
//-----
title = "";
```

```
description = isA()+ " calculates the flow deviation at the exit of a
particular blade segment. The correlation used is based on Carter and
Hughes. ";
```

```
usageNotes = isA()+ " can be added to each BladeSegment in a BladeRow
by calling the addSubelementToStack() function in a model file like so:
```

```
    myBladeRowName.addSubelementToStack( 'Deviation', 'S_Deviation' )
```

```
This Subelement requires non-zero values for solidity in order to
function properly. If the BladeRow this Subelement has been added to
does not have chord length and number of blades specified a warning
will be issued and execution will stop. ";
```

```
//-----
//      ++++++++ SET UP VARIABLES ++++++++
//-----
```

```
real m {
    value = 0.23; units = NONE;
    description = "slope of the line of deviation angle versus camber angle";
}
real theta {
    value = 0.*PI/180.; units = RAD;
    description = "camber angle for the blade segment";
}
```

```
real dev_st_10 {
    description = "design deviation_tc_10";
```



```

}
real beta_in;
// real deviation;
real dev_st;
real dev_mod;
real m_10;
real b;
real x;
real t_c;
real ddevqdinc;
    real camberDeg;
    real p1;
    real p2;
    real defl_star;
/* real Ksh {
value = 1.0; IOstatus = INPUT; units = NONE;
description = "blade shape parameter";
} */
    real Ktd;        // thickness correction

Option switchDes {
    allowedValues = { DESIGN, OFFDESIGN };
    description = "Design mode switch indicator [ DESIGN / OFFDESIGN]";
    rewritableValues = FALSE; // Enables converter optimization.
}

Option switchModel {
    allowedValues = { "AUNGIER", "CARTER", "HOWELL" };
    description = "Swtich between different deviation models";
    rewritableValues = FALSE; // Enables converter optimization.
}

//-----
// ++++++ SET UP PORTS, FLOW STATIONS, SOCKETS, AND TABLES ++++++
//-----

// SOCKETS
addInterface( "DEVANG" );

//-----
// ++++++ VARIABLE CHANGED METHODOLOGY ++++++
//-----
void variableChanged( string name, any oldVal ) {
    if ( name=="switchDes" ) {
        // do nothing
    }
}

//-----
// ++++++ PERFORM ENGINEERING CALCULATIONS ++++++
//-----
void calculate() {

```

```

        if ( switchModel == "CARTER" ) {
// curve fit from Creveling and Carmody, NASA CR-72427
        defl_star = camber*180./PI - dev_star + inc_star;
        p1 = (incidence*180./PI - inc_star)/defl_star;
        p1 = min(0.74,max(p1,-0.36)); // bracket to avoid extrapolation of Creveling-
Carmody curve
//      p1 = max(p1,-0.36);
//      if (p1 <= 0.74) {
//          p2 = (((((-3.011972E+00)*p1 + 3.531755E+00)*p1 + 9.064304E-01)*p1 -
2.619910E+00)*p1 + 7.536704E-01)*p1 + 4.450216E-01)*p1 + 7.154201E-05;
//      } else {
//          p2 = 0.11852*p1 + 0.15369;
//      }
        deviation = ( p2 * defl_star + dev_star )*PI/180.;
    }
    else if ( switchModel == "HOWELL" ) {
// curve fit adapted from Howell, Fluid Dynamics of Axial Compressors (1945)
        defl_star = camber*180./PI - dev_star + inc_star;
        p1 = (incidence*180./PI - inc_star)/defl_star;
        p1 = min(0.6,max(p1,-0.6)); // bracket to avoid extrapolation of Howell curve
//      p1 = min(1.8,max(p1,-0.46)); // bracket to avoid extrapolation of composite curve

        p2 = (((3.969704E-01)*p1 + 2.968064E-01)*p1 + 4.137201E-01)*p1 + 1.545309E-01)*p1
- 9.964312E-03; // Howell curve
//      p2 = ((((((9.341083E-03)*p1-1.838373E-01)*p1+7.061295E-01)*p1-9.003375E-
01)*p1+2.316127E-01)*p1+3.139581E-01)*p1; // composite curve
        deviation = ( p2 * defl_star + dev_star )*PI/180.;
    } else {
// Aungier's model for off-design deviation
        if (switchRotate == "NON_ROTATING") {
            beta_in = abs(FI_IR.alpha)*180./PI; }
        else {
            beta_in = abs(FI_IR.beta)*180./PI;
        }
        ddevqdinc = (1.0 + (solidity + 0.25*(solidity**4.0))*( ( beta_in/53.)**2.5 )/exp(3.1*solidity); //
Aungier eqn 6-76
//      dev_mod = dev_star + (incidence*180./PI - inc_star)*ddevqdinc + 10.0*(1.0 -
abs(FI_OR.Vm/FI_IR.Vm)); // Aungier eqn 6-77
        dev_mod = dev_star + (incidence*180./PI - inc_star)*ddevqdinc;
        deviation = dev_mod*PI/180.;
    }
} // end calculate
}

```

```

//=====
//          OTAC
//      Object-Oriented Turbomachinery Analysis Code
//
//  written by: Justin S. Gray, Scott M. Jones, and Thomas M. Lavelle
//
//=====

#include <InterpIncludes.ncp>

class ProfileLossModel extends Subelement {

//-----
//      ++++++++ DOCUMENTATION ++++++++
//-----
title = "";

description = isA()+ " is to be used as the base class for functioning
OTAC loss models. The loss model will return an estimated value for loss
based on the blade segment current conditions; the blade row solver will
enforce the condition that the loss estimated from this socket equals the
assumed, or actual, loss for the blade segment. The form of the loss is
set by the blade row attribute switchLossBasis, and may be either total
pressure loss, relative total loss parameter (omegaBar), or segment
adiabatic efficiency.
";

usageNotes = isA() + "";

addInterface("OTACLossModel");
socketRequired = FALSE;

//-----
//      ++++++++ SET UP VARIABLES ++++++++
//-----
real lossEstimated {
  value = 0.0; IOstatus = OUTPUT; units = NONE;
  description = "estimated loss for this blade segment";
}
real t_c {
  value = 0.08; IOstatus = INPUT; units = NONE;
  description = "blade max thickness-to-chord ratio";
}
real a_c {
  value = 0.5; IOstatus = INPUT; units = NONE;
  description = "location of max camber";
}

real beta1_choke {
  value = 60.0; IOstatus = INPUT; units = DEG;
  description = "flow angle at choke";
}
}

```

