

1-1-2013

## Modeling and Simulation Study of A Dynamic Gas Turbine System In A Virtual Test Bed Environment

Eshwarprasad Thirunavukarasu  
*University of South Carolina*

Follow this and additional works at: <https://scholarcommons.sc.edu/etd>



Part of the [Mechanical Engineering Commons](#)

---

### Recommended Citation

Thirunavukarasu, E.(2013). *Modeling and Simulation Study of A Dynamic Gas Turbine System In A Virtual Test Bed Environment*. (Master's thesis). Retrieved from <https://scholarcommons.sc.edu/etd/2254>

This Open Access Thesis is brought to you by Scholar Commons. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact [digres@mailbox.sc.edu](mailto:digres@mailbox.sc.edu).

MODELING AND SIMULATION STUDY OF A DYNAMIC GAS TURBINE SYSTEM  
IN A VIRTUAL TEST BED ENVIRONMENT

by

Eshwarprasad Thirunavukarasu

Bachelor of Engineering  
Anna University, 2010

---

Submitted in Partial Fulfillment of the Requirements

For the Degree of Master of Science in

Mechanical Engineering

College of Engineering and Computing

University of South Carolina

2013

Accepted by:

Dr. Jamil A. Khan, Advisor

Dr. Chen Li, Committee Member

Lacy Ford, Vice Provost and Dean of Graduate Studies

© Copyright by Eshwarprasad Thirunavukarasu, Spring 2013  
All Rights Reserved.

## DEDICATION

This work is dedicated to my parents Dad: Thirunavukarasu, Mom: Mahalakshmi; my dear sisters: Vijayashree Saravanan, Priya Ravi and my Grand Mom Sakunthala.

## **ACKNOWLEDGEMENTS**

Words cannot be true replicates of author feeling of gratitude towards to his adviser Dr. Jamil A.Khan, Professor and Chair of Mechanical Engineering at University of South Carolina, Columbia, who was instrumental in bringing out the best in me, through this research work. The author is very fortunate to have him as his guide and also deeply indebted to him for the financial aid he has provided to do this research studies. It is a great privilege and honor for the author to work with him.

The author would like to thank his defense committee member Dr. Chen Li for his comments, suggestions and time for reviewing this work. Special thanks to Dr. Roger A. Dougal, Professor and Chair Department of Electrical Engineering at University of South Carolina, Columbia for his guidance, insightful suggestion and also setting up work platform to perform a coordinated research study. This investigation is financially supported by Office of Naval Research under ESRDC consortium.

The author is grateful and wishes to express his sincere thanks to Dr. Ruxian Fang and Mr. R. Leonard from VTB Lab for their valuable help and guidance at every level of this research work.

The author deepest thanks goes to his dearest parents, for their hard work and sacrifices they did, to provide the author with best education. I extend my thanks to my lovable sisters for taking care of me and finally wish to thank my friends.

## **ABSTRACT**

Gas Turbine is a complex system and highly non linear in its overall performance. For power generation applications, it is essential to develop a reliable gas turbine model for simulating the impact on electric power generated under various load conditions. This research focuses on development of a dynamic gas turbine model to simulate both single shaft and twin shaft engines. The model is developed on a virtual test bed platform which is an advanced dynamic multidisciplinary simulation environment. The modeling approach starts by developing mathematical models for individual components of gas turbines based on the thermodynamic laws and is coupled together based on the Brayton-gas turbine cycle. Specifically, the compressor and turbine components are represented by manufacturer field test data and utilization of this data has increased the effectiveness of this simulation model. The developed gas turbine model is validated for the design, off-design and transient cases with available data from the literatures. Following the validation, the gas turbine model is applied to a cross-disciplinary co-simulation study. This is done by integrating the gas turbine model with a power generation and distribution system, and a thermal system. The purpose is to investigate the dynamic potential interaction that exists between the operation of the gas turbine engine and the electrical and thermal systems. Finally a variable speed parametric study is performed utilizing the developed gas turbine model. This study is done to demonstrate the opportunities available to improve part load efficiency of gas turbine, when it is operated under variable speed. Comparison of variable speed operation results of single

shaft and twin shaft gas turbine engine show that the efficiency increases as load decreases and the improvement is larger for single-shaft engines than for twin-shaft engines.

## TABLE OF CONTENTS

DEDICATION .....	iii
ACKNOWLEDGEMENTS.....	iv
ABSTRACT .....	v
LIST OF TABLES .....	ix
LIST OF FIGURES .....	x
LIST OF SYMBOLS .....	xii
CHAPTER 1 INTRODUCTION.....	1
1.1 MOTIVATION AND OBJECTIVES .....	2
1.2 THESIS STRUCTURE.....	4
CHAPTER 2 LITERATURE .....	6
2.1 THERMODYNAMIC PRINCIPLE OF GAS TURBINE .....	6
2.2 GAS TURBINE ENGINES.....	9
2.3 GAS TURBINE MODELING .....	11
2.4 VIRTUAL TEST BED ENVIRONMENT .....	25
2.5 CLOSURE.....	27
CHAPTER 3 MATHEMATICAL MODELING OF GAS TURBINE COMPONENTS .....	28
3.1 COMPRESSOR MODELING.....	28
3.2 COMBUSTOR MODELING .....	30
3.3 TURBINE MODELING .....	31
3.4 SHAFT MODELING.....	33



3.5 FUEL PUMP MODELING .....	34
3.6 INLET/EXHAUST MODELING.....	34
3.7 CLOSURE.....	34
CHAPTER 4 GAS TURBINE DYNAMIC MODELING AND ASSEMBLY .....	35
4.1 VTB RESISTIVE COMPANION FORM REPRESENTATION.....	35
4.2 OFF DESIGN FLOWCHART FOR VTB ARCHITECTURE.....	36
4.3 TRANSIENT FLOW CHART .....	39
4.4 MODEL ASSEMBLY-VTB .....	40
CHAPTER 5 INTEGRATION OF GAS TURBINE ENGINE WITH	
ELECTRICAL AND THERMAL SYSTEM.....	42
5.1 INTRODUCTION.....	42
5.2 CO-SIMULATION MODELING .....	44
CHAPTER 6 RESULT AND DISCUSSION .....	48
6.1 RESULTS OF VTB TWIN SHAFT GAS TURBINE MODEL.....	48
6.2 RESULTS OF GAS TURBINE DYNAMIC INTERACTION	
WITH ELECTRICAL AND THERMAL SYSTEM .....	57
6.3 RESULTS OF VARIABLE SPEED OPERATION STUDY OF GAS TURBINE .....	65
CHAPTER 7 CONCLUSIONS AND FUTURE SCOPE .....	73
REFERENCES .....	75
APPENDIX A –VTB CODE OF GAS TURBINE MODEL .....	80

## **LIST OF TABLES**

Table 6.1 Working parameters at design point .....	50
Table 6.2 Design point validation .....	51
Table 6.3 Transient boundary conditions.....	54
Table 6.4 Gas turbine operating conditions .....	59
Table 6.5 Thermal plant parameters and operating condition .....	60
Table 6.6 Gas turbine steady operation results .....	61
Table 6.7 PCMs' heat load and temperatures .....	61

## LIST OF FIGURES

Figure 2.1 (a) TS (b) PV graphical plot to represent a gas turbine cycle .....	8
Figure 2.2 (a) Single shaft gas turbine engine (b) Twin shaft gas turbine engine schematic .....	11
Figure 2.3 Performance simulation models .....	12
Figure 2.4 (a) Iteration procedure of a gas generator (b) Overall iteration procedure with free power turbine .....	17
Figure 2.5 (a) Typical compressor (b) Typical turbine map from literature.....	20
Figure 2.6 (a) Typical compressor map (b) Typical turbine map from Smooth C & T.....	22
Figure 2.7 Iteration procedure of single shaft gas turbine .....	24
Figure 2.8 (a) VTB user level hierarchy (b) VTB components .....	27
Figure 3.1 (a) Steps to convert compressor characteristic map into table format (b) Visualization of characteristic map with beta lines.....	30
Figure 3.2 (a) Flow steps to convert turbine characteristic map into table format (b) Visualization of beta lines from Smooth T software .....	33
Figure 4.1 Off design flow chart.....	39
Figure 4.2 Transient flow chart.....	40
Figure 4.3 Twin Shaft Gas turbine engine model in VTB schematic designer .....	41
Figure 5.1 Conceptual Layout of gas turbine-electrical-thermal system.....	44
Figure 5.2 VTB Layout of gas turbine-electrical-thermal system .....	45
Figure 6.1 Plot of off design results between Gas Turb software and VTB gas turbine model .....	53
Figure 6.2 (a) Mass flow rate of compressor versus power turbine shaft power (b) Compressor pressure ratio versus power turbine shaft power .....	54

Figure 6.3 Ramp fuel input given at the fuel pump .....	55
Figure 6.4 Comparative transient response plot of power turbine shaft power .....	55
Figure 6.5 Shaft power error versus time.....	56
Figure 6.6 Step change of combustor fuel flow rate.....	62
Figure 6.7 Variations of power in the systems .....	63
Figure 6.8 Heat generated from the PCMs .....	64
Figure 6.9 Temperature variations of the PCMs.....	64
Figure 6.11 Impact of variable speed operation of single shaft gas turbine efficiency .....	68
Figure 6.12 Part load efficiency point projection on a compressor map. ....	69
Figure 6.13 Relative efficiency improvement of single shaft gas turbine at variable speed .....	69
Figure 6.14 Variable speed operational for twin shaft gas turbine engine .....	71
Figure 6.15 Torque plot between (a) Single shaft (b) Twin Shaft .....	71

## LIST OF SYMBOLS

$B$	Overall history vector
$b_o$	History vector
$c$	Compressor
$des$	Design point condition
$f$	Fuel
$G$	Conductance Matrix
$GasTurb$	Gas turbine software
$h$	Enthalpy (J/kg)
$HV$	Heating value of fuel (J/kg)
$I$	Inertia (kg*m <sup>2</sup> )
$in$	Inlet
$isen$	Isentropic process
$L$	Load of gas turbine normalized to design load
$m$	Mass flow rate (kg/s)
$N$	Speed (rpm)
$N\%$	Ratio of shaft speed to design speed
$opt$	optimized condition
$P$	Pressure (Pa)
$P_o$	Ambient pressure (Pa)
PCM	Power conversion module

$PR$	Pressure ratio
$\Delta PR$	Pressure difference
$Q$	Volumetric flow rate ( $\text{kg}/\text{m}^3$ )
$R$	Gas constant
$RC$	Resistance companion
rpm	revolution per minute
$t$	Turbine
$T_o$	Ambient temperature (K)
$VTB$	Virtual test bed
$V$	Across variable vector
$\tau$	Torque (Nm)
$\tau_{drive}$	Source torque (Nm)
$\tau_{driven}$	Load torque (Nm)
$\eta$	Efficiency
$\beta$	Beta-auxiliary coordinate useful for representing table format of maps
$\omega$	Speed (rad/s)
$\gamma$	Ratio of specific heat
$\theta$	Temperature correction factor
$\delta$	Pressure correction factor

# **CHAPTER 1**

## **INTRODUCTION**

Gas Turbine engine is currently the most sought machine for the purpose of power generation and propulsion application. Its economical viability to fit as one technology solution for multiple energy sources is its key inherent advantage. The gas turbines had its inception in 1791 by John Barber who patented the technology. Then onwards it has undergone various stages of development in various applications. The first gas turbine generator was set up in 1939 and it operated for 63 years. Marine vessel operators also found its use during the world war and implemented it. Depending on the usage the current gas turbine can be classified into two types: aircraft/propulsion engines (first application of gas turbine engine) and power generation engines. Under aircraft/propulsion engines there are Turbo-Fan, Turbo-Jet, Turbo-Prop, Turbo-Shaft, Ram-Jet engines etc with each having their own advantages. Similarly power generation engines can be broadly divided into Single Spool, Twin Spool, Three Spool Engines, and Sequential Combustion Engines. The above types of gas turbine engine in general are complex machines with several independent components working in unison and each with its own performance characteristics. The performance characteristics of the gas turbine are highly non-linear and depend on each gas turbine component aerodynamic/thermodynamics. This performance characteristic also directly affects the dynamic operation behavior of gas turbine engines. Due to this characteristic aspect, it

becomes important to predict the gas turbine engine behavior at early design stages and before its operation for a particular application. So to predict the gas turbine behavior, the role of cost effective gas turbine simulation model became important, which in late 1950s started with coding simple gas turbine thermodynamic relationships to understand the effect of different parameters.

The main reason to develop a simulation model is because in a traditional approach building a new engine model is time consuming and expensive while an effective simulation model can be developed without any prototypes being needed at the very early stages of the design. The overall goal of this project was to develop a simulation model of gas turbine engine using manufacturer characteristic map. This model would predict the engine performances at design point, off design and transient situations with high accuracy. A variable speed parameter study is also performed with the developed gas turbine model. Moreover this developed gas turbine model will be extended to a co-simulation with an electrical system and a thermal system. The simulation platform used for developing the gas turbine model is a Virtual Test Bed environment (VTB) and it is currently developed at University of South Carolina. The VTB simulation tool can effectively handle this development and simulation of gas turbine engine and related works. The simulation model developed in this study can help the gas turbine engine community to address the issue of efficient power generation and distribution.

### **1.1 Motivation and Objectives**

High performance gas turbine engines are being developed to assist several applications like ships and power plants, because of their high power to weight ratio. Gas



turbine engines is proven to be the best means for power production because of their low manufacturing cost, compactness, modularity and flexibility of design. Other advantages of gas turbine include lower start-up time and shorter response time to accept peak load; low maintenance requirement and less pollution. As a highly non linear model by nature, the computer models needed for predicting the gas turbine engines have evolved from a low-level thermodynamic model prediction in late 20th century to current versatile methods of simulation with accurate controls to account the several key advantages of gas turbines. This research will introduce advanced system level simulation environment for gas turbine engines with following objectives.

- 1) To study and develop a non-linear dynamic gas turbine engine model.
- 2) To use the developed engine model to simulate the steady state and transient performance of the selected engine configuration.
- 3) To integrate the developed gas turbine model with multi-disciplinary systems. This research work will couple the mechanical-thermal-electrical system.
- 4) To make a parametric study with the developed gas turbine engine model and standard simulation model.

To elaborate on the objectives made above, firstly objectives 1 and 2 focus on developing the backbone mathematical model of the gas turbine system in a virtual test bed environment and validating their performances to a standard gas turbine performance tool. The approach to accomplish this objective was to develop mathematical models of individual components of gas turbines using characteristic performance data and finally to utilize the gas turbine component map matching method to complete simulation set up. The model was developed on a virtual test bed which is an advanced dynamic simulation

environment. The third objective was set keeping in mind that the demand for power generation is increasing and there is a need to understand the potential interaction between power generation and distribution system for efficiency improvement. In a larger power generation system i.e. involving many sub-systems or independent system working in conjunction; simplification in any of the particular system at early stage of design studies, leads to losing the important parameter essentials of that particular system. So in this study the gas turbine engine will be connected to an electrical and thermal system and their interaction behavior will be studied. A comprehensive procedure to setup this system will also be shown. The fourth objective was set forth to show the significant opportunities available to improve part-load efficiency of the gas turbine engine in those certain electrical power generation applications that permit variable speed operation. This study will understand the aspect of single shaft and twin shaft gas turbine engine performance for part load efficiency improvement. The study will utilize Gas turbine simulation software in addition to developed VTB gas turbine model. The other purpose of this research work shows the capability of Virtual Test Bed (VTB) simulation environment in handling wide range of simulations. The key advantage of VTB simulation environment is that it can effectively simulate a dynamic behavior of heterogeneous and complex systems. This simulation environment is currently being developed, to efficiently simulate the real world system application.

## **1.2 Thesis Structure**

The thesis is divided into several chapters with the results chapter at the end. This assists the reader of the thesis development to achieve the research goals. The outlines

shown below are the topics one will expect from the chapters that will be discussed in detail.

Chapter 2 provides the literature review of gas turbine engine. This chapter will detail the gas turbine thermodynamics, gas turbine system modeling and virtual test bed architecture.

Chapter 3 describes the mathematical modeling of gas turbine components.

Chapter 4 discusses the off design and transient architecture development for VTB environment. This chapter also details the dynamic model assembly and simulation methodology.

Chapter 5 describes the co-simulation modeling and architecture set up in VTB framework.

Chapter 6 presents the results of this research work. Result discussion starts with comparison of VTB model results with well-established gas turbine software used in the industry. Next, result discussion of co-simulation system that utilizes the gas turbine, electrical and thermal system. Finally discussion about the variable speed operation of gas turbine.

Chapter 7 has the conclusions and recommendation from the author's point of view.

## **CHAPTER 2**

### **LITERATURE**

Gas Turbine is a complex system and highly nonlinear in its overall performance. In general to gain insight into electric power quality under various load conditions at early design phase, it is essential to create a high quality performance model of gas turbine. Analytic/simulation model studies of gas turbine started around 1950 with NASA leading the pioneering work.

This chapter provides an overview of gas turbine engine technology with reference of several research publications and books. The reader will be exposed to key principle and theories, elaborating the gas turbine modeling methodology that is adapted for this project work. To give a grip to the reader, the discussion below will be as follows, firstly about the thermodynamic principles of gas turbine, next about the types of gas turbine and simulation modeling methods. Then discussion on key aspects of VTB simulation environment and literature on integrated simulation environment.

#### **2.1 Thermodynamic Principle of Gas Turbine**

To assess the gas turbine engine, its governing principle must be known first. The working fluid of gas turbine engine is gas and its property changes throughout the working cycle. A cycle can be defined as a process in which the gas in the system undergoes changes to reach the final state. Several gas turbine parameters otherwise known as state variable such temperature, pressure, mass flow rate undergoes a change

and satisfy the thermodynamic laws to complete the cycle. The co-relation that exists between the various state variables in a cycle can be found using the thermodynamic 1st law and 2nd laws. The properties of the gas are available through property table or curve fitted relations. The work done and heat transferred in this cycle are path functions (does not depend upon the state of the gas). The other parameters such as enthalpy which is a measure of available energy of the system, and entropy which is measure of systems unavailable work help to define the state of gas in the system.

The thermodynamic working cycle [5] of a gas turbine engine is known as the Brayton cycle. Figure 2.1 shows the processes of an ideal Brayton cycle on a T-S diagram (Figure 2.1 a) and a P-V diagram (Figure 2.1 b), respectively. As depicted in both diagrams 1-2 is an isentropic compression process, 2-3 is an isobaric combustion process, 3-4 is an isentropic expansion process, and 4-1 is an isobaric heat rejection process at atmospheric pressure. Physically process 1-2 takes place inside a compressor to increase the inlet pressure of incoming gas, 2-3 takes place in a combustion chamber where fuel is added to gas and combusted, 3-4 happens in a turbine where mechanical work is extracted from the gas, and 4-1 is exhaust where heat is rejected to reach the initial state. The work done by the cycle is represented by the shaded area as shown in the P-V diagram in Figure 2.1b.

The gas turbine thermodynamic cycle can be a closed or an open cycle. In a closed cycle the working fluid remains in the system and the high temperature exhaust gas is cooled to its original state before reentering the cycle. In an open cycle the working gas directly exits the system after passing through the turbine. There are many parameters that govern the gas turbine engine working conditions. Of those the important

thermodynamic cycle performance parameters that play vital role in gas turbine engines are the specific fuel consumption and thermal efficiency.

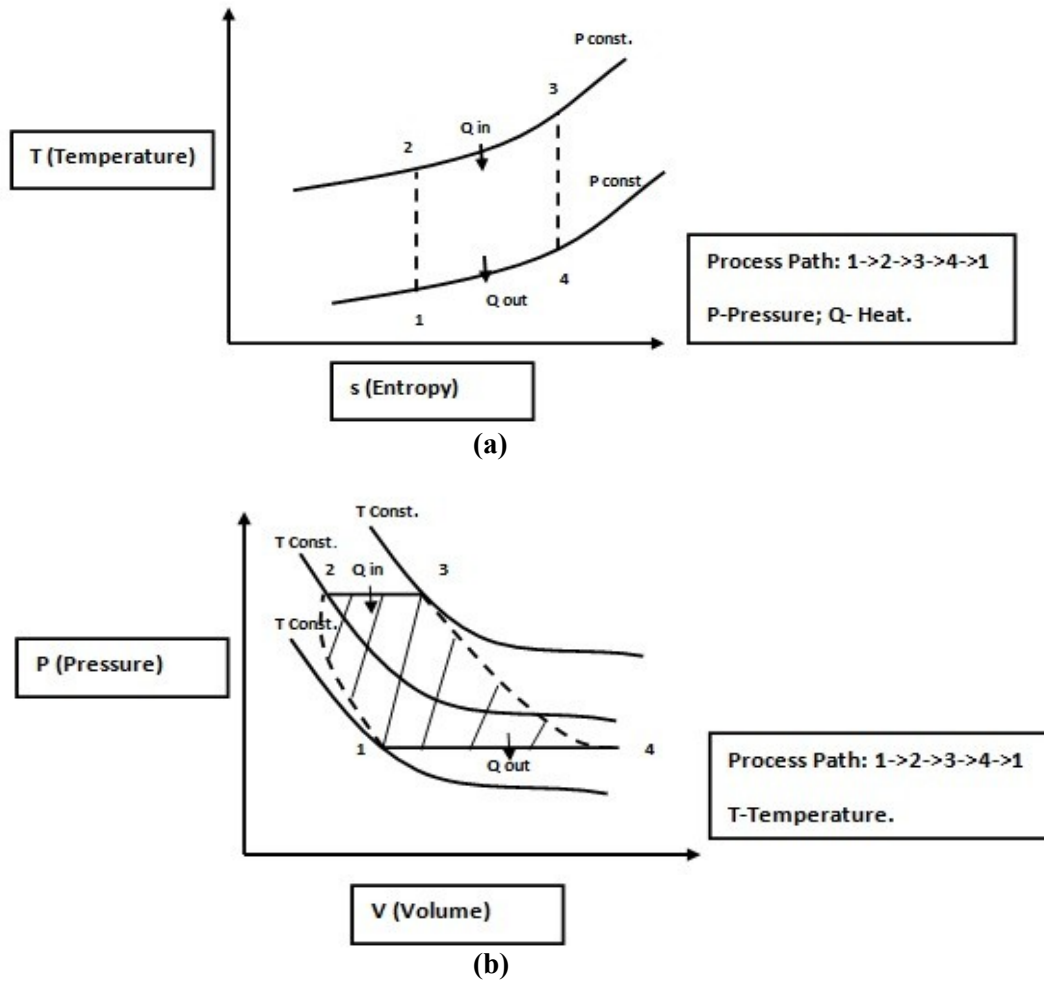


Figure 2.1 (a) T-S and (b) P-V diagrams of a gas turbine cycle [5].