```

/*    real Ksh {
    value = 1.0; IStatus = INPUT; units = NONE;
    description = "blade shape parameter";
}*/

FlowStation Fs_sonic {
    description = "sonic flow conditions at inlet total conditions";
}

    real Rec;
    real Deq;
    real wp_st;
    real rothalpy_in;
    real rothalpy_out;
    real circ_param;
    real deHaller;
    real alpha_st;
    real inc_st;
    real M1rel;
    real Rc;
//    real Rs;
    real Ksh1;
//    real inc_c;
    real inc_c_min;
//    real inc_s;
//    real inc_min;
    real wp_min;
    real wp;

    real inc;
    real camberDeg;

    real beta1_st;
    real beta2_st;
    real beta_in;
    real beta_out;

    real loss_param;
    real xi;
    real KRe;
    real err1;
    real kappa1;
    real kappa2;
    real Kti;           // thickness correction
    real n;
    real q;
    real m;
    real m1;
    real K1;
    real K2;
    real s_h;

```

```

real gam;
real aa;
real bb;
real cc;
real dd;
real ee;
real ff;
real Vmaxq1;
real delPq;
real delPq1;
// real Mcrit;
real sMcrit = 1;
real sVmax = 1.;
real wmaxqw1star;
real W1qW2_st;
real a1;
real WmaxqW1;
real WstarqW1;
real term2;
real p;
real inc_st_10;
real arg1;

SecantSolver iter1 {
    maxDx = 1;
    tolerance = 1.E-5;
    perturbSize = 0.01;
    maxIters = 50;
}

void calculate() {

    deHaller = abs(FI_OR.Vrel / FI_IR.Vrel); // deHaller number
    rothalpy_in = FI_IR.hs + (FI_IR.Vrel**2 - FI_IR.U**2)/2./32.174/778.1693;
    rothalpy_out = FI_OR.hs + (FI_OR.Vrel**2 - FI_OR.U**2)/2./32.174/778.1693;
    circ_param = 2.*abs(FI_IR.radius*FI_IR.Vtheta
FI_OR.radius*FI_OR.Vtheta)/(solidity*(FI_IR.radius+FI_OR.radius)*FI_IR.Vrel);

    inc = incidence*180./PI;
    t_c = tonc; // thickness-to-chord from parent
    if (parent.parent.switchRotate == "NON_ROTATING") {
        beta_in = abs(FI_IR.alpha); }
    else {
        beta_in = abs(FI_IR.beta);
    }
    beta_out = beta_in - deflection;
    kappa1 = bladeInletAngle*180./PI;
    kappa2 = bladeExitAngle*180./PI;
    Ksh1 = min(Ksh, 1.0); // Aungier page 136

//-----

```

```

// Minimum profile loss from Aungier, Axial-Flow Compressors
//-----

//      if (parent.switchBladeAngleSign == "POSITIVE") {
//          beta1_st = (inc_star + kappa1)*PI/180.;
//          beta2_st = (dev_star + kappa2)*PI/180.;
//      }
//      else {
//          beta1_st = -(inc_star + kappa1)*PI/180.;
//          beta2_st = -(dev_star + kappa2)*PI/180.;
//      }
wmaxqw1star = 1.12 + 0.61*( (cos(beta1_st)**2 )/solidity )*(tan(beta1_st) -
tan(beta2_st)); // Aungier eqn 6-35
Deq = wmaxqw1star*cos(beta2_st)*FI_IR.Vm/cos(beta1_st)/FI_OR.Vm;
// Aungier eqn 6-36
Deq = min (2.0, Deq); // is this needed? RKD 4-25-2014
K1 = 0.0073; // Aungier page 150 changed from 0.004 RKD 5-6-2014 and from
0.0073 to 0.01 for NACA 8-stage only!
s_h = 1.0 / solidity / aspectRatio; // spacing-to-height
Rec = FI_IR.rhos * FI_IR.Vrel * chord/12. / FI_IR.mus; // using def given by Koch & Smith
if (Rec < 2.5E5) {
    KRe = sqrt(2.5E5/Rec) - 1.0; } // Aungier eqn 6-101
else {
    KRe = (log10(2.5E5)/log10(Rec))**2.58 - 1.0; } // Aungier eqn 6-102
K2 = 1.0 + s_h*cos(beta_out) + 0.004*KRe/K1; // Aungier eqn 6-100
W1qW2_st = Deq/wmaxqw1star; // Aungier eqn 6-36
loss_param = K1 * (K2 + 3.1*(Deq -1.0)**2 + 0.4*(Deq -1.0)**8); // RHS of Aungier
eqn 6-46
wp_st = 2. * loss_param * solidity / cos(abs(beta2_st)) / (W1qW2_st)**2; // LHS of
Aungier eqn 6-46

wp_min = wp_st * (1.0 + (inc_minloss - inc_star)/Rs)**2); // Aungier eqn 6-58

if (FI_IR.MNrel > Mcrit) {
    M1rel = min(1.0, FI_IR.MNrel);

    if (switchAirfoilType == "NACA65") { // Aungier page 132
        a1 = 0.0117;
    } else {
        a1 = 0.007; // default to C4 circular arc
    }

    WmaxqW1 = 1.12 + 0.61*((cos(beta_in)**2)/solidity)*(tan(beta_in)-tan(beta_out));
// Aungier eqn 6-38
    if (incidence*180./PI >= inc_star) {
        WmaxqW1 = WmaxqW1 + a1*(incidence*180./PI-inc_star)**1.43;
        // Aungier eqn 6-38
    }
    WstarqW1 = Mcrit/M1rel*WmaxqW1;
    term2 = Ksh1*((1.0 - Mcrit/M1rel)*WmaxqW1)**2;
    wp_min = wp_min + term2;
}

```

```

//-----
// Off-design loss from Aungier
//-----

    if (inc < inc_minloss) {
        xi = (inc - inc_minloss)/(inc_s - inc_minloss); } // Aungier eqn 6-78
    else {
        xi = (inc - inc_minloss) / (inc_minloss - inc_c); } // Aungier eqn 6-79
    if (xi > 1.0) {
        wp = wp_min*(2.0 + 2.0*(xi - 1.0)); } // Aungier eqn 6-82
    else if (xi < -2.0) {
        wp = wp_min*(5.0 - 4.0*(xi + 2.0)); } // Aungier eqn 6-81
    else {
        wp = wp_min*(1.0 + xi**2); } // Aungier eqn 6-80