Specific fuel consumption (SFC) is defined as the ratio of fuel burnt to power delivered. It is important to reduce the SFC in gas turbine engine to ensure better efficiency. Thermal efficiency is the ratio produced power to the heat added to the system. The thermal efficiency in general can be increased by increasing the working pressure of the gas in the system or increasing the turbine working temperature. The

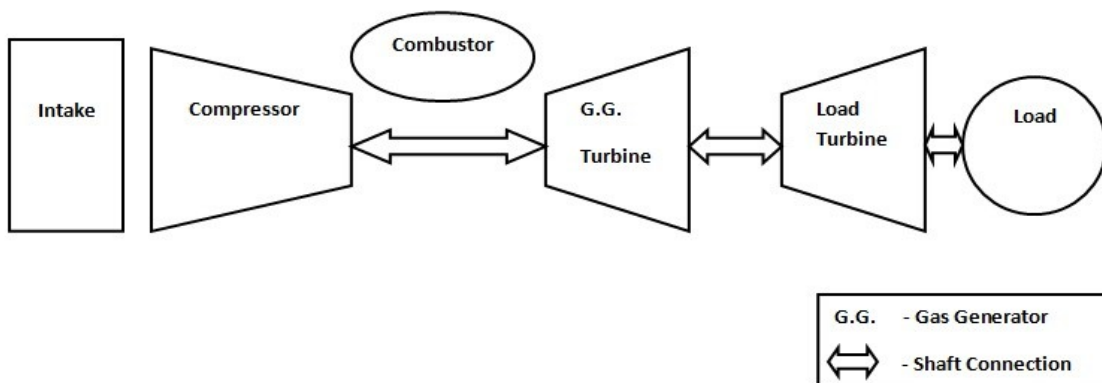
thermal efficiency is an important contributor to the overall efficiency of the system under consideration (e.g. for power generation system applications).

The thermal efficiency of the simple Brayton cycle can be increased by adding additional bottoming cycles or heat recovery cycles to the original simple cycle. The bottoming cycle uses the exhaust heat from the original cycle and acts as an additional source to the original cycle. The heat recovery cycles uses the concept of recycle of exhaust heat within the same cycle and this process also known as regenerative/recuperative cycle. In addition to above methods, there are other modifications which can be done to increase the Brayton cycle efficiency such as a turbo fan etc.

## **2.2 Gas Turbine Engines**

There are several types of gas turbine engines based on its applications and works on the basis of thermodynamic cycle explained above. Broad classification can be made based on the number of engine shafts, e.g., either single shaft or twin shaft engine. A single shaft engine has the compressor, turbine and load on the same shaft. For a twin shaft engine, the compressor and the gas generator turbine are on the same shaft whereas the power turbine and loads are on a separate shaft. The layout of a single and twin shaft are shown below in Figure 2.2. The working principle is similar for both cases with ambient air (working fluid) pass from free stream into compressor inlet which then increases the pressure of air. This pressurized air is burnt with fuel inside the combustor. This high pressure and high temperature gas mixture is expanded in turbine to get work. The advantages of a single shaft engine are its compact in size and have high efficiency.

It requires less time to accelerate or decelerate the single shaft engine based on the load fluctuations. On other hand the main advantage of twin shaft is its autonomous operation of power turbine which provides higher flexibility for overall gas turbine operation. Operational comparison of single shaft versus twin shaft in terms of part load; a typical gas turbine engine is designed to operate at higher efficiency (design load efficiency) which is usually around the design point. So running a gas turbine at part load quickly lower the overall fuel efficiency (noting engine efficiency is a function of speed, load) since the compressor and turbine are reaching lower efficiency independently in this operation. So if the gas generating system is separated from the load system i.e. a twin shaft engine where the gas generator system is separated from the power turbine (load turbine), then the resulting overall efficiency of the engine will be higher at part load [33]. On other hand though the single shaft doesn't benefit much at part load, if its properly designed with its speed controlled then overall part load efficiency of single shaft also can be improved.





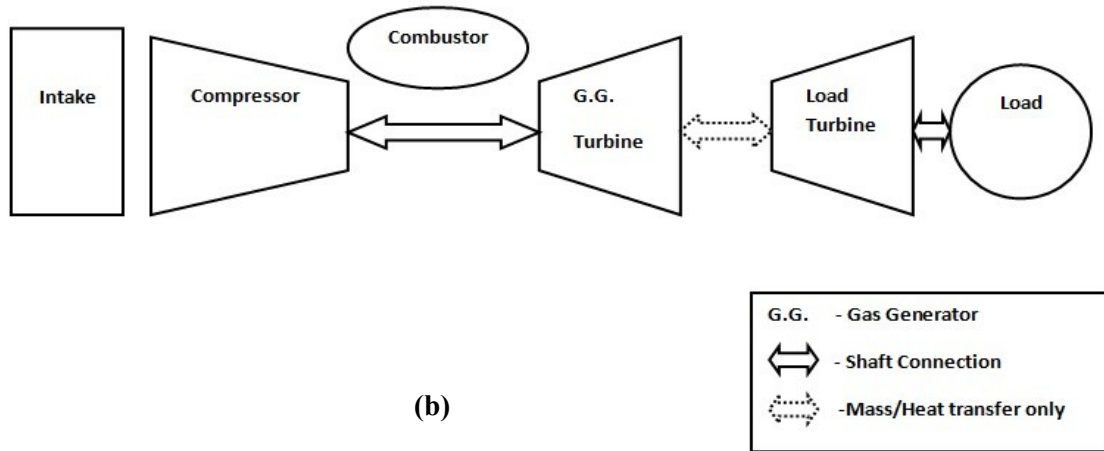


Figure 2.2 Schematic of (a) single Shaft gas turbine engine and (b) twin shaft gas turbine engine.

### 2.3. Gas Turbine Modeling

The needs for the exploring the performance of gas turbine to reduce the initial cost of designing new engines had led to the raise of several dynamic simulation models. These models can be broadly classified as analog models, hybrid models, and digital simulation models. Analog models played key role in early 19th century with the notable work by Larowe, Spencer and Saravanamutto et al. [2][5]. They developed a full operating range of gas turbine analog model. The analog models supported the development of control strategies for gas turbine engine in the early stages. Later the hybrid models further refined these analog gas turbine models with better accuracy. The dawn of digital computer simulation models have overtaken the ability of analog and hybrid models with its speed and capability of complex calculation. Currently digital computer simulation model are widely used to effectively capture the gas turbines nonlinearities and secondary effects.

The methods used to perform off-design simulation in digital computer models range from a single component matching method to a complex rule based expert system. As the complexity increases, the accuracy of the simulation model increases, but there is a proportional decrease of the end user usage. This is because end user with complete background knowledge of that simulation model can only work on this complex platform. Figure 2.3 plots the available methods for off design simulation of gas turbines and the accuracy with respect to complexity of the models.

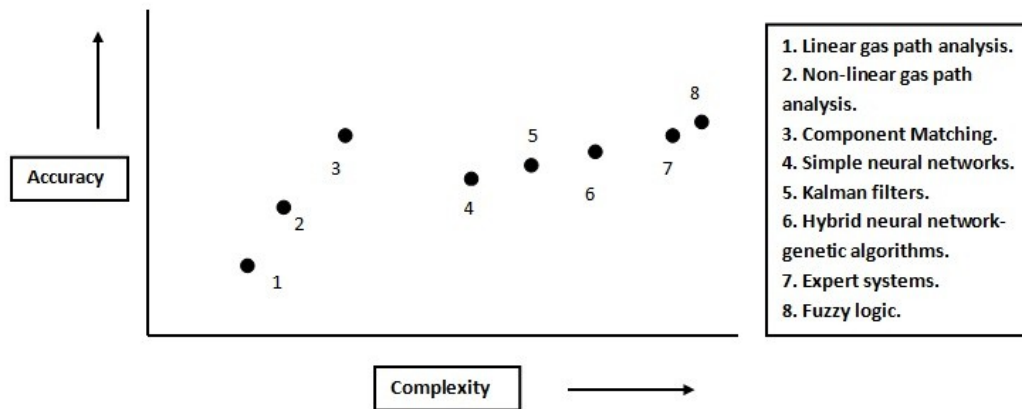


Figure 2.3 Performance simulation models (adapted from [13][27]).

The simulation methods shown in the above Figure 2.3 can work either as a linear system model or non-linear system models. A system is said to be linear, if the superposition principle applies and has only first order relations representing the system. The linear system lumps several parameters into a single time constant relation and considers only the important parameter while solving the governing equation. Linear systems are valid only to certain extent and suffer when it comes to transient system simulation, since only few dynamic parameters are available to understand a sudden off shoot behavior. But for non-linear model superposition principle does not apply and

governing equations of the system are of higher order. So a systematic approach is needed in order to solve the equations and get converged solution. And non-linear systems can give highly accurate dynamic system results. In the following paragraphs, a concise discussion on the methods shown in Figure 2.3 of gas turbine simulation will be discussed with their advantages and disadvantages. The methods were also clearly discussed in the thesis [13] and referred here for literature assistance to understand different gas turbine performance simulation models.

Gas path analysis is a type of gas turbine simulation method and was effectively used in industries in the past. Gas path analysis come in the forms of inverse, linear and nonlinear methods [13] and can be used as performance prediction method or as a performance diagnostic method with reliable result accuracy. Gas path analysis assumes that, when there is a small change in independent variables, it will thereby affect the dependent variables in a linear fashion. The governing equations connecting the independent and dependant variables must be solved to get the converged solution. The independent parameters are usually efficiency and flow capacity and dependant variables are pressure, temperature etc [13]. The main advantage of this method is less computation time. The main drawback of this method is still the characteristic maps of compressor and turbine is required to run the simulation calculations. Reference for this method can be taken from research papers of Mathioudakis et al. [29] and Simon et al. [44].

Component matching is one of the reliable and accurate methods for gas turbine simulation. This method of simulation had been widely accepted in industries. The principle behind component matching method is flow compatibility and work compatibility i.e. the conservation of mass in the system and work matches between

turbine, compressor and load. The solution converges when those criteria are satisfied. The main advantage of this method is the ease of implementation and can be flexible for end user [16]. Prominent works in this method was done by Saravanamutto [5], Muir [32], Walsh and Fletcher [36], Kurzke [14-16]. Details of component matching method will be discussed in later chapters, as this method also forms the basis of this research project. Another method which has similarities with component matching method for performance analysis is the stage stacking method. In this method a compressor characteristic data is predicted using stage wise data available after processing from the running conditions [46]. This characteristic data can be used to design study new gas turbine engines.

The Artificial Neural Network (ANN) method is an innovative gas turbine simulation method that is currently in stages of development. The artificial neural network concept is based on biological neuron interactions inside the brain. The artificial neural network can work effectively, when it is trained with all possible sets of desired inputs and desired outputs. The ANN design can range from a single layer perception neural network to complicated perception neural network. The advantage of ANN is only when all the possible data cases are available otherwise it loses its convergence (prediction) when a new boundary condition is applied [4]. References for this method were taken from research papers of Andre Lazzaretto and Andrea Toffolo [1], Sampath et al. [41].

The Kalman filter is a type off-design simulation method for gas turbine performance simulation. The Kalman filter is an iterative method with a number of measurements in function of time and these set of measurement is known as Kalman

filter gains. These measurements prediction will have error associated and it is reduced after several prediction with continuous iteration. The main disadvantage of kalman filters is that it is lesser accuracy than component matching method; though we can get quicker converged solution [13] [28].

Fuzzy logic method is a simulation method mainly used for gas turbine engine health monitoring purpose. This method supports the study on engine characteristics when there are high uncertainties in the measured data and is found to be very effective way of prediction when there are more data. The research paper of Gunetti et al.[10], Kyriazis et al.[25] were surveyed for this method and details regarding this method can be referred from those papers.

Computational Fluid Dynamics (CFD) is an in depth finite element or finite difference modeling methods, mainly used to understand fluid flow characteristics of a particular component. The CFD method presently can handle complex models. In gas turbine applications it's specifically concentrated on analyzing gas turbine components including compressor, combustor, and turbine etc independently in order to understand their flow and heat transfer. CFD has capabilities to accurately predict the engine behavior that is working in a dynamic environment. Though CFD is able to produce accurate results, it is not often used for off-design simulation of gas turbines as it is not suitable at system-level. Many researchers have worked in this area. The work of Tomita et al. [48] can be referred for further understanding on this method.

Lastly, Wittenberg's method is also an advanced gas turbine simulation method and can predict the gas turbine off-design characteristics without using component maps.

The construction of this model requires thermodynamic relationship and good assumptions of boundary conditions to solve the governing equations. This method had inception in 1976 and further review about this method can be referred from Philip P. Walsh and Paul Fletcher [36].

According to Saravanamutto [5] and Kurzke [15-18], component map matching method is better suited for non-linear off-design simulation of gas turbines mainly because of its flexible approach, reliability, less complex and better accuracy. Both have done extensive work from a rigorous hand written calculations to complex highly accurate digital computer simulations of gas turbine. Based on their work summary of component matching modeling method one thing was in common; both solve mathematical non-linear set of gas turbine system equations iteratively. A detail review of Sarvanamutto's and Kurzke work on component matching will be presented in the following.

To brief about Saravanamutto [5] work of gas turbine simulation, firstly he brought forward the idea of component matching for single and twin spool engines. According to him the component matching procedure of gas turbine depends on compatibility of flow and compatibility of work as discussed earlier. And it must be satisfied at all the time in steady state simulation cases, while for transient case compatibility of work need not to be satisfied. Due to this fact transient simulation component matching procedure requires iteration to reach a steady state. Saravanamuttoo simulated a gas turbine with several standard assumptions. The first assumption was to use a non-dimensional mass flow parameter for turbine and the second assumption was to use constant specific heat. The significance of these assumptions is to reduce the

simulation complexity. The flow chart (Figure 2.4) developed by Saravanmutto shows the clear procedure of handling the gas turbine simulation model and to generate the equilibrium point or running line of a twin-shaft gas turbine with free power turbine.

The flow chart gives the details about the gas generator matching (a) and free turbine matching (b) methodologies. The important iteration step in the flow chart is guess values. Both the pressure ratio across the combustor (a) and the compressor operating point (on the chosen speed) (b) must be updated after their initial guesses.

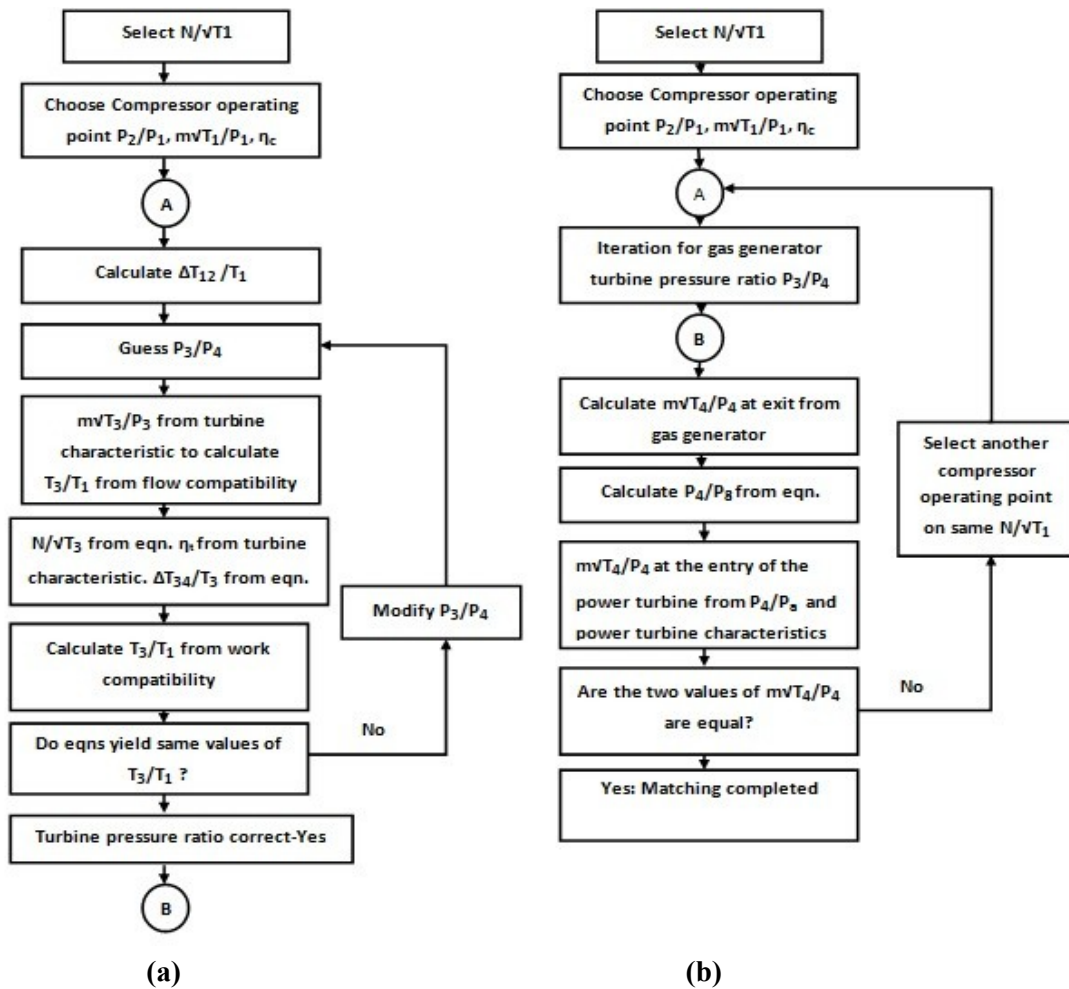


Figure 2.4 (a) Iteration procedure of a gas generator (b) Overall iteration procedure with free power turbine [5].

The Details of gas-generator matching (Figure 2.4a) are discussed as following.

- Select a speed line on the compressor map.
- The corresponding point on the turbine characteristic is obtained from consideration of compatibility of rotational speed and mass flow.
- With matched compressor and turbine characteristics, check whether the generated work corresponding to the selected operating point of the turbine is compatible with the required work from the compressor and the load.

For a twin-shaft gas turbine, the matching of gas generator and free power turbine is also needed. The other parameters are derived from governing equations and characteristic maps. The system converges to an equilibrium operating point after several iterations and after satisfying compatibility conditions. Connections of all the equilibrium running points (Figure 2.6) make equilibrium operating line (Figure 2.6) of the gas turbine. The flow chart shown in Figure 2.4b is an overall flow chart of a twin-shaft gas turbine. The key criteria (Figure 2.4 b) for component matching (involving GG and free PT) are the mass flow rate leaving the gas generator must balance power turbine mass flow rate. The equation 1.1 and 1.2 shown below are the compatibility of flow and work in simple terms.

*Mass flow into a node in a system*

$$= \text{Mass flow out from node in a system} \quad (1.1)$$

*Work produced by system = Work absorbed by the system loads (1.2)*



Following Saravanamutto's work, several other works using component matching method had been done. To take component matching method of gas turbine to the next level in the simulation algorithm, Jaw and Garg [43]; Kurzke [43] implemented a numerical iteration method called the Newton-Raphson iteration method into their gas turbine simulation model.

Newton-Raphson iteration method had capability to handle non-linear system equation and was found to be very effective in terms of flexibility. Kurzke had developed this component matching method further with this Newton iteration methodology and created an advanced gas turbine simulation environment named GAS TURB [15-16]. GAS TURB software is used both in industries and for academic purpose. Kurzke utilizes the component characteristic maps and component matching method to perform the gas turbine simulation. He also stressed the importance of component maps [17] and its usage to get highly accurate simulation results. His gas turbine simulation software architecture had similarities with Saravanamutto's architecture as discussed earlier. He further polished his gas turbine component matching procedure with the need of equivalent number of input guesses and iteration errors while solving the system equations.

According to Kurzke the technique of equivalency in number of inputs guesses and errors proved to increase the quality of the component matching process. Kurzke also used the principles of work compatibility and flow compatibility equations which Saravanamutto framed in order to reach an operating point in gas turbine simulation. The characteristic maps used by him were in terms of looking up table. Dedicated softwares used to convert this characteristic map into table format were also developed by Kurzke. These softwares are called as Smooth C and Smooth T [15]. In the following, a general

discussion about the compressor and turbine map and further followed by Kurzke gas turbine component map procedure will be discussed. [The maps shown below were taken from literature maps list provided by Kurzke along with the Smooth C and Smooth T software.]