//    wp = 0.0626;
//    lossEstimated = wp;
//    lossEstimated = min(wp,0.5); // was 0.5 RKD 10-17-2014
//    lossEstimated = 0.0;
//cout << "1 " << lossEstimated << " " << wp_min << " " << wp_st << " " << Deq << " " << beta1_st << endl;
//cout << "2 " << inc_minloss << " " << inc_star << " " << Mcrit << " " << Ksh << " " << dev_star << " " << Rs
<< endl;
    }
}

```

```

//=====
//          OTAC
//      Object-Oriented Turbomachinery Analysis Code
//
//  written by: Justin S. Gray, Scott M. Jones, and Thomas M. Lavelle
//
//=====

#include <InterpIncludes.ncp>

class EndwallLossModel extends Subelement {

//-----
//      ++++++++ DOCUMENTATION ++++++++
//-----
title = "";

description = isA()+ " is to be used as the base class for functioning
OTAC loss models. The loss model will return an estimated value for loss
based on the blade segment current conditions; the blade row solver will
enforce the condition that the loss estimated from this socket equals the
assumed, or actual, loss for the blade segment. The form of the loss is
set by the blade row attribute switchLossBasis, and may be either total
pressure loss, relative total loss parameter (omegaBar), or segment
adiabatic efficiency.
";

usageNotes = isA() + "";

socketRequired = FALSE;

addInterface("OTACLossModel");

//-----
//      ++++++++ SET UP VARIABLES ++++++++
//-----
    real lossEstimated {
        value = 0.0; IOstatus = OUTPUT; units = NONE;
        description = "estimated loss for this blade segment";
    }
    real e_c {
        value = 0.01; IOstatus = INPUT; units = NONE;
        description = "tip clearance-to-chord ratio";
    }
    real torque {
        value = 0; IOstatus = OUTPUT; units = NONE; //ft-lbf
        description = "blade torque";
    }
/*
    real chord {
        value = 0; IOstatus = INPUT; units = NONE; //in
        description = "blade clearance";
    } */
}

```



```

real radius_in {
    value = 0; IOstatus = OUTPUT; units = NONE; //in
    description = "radius of stream at inlet to bladerow";
}
real radius_out {
    value = 0; IOstatus = OUTPUT; units = NONE; //in
    description = "radius of stream at outlet to bladerow";
}
real rho_in {
    value = 0; IOstatus = OUTPUT; units = NONE; //lbm/ft^3
    description = "density of stream at inlet to bladerow";
}
real rho_out {
    value = 0; IOstatus = OUTPUT; units = NONE; //lbm/ft^3
    description = "density of stream at outlet to bladerow";
}
real rho_average {
    value = 0; IOstatus = OUTPUT; units = NONE; //lbm/ft^3
    description = "average density of stream at outlet to bladerow";
}
real Vm_in {
    value = 0; IOstatus = OUTPUT; units = NONE; //ft/s
    description = "meridinal velocity at inlet to bladerow";
}
real Vm_out {
    value = 0; IOstatus = OUTPUT; units = NONE; //ft/s
    description = "meridinal velocity at outlet to bladerow";
}
real Vtheta_in {
    value = 0; IOstatus = OUTPUT; units = NONE; //ft/s
    description = "tangential velocity at inlet to bladerow";
}
real Vtheta_out {
    value = 0; IOstatus = OUTPUT; units = NONE; //ft/s
    description = "tangential velocity at outlet to bladerow";
}
real Z {
    value = 20; IOstatus = OUTPUT; units = NONE;
    description = "numbers of blades in blade row";
}
real staggerTip {
    value = 0; IOstatus = OUTPUT; units = DEG;
    description = "stagger angle AT BLADE TIP";
}
real V_leak {
    value = 0; IOstatus = OUTPUT; units = NONE; //ft/s
    description = "leakage velocity of tip flow";
}

real W_leak {
    value = 1; IOstatus = OUTPUT; units = NONE; //lbm/s
    description = "leakage mass flow";
}

```

```

real W_stage {
    value = 1; IOstatus = OUTPUT; units = NONE; //lbm/s
    description = "mass flow across stage";
}
real dP_blade {
    value = 1; IOstatus = OUTPUT; units = NONE; //lbf/in2
    description = "overall pressure drop";
}
real dP_overall {
    value = 1; IOstatus = OUTPUT; units = NONE; //lbf/in2
    description = "overall pressure drop";
}
real dP {
    value = 1; IOstatus = OUTPUT; units = NONE; //lbf/in2
    description = "pressure drop at span location";
}

real K1_leak = 0.816;
real endloss;
real s_Re;
real V_in;
real V_out;
real Reynolds;
real chord_local;
int firstPass = TRUE;

void calculate() {

    //-----
    // Endwall loss from Aungier - Axial Flow Compressor - pg 146-147
    //-----
    //use flow properties at tip if a rotor and at hub if stator

    if (switchRotate == "ROTATING") {
        radius_in = bladeSegments[numberOfStreams-1]->FI_IR.radius;
        rho_in = bladeSegments[numberOfStreams-1]->FI_IR.rhos;
        Vm_in = bladeSegments[numberOfStreams-1]->FI_IR.Vm;
        Vtheta_in = bladeSegments[numberOfStreams-1]->FI_IR.Vtheta;
        chord_local = bladeSegments[numberOfStreams-1]->chord;

        radius_out = bladeSegments[numberOfStreams-1]->FI_OR.radius;
        rho_out = bladeSegments[numberOfStreams-1]->FI_OR.rhos;
        Vm_out = bladeSegments[numberOfStreams-1]->FI_OR.Vm;
        Vtheta_out = bladeSegments[numberOfStreams-1]->FI_OR.Vtheta;
        staggerTip = bladeSegments[numberOfStreams-1]->stagger;
    } else {
        radius_in = bladeSegments[0]->FI_IR.radius;
        rho_in = bladeSegments[0]->FI_IR.rhos;
        Vm_in = bladeSegments[0]->FI_IR.Vm;
        Vtheta_in = bladeSegments[0]->FI_IR.Vtheta;
        chord_local = bladeSegments[0]->chord;
    }
}

```

```

        radius_out = bladeSegments[0]->Fl_OR.radius;
        rho_out = bladeSegments[0]->Fl_OR.rhos;
        Vm_out = bladeSegments[0]->Fl_OR.Vm;
        Vtheta_out = bladeSegments[0]->Fl_OR.Vtheta;
        staggerTip = bladeSegments[0]->stagger;
    }