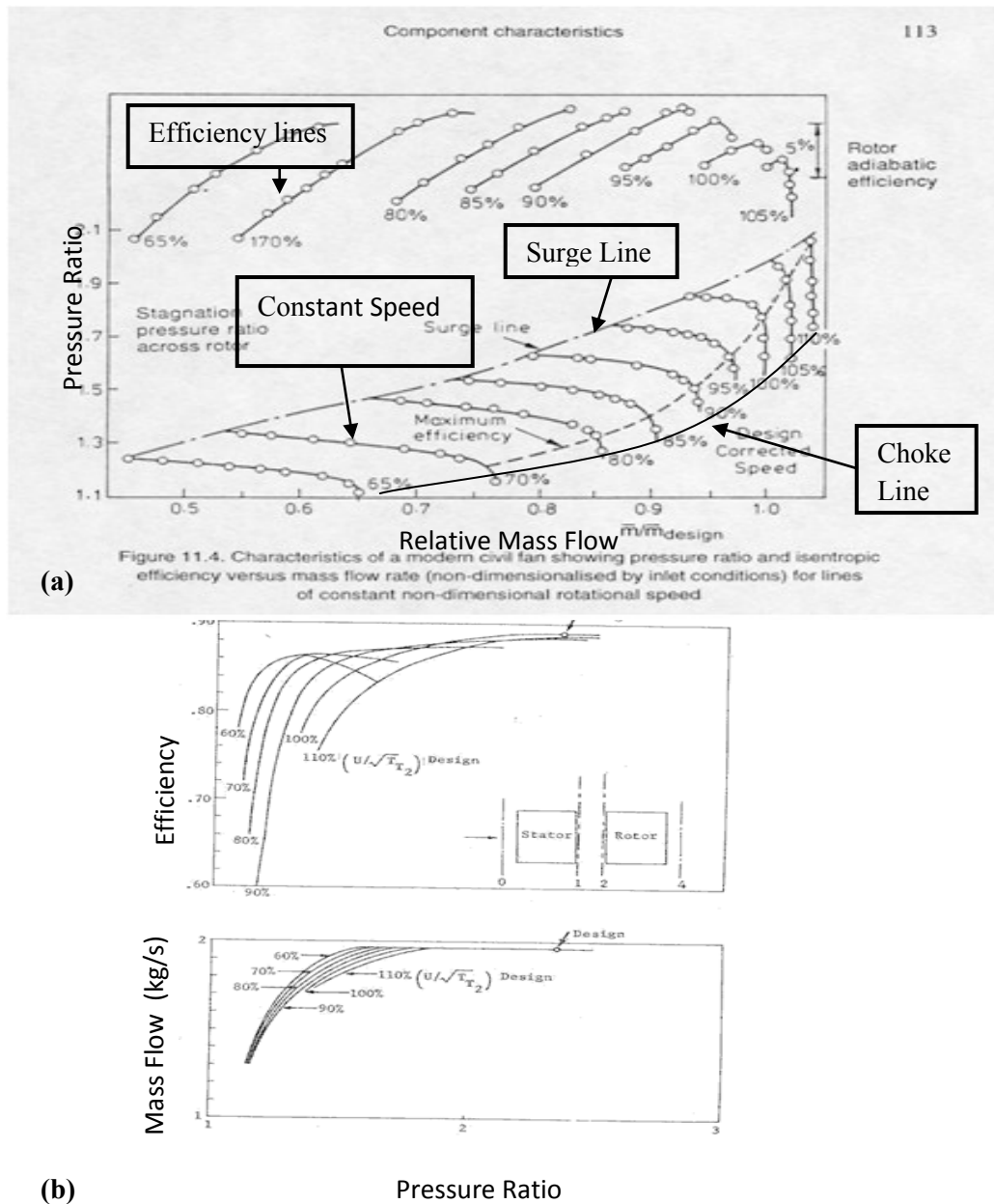
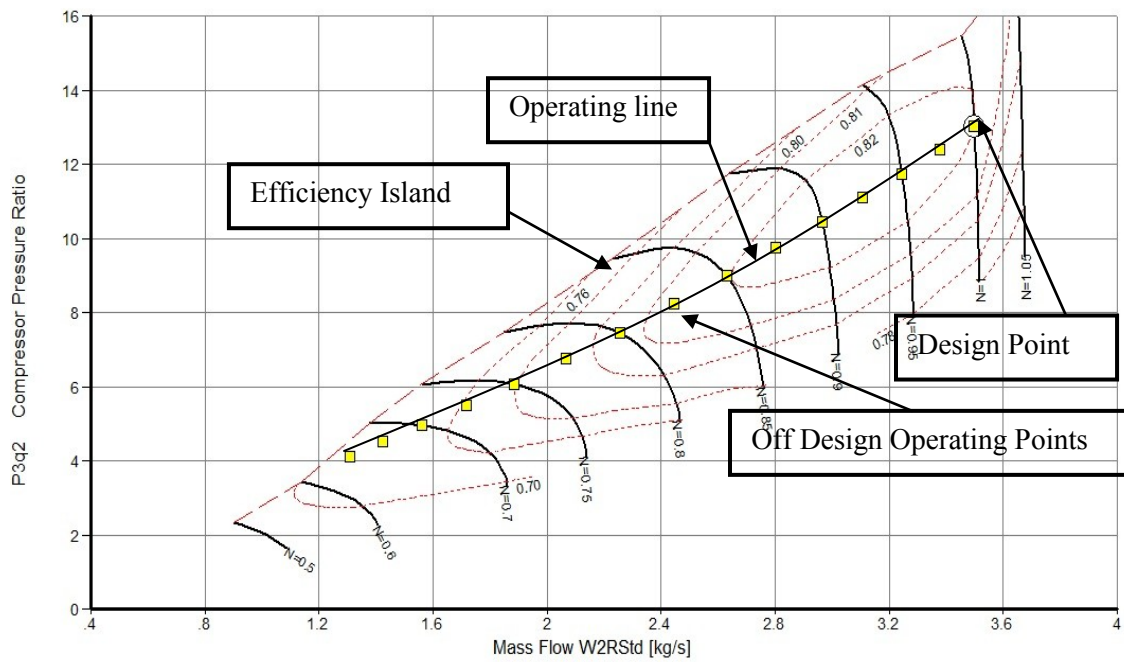


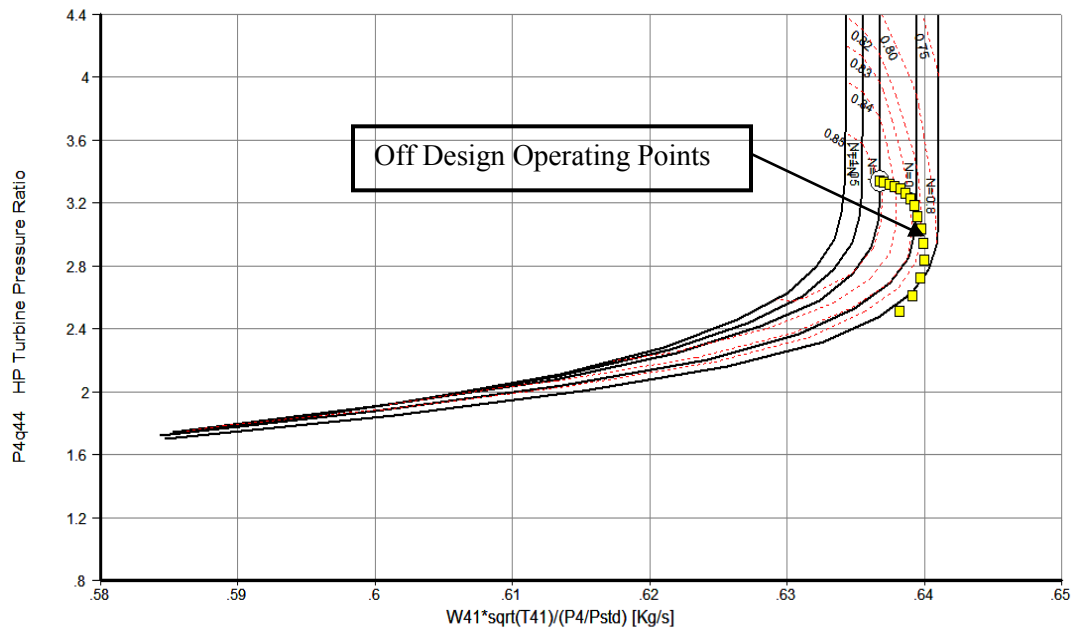
Figure 2.5 (a) Typical compressor map (b) Typical turbine map from literature [15]

Figure 2.5 illustrates the typical compressor and turbine maps which are usually available only to end users by the manufacturer (deemed property). The compressor map shown in Figure 2.5 (a) highlights the key terms. The compressor map is developed from field test data and usually plotted in terms of mass flow rate, pressure ratio, and efficiency. From the compressor map figure, we can see that the constant speed lines start horizontally from the left and drop vertically at the right side. The pressure ratio and mass flow rate increase with increase in speed as depicted in the plot. The surge line is defined simply as reversal of air flow at the compressor individual stages which causes blockage of discharge flow. The choke line is an operating zone where the compressor efficiency drops very low. The turbine map shown Figure 2.5 (b) representation is slightly complex on comparison with compressor map. It is usually have axis plot in terms of mass flow rate, pressure ratio, efficiency and work. Figure 2.6, shows a representation of compressor map and turbine map from Smooth C and T map conversion softwares (Fig. 2.6 maps were utilized for this work). The design point, operating point and operating line are shown in the compressor map Figure 2.6 (a).

Example of single shaft engine matching procedure shown in Figure 2.7 by Kurzke will be described in the following paragraphs and was taken from Kurzke GAS Turb manual[16] [43]. The single shaft component matching of kurzke model starts with a selection of a rotational speed and input of variables (boundary conditions) at compressor side such as inlet mass flow, inlet pressure, inlet temperature and final exit conditions.



**(a)**



**(b)**

Figure 2.6 (a) Typical compressor map (b) Typical turbine map from Smooth C and T [15].

Then using thermodynamic relationships and compressor characteristics curves the calculation of compressor process are performed. The output results of the compressor acts as inputs to the combustor and with combustor governing equations, the next step of calculations are performed. Then finally turbine calculations are done with combustor outlet conditions, turbine characteristic map data and thermodynamic relations. The error calculations simultaneously are made at every inlet and outlet section of the gas turbine components, for example the inlet mass flow (calculated value) of the turbine must match the mass flow value got from turbine characteristic curve (map data) for chosen speed. All these errors are reduced to zero iteratively and that particular operating point is said to be converged operating point for gas turbine system. A simple flowchart representation of single shaft is shown in Figure 2.7.

The characteristic maps used by Kurzke in his component map matching method, however were not readily available for simulation purposes (since it is a deemed property of the manufacturer). Kurzke applied scaling techniques to scale the available maps from the literature to suite to particular engine configuration and run the simulation. This scaling technique was also used by DYNGEN [43] software tool in 1975 and also recent studied (2001) by Kong and Ki [23]. Overall, the scaled maps yielded close enough results on comparison with working model. The scaling laws are show below.

$$\text{Pressure Scaling Equation} = \frac{((PR)_{DP} - 1)}{((PR)_{MapDP} - 1)} * ((PR)_{Map} - 1) + 1 \quad (1.3)$$

$$\text{Mass Scaling Equation} = \frac{m_{DP}}{m_{MapDP}} * m_{Map} \quad (1.4)$$

$$\text{Efficiency Scaling equation} = \frac{\eta_{DP}}{\eta_{MapDP}} * \eta_{Map} \quad (1.5)$$

$$\text{Speed Scaling Equation} = \frac{N_{DP}}{N_{MapDP}} * N_{Map} \quad (1.6)$$

Where DP – Design Point data, MapDP – Map design point data

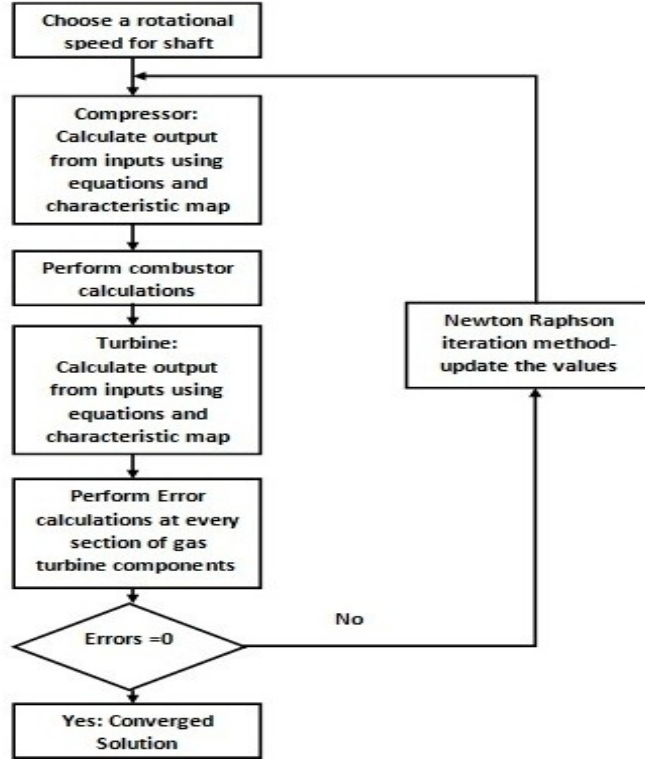


Figure 2.7 Iteration procedure of single shaft gas turbine [adapted from Kurzke GAS Turb manual] [16].

Further to get additional in depth information about the component maps usage for gas turbine simulation environment a paper by Qusai Z Al-Hamdan and Munzer S.Y.Ebaid [37] have detailed a different approach. According to them mathematical modeling using computational techniques was considered to be economical as we described earlier. The first part of their paper shows the gas turbine modeling and the second part details a component matching by super imposing the turbine performance characteristics map on compressor characteristics map and thus identifying the operating

range of gas turbine in a new co-ordinate. The superimposed maps gave a new dimension of using component maps for gas turbine performance modeling and development. Work by Soon Kiat et al. [45] and J H Kim et al. [22] have given in detail description of thermodynamic gas turbine simulation models to understand the transient behavior in simulation.

The following is a brief introduction, on the coding language that is to be used for gas turbine. This was also first started by NASA and several people started working on it from late 90's. Researchers, John A Reed and Abdollah A.Afjeh [19] have developed Onyx a java based programming language (Object oriented programming-OOP) in application of gas turbine engine performance modeling. They addressed the need to develop a flexible software simulation system that is capable of performing advance multidisciplinary analysis. OOP effectively decreases the downtime for code debugging and also reduce the programming difficulties. Several platforms like MatLab, Fortran, Java, C# etc are used for coding. This research project platform was using C# programming language which is also an object oriented programming environment.

## **2.4 Virtual Test Bed Environment**

The current project simulation environment known Virtual test bed (VTB) is being developed by electrical engineering department of USC. The VTB research objective is to create virtual prototype of the real world systems in a simulation platform. The VTB application was based on the theories of the resistive companion modeling technique and is described in detail chapter 4. VTB provide a well-defined three level user hierarchy with each level of the user is considered a distinct level Figure.2.8. From application level the standard user avails the existing libraries and executes the

simulations using pre existing method. From modeler level the users have the ability to modify and design the preexisting models and then execute the simulation. Users at solver designer level have the capability of defining a new method for interpreting system schematics and also for creation of its corresponding models. The VTB graphical user interface consists of a Schematic designer, entities designer, databases (libraries) and external tools. This research work stood partly application level and partly at modeler level. Other detailed explanation on this simulation environment can be taken from thesis work of Lovett [47].

The VTB simulation environment had been adapted for various simulation works in different disciplines; example work by Fang et al. [38-40] on VTB framework will be discussed here. In one of the papers Fang et al. have the addressed future electric navy ships effective heat dissipation issue using VTB dynamic simulation bed. The navy ships depend on power electronic components, high power sensors and advanced weapon system inside the ship. These components increases overall heat load of the total ship system. According to them to understand the issue of heat load distribution in ship system, an effective simulation platform considering both electrical system and thermal system must be present in order to effectively understand the overall system behavior and VTB proves to be better choice because of its ability. Their approach utilizes a hybrid electrical power model which consists of solid oxide fuel cell/ gas turbine (GT) and ship cooling system model. They analyzed dynamic simulation response of the coupled system for simple application scenario and this revealed several important system interaction parameters in combined platform. In this way they also increased the



confidence of using VTB for higher system level simulation for multidisciplinary application.

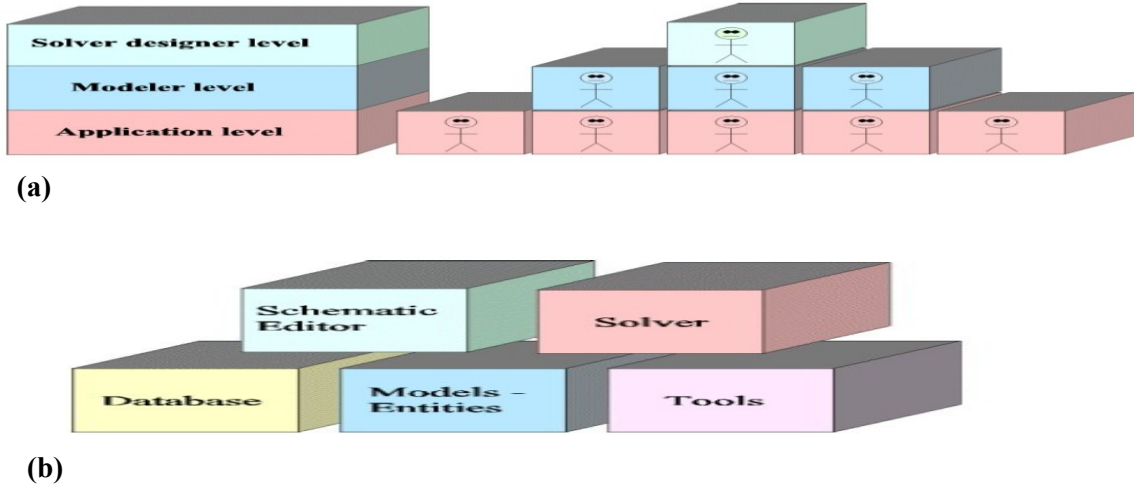


Figure 2.8 (a) VTB user level hierarchy (b) VTB components taken from thesis work of Lovett [47]

## 2.5 Closure

Gas turbine modeling and simulation is complex task involving in-depth technical details. The gas turbine simulation model varies from simple thermodynamic relationships gas turbine programs to a very complex CFD analysis. Various works done in gas turbine simulation methods were discussed and work done by Saravanamutto [5], Kurzke [14-17] using component matching described above forms the basis in this research project. Further to understand the integrated simulation environment work done by Fang et al.[39] was referred. All efforts have been made by the author to acknowledge all possible research publications related to the subject of interest and omissions are unintentional. Any other literature review which is found to be relevant to topic will be included in the further sections.

## CHAPTER 3

### MATHEMATICAL MODELING OF GAS TURBINE COMPONENTS

Mathematical models give an insight into the engine characteristics and model physics during simulation. The conservation laws and equations of motion act as starting point to develop this dynamic math-model which consists of working fluid and rotational components. Gas turbine engine overall performance is determined by its main components the compressor, combustor, turbine, intake/exhaust and engine auxiliaries. The needed requirements of mathematical simulation model were summarized by Saravanamutto et al [5]. According to him, the mathematical model developed is required to be flexible enough, readily understandable, and must give reliable results.

#### 3.1 Compressor Modeling

The compressor modeled here uses component performance map and energy relations. The fluid enters the compressor at one end and leaves with high pressure at other end. The momentum change in the fluid and rise in pressure is due external torque acting on it. The characteristic equations of the compressor are as follows [43]:-

$$\dot{W}_c = \dot{m} * (h(T_{in}) - h(T_{out})) \quad (3.1)$$

$$\tau_c = \frac{\dot{W}_c}{\omega} \quad (3.2)$$

$$T_{out} = T_{in} + \frac{T_{in}}{\eta_{c,isen}} \left( PR_C^{\frac{(\gamma-1)}{\gamma}} - 1 \right) \quad (3.3)$$

$$PR_C = \frac{P_{out}}{P_{in}} \quad (3.4)$$

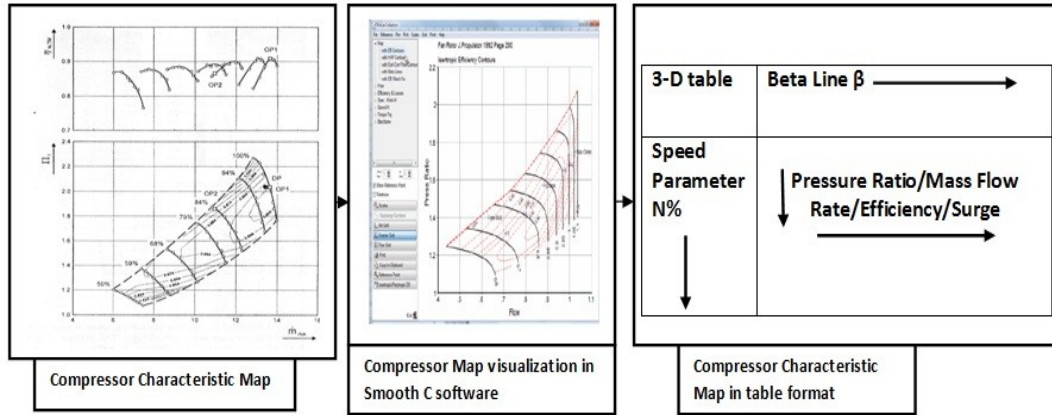
Eq.(7) represents the compressor work and derived from energy conservation with no heat transfer. The enthalpy data for air as a function of temperature is found from published curve relations found in Gas Turb details 5 software [16]. Eq.(8) presents the torque equation and Eq.(9) uses the gas law to calculate the outlet temperature. Eq.(10) gives pressure ratio relation for compressor.

The compressor characteristics are represented in terms of component maps. The advantage of using a component map is that it includes all the losses for a particular design case. A flexible way to represent a component map in simulation is in the form of tables. This is done by the software Smooth C [16]. The software output of table format is unique and avoids the ambiguity of vertical and horizontal speed line characteristics representation of the map, thus better support simulation process. The software representation of compressor characteristics in table format as function of auxiliary coordinate ( $\beta_c$ ) and relative speed is shown below in Figure. 3.1.

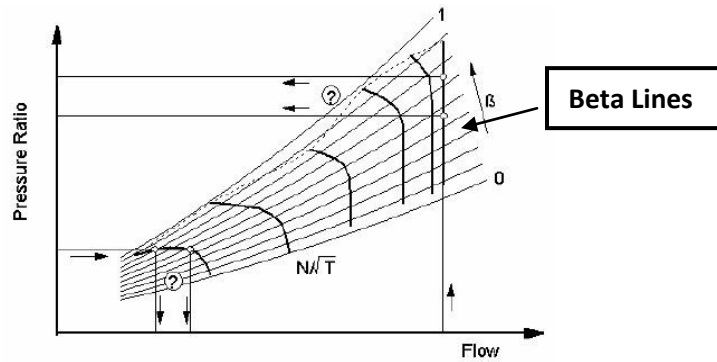
$$PR_C = Fn(\beta_c, \%N_c) \quad (3.5)$$

$$\dot{m} \frac{\sqrt{\theta}}{\delta} = Fn(\beta_c, \%N_c) \text{ where } \theta = \frac{T_o}{288.15} \quad \delta = \frac{P_o}{101325} \quad (3.6)$$

$$\eta_{c,isen} = Fn(\beta_c, \%N_c) \quad (3.7)$$



(a)



(b)

Figure 3.1 (a) Steps to convert compressor characteristic map into table format. (b) Visualization of Characteristic map with Beta Lines taken from Smooth C manual [15].

The modeling is complete after the compressor equations are discretized to get the form of the Resistive companion (RC) format needed for VTB simulation. The RC method is discussed in section 4.

### 3.2 Combustor Modeling

The combustor performance is determined by the amount of heat generated and pressure loss across the chamber. For known inlet conditions, the outlet temperature and pressure loss is calculated based on the equations shown below. Eq.(14) represents the

ratio of change in enthalpy to heat added to the system and defines the combustion chamber efficiency-

$$\eta_b = \frac{1}{\dot{m}_f * HV} (\dot{m}_{out} * h(T_{out}) - \dot{m}_{in} h(T_{in})) \quad (3.8)$$

$$P_{out} = P_{in}(1 - \Delta PR_b)$$

$$\text{where } \Delta PR_b = (\Delta PR_b)_{des} * \left( \frac{\frac{\dot{m}_{in} * \sqrt{T_{in} R}}{P_{in}}}{\left( \frac{\dot{m}_{in} * \sqrt{T_{in} R}}{P_{in}} \right)_{des}} \right)^2 \quad (3.9)$$

$$\dot{m}_{out} = \dot{m}_{in} + \dot{m}_f \quad (3.10)$$

The enthalpy data for hot gas as a function of temperature is found from published curve relations found in Gas Turb details 5 software. Eq.(15) gives the combustor exit pressure calculation. And Eq.(10) calculates the outlet mass flow rate. Likewise, equations (14) - (16) are discretized to get the form of needed RC format for VTB simulation.

### 3.3 Turbine Modeling

Similar to the compressor modeling, turbines modeled here also employ component performance maps Fig. 2 and energy/momentum relations. High pressure and temperature gas expands through the turbine and generates power to rotate shaft. The power developed is used to run the compressor and other auxiliary components. The governing equations of the turbine are as follows:-

$$\dot{W}_T = \dot{m} * (h(T_{in}) - h(T_{out})) \quad (3.11)$$

$$\tau_T = \frac{\dot{W}_T}{\omega} \quad (3.12)$$

$$T_{out} = T_{in} - T_{in} * \eta_{T,isen} * \left( 1 - \left( \frac{1}{PR_T} \right)^{\frac{(\gamma-1)}{\gamma}} \right) \quad (3.13)$$

$$PR_T = \frac{P_{out}}{P_{in}} \quad (3.14)$$

Eq.(17) calculates the turbine work. It is derived from energy conservation with no heat loss to the ambient. Similarly the enthalpies are calculated from curve fit correlation. Eq.(18) gives the torque relation and Eq.(19) uses the gas law to calculate the outlet temperature. Eq.(20) defines the pressure ratio.

Following the same procedure as described above, the turbine characteristic map is converted to table form and by using an auxiliary co-ordinate and the relative speed.

$$PR_T = Fn(\beta_T, \%N_T) \quad (3.15)$$

$$\dot{m} \frac{\sqrt{\theta}}{\delta} = Fn(\beta_T, \%N_T)$$

$$\text{where } \theta = \frac{T_o}{288.15} \quad \delta = \frac{P_o}{101325} \quad (3.16)$$

$$\eta_{T,isen} = Fn(\beta_T, \%N_T) \quad (3.17)$$

Figure 3.2 illustrates the conversion procedure of conversion and turbine map representation from Smooth T.

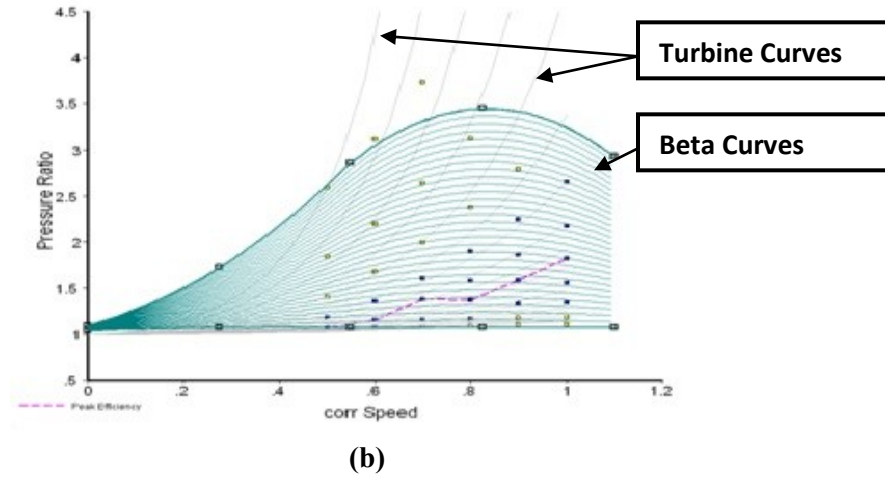
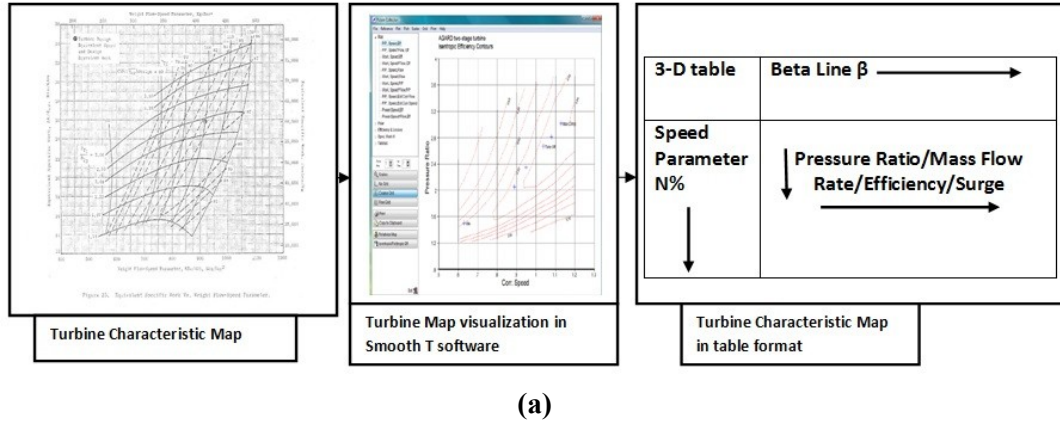


Figure 3.2 (a) Flow steps to convert turbine characteristic map into table format (b) Visualization of Beta Lines from smooth T software.

### 3.4 Shaft Modeling

To predict the transient behavior of the gas turbine, the time constants associated with rotor inertia have to be considered. A physical model of the rotor is employed in this study. The governing equation is shown below:-

$$\frac{d\omega}{dt} = \frac{(\tau_{drive} - \tau_{driven})}{I} \quad (3.18)$$

During transient process unbalanced torque applied on the shaft causes the rotor to accelerate or decelerate.

### 3.5 Fuel Pump Modeling

An externally driven pump is used to fuel into the inlet duct of combustor. The pumping power is a function of the pressure difference, volume flow rate and pump efficiency. It is defined by the following equation.

$$P_{Pump,power} = \frac{\Delta P_{Tot.Pressure} * Q}{\eta_{pump}} \quad (3.19)$$

### 3.6 Inlet/ Exhaust Modeling

Modeling of the inlet and exhaust ducts are the same and follows the below relation to account for pressure loss.

$$P_{out} = P_{in}(1 - \Delta PR_b)$$

$$\text{where } \Delta PR_b = (\Delta PR_b)_{des} * \left( \frac{\frac{m_{in} * \sqrt{T_{in}R}}{P_{in}}}{\left( \frac{m_{in} * \sqrt{T_{in}R}}{P_{in}} \right)_{des}} \right)^2 \quad (3.20)$$

### 3.7 Closure

Gas turbine basically converts the fuel energy into useful work. It generally consists of the components briefed above and works on Brayton thermodynamic principle discussed earlier. The air enters through the inlet air is compressed in the compressor and fuel is added to increase the temperature of the pressurized gas. Then it is expanded in turbine to produce power. Next chapters will brief the VTB form of representation of the above equations (i.e. Resistive companion format) and off design/ transient performance architecture for VTB.



## CHAPTER 4

### GAS TURBINE DYNAMIC MODELING AND ASSEMBLY

#### 4.1 VTB Resistive Companion Form Representation

The governing equation of various entities in the system that is to be solved is represented in resistive companion form. The resistive companion method supports a way to represent the natural conservation laws (i.e. energy flow into and out of the system) by defining a pair of across and through variables at each terminal of the entity in the system. And this is the procedure followed for developing natural port of each component for a system in VTB. Then the independent components in the system communicate with the VTB solver by forming components conductance matrix and history vector needed for every iteration the simulation is carried out. The generic resistive companion (RC) format required while using solver in VTB is shown below:-

$$I(t) = G.V(t) - B(t - h) \quad (4.1)$$

where  $I(t)$  is the through variable vector,  $G$  is the conductance matrix,  $V(t)$  across variable vector,  $B(t-h)$  history vector of device and  $h$  is simulation time step. For example the outline representation of RC method for a mechanical component with through variables (mass flow rate  $\dot{m}$ , rate of heat transfer  $\dot{q}$ , torque  $\dot{\tau}$ ) and corresponding across variables (Pressure  $P$ , temperature  $T$ , rotational speed  $\dot{\omega}$ ) is shown below. The intermediate variables that depends on the through and across variable are not shown. The representation is a generalized form as shown below,

$$\begin{pmatrix} \dot{m} \\ \dot{q} \\ \tau \end{pmatrix}_t = (G)_{t-h} * \begin{pmatrix} P \\ T \\ \omega \end{pmatrix}_t - \begin{pmatrix} b_o \\ b_1 \\ b_2 \end{pmatrix}_{t-h} \quad (4.2)$$

For above RC form, the discretized VTB representation of the conductance matrix is

$$(G)_{t-h} = \begin{pmatrix} \frac{\partial \dot{m}}{\partial P} & \frac{\partial \dot{m}}{\partial T} & \frac{\partial \dot{m}}{\partial \omega} \\ \frac{\partial \dot{q}}{\partial P} & \frac{\partial \dot{q}}{\partial T} & \frac{\partial \dot{q}}{\partial \omega} \\ \frac{\partial \tau}{\partial P} & \frac{\partial \tau}{\partial T} & \frac{\partial \tau}{\partial \omega} \end{pmatrix} \quad (4.3 a)$$

and the history vector is:

$$b_o(t-h) = -\dot{m}(t-h) + \frac{\partial \dot{m}}{\partial P} P(t-h) + \frac{\partial \dot{m}}{\partial T} T(t-h) + \frac{\partial \dot{m}}{\partial \omega} \omega(t-h) \quad (4.3 b)$$

$$b_1(t-h) = -\dot{q}(t-h) + \frac{\partial \dot{q}}{\partial P} P(t-h) + \frac{\partial \dot{q}}{\partial T} T(t-h) + \frac{\partial \dot{q}}{\partial \omega} \omega(t-h) \quad (4.3 c)$$

$$b_2(t-h) = -\omega(t-h) + \frac{\partial \tau}{\partial P} P(t-h) + \frac{\partial \tau}{\partial T} T(t-h) + \frac{\partial \tau}{\partial \omega} \omega(t-h) \quad (4.3 d)$$

The solver in VTB uses multivariable newton raphson iteration method and this iteration has been utilized in the designing the off design VTB flow chart and is discussed in the next section.

## 4.2 Off design flowchart for VTB architecture

The off design procedure and transient procedure for VTB simulation environment were adapted from Saravanamutto [5], Kurzke [16], Shaun R. Gaudet [43]. In general gas turbine off design simulation requires iteration of several variables. The iteration converges to a particular operating point when it satisfies the compatibility of flow and compatibility of work with the given boundary condition. The set of these operating

points together form an operating line Fig.2.6. (a) which is the aim of off-design study.

The general off design flow chart for twin shaft gas turbine engine is shown in Figure 4.1.

For a set of given input/boundary conditions, the engine reaches particular operating point when it satisfies the following three compatibility conditions:-

- (i) Balance in mass flow between all components-.
- (ii) Balance in work between the compressor/load and turbine on the same shaft-.
- (iii) Balance in speed between the compressor/load and turbine on the same shaft-.

The compatibility requirements are explicitly the component matching conditions [37] (i.e. of the gas turbine components). They should be met to solve the gas turbine governing equations which becomes a non linear algebraic loop when put together and needs to be solved iteratively. As stated earlier Newton raphson iteration (VTB solver) is used in this analysis.

Component matching conditions:

Shaft Speed of compressor and turbine must be equal

$$N_C = N_T = N \quad (4.4)$$

Turbine mass flow is only the sum of air mass flow from the compressor outlet and fuel mass flow at the combustor [No bleed]

$$\dot{m}_{turbine\ inlet} = \dot{m}_{compressor.\ air} + \dot{m}_{combustor\ fuel} \quad (4.5)$$

Assuming pressure loss across the combustion chamber is a constant small percentage of combustor inlet pressure.

$$P_{cb-out} = P_{cb-in}(1 - \Delta PR_b) \quad (4.6)$$

Power flows in balance.

$$\dot{W}_T = \dot{W}_C + \dot{W}_l \quad (4.7)$$

Off design flow chart shown below Figure 4.1 is for twin shaft engine case and can be adapted for other types of gas turbine engine. This flow chart simulation starts with the initial guesses beta compressor, beta turbine, turbine inlet temperature, beta power turbine, relative power turbine speed. The governing equations of the gas turbine engine that was discussed earlier were solved after this initial guesses. This initial guesses will be updated at every step of Newton-raphson iteration. In particular the beta variable associated with the compressor and turbine, gets updated by reverse interpolating new solved pressure ratio value of newton raphson iteration. Another observation, the characteristic map discussed before is represented in the form of tables, but the VTB resistive companion format needs a governing relation or equation for example a relation between the through variable like mass flow and across variable like pressure. To satisfy that criteria, a piece wise linear approach had been used in compressor and turbine component to relate its table values (characteristic map data). Since Newton raphson iteration automatically updates the guess, this linear equation also gets updated at every step of iteration. And, another important methodology that was adapted from Kurzke for this flow chart is that the number of guesses made and number of iteration errors were the kept the equal in this algorithm.

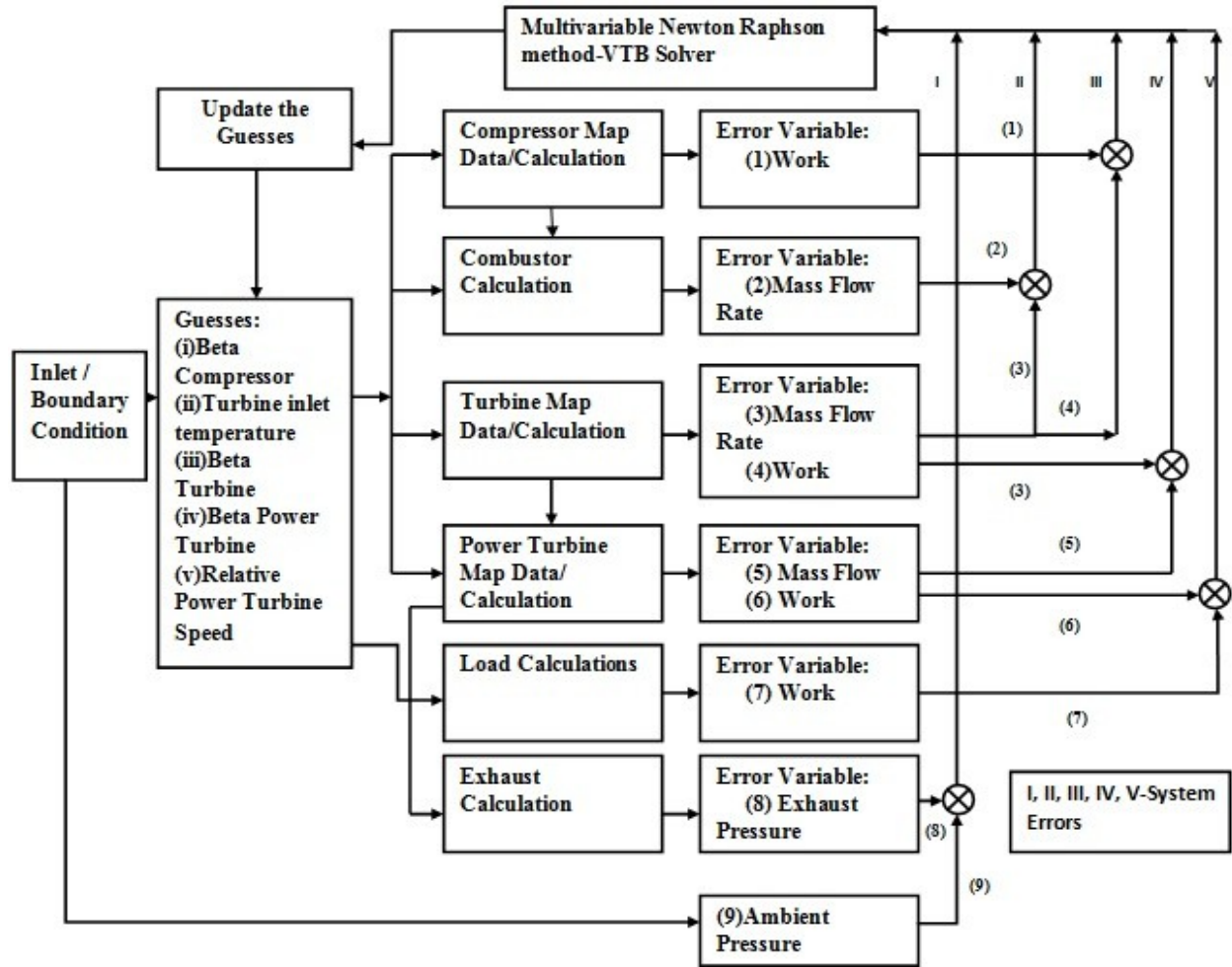


Figure 4.1 Off design flow chart.

### 4.3 Transient Flow Chart

The engine is in transient state when the given control input changes and engine takes finite time to reach the next steady state. The transient behavior, of acceleration or deceleration of the engine can be achieved normally by changing the fuel flow, changing load, inlet mass flow change etc. This phenomenon requires a set additional time dependant equations (shaft equations) into the steady state algorithm. The transient calculation starts at a given steady state and proceeds with the perturbation of fuel flow in current case study to reach the next steady state point. The flow chart shown in Figure 4.2

gives a clear picture of the transient phenomenon. Detailed transient performance calculation flow chart is same as the off-design calculation chart discussed previously, but includes shaft conditions.

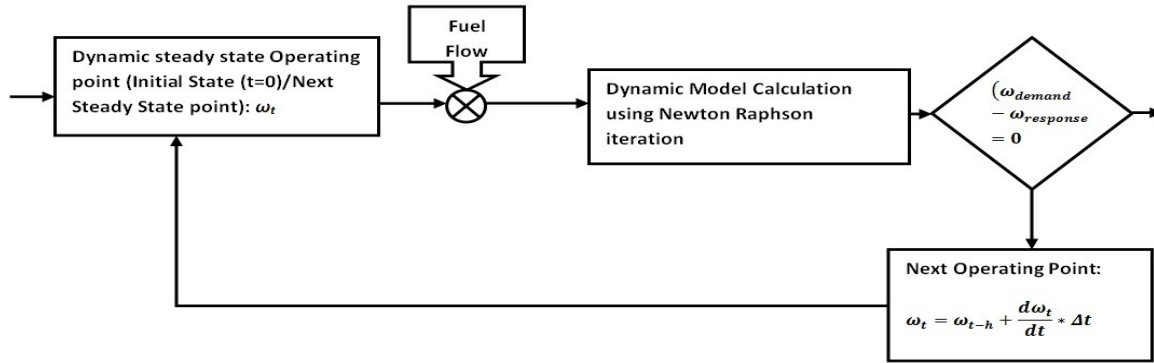


Figure 4.2 Transient flow chart.

#### 4.4 Gas Turbine Model Assembly in VTB

Model assembly consists of stacking the components together as per fundamental thermodynamic principles. For given input conditions, the inter-relationship between various components governs the engine steady state and transient phenomenon. The VTB architectural representation of the gas turbine system is shown Figure 4.3:-

Gas turbine engine has many engine configurations. For the present case studies a single spool and twin spool with free power turbine engine (Appendix A details the VTB code) for power generation purpose is taken into consideration. The components that are modeled in VTB for a twin shaft engine (assembled) are shown in Figure 4.3. It consists of Source/intake, compressor, combustor, fuel pump, gas generator turbine, power turbine, load, exhaust, sink. Intake/Exit boundary conditions are provided by the source and sink components. The details about each component were previously discussed.