    W_stage = 0;
    int i;

    i = 0;
    for( i=0; i < numberOfStreams; ++i ) {
        W_stage = W_stage + bladeSegments[i]->Fl_OR.W;
    }

    rho_average = ( rho_in + rho_out ) / 2.;

    torque = PI*( clearance /12. )*( ( radius_in* rho_in * Vm_in /12. ) + ( radius_out*
rho_out * Vm_out /12. ) ) * abs( ( radius_out* Vtheta_out /12. ) - ( radius_in* Vtheta_in /12. ) ) /
32.174; // Aungier eqn 6-85

    dP_blade = 12. * torque / (numberOfBlades * radius_in * clearance * chord_local * cos
(staggerTip) ); // Aungier eqn 6-86
    //fixme n_row need to determine easy fix is a stage entering term
    V_leak = K1_leak * sqrt( 144.* 32.174 * 2.0 * dP_blade / rho_average ) / ( N_row**0.2 );
    // Aungier eqn 6-87

    W_leak = rho_average * V_leak * numberOfBlades * clearance * chord_local * cos (
staggerTip) / 144.; // Aungier eqn 6-88

    dP_overall = dP_blade*W_leak/W_stage; // Aungier eqn 6-89
    real m;
    real b;

    //linearly vary dp across the span to make integrated dP equal that of the dP of the last -
see Aungier page 147
    //check this - I did this differently the first time - see commented logic
    real dPtip = (2.0*dP_overall); /*(bladeSegments[numberOfStreams-1]->Fl_OR.radius -
bladeSegments[numberOfStreams-1]->Fl_OR.radius) / (bladeSegments[numberOfStreams-1]-
>Fl_OR.radius - bladeSegments[0]->radiusInnerBlk);

    if (numberOfStreams > 1) {
        real ri = (bladeSegments[0]->Fl_OR.radius + bladeSegments[0]->Fl_IR.radius)/2;
        real ro = (bladeSegments[numberOfStreams-1]->Fl_OR.radius +
bladeSegments[numberOfStreams-1]->Fl_IR.radius)/2;
        m = (Fl_OR.radius + Fl_IR.radius)/2.;
        if (switchRotate == "ROTATING") {
            b = (m - ri)/(ro-ri);
            b = min(max(b,0.),1.);
        } else {
            b = (ro - m)/(ro-ri);
        }
        dP = dPtip* b;
    }

```

```

        //cout<<"b = "<<b<<endl;
    }
    else {
        dP = dP_overall;
    }

    //cout<<"dP = "<<dP<<endl;
    //fixme - Pt delta is absolute. Need to convert to relative.
    if ( parent.switchLossBasis == "PTOT" ) {
        lossEstimated = min(dP/(FI_IR.Pt - FI_IR.Ps),0.5);
    }
    else if ( parent.switchLossBasis == "PTOTREL" ) {
        lossEstimated = min(dP / (FI_IR.PtRel - FI_IR.Ps),0.5);
    }

    //cout << " endwall " << N_row << " " << iPass << " " << b << " " <<dP << " " << lossEstimated << " " <<
    dP_blade << " " << dP_overall << " " << dPtip << endl;
    //        lossEstimated = 0.0;

        //        cout<<"lossEstimated = "<<lossEstimated<<endl;

    if (firstPass) { // ignore loss calculation until all streams have been calculated
        firstPass = FALSE;
        lossEstimated = 0.0;
    }

    } // end calculate
}

```

```

//=====
//          OTAC
//      Object-Oriented Turbomachinery Analysis Code
//
//  written by: Justin S. Gray, Scott M. Jones, and Thomas M. Lavelle
//
//=====

#include <InterpIncludes.ncp>

class ShockLossModel extends Subelement {

//-----
//      ++++++++ DOCUMENTATION ++++++++
//-----
title = "";

description = isA()+ " is to be used as the base class for functioning
OTAC loss models. The loss model will return an estimated value for loss
based on the blade segment current conditions; the blade row solver will
enforce the condition that the loss estimated from this socket equals the
assumed, or actual, loss for the blade segment. The form of the loss is
set by the blade row attribute switchLossBasis, and may be either total
pressure loss, relative total loss parameter (omegaBar), or segment
adiabatic efficiency.
";

usageNotes = isA() + "";

addInterface("OTACLossModel");
socketRequired = FALSE;

//-----
//      ++++++++ SET UP VARIABLES ++++++++
//-----
real lossEstimated {
  value = 0.0; IOstatus = OUTPUT; units = NONE;
  description = "estimated loss for this blade segment";
}
real rLE_c {
  value = 0.00687; IOstatus = INPUT; units = NONE;
  description = "blade LE radius-to-chord ratio";
}

real w_shock;
real inc;
real nu1;
real nu2;
real solidityp;
real sigma;
real xnorm;
real phi_s;

```

```

real xi_c;
real Ptr_shock;
real Mpass;
real M2;
real kappa1; // blade metal angle at inlet
real kappa2; // blade metal angle at exit

real NormalShockPtRatio(real M1) {
  real PtRatio;
  real gam = 1.4;
  real M2 = sqrt(((gam-1.)*M1**2+2.)/(2.*gam*M1**2-(gam-1.)));
  PtRatio = ((1.+(gam-1.)/2.*M2**2)/(1.+(gam-1.)/2.*M1**2))**(gam/(gam-1.))*(2.*gam*M1**2-
(gam-1.))/(gam+1.);
  return PtRatio;
}

real PrandtlMeyerAngle (real M) {
  real nu;
  real gam = 1.4;
  nu = sqrt((gam+1.)/(gam-1.))*atan(sqrt((gam-1.)*(M**2-1.)/(gam+1.))) - atan(sqrt(M**2-1.));
  return nu;
}

real t_c {
  value = 0.08; IOstatus = OUTPUT; units = NONE;
  description = "blade max thickness-to-chord ratio";
}

real camber_up {
  value =0; IOstatus = OUTPUT; units = NONE;
  description = "Upper surface camber angle";
}

SecantSolver iterMn {
  description = "solver for Prandtl-Meyer expansion";
  maxDx = 5;
  tolerance =1.E-5;
  perturbSize = 0.01;
  maxIters = 50;
}

  real camberDeg;
  real gam;
  real aa;
  real circ_param;
  real bb;
  real cc;
  real dd;
  real ee;
  real ff;
  real Vmaxq1;
  real delPq;
  real delPq1;

```