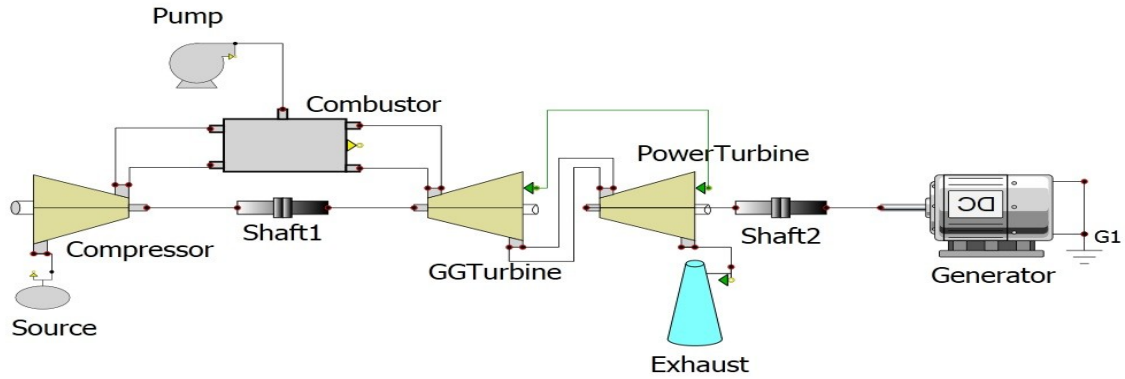


Figure 4.3: Twin Spool Gas turbine engine model in VTB Schematic designer.

Systems that will be included with gas turbine system for studies are the thermal system and electrical system (for co-simulation studies). The objective of this co-simulation study is to integrate the developed gas turbine model with the electrical and thermal system models and to study their interaction behavior during transient events. In this way also prove the VTBs multi-disciplinary system level simulation capability. An example simulation is given in the results section to illustrate the working of these systems in conjunction. The next section will describe about each of systems that will be connected with gas turbine system in detail.

## **CHAPTER 5**

### **INTEGRATION OF THE GAS TURBINE WITH ELECTRICAL AND THERMAL SYSTEM**

#### **5.1. Introduction**

The gas turbine plays a vital role as a prime mover on naval ships or a power generator for its size and high power-to-weight ratio. For example, in navy's future all-electric ship design, the role of gas turbine is different. Instead of mechanically powering a propeller, the gas turbine is dedicated to electrical power generation. The electrical power is then sent to a common electrical bus for allocation to both propulsion and non-propulsion electrical loads. Such a configuration provides the best option to meet the requirements for ship's survivability, re-configurability and flexibility. Meanwhile, thermal issues also become critical and need to be addressed with the electrical system for future all-electric naval ships, as heat generation increases drastically accompanying with the increase in electrical power requirements for high power sensors, etc [39-40]. Thus, the behavior of the gas turbine is closely coupled with the electrical system and thermal system. Advanced simulation is important for the early stage design and analysis of the dynamic interactions between the gas turbine and coupled systems.

Most simulation works are concentrated on component or subsystem levels, and usually are confined with particular discipline such as electrical systems or thermal



systems. A few simulation works were done on the system level and across disciplines. In the work of Norman et al. [35], the dynamic interaction between aircraft gas turbine engine and electrical system was simulated. The relationship between the operation of the engine and the behavior of the aircraft electrical power distribution system were discussed. Chiocchio et al. [39] performed a co-simulation to study the transient interactions between an electrical system and a thermal system. The results gave some insights into the dynamic interactions between the two systems, and revealed some phenomena that cannot be captured without a co-simulation.

To enable the aforementioned simulation between the gas turbine and the coupled electrical and thermal systems in this study, first a high fidelity gas turbine model was created by employing component performance maps for both the compressor and the turbine. This is a twin-shaft gas turbine engine model developed for power generation purpose and also validated with GasTurb a commercial simulation software. Both steady-state and transient operations were evaluated for this model. The other system that will be utilized in this study is a thermal system model developed on VTB platform. This model was developed for the thermal management of ship's electronic system. It consists of two essential cooling schemes typically used in ship's thermal management, e.g., a freshwater cooling scheme and a chilled water cooling scheme. In this study, the freshwater cooling scheme is used to dissipate heat generated from the electrical system. Another system that will be used is an electrical system which serves as power generation and distribution.

Detail modeling methodology will be described below and will discuss about the connection of gas turbine, thermal and electrical system.

## 5.2 Co-Simulation Modeling

A conceptual layout of the gas turbine-electrical-thermal system is shown in Figure 5.1. The gas turbine engine drives a generator for electrical power generation. After power conversion, the electrical power is sent to an electrical DC bus. The DC bus then distributes electrical power to the electrical loads after further conversion. The thermal system is used to cool the electronic components (the power converters, in this study). The interaction between the power system and the thermal system are implemented through a thermal port on each power converter model. Thermal losses resulting from the efficiency calculation of each power converter serve as the forcing function for the thermal system. The heat generated from the power converters is transferred to the heat sinks in the thermal system. Heat sink temperature is assumed to be the same as that of the electrical component being cooled.

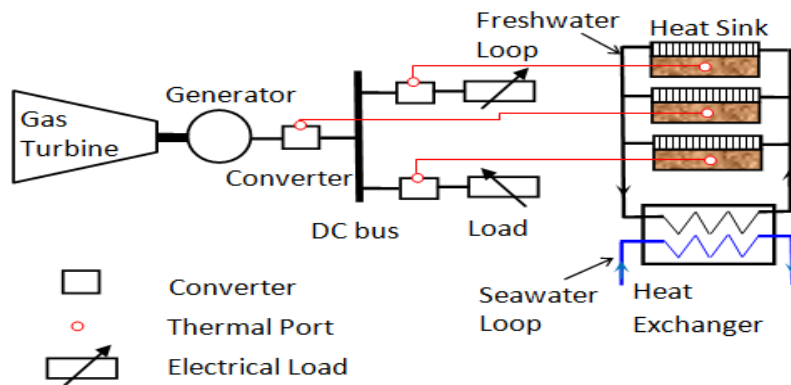


Figure 5.1 Conceptual Layout of gas turbine-electrical-thermal system

The configuration of the electrical system is illustrated at the upper right portion of Figure 4.5. The electrical system in this simulation is a simplified system. It consists of generator, converters, and electrical loads. The electrical system is coupled with the gas

turbine via the shaft in between the power turbine and the generator. Electrical power produced by the gas turbine/generator is supplied to a DC bus after conversion. From there the power is further converted by DC/DC or DC/AC converters and distributed to power consumption devices. In the present simulation, resistive programmable load models are employed. As shown in Figure 4.5, the power converter from the generator to the DC bus is denoted as PCM-1. Similarly, the power converters from the DC bus to the two electrical loads are denoted as PCM-2 and PCM-3, respectively. Those denotations will be used to identify the power converters in the following result analysis section.

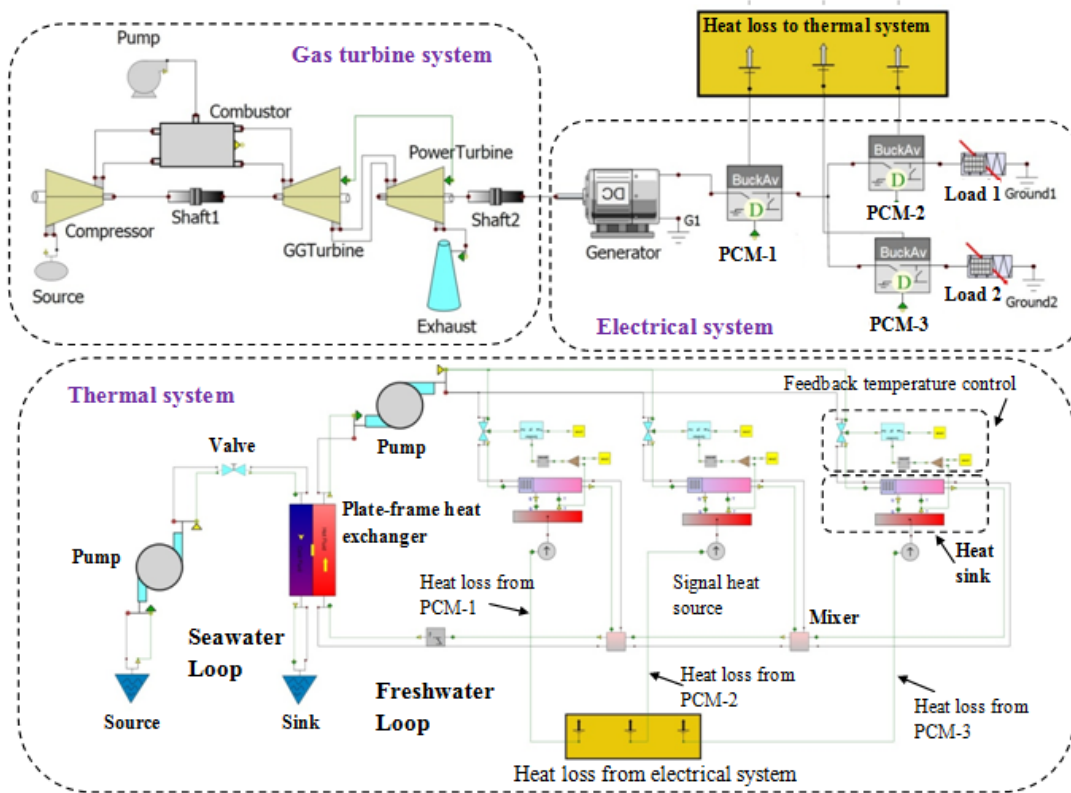


Figure 5.2 VTB Layout of gas turbine-electrical-thermal system

The thermal system model is illustrated in the lower half of Figure 5.2. It is on a separate layer of the schematic for succinctness of the layout, whereas the gas turbine system

model and the electrical system model are on the same layer. The connections between layers are through subsystem connectors. In the co-simulation, the thermal system is coupled with the electrical system through the thermal ports on each power converter modules as described earlier.

The thermal system consists of two loops. One is a closed freshwater loop and the other is an open seawater water loop. Each loop has a dedicated pump for fluid circulation. The two loops exchange heat through a plate-frame heat exchanger. The freshwater loop is mainly used for the power conversion module (PCM) cooling. It is consisted of three parallel branches. Each branch has a heat sink, which is used to retrieve heat from a power conversion module in the electrical system. Cooling of each PCM is achieved by passing freshwater through the heat sinks. Freshwater gets hot at the exit of each branch. Then the mixed hot freshwater is cooled by dissipating heat to the seawater loop. The seawater loop is configured as an open-loop in this co-simulation. It will be a closed centralized loop in the ship's whole cooling system.

In this co-simulation work, only the heat generated from those power converters are dissipated into the thermal system. The heat losses are computed from their instantaneous component through power values by multiplying it with an energy loss factor between 5% and 8%.

Similar to the approach used in [39-40], a simple feedback controller is implemented into the thermal system through a signal valve and a valve controller as shown in Figure 4.5. The goal of the controller is to maintain the PCM temperatures at desired values. The controller checks the value of each PCM temperature in every time step. When the

temperature is greater than the desired value, the valve controller will change the opening of the valve. Correspondingly, the flow rate of freshwater through the heat sink will be changed.

Mathematical description and model development of major thermal components, such as the plate-frame heat exchanger, plate-fin heat sink, and fluid mixer, are described in detail in [39-40]. Other hydraulic components used in the thermal system including valves, water reservoirs, pumps, and pipes were modeled and validated on component level. The validation of the thermal system can be found in [39-40].

## **CHAPTER 6**

### **RESULT AND DISCUSSION**

#### **6.1 Results of VTB twin shaft gas turbine model**

##### **6.1.1 Design Point Validation and Analysis**

To get a reasonably accurate gas turbine model in the VTB environment, the following standard assumptions are made:

- (i) Intake is dry air i.e. relative humidity is zero.
- (ii) Pressure loss at the intake and pump are neglected.
- (iii) Heat transfer effects from components to ambient are not taken into consideration.
- (iv) In the combustor chamber combustion efficiency is constant.
- (v) Volume dynamic of the combustor is not taken into consideration for transient simulation.

As stated earlier, the component maps of compressor and turbines are rarely available during early stage engine design since they are proprietary of the manufacturer. So the maps used in VTB models are the ones available from the Gas-turb and smooth C & T software. These maps are scaled down to design point conditions of the particular case dealt. But can be updated with accurate maps whenever available.

The component models previously discussed have been implemented in VTB environment for simulation studies. Design point validation is done to check the accuracy of VTB programming of the independent components. The validation technique used for the current study had been adopted by many industries/research labs. The idea is to use commercial software. The main reason for using it is because of the non-availability of experimental data in open literature to match the simulation results. The software used in this study is Gas-turb and is also developed/distributed by Kurzke. This software has the capability to perform a wide variety of gas turbine simulations and is being widely used for several applications.

The example application taken for comparison was a twin-shaft engine model for power generation. The base model for comparison was also available in Gas-turb software and was simulated for validation purpose. After giving the VTB dynamic model some design specification (table 6.1) (identical simulation boundary conditions was set in Gas-turb software also). It was made to run at design point condition to produce its own performance data. These results were then compared with Gas-turb (table 6.2).

Analyzing the results generated by the VTB model with the Gas-turb data, discrepancies were found at the hot section combustion chamber but only with minimal error (combustor outlet temperature-0.39%). The reason for this error was due to additional handling on fuel burning taken by Gas-turb software and takes into account of combustion products on the thermodynamic properties. It is to be noted, the contribution by this effect is very less as seen in the results. So the design point results fell well within the acceptable range on comparison, if dissociation effects of fuel were not taken into consideration.

Table 6.1: Working parameters at design point.

Design Point Specification of twin shaft gas turbine engine.	
Ambient pressure and temperature	$P_0=101325 \text{ Pa}$ $T_0=288.15 \text{ K}$
Rotational Speed : Gas Generator : Power Turbine	38000 rpm 20000 rpm
Combustor Outlet Temperature	1450 K
Compressor Mass Flow Rate	3.50 kg/s
Compressor Pressure ratio	13
Compressor Outlet temperature	658 K
Fuel Mass Flow rate	0.080 kg/s
Fuel Heating value	43.124 MJ/kg
Gas Generator Turbine Outlet Temperature	1148.62 K
Power Turbine Outlet Temperature	868.22 K
Power Turbine Shaft Power delivered	1187 KW



Table 6.2: Design point Validation

Design Point Validation of twin shaft gas turbine engine.						
S.No	Component	Parameter	Units	GasTurb	VTB Model	Error %
1	Compressor	Inlet.Mass Flow	kg/s	3.4999	3.4942	0.16
		Inlet.Temperature	K	288.15	288.15	0
		Inlet.Pressure	kPa	101.325	101.325	0
		Exit.Mass Flow	kg/s	3.4999	3.4942	0.16
		Exit.Temperature	K	657.9915	658.03	0
		Exit.Pressure	kPa	1317.225	1316.073	0.08
		Pressure Ratio		13	12.9886	0.08
2	Combustor	Exit.Mass Flow	kg/s	3.5796	3.5738	0.16
		Exit.Temperature	K	1450	1452.791	0.19
		Exit.Pressure	kPa	1277.708	1276.439	0.09
		Fuel Flow	kg/s	0.07968	0.07968	0
3	Turbine	Exit.Mass Flow	kg/s	3.5796	3.5738	0.16
		Exit.Temperature	K	1148.622	1152.446	0.33
		Exit.Pressure	kPa	382.91	383.389	0.12
		Pressure Ratio		3.3368	3.3294	0.22
4	Power Turbine	Exit.Mass Flow	kg/s	3.5796	3.5738	0.16
		Exit.Temperature	K	868.2211	871.6608	0.39
		Exit.Pressure	kPa	102.3383	102.3385	0
		Shaft Power Deliv.	KW	1187.096	1188.885	0.15
		Pressure Ratio		3.741612	3.7463	0.12
5	Exhaust	Exit.Temperature	K	288.15	288.15	0
		Exit.Pressure	kPa	101.325	101.325	0

### **6.1.2 Off Design Point Validation and Analysis**

Off design point calculation follows the off design flow chart and studied to validate the steady state off design operating points generated by the VTB model. The validation strategy was the same as the design point study but with different off-design boundary conditions. The comparative results were plotted in two different ways to get better understanding of the generated off design points. A plot on compressor map, with off-design operating points of both VTB and Gas-turb was made (Fig.6.1), followed by a comparative plot of important parameters throughout the operating line Figure 6.2.

Looking into the compressor map plot, one can identify the off design operating point matched with the Gas-turb off-design results. The off design operating trend matched well because the load characteristics curve was not taken into account in both the simulation environments. The comparative charts made for different off design speeds with respect to shaft power delivered by power turbine also showed minimal error. The trend of all the plots in comparative plot is similar because the speed was decreasing and hence the trends of inlet mass flow rate, pressure ratio, combustor outlet temperature, fuel efficiency were decreasing.

### **6.1.3 Transient Validation and Analysis**

The transient behavior validation was done to verify the response of VTB simulation for given a input change. The strategy for the transient behavior validation still benchmarks the Gas-turb software. The transient simulation is made with changes in fuel flow rate. A ramp fuel flow is given as input to study the transient behavior at 10% from the design speed.

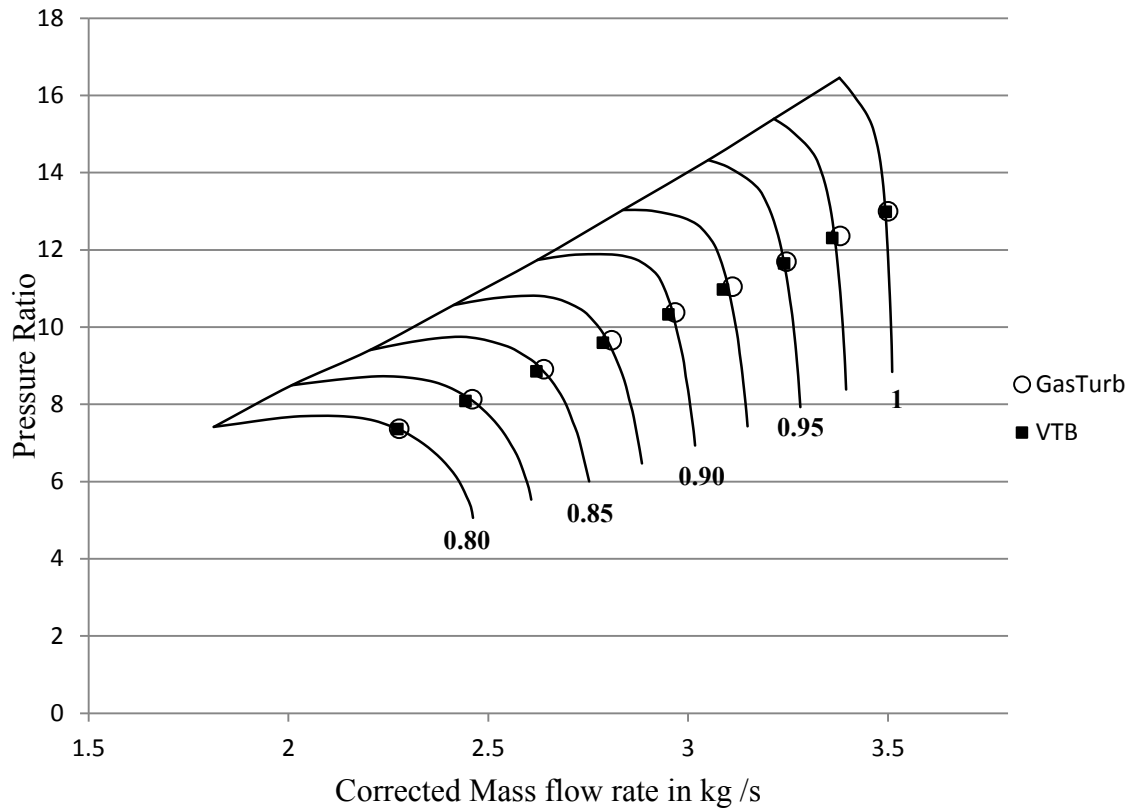
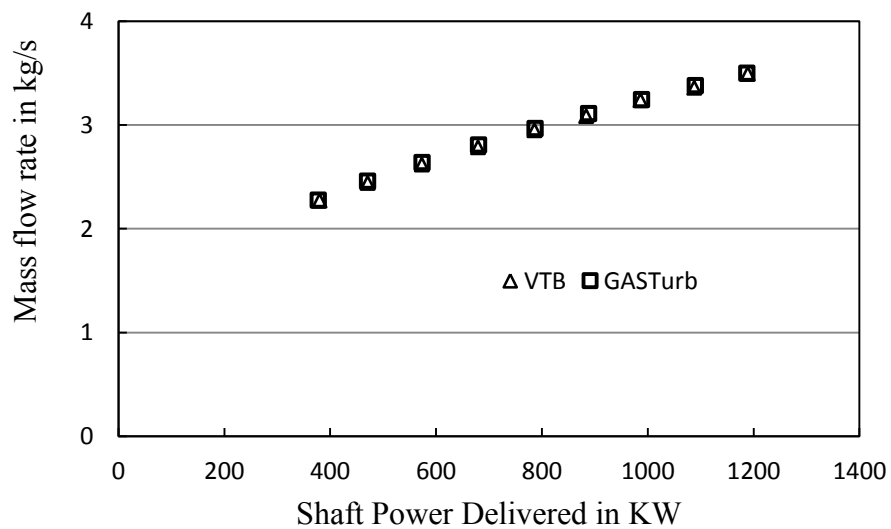
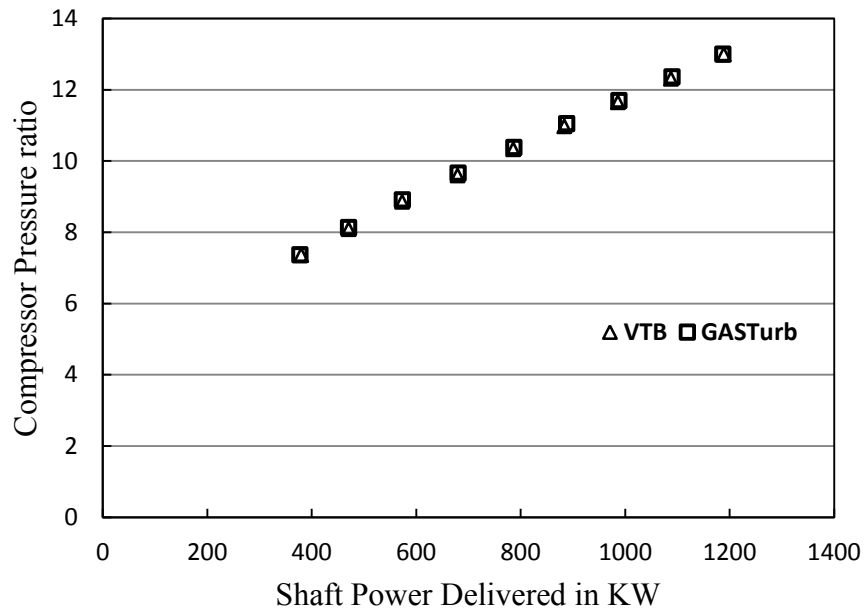


Figure 6.1 Plot of Off design results between gas turb software and VTB gas turbine model.