```

//      real Mcrit;
//      real M1rel;
//      real err1;
//      real psi;
//      real pitch;
//      real Ru;
//      int firstPass = TRUE;

void calculate() {

    kappa1 = bladeInletAngle;
    kappa2 = bladeExitAngle;

    t_c = tonc; // thickness-to-chord from parent

    M1rel = min(Mcrit,FI_IR.MNrel);

    if (FI_IR.MNrel < 1) {
        w_shock = 0.0; //the loss between Mcritical and 1 is bookkept in the profile loss - per Aungier
    }
    else {

//-----
// Prandtl Meyer Angle calculation and estimate surface Mach number
//-----
//      M1_loc = min(FI_IR.MNrel, 1.6);
//      nu1 = PrandtlMeyerAngle(FI_IR.MNrel);

        solidityp = solidity * (1.-2.*rLE_c);
        sigma = atan(sin(kappa1)*sin(camber/2.)/(solidityp + cos(kappa1)*sin(camber/2.)));
        xnorm = 1. + sin(2.*sigma)/tan(camber/2.) - cos(2.*sigma);
        phi_s = 2.*atan((1.-cos(camber/2.))/sin(camber/2.) + (t_c - 2.*rLE_c)/(1. - 2.*rLE_c));
        xi_c = asin(sin(phi_s/2.)*(xnorm - 1.));
        nu2 = nu1 + inc + camber/2. + xi_c;

//-----
// Aungier shock geometry calculation
        camber_up = 4.*atan(tan(camber/4.) + t_c);
        psi = 90. - (camber_up*180./Pi/2.) - (kappa1*180/Pi) + (camber*180./Pi/2.); //equation 6-68
        //cout<<"psi degrees = "<<psi<<endl;
        psi = psi*Pi/180.; //convert to radians
        pitch = chord/solidity;
//      Ru = chord*sin(camber_up/2.)/2.; //equation 6-61
//      Ru = chord/sin(camber_up/2.)/2.; //equation 6-61
        phi_s = atan((pitch*cos(psi))/(pitch*sin(psi) + Ru));
        nu2 = nu1 + phi_s;

//-----

        // iterate to find M2 given nu2 (P-M expansion)
//      M2 = 1.546448;
//      real MnOut = 1.5;
//      real nu = 0.0;
        iterMn.initialize( MnOut );
        do {

```

```

nu = PrandtlMeyerAngle(MnOut);
MnOut = iterMn.iterate( nu - nu2 );
} while ( !( iterMn.isConverged() ) && !( iterMn.errorType ) );

M2 = MnOut;

//-----
// Use average Mach number to estimate the pressure loss across normal shock
//-----
Mpass = sqrt(FI_IR.MNrel*M2);           //Aungier 6-71
Ptr_shock = NormalShockPtRatio(Mpass);

if ( parent.switchLossBasis == "PTOT" ) {
    //w_shock = FI_IR.PtRel*(1.-Ptr_shock) / (1. - FI_IR.Pt/FI_IR.PtRel);
    w_shock = min(FI_IR.PtRel*(1.-Ptr_shock)/(FI_IR.Pt - FI_IR.Ps),0.5);
}
else if ( parent.switchLossBasis == "PTOTREL" ) {
    w_shock = min(FI_IR.PtRel*(1.-Ptr_shock) / (FI_IR.PtRel - FI_IR.Ps),0.5);
}

} // end else

lossEstimated = w_shock;
// lossEstimated = 0.0;
// cout << " shock loss " << FI_IR.MNrel << " " << lossEstimated << " " << Ptr_shock << " " << Mpass
<< " " << nu2 << " " << phi_s << " " << nu1 << " " << camber << " " << Mcrit << " " << M1rel << " " <<
camber_up << endl;
// cout << "shock loss = " << lossEstimated << endl;
}
}

```



```

//=====
//          OTAC
//      Object-Oriented Turbomachinery Analysis Code
//
//                      Stall Socket written by: Russ Denney
//
//=====

#include <InterpIncludes.ncp>

class Stall extends Subelement {

real AR; // diffuser area ratio
real VR; // diffuser velocity ratio
real Lqb1; // diffuser length to width ratio
real WRElimit; // limiting equivalent relative velocity ratio
real WREact; // actual equivalent relative velocity ratio
real t_c; // thickness to chord
real Deq; // equivalent diffusion factor
real deHaller; // deHaller number
real wmaxqw1; // airfoil surface velocity ratio Wmax/W1
real a1 = 0.0117; // value for NACA 65-series airfoils (Aungier p. 132)
real k_Stall = 1.05; // fudge factor on stall limit
real beta_in;
real beta_out;
real inc;

void calculate() {
    inc = incidence*180./PI;
    if (parent.parent.switchRotate == "NON_ROTATING") {
        beta_in = abs(FI_IR.alpha);
    } else {
        beta_in = abs(FI_IR.beta);
    }
    beta_out = beta_in - deflection;
    deHaller = FI_OR.Vrel / FI_IR.Vrel; // deHaller number
    if (switchAirfoilType == "NACA65") { // Aungier page 132
        a1 = 0.0117;
    } else {
        a1 = 0.007; // default to C4 circular arc
    }
    if (inc > inc_star) {
        wmaxqw1 = 1.12 + 0.61*( (cos(beta_in)**2)/solidity)*(tan(abs(beta_in)) - tan(abs(beta_out))) +
a1*(inc-inc_star)**1.43; // Aungier eqn 6-38
    } else {
        wmaxqw1 = 1.12 + 0.61*( (cos(beta_in)**2)/solidity)*(tan(abs(beta_in)) - tan(abs(beta_out)));
    }
//    Deq = wmaxqw1/deHaller; // Aungier 6-30
    Deq = wmaxqw1*cos(beta_out)*FI_IR.Vm/cos(beta_in)/FI_OR.Vm; // Aungier eqn 6-
36

    Lqb1 = camber*solidity/(2.*sin(camber/2.)*cos(stagger)); // Aungier 9-15
    Lqb1 = max(Lqb1, 1.1); // Aungier 9-18
    AR = 1. + 0.4*(Lqb1)**0.65; // Aungier 9-13
}
}

```

```

VR = 1./AR; // Aungier 9-14
t_c = tonc;
WRElimit = (0.15+11.0*t_c)/(0.25+10.0*t_c)*VR; // Aungier 9-17
if (Deq >= 2.2) {
    WRElimit = ((2.2/Deq)**0.6)*WRElimit; // Aungier 9-19
}
WRElimit = WRElimit * k_Stall; // added RKD 5-25-2014
WREact = sqrt((FI_OR.PtRel - FI_OR.Ps)/(FI_IR.PtRel - FI_IR.Ps)); // Aungier 9-16

if (WREact < WRElimit) {
    stall = 1;
}
else {
    stall = 0;
}

}

} //End Stall Socket

//-----
// Check Aungier Stall Criteria
//-----
int ij = 0;
int jj = 0;
string name;
string stageRotor;
string stageStator;
string stall_segments1;
string stall_segments3;
int stall_count = 0;
int bladeStall = 0;
real cmpLength = 0;

void checkStall() {

    stall_count = 0;
    stall_segments1 = "";
    stall_segments3 = "";

    //-----
    // Criterion 1: Discharge (Overall) PR slope near zero
    //-----
    if (PR_out < PR_old) {
        stall_segments1 = stall_segments1 + "crit1 overallPR,";
        stall_flag = 1;
    }
    PR_old = PR_out;

    //-----
    // Criterion 3: StallSocket_Aungier.int
    //-----
    string BladeRows[] = list("BladeRow", TRUE);

```

```

for (ij=0; ij < BladeRows.entries(); ++ij) {
    bladeStall = 0;
    for( jj=0; jj < OTACdefaults.numberOfStreams; ++jj ) {
        name = BladeRows[ij] + ".bladeSegment_" + toStr(jj+1);
        if (name->stall == 1) {
            stall_segments3 = stall_segments3 + name + " crit3,";
            stall_count = stall_count + 1;
            bladeStall = bladeStall + 1;
        }
    }

    // 40% of one blade must be stalled
    if (bladeStall/OTACdefaults.numberOfStreams >= .4) {
        stall_flag = 1;
        //out << "blade stall on " << BladeRows[ij] << endl;
    }

    // Calculate length of the compressor
    jj = OTACdefaults.midStream;
    name = BladeRows[ij] + ".bladeSegment_" + toStr(jj+1);
    cmpLength = cmpLength + name->chord_in * cos(name->stagger);
    //cout << cmpLength << endl;
}

// 10% of machine must be stalled
if (stall_count/(OTACdefaults.numberOfStreams*BladeRows.entries()) >= .1) {
    stall_flag = 1;
}
else {
    stall_count = 0;
}

/*if (loss_out < minloss_out) {
    minloss_out = loss_out;
}
cout << "minloss_out= " << minloss_out <<endl;
cout << "loss_out = " << loss_out <<endl;
if (loss_out > 1.1 * minloss_out) {
    stall_segments1 = "loss crit";
    stall_flag = 1;
}*/
}

```

APPENDIX D: MAP GENERATION CODE

```
// *****
// * Analysis.run
// * Modeled by: Andrew Miller
// * based on code by Russell Denney
// * Aerospace Systems Design Laboratory
// * Georgia Institute of Technology
// * Atlanta, GA 30332
// * April 7, 2015
// *****

//-----
//      SET THERMO PACKAGE
//-----

setThermoPackage( "GasTbl" );

//-----
//      INCLUDE OTAC INTERPRETED ELEMENTS AND FILES
//-----
#include <elements\OTAC.fnc>
#include <elements\OTACstart.int>
#include <elements\Expander.int>
#include <elements\Reducer.int>
#include <elements\BladeRowASDL.int>
#include <elements\BladeSegmentASDL.int>
#include <elements\OTACPerfStationAvg.int>
#include <elements\OTACPerfStageASDL.int>
#include <elements\OTACPerfMachine.int>
//-----
//      INCLUDE LOSS MODEL ELEMENTS AND FILES
//-----
#include <elements\AirfoilASDL.int>
#include <elements\DeviationASDL.int>
#include <elements\OTACLossModel_ProfileNew_Aungier.int>
#include <elements\OTACLossModel_EndwallAungier.int>
#include <elements\OTACLossModel_ShockAungier.int>
#include <elements\StallSocket_Aungier.int>
//#include <elements\Blockage.int>
//-----
//      INCLUDE OTHER FILES
//-----
#include <elements\print_macros.fnc>
```

```

#include <elements\solver_macros.fnc>
#include <view\OTACpage4.view>
pvStream.filename = "out/Param_4stg_Map.out";
#include <view\rows.view>

OTACdefaults {
    numberOfStreams = 3;
}
OTACdefaults.calculate();

//-----
// Model of Compressor
//-----
#include <models\ParametricModel.int>
//#include <models\ParametricNACA4Stage.mdl>
//#include <models\Parametric5Stage.mdl>
//#include <models\NACA_5Stage.mdl>

//-----
// Add Performance Element
//-----
PERF {}

#include <view\MapData.view>
#include <elements\checkStall.int>
#include <elements\Triangle.int>
#include <elements\BigOutput2.int>
#include <elements\saveGeometry.int>
#include <elements\runSpeedline.int>
real step_size = 0.1;
int stall_flag = 0;
real PR_old = 0;
int solverCount;
    OutFileStream DOEout;
    DOEout.open("out\\DOEtempout.int");

//-----
// Isentropic FlowStation for Efficiency Calculation
//-----
FlowStation OutIsen {
    description = "Fictitious isentropic flowstation";
}

//-----
// Turbomachinery Analysis, not Cycle Analysis
//-----

```

```

OTACenable();

//-----
//          RUN THE MODEL DESIGN POINT
//-----
setOption( "switchDes", "DESIGN" );
autoSolverSetup();
solver.removeDependent( "rotorShaft.integrate_Nmech" );
solver.solutionMode = "ONE_PASS";

run();

real PR_out = (reducer.Fl_O.Pt/expander.Fl_I.Pt);
checkStall();
if(stall_flag == 0){
    cout << "point works!" << endl;
    solver.solutionMode = "STEADY_STATE";
    run();
}

//printPride();
OTACpage.display();
//PERF.print_perf();
Triangles();