(a)



(b)

Figure 6.2 (a) Mass flow rate compressor versus power turbine shaft power (b) Compressor pressure ratio versus power turbine shaft power.

Table 6.3: Transient boundary conditions

Transient Boundary Conditions	
Start time	1 second
Stop time	2 seconds
Initial Fuel Flow	0.0796796
Final Fuel Flow	0.05669

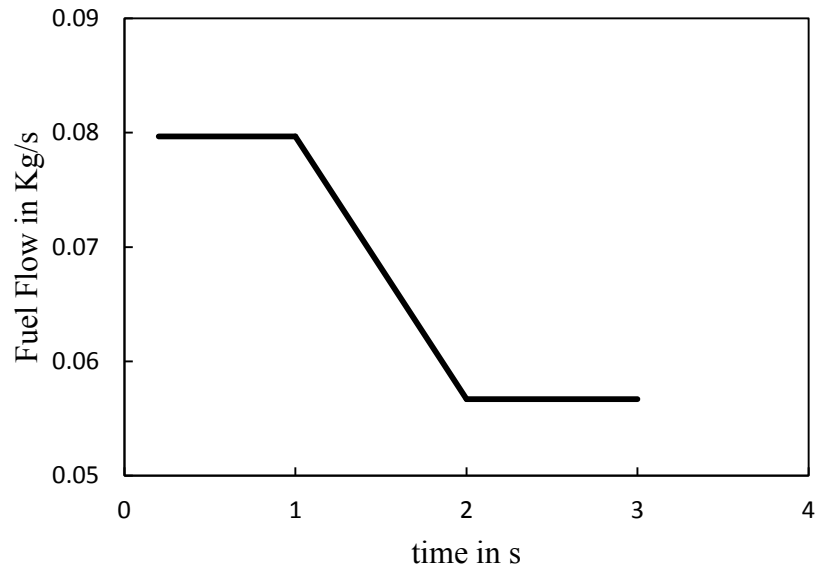


Fig. 6.3 Ramp fuel input given at the fuel pump.

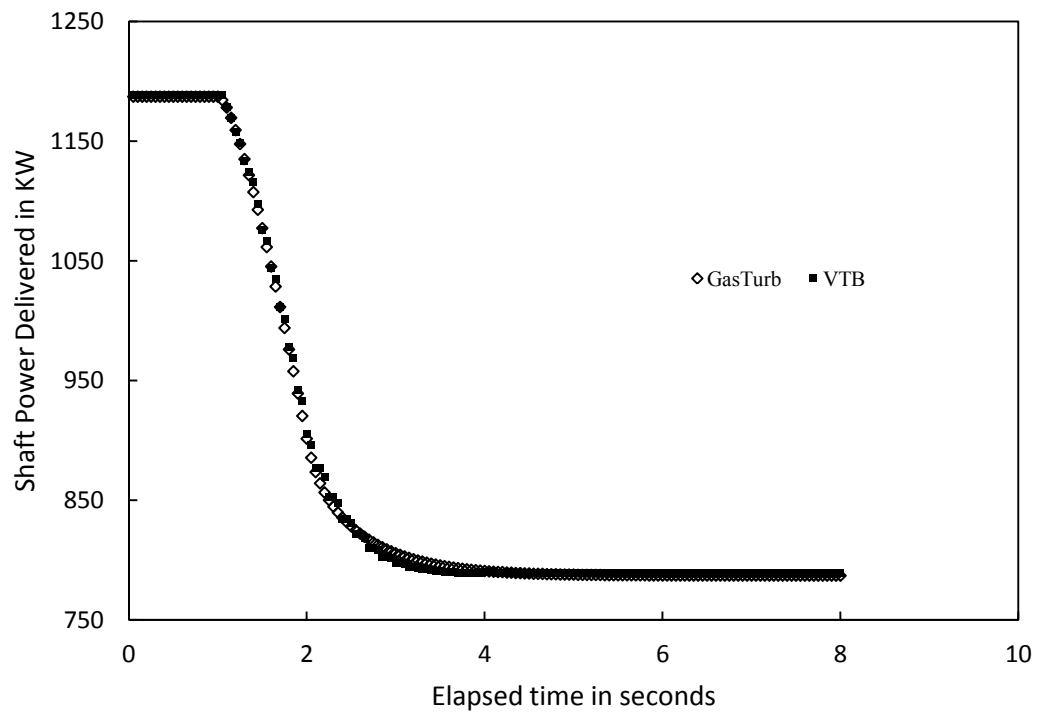


Fig.6.4 Comparative transient response plot of power turbine shaft power.

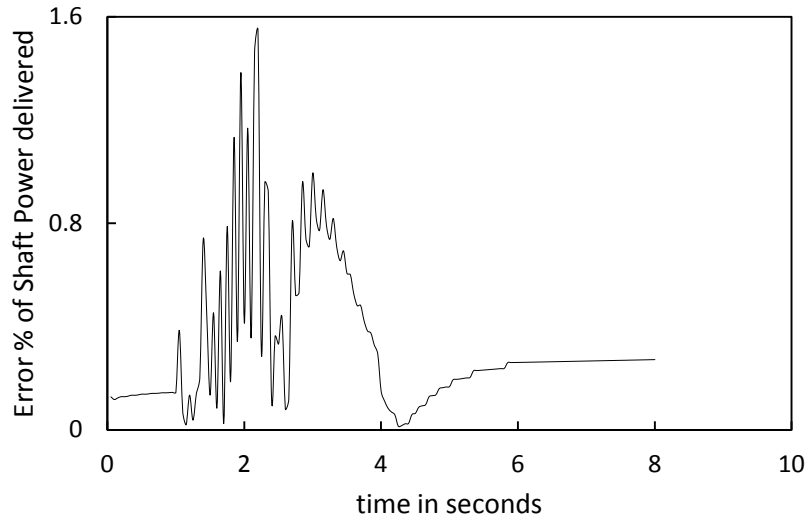


Fig 6.5. Shaft power error versus time.

From this result section we can conclude the following with respect to part of the initial objectives set,

- (1) Represented each independent components of gas turbine by thermodynamic relationships.
- (2) Used characteristics maps of compressor and turbine in the form of tables.
- (3) Framed overall steady state and transient matching phenomenon for gas turbines in VTB environment.
- (4) Developed Design point, off design and transient calculation and verified it with the Gas turb software.
- (5) Have further increased the confidence of using the VTB for system level prediction of mechanical systems.

In the next section we will further show the ability of developed VTB gas turbine model working with other system and show the results on co-simulation performance.

## **6.2 Results of Gas turbine Engine Dynamic Interaction with Electrical and Thermal System**

In this section the developed and verified twin shaft gas turbine model is now connected to thermal-electrical system. This enables to study their interaction behavior between system during a steady state and transient events. An representative simulation is presented to illustrate the working of these systems in conjunction. The objective of this analysis has been stated earlier, in brief this section addresses importance of potential multi-disciplinary systems interaction.

As quoted in an earlier section, with example, in navy's future all-electric ship design, the gas turbine engine is dedicated to electrical power generation. The power is then sent to a common electrical bus for allocation to both propulsion and non-propulsion electrical loads. Thus the gas turbine engine is dynamically coupled with the electrical system, and even with the thermal system, which is usually critical for the electrical system design. It has becoming increasingly important to understand the interactions that exist between the operation of the engine and the behavior of the electrical and thermal systems. This section presents a co-simulation approach for cross-disciplinary simulations. Such an approach is implemented by integrating a twin-shaft gas turbine model, with a power generation and distribution system, and a thermal system. In this study, the thermal system is mainly used to manage the heat generated by the power converters in the electrical system. This paper discusses potential interactions that could

take place during a dynamic disturbance of the fuel flow to the gas turbine engine. Preliminary simulation results for the dynamics of gas turbine power generation, power redistribution between the electrical loads, temperatures of power converters are presented to demonstrate the modeling and simulation capability, as well as illustrating the opportunities for further research. The assembly of this co-simulation system was discussed in the previous chapters.

### **6.2.1 Controls for the Co-simulation and Time Step**

Several control strategies are applied to the hybrid system in order for this co-simulation to perform under transient conditions.

- (1) The gas turbine system is made to start from the rated speed directly, thus start-up process is not included in the current simulation.
- (2) The fuel flow rate to the combustor can be controlled by the fuel supply pump, where a fuel supply scheme can be programmed.
- (3) The power supplied to the electrical loads is also programmable, so that different load modes can be studied for system dynamic behavior investigation.
- (4) As described in section 5.2, the temperatures of the PCMs can be controlled at a desired level by the feedback controller in the thermal system.

The time step is another important computational aspect of this coupled co-simulation. All the three systems have a vast difference in time step. In current simulation, a uniform time step of one millisecond is adopted for all three systems. In following, based on the configuration as described in Figure 5.2, an example co-simulation is implemented to



investigate the typical responses of the coupled systems. Both steady state and dynamic behavior of the complex system are evaluated.

### 6.2.2 Steady State Operating Point and Parameters

The steady state operating conditions for the gas turbine system are summarized in Table 6.4. Those conditions are used as boundary conditions and parameters to solve the system model. In this simulation, the rated power of the gas turbine is 1.18 MW. The needed fuel flow rate, gas flow rate, gas pressure and temperature at each component, etc., will be calculated based on these settings.

Table 6.4: Gas turbine operating conditions

Parameter	Value	Unit
Compressor inlet pressure	1.013	bar
Compressor inlet temperature	288.15	K
Combustor efficiency	99%	-
Combustor pressure loss factor	5%	-
Fuel heating value	43124	kJ/kg
Gas generator turbine inlet	1450	K
Gas generator turbine rated	38000	Rpm
Power turbine rated speed	20000	Rpm

The thermal system is designed to have a rated cooling capability of 200 kW to dissipate the heat loss from the power converter modules. The parameters and operating conditions of the thermal plant are listed in Table 6.5. The geometrical parameters of

each heat sink, which is used to cool the PCMs, are designed differently according to their rated cooling capacities.

Table 6.5: Thermal plant parameters and operating conditions

Parameter	Value	Unit
System initial temperature	25.0	°C
Rated freshwater flow rate	1.22	kg/s
Rated seawater flow rate	1.51	kg/s
Heat sink surface area for PCM-1	0.35	m <sup>2</sup>
Heat sink surface area for PCM-2	0.52	m <sup>2</sup>
Heat sink surface area for PCM-3	0.27	m <sup>2</sup>

### 6.2.3 Steady State Operation Results

Based on the above specific operation condition, the steady state simulation results for the gas turbine are summarized in Table 6.6. At the design point, the gas generator turbine produces a power of 1305 kW, which is used to drive the compressor. The power turbine yields an output power of 1.172 MW, corresponding to a fuel flow rate of 0.0797 kg/s.

For the thermal system, the total heat dissipated from the power converters to the thermal plant is around 150 kW. Heat generated from each PCM and their corresponding temperatures are listed in Table 6.7. As stated earlier, the heat loss from each PCM is proportional to their electrical power being converted, thus the PCM-1, which is the one after the generator, has a larger value of heat generation. The temperatures of each PCM

are different, since the fresh water flow rate passing through each corresponding heat sink is designed to be different in this simulation.

Table 6.6: Gas turbine steady operation results

Parameter	Value	Unit
Compressor outlet pressure	1297	kPa
Compressor outlet temperature	655	K
Compressor pressure ratio	13	-
Compressor gas flow rate	3.46	kg/s
Combustor fuel flow rate	0.078	kg/s
Combustor gas temperature at	1442	K
Gas generator turbine inlet	1258	kPa
Gas generator turbine pressure	3.33	-
Gas generator turbine power	1305	kW
Power turbine inlet	1142	K
Power turbine inlet pressure	377	kPa
Power turbine outlet	822	K
Power turbine pressure ratio	3.69	-
Power turbine shaft power	1172	kW

Table 6.7: PCMs' heat load and temperatures

Parameter	Value	Unit
Heat loss from PCM-1	86.4	kW
Heat loss from PCM-2	25.7	kW
Heat loss from PCM-3	37.7	kW
Total heat loss of the thermal	150	kW
Temperature of PCM-1	78	°C
Temperature of PCM-2	51	°C
Temperature of PCM-3	62	°C
Freshwater flow rate of PCM-1	0.65	kg/s
Freshwater flow rate of PCM-2	0.24	kg/s
Freshwater flow rate of PCM-3	0.32	kg/s

### 6.2.4 Dynamic Operating Results

To investigate the dynamic responses of the complex system, a step change of the combustor fuel flow rate is applied to the system as a disturbance from the steady state. Figure 6.6 shows the step change of fuel flow rate from 0.078 kg/s to 0.068 kg/s at  $t = 400$  sec. Its effects on all the three systems are simulated. Specifically, the corresponding changes of gas turbine power generation, power redistribution between the electrical loads, PCMs' heat losses and temperatures are visualized and analyzed in the following.

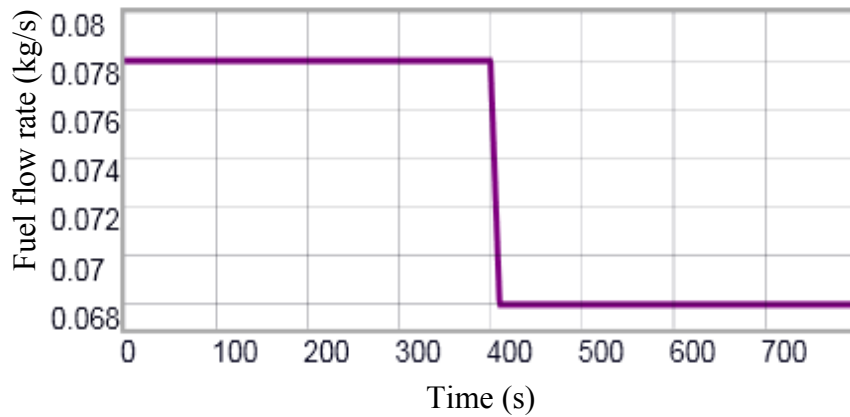


Figure 6.6 Step change of combustor fuel flow rate

The corresponding variations of the power output from the gas turbine and the electrical power redistribution between the two electrical loads are plotted in Figure 6.7. As it can be seen from the figure, starting at  $t = 400$  sec, all the three powers are decreasing following the fuel flow change. For the gas turbine, during the transient change of fuel flow, the compatibility of work between the components can no longer be applied. The power generated from the gas turbine drops immediately with the fuel flow, whereas it takes around 100 sec for the electrical power of the loads to reach at a new

steady state. The lag is mainly because of the rotor inertia and the inductance in the electrical system.

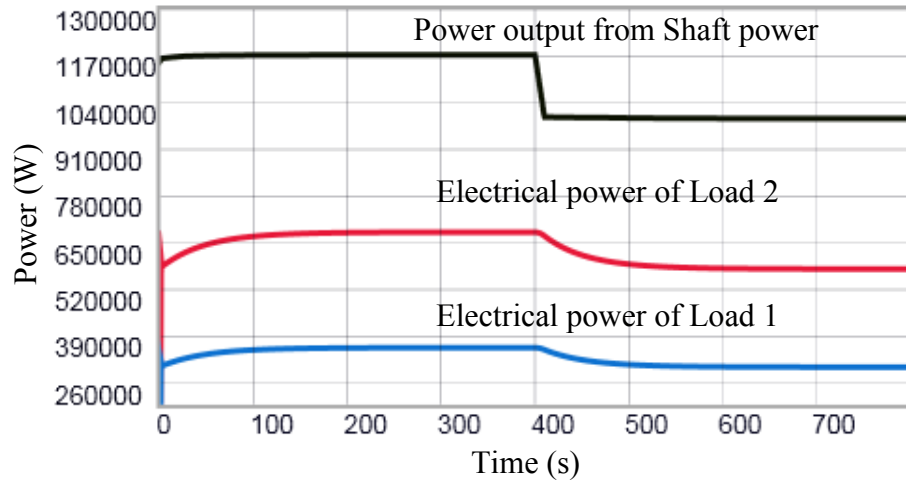


Figure 6.7 Variations of power in the systems

Heat generated from the power converters also experiences variations. The dynamic changes of heat loss from the power converters are plotted in Figure 6.8. Responding to the step change in fuel flow, heat generated from all the three power converters experiences a gradually decrease. The amount of decrease depends on the electrical power passing through it. For example, PCM-1 experiences a larger heat loss variation as its through power is almost the sum of that of PCM-2 and PCM-3.

Heat generated from the power converters are effectively regulated by the thermal system. In response to the step change of the fuel flow, the thermal system also experiences some dynamics, as the change of fuel flow leads to a change of the thermal load. The total thermal load from the electrical system to the thermal system is reduced by  $\sim 22.5$  kW with response to the fuel flow change. The temperature variations of each power converter are shown in Figure 6.9. The temperature decreases are around  $8^{\circ}\text{C}$ ,  $4^{\circ}\text{C}$  and  $6^{\circ}\text{C}$  for PCM-1, PCM-2 and PCM-3, respectively. The even longer lag time (around

200 sec) for the temperatures to reach at a new steady state is mainly due to the thermal mass of the heat sink together with the fresh water inside the heat sink.

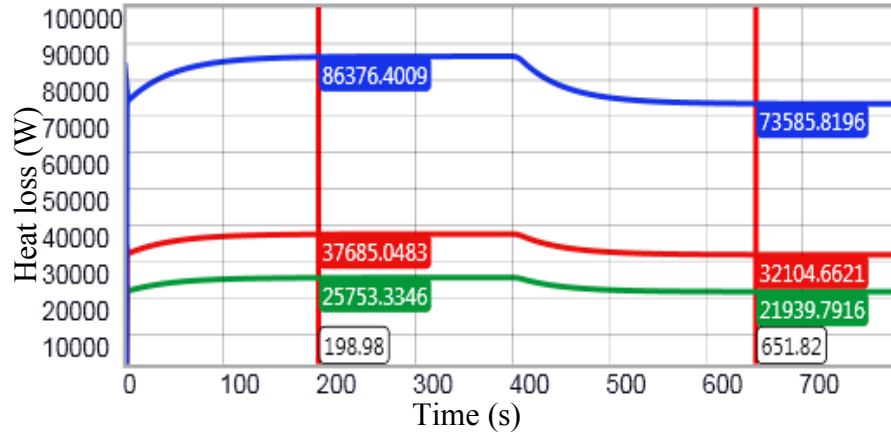


Figure 6.8 Heat generated from the PCMs

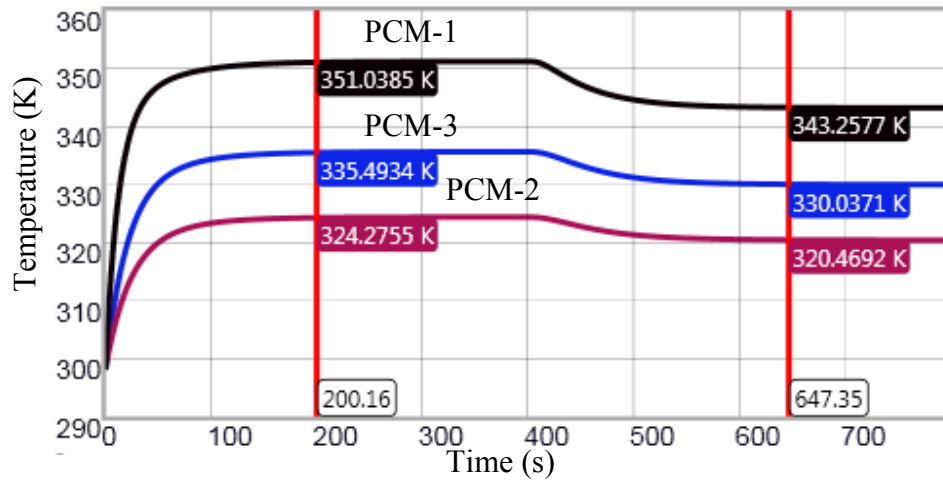


Figure 6.9 Temperature variations of the PCMs

From the above analysis we can see the approach of the co-simulation study. In brief, a co-simulation scenario has been implemented by integrating a twin-shaft gas turbine system with an electrical system and a thermal system on VTB platform. Model developments and system configurations for the co-simulation were described in detail. First, a steady-state simulation for the complex system is performed, and then a step

change of combustor fuel flow rate is applied to the system as a disturbance. Dynamic responses between the systems are simulated accordingly. Especially, the transients of gas turbine power generation, power redistribution between the electrical loads, PCMs' heat losses and their temperatures are analyzed in detail. The variations of those variables reveal different system responses to the dynamic disturbance. In addition to the fuel flow change, system dynamics caused by some other disturbances such as opening a bleeding valve in the gas turbine system, changing the electrical load in the electrical system, or changing the cooling water flow rate in the thermal system, etc., can also be investigated using this co-simulation scenario.