PR_out = (reducer.Fl_O.Pt/expander.Fl_I.Pt);
OutIsen.setTotalSP(expander.Fl_I.entropy,reducer.Fl_O.Pt);
real eff_out = (OutIsen.ht-expander.Fl_I.ht)/(reducer.Fl_O.ht-expander.Fl_I.ht);
real loss_out = (reducer.Fl_O.ht-expander.Fl_I.ht)*(1-eff_out);

cout<<"---- Design Point ----"<<endl;
    cout << "    W = " << start.W << " lb/s \n";
    cout << "    N = " <<rotorShaft.Nmech<< " RPM \n";
    cout << "    PR = " << PR_out << " \n";
    cout << "    eff = " << eff_out << " \n";
    cout<<endl;
    cout<<endl;

BigOutHeadings();
BigOutData();
PR_old = 0;
checkStall();
cout << "stalled? " << stall_flag << endl;
    DOEout << rtip_in << " " << cmpLength << " " << rhqrt_out << " " << PR_out << " " <<
eff_out << " ";
if (stall_flag == 1){
    DOEout << "0 0 ";
}

```

```

//-----
//      SAVE GEOMETRY FOR OFF-DESIGN
//-----
saveGeometry();

//-----
//      REPEAT DESIGN POINT IN OFF-DESIGN MODE
//-----

setOption( "switchDes", "OFFDESIGN" );
autoSolverSetup();
solver.removeIndependent( "rotorShaft.ind_Nmech" );
solver.removeDependent( "rotorShaft.integrate_Nmech" );

run();

OTACpage.display();
//printPride();
//printPrideOTAC();
//saveIndependents();

PR_out = (reducer.Fl_O.Pt/expander.Fl_I.Pt);
OutIsen.setTotalSP(expander.Fl_I.entropy,reducer.Fl_O.Pt);
eff_out = (OutIsen.ht-expander.Fl_I.ht)/(reducer.Fl_O.ht-expander.Fl_I.ht);
loss_out = (reducer.Fl_O.ht-expander.Fl_I.ht)*(1-eff_out);

cout<<"---- Off-Design Point ----"<<endl;
    cout << "    W = " << start.W << " lb/s \n";
    cout << "    N = " << rotorShaft.Nmech<< " RPM \n";
    cout << "    PR = " << PR_out << " \n";
    cout << "    eff = " << eff_out << " \n";
    cout<<endl;
    cout<<endl;

BigOutData();
checkStall();
//quit();

//-----
//      RUN COMPLETE MAP
//-----

cout << " ***** Running Map ***** " << endl;

real startFlow;
real stallFlow;
real chokeFlow;

```

```

// Set Speed Lines for Maps
real TB_Speed [] = { 12605., 11975., 11344., 10084., 8824.};//, 7563., 6302., 5042. };

// Set Starting Flows for each Speed Line with Design vane angles
Table TB_startW( real PS ) {
  PS = { 12605., 11975., 11344., 10084., 8824.};//, 7563., 6302., 5042. }
  y = { 67.5, 64., 58., 52., 30.};//, 25., 20., 10. }
// y = { 67.5, 60., 55., 45., 35., 25., 20., 10. }
}

// Set Choked Flows for each Speed Line with Design vane angles
Table TB_chokeStop( real PS ) {
  PS = { 12605., 11975., 11344., 10084., 8824. }
  y = { 69.5, 64.5, 61.85, 51.35, 70. }
// y = { 68.45, 64.4, 59.5, 48.2, 38.35, 30.8, 24.9, 19.4 }
}

int iSpeed;
//-----
// Loop through Speed Lines
//-----
for (iSpeed = 2; iSpeed < TB_Speed.entries(); iSpeed++) {
  rotorShaft.Nmech = TB_Speed[iSpeed];
  startFlow = TB_startW(rotorShaft.Nmech);
  chokeFlow = TB_chokeStop(rotorShaft.Nmech);

runSpeedline();

}

//-----
// function to run the speedlines for the map
//-----

//Define step size on map
real step_back = 0.1; //Flow increment to step towards stall
real step_forward = 0.05; //Flow increment to step towards choke
int step;
real speed;
//int stall_flag = 0;

//int bladeStall;
real minloss_out = 1000;

string BladeRows[];
//real PR_old;

```



```

void runSpeedline() {
    cout << " ***** Running Speedline ***** " << rotorShaft.Nmech << endl;
    start.W = startFlow;

    //-----
    // Decrease Flow to Stall
    //-----
    stall_flag = 0;
    step = 0;
    PR_old = 0;
    while (stall_flag == 0){
        cout<<"----Decreasing flow----"<<endl;

        start.W = startFlow-step*step_back;

        run();
        PR_out = (reducer.Fl_O.Pt/expander.Fl_I.Pt);
        OutIsen.setTotalSP(expander.Fl_I.entropy,reducer.Fl_O.Pt);
        eff_out = (OutIsen.ht-expander.Fl_I.ht)/(reducer.Fl_O.ht-expander.Fl_I.ht);
        loss_out = (reducer.Fl_O.ht-expander.Fl_I.ht)*(1-eff_out);

        //-----
        // Check Aungier Stall Criteria
        //-----
        checkStall();

        //-----
        // Outputs
        //-----
        cout << "    N = " << rotorShaft.Nmech << " RPM \n";
        cout << "    W = " << start.W << " lb/s \n";
        cout << "    PR = " << PR_out << " \n";
        cout << "    eff = " << eff_out << " \n";
        cout << stall_segments3 << stall_segments1 << endl;

        mapOut.update();
        BigOutData();
        // stageCalc();
        // checkLimits();

        // for next iteration
        stall_segments1 = "";
        stall_segments3 = "";
        step++;
    }
    minloss_out = 2*minloss_out;
}

```

```

while (step > 1) {
    start.W = startFlow-step*step_back;
    run();
    step--;
}

//-----
// Increase Flow to Choke
//-----
step = 0;

while (eff_out > 0 && start.W < chokeFlow) {
    cout<<"----Increasing flow-----"<<endl;

    start.W = startFlow + step*step_forward;
    run();

    PR_out = (reducer.Fl_O.Pt/expander.Fl_I.Pt);
    OutIsen.setTotalSP(expander.Fl_I.entropy,reducer.Fl_O.Pt);
    eff_out = (OutIsen.ht-expander.Fl_I.ht)/(reducer.Fl_O.ht-expander.Fl_I.ht);
    loss_out = (reducer.Fl_O.ht-expander.Fl_I.ht)*(1-eff_out);

    // Outputs
    cout << "   N = " << rotorShaft.Nmech << " RPM \n";
    cout << "   W = " << start.W << " lb/s \n";
    cout << "   PR = " << PR_out << " \n";
    cout << "   eff = " << eff_out << " \n";

    mapOut.update();
    BigOutData();