### **6.3 Results of Variable Speed Operation Study of Gas turbines**

In electrical power generation applications, both single-shaft and twin-shaft gas turbines run at fixed speed, the design speed. The reason for this is that the turbine speed has to match with the frequency of ac system, typically 50 or 60Hz. With constant speed operation, gas turbines are well known for lower part-load efficiency. When the load is 20% of rated power, the specific fuel consumption is typically about 1.9 of rated fuel consumption at rated output power. To improve the part-load efficiency, recently, variable speed operation is another emerging method, mainly in fuel cell gas turbine hybrid system. In gas turbine electrical power generation, variable speed operation is realized by a controlled ac-dc rectifier in dc system applications to maintain the desired dc bus voltage. Or in ac system, the desired ac bus voltage is achieved by the combination of ac-dc rectifier (either controlled or uncontrolled) and dc-ac inverter. Mura [33] described an example of variable speed operation of gas turbine in ac power system

where the desired ac bus voltage is controlled by the inverter only and the rectifier is uncontrolled diode rectifier.

Many works related to variable speed operation of gas turbine have been done [5][50] this investigation was to study the role of engine speed on the part-load efficiency including a comparative performance results of both single-shaft and twin-shaft gas turbine. The investigation of variable speed operation of both single-shaft and twin-shaft gas turbines shows significant opportunities to improve part-load efficiency in electrical power generation applications that permit variable speed operation. Efficiency improvement increases as load decreases and the improvement is larger for single-shaft than twin-shaft. For example, one data point shows the fuel efficiency can be improved by 14% for single-shaft gas turbines, but only 2% for twin-shaft gas turbines when load power is 20% of rated turbine power.

In this study the VTB developed gas turbine model and GAS Turb software was utilized. Both the single shaft and twin shaft model of gas turbine was used. VTB gas turbine model was used conjunction with the Gas Turb software gas turbine model in order to show that the thermodynamic relations and characteristic maps discussed earlier formed the back ground frame work for this part load study. Further to show the VTB gas turbine model capability of performing part load efficiency optimization study. In the following, firstly the VTB developed gas turbine model part load optimal results will shown followed by an in-depth discussion GAS Turb model part load optimal results. Finally to note, the VTB gas turbine can also be termed as a semi-theoretical model, since VTB environment basically solves the system equations for given boundary conditions. VTB gas turbine model is termed as semi-theoretical with respect to complete theoretical



model, since it combines both the characteristic map data and the governing thermodynamic relations. The parlance of semi-theoretical will be used for VTB model results discussions below.

### 6.3.1 Variable Speed Results of Single-shaft Gas Turbine

Following the gas turbine modeling procedure previously discussed was applied and investigated a 5.71MW, 15808RPM single-shaft gas turbine on the VTB bed gas turbine model and again independently using Gas Turb software gas turbine model.

$$\eta_{opt-1shaft} = 28.06L^3 - 71.01L^2 + 64.44L + 12.48 \quad (6.1)$$

$$N_{opt-1shaft} = 1.4931L^4 - 4.1703L^3 + 4.0111L^2 - 1.3682L + 1.0342 \quad (6.2)$$

The above equation is curve fitted with optimized part load performance results using VTB gas turbine model and also plotted on Fig. 6.11 for comparison with simulation results. From the plot Fig. 6.11, we can see VTB results have similar trend as the simulation results. The VTB results (semi-theoretical results) were solved for a standard gas turbine running condition. But the GAS turb simulation results for this case study considered several important parameters to replicate a gas turbine engine in working environment condition. The several simulation parameters that were considered in this case study in GAS turb simulation were bleed, inlet/outlet losses, combustor part load calculation, shaft efficiency, component geometric properties and other standard loss calculations. This was also the reason behind for curves of VTB results on the higher side on comparison with the simulation results plotted. Otherwise, the overall graphical trend of the VTB plot provides expected background picture for a variable speed operation of gas turbine for part load efficiency improvement.

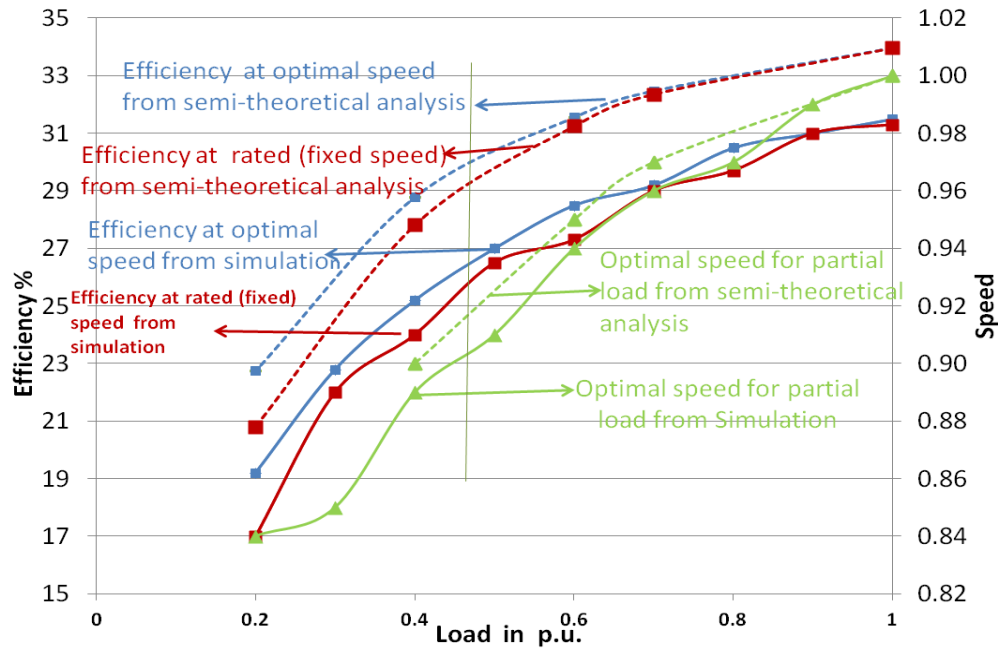


Figure 6.11 Impact of variable speed operation of single-shaft gas turbine efficiency

Looking closely into the simulation results also show, the impact of variable speed operation on single-shaft gas turbine part load efficiency as illustrated in figure 6.11. In fig. 6.11, the green curve is the optimal speed corresponding to the optimal efficiency. The fixed speed efficiency curve is part-load efficiency at design speed,  $N=1.0$  p.u. As can be seen on fig.6.11, for lower power load demand, the gas turbine is more efficient at lower speed, as compared to fixed design speed. For this specific single-shaft gas turbine, the maximum absolute efficiency gain is 2.47% and its relative efficiency increased 14.77% at load with 0.2 p.u. where the optimal speed is 0.84 p.u. The plots of relative efficiency and optimal points of single shaft is shown below in Figure 6.12-13. This gives a clear picture of the single shaft engine optimal efficiency working conditions.

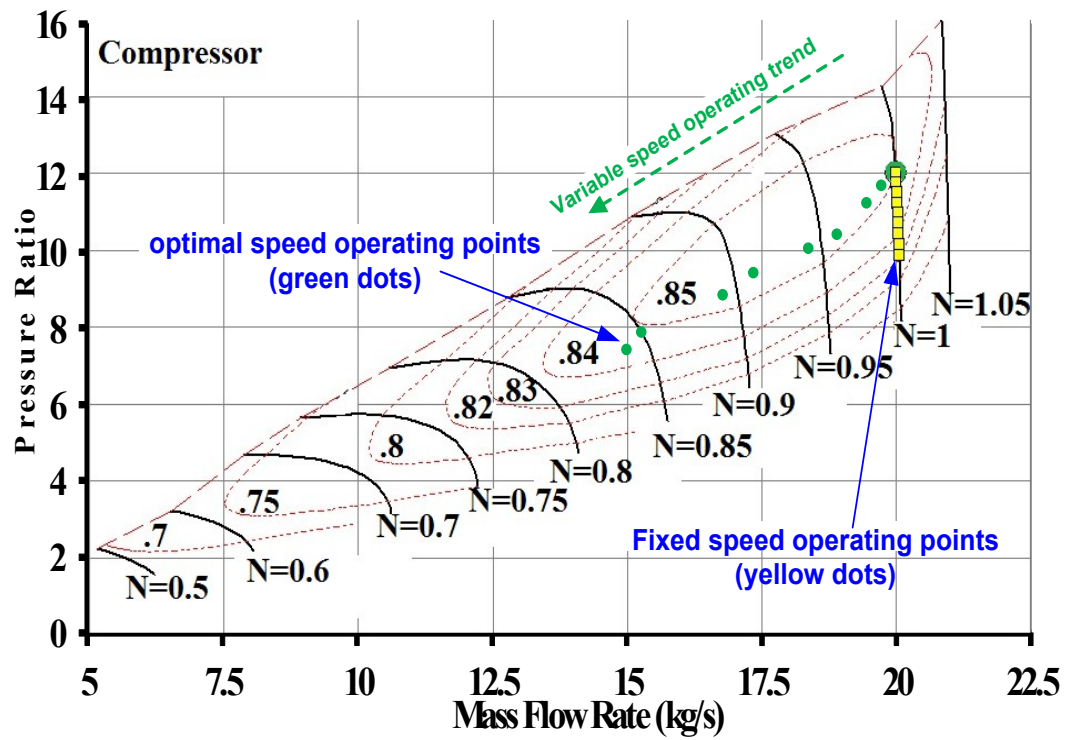


Figure 6.12 Part load efficiency point projection on a compressor map.

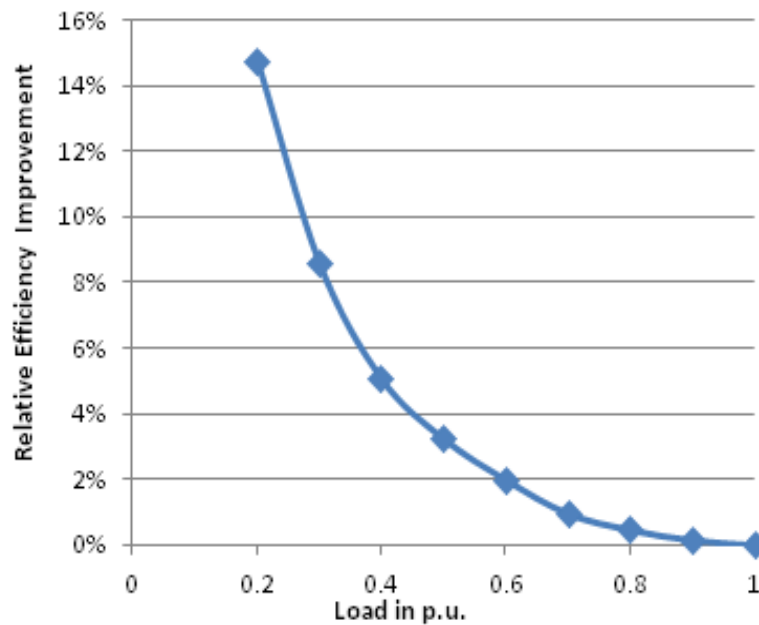


Fig 6.13 Relative efficiency improvement of single-shaft gas turbine at variable speed

### 6.3.2 Variable Speed Results of Twin-shaft Gas turbine

Following the gas turbine modeling procedure previously discussed was applied and investigated a 0.96MW 20000RPM twin-shaft gas turbine on both the VTB bed and Gas Turb software gas turbine model.

$$\eta_{opt-2shaft} = -13.323L^2 + 31.1L + 17.372 \quad (22)$$

$$N_{opt-2shaft} = -19.89L^4 + 54.99L^3 - 55.1L^2 + 24.16L - 2.96 \quad (23)$$

The above equation is curve fitted with optimized results of VTB (semi-theoretical results) gas turbine model and also shown on fig. 6.14 for comparison with simulation results. And explanation similar to VTB results of single shaft also applies in the case of twin shaft engine also.

The simulation results of variable speed study of twin shaft engine are also projected in figure 6.14. One can observe that lower load power demand, the gas turbine is more efficient at lower speed, as compared to fixed design speed. For example at speed  $N=0.8$  the optimal efficiency increases from 15.03% to 15.8% at 20% of rated load. And other rated loads also have negligible increase in optimal efficiency. The reason for negligible optimal efficiency percentage increase for twin shaft on comparison with single shaft was because only power turbine of twin shaft engine had the direct effect of the load demand. This can be seen from the torque plot in figure 6.15 where there significant operation torque difference for single shaft and twin shaft for constant speed and variable speed. For the twin shaft torque plot with power turbine at fixed speed, the gas turbine is already at variable speed and this defines the gas generator is already operating at higher efficiency for change in load. So, this is the reason why only the

power turbine gets the maximum benefit in a twin shaft while operating at variable speed for part load efficiency improvement.

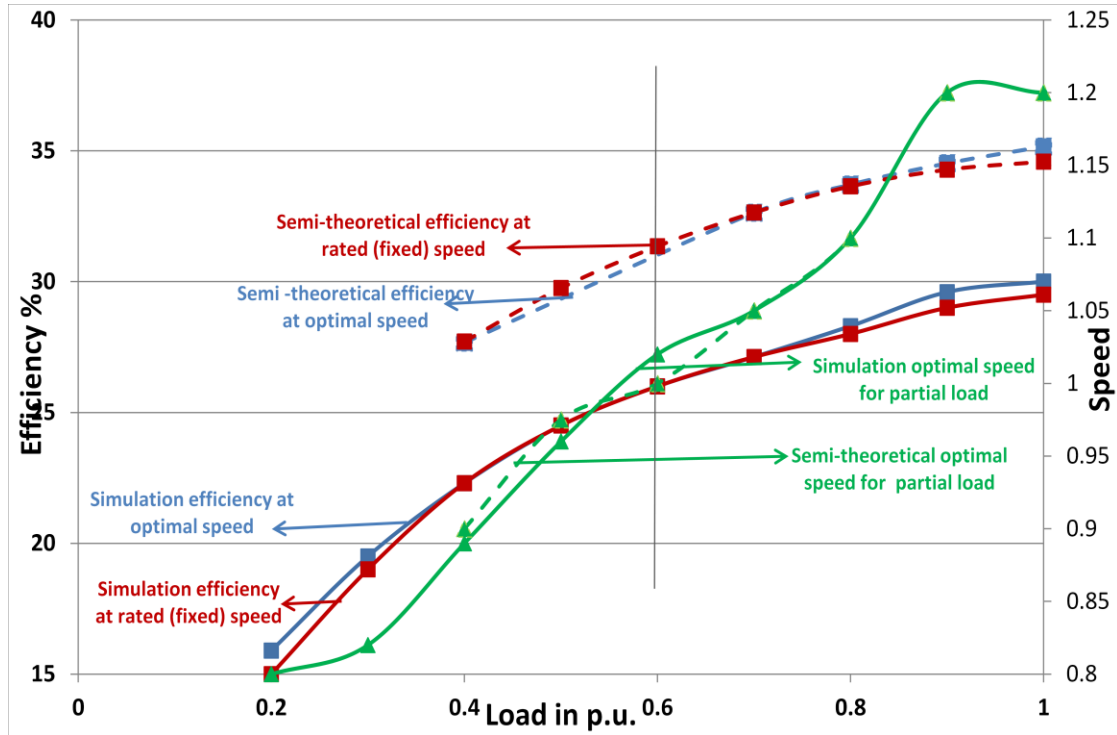


Fig. 6.14 Variable speed operational plot for a twin shaft gas turbine engine.

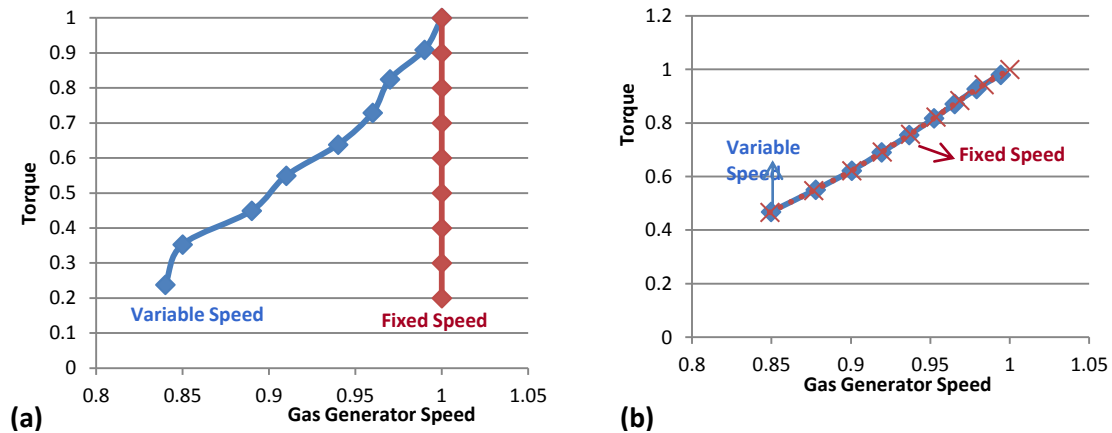


Fig. 6.15. Torque plot between (a) single shaft and (b) twin shaft.

The conclusion that we can come from above graphs and simulation studies is that variable speed operation has great benefit in improving part-load efficiency of gas turbine. The single shaft gas turbine engine can yield a better torque-speed characteristic provided the shaft speed is adjusted to the power output demand (i.e. to have control over the compressor efficiency curves). In contrast, the twin-shaft gas turbine doesn't benefit that much from variable speed operation as much as the single-shaft gas turbine. Because only the free power turbine itself benefits from the variable speed operation. Considering about 2/3 of produced power of gas turbine consumed by the compressor and the turbine itself running at a relatively high efficiency range, improving the free turbine efficiency really doesn't contribute much to improve the whole twin-shaft gas turbine efficiency. But overall, the variable speed operation does improve the part-load efficiency.

## **CHAPTER 7**

### **CONCLUSION AND FUTURE SCOPE**

This research has accomplished the following,

1. Developed a gas turbine engine model using component map matching.
2. Represented each gas turbine component by thermodynamic relationships
3. Understood the utilization of characteristics maps of compressor and turbine and implemented them for simulation.
4. Developed the gas turbine simulation architecture for design, off design, and transient simulation for VTB environment.
5. Working model of the gas turbine on VTB platform was shown and specific twin shaft case model was simulated.
6. Validated the VTB twin shaft gas turbine model with industry recognized gas turbine simulation software.
7. Performed a variable speed operation for cases of single shaft and twin shaft utilizing both VTB gas turbine model/ Gas Turb software and studied the possibility of increasing part load efficiency of a gas turbine engine.

8. Co-simulation studies highlighted the importance of dynamic parameters of different non-linear system when they are connected.
9. Finally increased the confidence of virtual test bed simulation environment and its use in future multi-disciplinary system level studies.

#### Future Scope

The project can be extended to the following

1. Further streamlining the VTB simulation environment to increase the robustness/adaptability of the gas turbine model implemented.
2. Expand the co-simulation study and develop it for an in-depth analysis.
3. VTB variable speed operation study of single shaft and twin shaft engines which was concentrated near the design point conditions can be further expanded to lower speed/start up regions of gas turbine.
4. Developing a robust control system for the gas turbine engine implemented in VTB.
5. Adding other standard industrial parameters for the gas turbine engine simulation model to make it effective and to satisfy various specific applications needs.



## REFERENCES

- [1] Andrea Lazzaretto and Andrea Toffolo, "Analytical and Neural Network Models for gas Turbine Design and Off-Design Simulation," *Int. J. Applied Thermodynamics* (173-182), December 2001.
- [2] Azmatulla Khan, "Simulation and robust control of marine gas turbines," thesis, IIT, India, July 2004.
- [3] Boyce M.P, "Gas Turbine Engineering Handbook," 3rd edition. Gulf Professional Publishing, 2006.
- [4] Chiras N, Evans C and Rees D, "Non linear gas turbine modeling using feed forward neural networks," *Proceeding of ASME Turbo Expo 2002*, ASME, Amsterdam, The Netherlands, GT-2002-30035, 2002.
- [5] Cohen.H, G. F. C. Rogers, H. I. H. Saravanamuttoo; "Gas Turbine Theory",1996.
- [6] Drummond,C.K.,Follen,G.J and Putt, C.W.Gas, "Turbine System Simulation: An object oriented approach," NASA technical memorandum, 106044,Lewis research centre,Cleveland, USA,1992.
- [7] Edward,W.O., and Taylor, B.L., "Dynamics of turbojet engine considered as Quasi static system", NACA-TN-2091,1950.
- [8] Eshwarprasad T,Ruixian Fang, Jamil A Khan and Roger Dougal, " Modeling and simulation of gas turbine on a virtual test bed," *Proceedings of the International Mechanical Engineering Congress and Exposition IMECE 12*, Texas, US IMECE 2012-87919,Nov 9-15,2012.
- [9] Giampaolo. T, "Gas Turbine Handbook Principles and Practices, 3rd edition," Fairmont Press, 2006.
- [10] Gunetti. P, Millis A and Thompson H, "A distributed intelligent agent architecture for gas turbine engine health management," 46th AIAA Aerospace Science meeting and Exhibit, AIAA, Reno, Nevada, USA, AIAA-2008-883, Jan 7-10, 2008.
- [11] Horlock H.J, "Aero-engine derivative gas turbines for power generations: thermodynamic and economic perspectives," *ASME Journal of Engineering for Gas Turbines and Power*, vol. 119, pp. 119-123, 1997.