    // for next iteration
    pctFlow = pctFlow + step_forward;
    step++;
}

// Step back to startFlow to help convergence
while (step > 1){
    start.W = startFlow - 2*step*step_forward;
    run();
    step--;
}
}

```

REFERENCES

- [1] Schutte, J.S., SIMULTANEOUS MULTI-DESIGN POINT APPROACH TO GAS TURBINE ON-DESIGN CYCLE ANALYSIS FOR AIRCRAFT ENGINES, Georgia Institute of Technology, May 2009.
- [2] Bolemant, M., and Peitsch, D., "An Alternative Compressor Modeling Method Within Gas Turbine Performance Simulations," DocumentID: 340047, Deutscher Luft- und Raumfahrtkongress 2014.
- [3] Follen, G., and auBuchon, M., "Numerical Zooming Between a NPSS Engine System Simulation and a One-Dimensional High Compressor Analysis Code," NASA/TM-2000-209913, April 2000.
- [4] Smith, Jr., L.H., "Axial Compressor Aerodesign Evolution at General Electric," ASME Journal of Turbomachinery, Vol. 124, pp. 321-330, July 2002.
- [5] Hendricks, E.S., A MULTI-LEVEL MULTI-DESIGN POINT APPROACH FOR GAS TURBINE CYCLE AND TURBINE CONCEPTUAL DESIGN, (Unpublished) Georgia Institute of Technology, November 2014.
- [6] "Aircraft Propulsion System Performance Station Designation and Nomenclature," in Aerospace Recommended Practice (ARP) 755B, Society of Automotive Engineers, 1994.
- [7] Gallimore, S.J., "Axial flow compressor design," Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 1999 213:437, May, 1999.
- [8] Cumpsty, N., Jet Propulsion, 2nd ed., Cambridge University Press, 2003.
- [9] Fowler, T.W., Jet Engines and Propulsion Systems for Engineers, Chapter 1, GE Aircraft Engines, 1989.
- [10] Fowler, T.W., Jet Engines and Propulsion Systems for Engineers, Chapter 3, GE Aircraft Engines, 1989.
- [11] Jones, S.M., "An Introduction to Thermodynamic Performance Analysis of Aircraft Gas Turbine Engine Cycles Using the Numerical Propulsion System Simulation Code," NASA/TM-2007-214690, March 2007.

- [12] Reed, J. A., and Afjeh, A. A., "An Interactive Graphical system for Engine Component Zooming in a Numerical Propulsion System Simulation," AIAA 95-0118, January 1995.
- [13] Aungier, R.H., Axial-Flow Compressors: A Strategy for Aerodynamic Design and Analysis, ASME Press, 2003.
- [14] Koch, C.C., and Smith, Jr., L.H., "Loss Sources and Magnitudes in Axial-Flow Compressors," ASME Journal of Engineering Power, July 1976.
- [15] Molinari, M., and Dawes, W. N., "Review of evolution of compressor design process and future perspectives," IMechE Vol. 220 Part C: JMES 298, March 2006.
- [16] Day, I.J., "Axial Compressor Performance During Surge," AIAA DOI: 10.2514/3.23760, 1991.
- [17] Miller, A. S., Denney, R. K., and Tai, J. C., "Compressor Stall Criteria Using Object-Oriented Turbomachinery Analysis Code (OTAC)," ASME GT2015-43680, June 2015.
- [18] Gill, A., A Comparison Between Stall Prediction Models for Axial Flow Compressors, Stellenbosch University, April 2006.
- [19] Sandercock, D.M., Kovach, K., and Lieblein, S., Experimental Investigation of a Five-Stage Axial Flow Research Compressor with Transonic Rotors in All Stages, Vol I – Compressor Design, NACA RM E54F24, 1954.
- [20] Kovach, K. and Sandercock, D.M., Experimental Investigation of a Five-Stage Axial Flow Research Compressor with Transonic Rotors in All Stages, Vol II – Compressor Overall Performance, NACA RM E54G01, 1954.
- [21] Sandercock, D.M., and Kovach, K., Experimental Investigation of a Five-Stage Axial Flow Research Compressor with Transonic Rotors in All Stages, Vol III – Interstage Data and Individual Stage Performance Characteristics, NACA RM E56G24, 1956.
- [22] Sandercock, D.M., Experimental Investigation of a Five-Stage Axial Flow Research Compressor with Transonic Rotors in All Stages, Vol IV – Blade Element Performance, NACA RM E57B12, 1957.
- [23] Kovach, K., and Sandercock, D.M., Aerodynamic Design and Performance of Five-Stage Transonic Axial-Flow Compressor, Journal of Engineering for Power, July 1961.

- [24] Farokhi, S., Aircraft Propulsion, John Wiley & Sons, New York, 2009.
- [25] Cohen, H., Rogers, G.F.C., and Saravanamuttoo, H.I.H., Gas Turbine Theory, 3rd ed., John Wiley & Sons, New York, 1987.
- [26] Wilson, D.G., and Korakianitis, T., The Design of High-Efficiency Turbomachinery and Gas Turbines, 2nd ed, Prentice Hall, New Jersey, 1998.
- [27] Denney, R., Kestner, B., and Mavris, D.N., "Compressor Modeling and Simulation Using Object-Oriented Turbomachinery Analysis Code (OTAC)," AIAA 2014-3927.
- [28] Nichols, L.D., and Chamis, C.C., "Numerical Propulsion System Simulation: An Interdisciplinary Approach," AIAA 1991-3554.
- [29] Jones, S.M., "Development of an Object-Oriented Turbomachinery Analysis Code Within the NPSS Framework", NASA TM 2014-216621.
- [30] Kulkarni, S., Development of a Methodology to Estimate Aero-Performance and Aero-Operability Limits of a Multistage Axial Flow Compressor For Use in Preliminary Design, Case Western University, January 2012.
- [31] Johnsen, I.A., and Bullock, R.O., Aerodynamic Design of Axial-Flow Compressors, NASA SP-36, 1965.
- [32] Mavris, D., "Introduction to Decision Making and Multi-attribute Decision Making (MADM)", AE 6373 Lecture Notes, Georgia Institute of Technology, Atlanta, GA, 2014 (unpublished).
- [33] Brandt, D.E., and Wesorick, R.R., "GE Gas Turbine Design Philosophy", GE Industrial & Power Systems, Schenectady, NY.