- [12] Hosseini S.H, Khaledi H and Soltani M R, "New model based gas turbine fault diagnostics using ID engine model and nonlinear identification algorithms," ASME, GT-2009-59439 Jun 8-12, 2009, pp. 575-585.
- [13] Janitha Kanishka Suraweera,"Off-design performance prediction of gas turbines without use of compressor or turbine characteristics", Master thesis, Aerospace Engineering, Carleton University, Ottawa, Ontario,2011.
- [14] Joachim Kurzke, "Advanced User-Friendly Gas Turbine Performance Calculations On The Personal Computer", ASME 95-GT-147, ASME 1995.
- [15] Joachim Kurzke,"Smooth C 8.2and Smooth T 8.2 Map Softwares and manual",2011.
- [16] Joachim Kurzke,"GasTurb 11 Software/Details and manual",2011
- [17]Joachim Kurzke, "The Importance of Component Maps for Gas Turbine Performance Simulation", ISROMAC12-2008-20186, The 12th International Symposium on Transport Phenomena and Dynamics of Rotation Machinery", February 2008.
- [18] Joachim Kurzke and Claus Riegler, "A New Compressor Map Scaling Procedure for Preliminary Conceptional Design of Gas Turbines", ASME 2000-GT-0006, Proceedings of ASME IGIT Turbo Expo 2000, May 2000.
- [19] John A Reed and Abdollah A.Afjeh,"Computational Simulation of Gas Turbines: Part I-Foundation of component based models",ASME, paper# 99-GT-346.
- [20] John A. Reed and Abdollah A. Afjeh, "An object-oriented framework for distributed computational simulation of aerospace propulsion systems" Proceeding of the 4th USENIX conference on object oriented technologies and systems, Santa Fe, New Mexico, April 27-30, 1998.
- [21] Kim J H, Song T W, Kim T S and Ro S T, "Analysis of dynamic behavior of regenerative gas turbines," Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy, May 2001.
- [22] Kim J H, Song T W, Kim T S and Ro S T, "Model Development and transient behavior of heavy duty gas turbines," Journal of Engineering for gas turbines and power, ASME, July 2001, Vol. 123,pp.(589-594).
- [23] Kong C and Ki J, "Performance simulation of turboprop engine for basic trainer," 2001-GT-0391, Proceedings of ASME Turbo Expo 2001, New Orleans, Louisiana, USA, Jun 4-7, 2001.
- [24] Kyprianidis K.G. and Kalfas A.I., "Dynamic performance investigations of a turbojet engine using a cross-application visual oriented platform," Aeronautical Journal, March 2008.

- [25] Kyriazis A and Mathioudakis K, " Gas turbine fault diagnosis using fuzzy based decision fusion, " Journal of propulsion and power, "Vol.25, No.2 Mar-Apr 2009, pp. 335-343.
- [26] Lee A.S, Singh R and Probert S.D, "Modeling of the performance of a F100-PW229 equivalent engine under sea-level static conditions,"45th AIAA/other Joint Propulsion Conference and Exhibit, AIAA,Denver,Colorado,USA, AIAA 2009-5018, Aug 2-5, 2009.
- [27] Li, Y.G. " Performance analysis-based gas turbine diagnostics: a review," Journal of power and energy, Vol.216, Part A, Jul 2002, pp. 363-377.
- [28] Litt J.S " An optimal orthogonal decomposition method for kalman filter-based turbofan engine thrust estimation,"NASA Technical Memorandum, NASA/TM-2005-213864,Oct 2005.
- [29] Mathioudakis K, Stamatis A and Bonataki E, "Allocating the causes of performance deterioration in combined cycle gas turbine plants," Journal of Engineering for gas turbines and power, Vol.124, Apr 2002, pp. 256-262.
- [30] Mirandola A and Macor A, "Full load and part load operation of gas turbine-steam turbine combined plant," ISEC, Vol. 8-15, 1986.
- [31] Moran and Shapiro, "Fundamentals of Engineering thermodynamics," 5th edition.
- [32] Muir D.E et al " Health monitoring of a variable geometry gas turbines for the canadian navy," Journal of Engineering for gas turbine and power,Vol. 111, Apr 1989, pp. 244-250.
- [33] Mura F, De Doncker R W, Persigehl B, Jeschke P and Hameyer K, "Analysis of a gearless medium-voltage variable speed gas turbine," VGB PowerTech, vol. 91, No. 4, pp. 39-43, 2011.
- [34] Moran and Shapiro, "Fundamentals of Engineering thermodynamics," 5th edition.
- [35] NLR GSP: Gas turbine simulation program, <http://www.gspteam.com>,2006.
- [35] Norman P J, Galloway S J, Burt G M, Hill J E, Trainer D R, " Evaluation of dynamic interactions between aircraft gas turbine engine and electrical system," 4th International conference on power electronics, machines and drives, York, UK, pp. 671-695, 2-4 April 2008.
- [36] Philip P. Walsh and Paul Fletcher, "Gas turbine performance" Second Edition.
- [37] Quasi Z. Al-Hamdan, Munzer S. Y. Ebaid; "Modeling and Simulation of a Gas Turbine Engine for Power Generation," Journal of engineering for gas turbines, ASME, Vol.128, April 2006.

- [38] Ruixian Fang, W. Jiang, A. Monti, M. Zerby, G. Anderson, P Bernotas, J. Khan, "System-Level Dynamic Thermal Modeling and Simulation for an All-Electric Ship Cooling System in VTB", 2007 IEEE Electric Ship Technologies Symposium, pp462-469, Arlington, VA, May 2007.
- [39] Ruixian Fang, Wei Jiang, Jamil Khan, Roger Dougal, "System-Level Thermo Modeling and Co-simulation with Hybrid Power System for Future All Electric Ship", 2009 IEEE Electric Ship Technologies Symposium, pp.547-553, Baltimore, April 2009.
- [40] Ruixian Fang, Wei Jiang, Jamil Khan, Roger Dougal, "Thermal Modeling and Simulation of the Chilled Water System for Future All Electric Ship", 2011 IEEE Electric Ship Technologies Symposium, Alexandria, VA, April 10-13, 2011.
- [41] Sampath S., Gulati and Singh R, " Artificial intelligence techniques for gas turbine engine fault diagnostics," 38th AIAA/other Joint propulsion conference and exhibit, AIAA, Indianapolis, Indiana, USA, AIAA 2002-4308, July 7-10, 2002.
- [42] Sellers,J.F and Daniele C.J. "DYNGEN- A program for calculating steady state and transient performance of turbo-jets and turbo fan engines," NASA TN D-7901,1975.
- [43] Shaun R. Gaudet, "Development of Dynamic Modeling and Control System Design Methodology for Gas Turbines", Department of Mechanical and Aerospace Engineering, Carleton University, December 2007.
- [44] Simon D. L Bird, J Davison C, Volponi A and Iverson R. E, "Benchmarking gas path diagnostic methods: public approach," Proceeding of ASME Turbo Expo 2008: Power for Land, Sea and Air, Berlin, Germany, GT2008-51360, Jun 9-13, 2008.
- [45] Soon Kiat Yee, Jovica V.Milanovic, and F. Michael Hughes,"Validated models for gas turbines based on thermodynamic relationships," Journal: IEEE transaction of power systems, February 2010, Paper No. TPWRS-00411-2009.
- [46] Steinke, R.J, "STGSTK-a computer code for predicting multistage axial flow compressor performance by a mean line stage stacking method," NASA technical paper, NASA, 1982.
- [47] Teems E. Lovett, "Design and Implementation of a Multi-Discipline Simulation Environment", University of South Carolina 2002.
- [48] Tomita, J. T et al, "Nacelle design for mixed turbofan engines," Proceeding of ASME Turbo Expo 2006: Power for Land, Sea and Air, ASME, Marcelona, Spain, GT-2006-91212, May 8-11, 2006.
- [49] Wei Jiang, Jamil Khan, Roger A. Dougal, "Dynamic Centrifugal Compressor Model for System Simulation", Journal of Power Sources 158 (2006) 1333-1343, October 2005.

[50] Zhang N and Cai R, "Analytical solutions and typical characteristics of part-load performances of single-shaft gas turbine and its cogeneration," *Energy Conversion and Management*, vol. 43, pp. 1323-1337, 2002.

## APPENDIX A

VTB coding of Gas turbine model (Compressor, Combustor and Turbine)

////Maps used from Gas Turb Software [16] 99 NASA TM 101433 Example Compressor Map, 99 High Work Low Aspect Ratio Turbine NASA TM83655. Example GG Turbine Map, 99 A G A R D two-stage turbine Example P.Turbine Map.

///Language used C# in .Net framework/////

///// Model coding of the Compressor Component ///

Public class Compressor : NaturalEngine //uses Natural solver i.e. uses conservation principles

{

#region Data Members

//Natural Ports: /////Compressor Interaction Ports

private IInteractionPoint m\_oAirInlet;

private IInteractionPoint m\_oAirOutlet;

private IInteractionPoint m\_oTemperatureInlet;

private IInteractionPoint m\_oTemperatureOutlet;

private IInteractionPoint m\_oShaft;

//Internal natural ports:

private IInteractionPoint m\_oOutletHeatFlowRate;

//Parameters: //////////Compressor Parameters

private IInteractionPoint m\_oEfficiencyTable;

private IInteractionPoint m\_oMassFlowTable;

```

//////////////////////////////////// private IInteractionPoint
m_oPressureRatioTable;

    private IInteractionPoint m_oRatedTemperature;

    private IInteractionPoint m_oRatedSpeed;

    private IInteractionPoint m_oFAR;// Fuel Air ratio

    private IInteractionPoint m_oStdTemp;

    private IInteractionPoint m_oIdealAirTable;

    private IInteractionPoint m_oEnthalpyAir;

    private IInteractionPoint m_oNullSpeed;

    private IInteractionPoint m_oNullBeta;

    private IInteractionPoint m_oNullPin;

    private IInteractionPoint m_oNullRelSpeed;

    private IInteractionPoint m_oNullTin;

    private IInteractionPoint m_oBetaOffset;

    //States//For comparison the current time step/previous time step simulation values

    private IInteractionPoint m_oLastBeta;

    //Viewables ///Values to be plotted at the output window///

    private IInteractionPoint m_oWork;

    private IInteractionPoint m_oBeta;

    private IInteractionPoint m_oEfficiency;

    private IInteractionPoint m_oInletPressure;

    private IInteractionPoint m_oOutletPressure;

    private IInteractionPoint m_oMassFlow;

    private IInteractionPoint m_oInletTemperature;

```

```

private InteractionPoint m_oOutletTemperature;

private InteractionPoint m_oPressureRatio;

private InteractionPoint m_oInletN;

private InteractionPoint m_oSpeed;

private InteractionPoint m_oTorque;

//Calling the characteristic map table and other property table for interpolation ///

private BilinearInterpolation2 m_oEfficiencyInterpolation;

private BilinearInterpolation2 m_oMassFlowInterpolation;

private BilinearInterpolation2 m_oPressureRatioInterpolation;

private BilinearInterpolation2 m_oBetaInterpolation;

private BilinearInterpolation2 m_oIdealAirTableInterpolation;

private BilinearInterpolation2 m_oRelativePressureInterpolation;

private BilinearInterpolation2 m_oEnthalpyAirInterpolation;

private BilinearInterpolation2 m_oReverseEnthalpy;

//Variable declaration

private double m_dRelativeSpeedFactor;

private double m_dCp;

private double m_dmass;

#endregion

#region Constructor //Helps to link the component developed in the entity designer
with the engine

public Compressor(IEntity oEntity)

: base(oEntity)

```



```

{
    //Natural Ports:

    m_oAirInlet = GetIP("AirInlet");

    m_oAirOutlet = GetIP("AirOutlet");

    m_oTemperatureInlet = GetIP("TemperatureInlet");

    m_oTemperatureOutlet = GetIP("TemperatureOutlet");

    m_oShaft = GetIP("Shaft");

    //Internal natural ports:

    m_oOutletHeatFlowRate = GetIP("OutletHeatFlowRate");

    //Parameters:

    m_oEfficiencyTable = GetIP("EfficiencyTable");

    m_oMassFlowTable = GetIP("MassFlowTable");

    m_oPressureRatioTable = GetIP("PressureRatioTable");

    m_oRatedTemperature = GetIP("RatedTemperature");

    m_oRatedSpeed = GetIP("RatedSpeed");

    m_oFAR = GetIP("FAR");

    m_oStdTemp = GetIP("StdTemp");

    m_oNullSpeed = GetIP("NullSpeed");

    m_oNullBeta = GetIP("NullBeta");

    m_oNullPin = GetIP("NullPin");

    m_oNullRelSpeed = GetIP("NullRelSpeed");

    m_oNullTin = GetIP("NullTin");

    m_oBetaOffset = GetIP("BetaOffset");

```

```

    m_oIdealAirTable = GetIP("IdealAirTable");

    m_oEnthalpyAir = GetIP("EnthalpyAir");

    //States

    m_oLastBeta = GetIP("LastBeta");

    //Viewables

    m_oWork = GetIP("Work");

    m_oBeta = GetIP("Beta");

    m_oEfficiency = GetIP("Efficiency");

    m_oInletPressure = GetIP("InletPressure");

    m_oOutletPressure = GetIP("OutletPressure");

    m_oMassFlow = GetIP("MassFlow");

    m_oInletTemperature = GetIP("InletTemperature");

    m_oOutletTemperature = GetIP("OutletTemperature");

    m_oPressureRatio = GetIP("PressureRatio");

    m_oSpeed = GetIP("Speed");

    m_oTorque = GetIP("Torque");

    m_orelN = GetIP("relN");

    Linear = false;

    TimeDependent = true;

}

#endregion

#region NaturalEngine

    public override void OnSimulationStart()

```

```

{
    base.OnSimulationStart();

    //Setting up temperature equations//Part of the Jacobian
    SetJacobian(m_oTemperatureInlet, m_oOutletHeatFlowRate, 1.0);
    SetJacobian(m_oTemperatureOutlet, m_oOutletHeatFlowRate, -1.0);
    SetJacobian(m_oOutletHeatFlowRate, m_oTemperatureOutlet, -1.0);
}

public override void OnRunTimeChange(bool bParameterChanged) // To get the
value from the input command if the parameters gets updated.
{
    if (bParameterChanged)
    {
        string sPath;

        GetIPValue(m_oEfficiencyTable, out sPath);
        m_oEfficiencyInterpolation = LoadTable(sPath);

        GetIPValue(m_oMassFlowTable, out sPath);
        m_oMassFlowInterpolation = LoadTable(sPath);

        GetIPValue(m_oPressureRatioTable, out sPath);
        m_oPressureRatioInterpolation = LoadTable(sPath);

        m_oBetaInterpolation = LoadTable(sPath, true);

        GetIPValue(m_oIdealAirTable, out sPath);
        m_oIdealAirTableInterpolation = LoadTable(sPath);

        m_oRelativePressureInterpolation = LoadTable(sPath, true);
    }
}

```

```

        GetIPValue(m_oEnthalpyAir, out sPath);

        m_oEnthalpyAirInterpolation = LoadTable(sPath);

        m_oReverseEnthalpy = LoadTable(sPath, true);

        m_dRelativeSpeedFactor = sqrt(GetIPValue(m_oRatedTemperature) /
GetIPValue(m_oStdTemp)) / (GetIPValue(m_oRatedSpeed));

    }

}

public override void Step()// is the main body of the program and also used as
initialization step (null step)

    double dTemperatureInlet = GetAcross(m_oTemperatureInlet);

    double dTemperatureOutlet = GetAcross(m_oTemperatureOutlet);

    double dInletPressure = GetAcross(m_oAirInlet);

    double dOutletPressure = GetAcross(m_oAirOutlet);

    double dMassFlowrateValue = GetThrough(m_oAirInlet);

    double dMassFlowrateValueOutlet = GetThrough(m_oAirOutlet);

    double dShaft = GetAcross(m_oShaft);

    double dSpeed = (GetAcross(m_oShaft) / sqrt(dTemperatureInlet /
GetIPValue(m_oStdTemp)) * m_dRelativeSpeedFactor);

    double dTheta = dTemperatureInlet / 288.15;

    double dDelta = dInletPressure / 101325;

    double dtimestep = GetTimeStep();

    double dtime = GetCurrentTime();

    double dtime1 = GetCurrentStep();

```

```

double dPressureRatio = GetAcross(m_oAirOutlet) / GetAcross(m_oAirInlet);

double dBeta, dMassFlow, dEfficiency;

double dLastBeta = GetIPValue(m_oLastBeta);

try
{
    dBeta = dLastBeta;

    dMassFlow = GetThrough(m_oAirInlet);

    dEfficiency = m_oEfficiencyInterpolation.getValue(dLastBeta, dSpeed);

    m_dmass = m_oMassFlowInterpolation.getValue(dLastBeta, dSpeed) * dDelta
/ sqrt(dTheta);
}

catch (System.Exception e)
{
    RaiseException(e.Message);

    return;
}

double dP1 = m_oPressureRatioInterpolation.getValue(dBeta2, dSpeed);

double dMass1 = m_oMassFlowInterpolation.getValue(dBeta2, dSpeed) * dDelta /
sqrt(dTheta);

double dP2 = m_oPressureRatioInterpolation.getValue(dBeta1, dSpeed);

double dMass2 = m_oMassFlowInterpolation.getValue(dBeta1, dSpeed) * dDelta /
sqrt(dTheta);

```

//Enthalpy values as function of fuel-air ratio and temperature will be called from  
separate sub routine not shown here

```
double dPr1 = dP1;

double dPr2 = dP2;

if (dMass1 == dMass2)

{

    double dMassIn = m_oMassFlowInterpolation.getValue(dBeta2, dSpeed);

    SetJacobian(m_oAirInlet, m_oAirInlet, dMassIn * 1 / (101325 * sqrt(dTheta)));

    SetJacobian(m_oAirOutlet, m_oAirInlet, -dMassIn * 1 / (101325 * sqrt(dTheta)));

}

else

{

    double dSlope = (dPr1 - dPr2) / (dMass1 - dMass2);

    double dIntercept = dPr1 - dSlope * dMass1;

    //Jacobian for Pressure and Mass Flow

    SetJacobian(m_oAirInlet, m_oAirOutlet, 1 / (dSlope * dInletPressure));

    SetJacobian(m_oAirOutlet, m_oAirOutlet, -1 / (dSlope * dInletPressure));

    //set Bequiverlent

    SetBEquivalent(m_oAirInlet, dIntercept / dSlope);

    SetBEquivalent(m_oAirOutlet, -dIntercept / dSlope);

}

double dpr1 = m_oIdealAirTableInterpolation.getValue(dTemperatureInlet, z);

double dpr2 = dPressureRatio * dpr1;
```

```

double dTo2s = m_oRelativePressureInterpolation.getValue(z, dpr2);

double G = ((dT2 / dTemperatureInlet) - 1);///

double Lg1 = log((G * dEfficiency) + 1);

double Lg2 = log(dPressureRatio);

double y = Lg1 / Lg2;

double dGamma = 1 / (1 - y);

m_dCp = dMassFlow * (dH1 - dHC) / (dMassFlow * (dTemperatureInlet - dT2));

//Jacobian for temperature

double dTemperatureJacobian = 1 + (pow(dPressureRatio, (dGamma - 1) /
dGamma) - 1) / dEfficiency;

SetJacobian(m_oOutletHeatFlowRate, m_oTemperatureInlet,
dTemperatureJacobian);

//Jacobian for torque

double dShaftTemperatureJacobian = dMassFlow * m_dCp /

GetAcross(m_oShaft);

SetJacobian(m_oShaft, m_oTemperatureOutlet, dShaftTemperatureJacobian);

SetJacobian(m_oShaft, m_oTemperatureInlet, -dShaftTemperatureJacobian);

SetIPValue(m_oBeta, dBeta);

SetIPValue(m_oEfficiency, dEfficiency);

SetIPValue(m_oLastBeta, dBeta);

}

public override void PostStep() // Output –plot data after it is converged

{

```

```

double dInletPressure = GetAcross(m_oAirInlet);

double dOutletPressure = GetAcross(m_oAirOutlet);

double dInletTemperature = GetAcross(m_oTemperatureInlet);

double dOutletTemperature = GetAcross(m_oTemperatureOutlet);

double dMassFlow = GetThrough(m_oAirInlet);

double dSpeed = GetAcross(m_oShaft);

double dTorque = GetThrough(m_oShaft);

SetIPValue(m_oInletPressure, dInletPressure);

SetIPValue(m_oOutletPressure, dOutletPressure);

SetIPValue(m_oPressureRatio, dOutletPressure / dInletPressure);

SetIPValue(m_oInletTemperature, dInletTemperature);

SetIPValue(m_oOutletTemperature, dOutletTemperature);

SetIPValue(m_oMassFlow, m_dmss);

SetIPValue(m_oSpeed, dSpeed);

SetIPValue(m_oTorque, GetThrough(m_oShaft));

SetIPValue(m_oWork, dMassFlow * m_dCp * (dInletTemperature -
dOutletTemperature));

}

#endregion

#region Methods //This part reads several input tables to input into the code

private double[] ParseString(string sLine)

{

    string[] sValues = sLine.Split(',');

```



```

        return sValues.Select(oValue => double.Parse(oValue)).ToArray();
    }

    private BilinearInterpolation2 LoadTable(string sPath, bool bSwitchXandZ = false)
    {
        StreamReader oReader = null;

        List<Point3D> oData = new List<Point3D>();

        try
        {
            oReader = new StreamReader(new FileStream(sPath, FileMode.Open));

            double[] rgdBeta = ParseString(oReader.ReadLine());

            while (!oReader.EndOfStream)
            {
                double[] rgdData = ParseString(oReader.ReadLine());

                for (int i = 1; i < rgdBeta.Length; i++)
                {
                    oData.Add(new Point3D(rgdBeta[i], rgdData[0], rgdData[i]));
                }
            }
        }
        catch
        {
            RaiseException("Unable to read file");

            return null;
        }
    }

```

```

    }
    finally
    {
        if (oReader != null)
        {
            oReader.Close();
        }
    }
    if (bSwitchXandZ)
    {
        Point3D[] oSwitchedData = oData.Select(oPoint => new Point3D(oPoint.dY,
oPoint.dZ, oPoint.dX)).ToArray();
        return new BilinearInterpolation2(oSwitchedData);
    }
    return new BilinearInterpolation2(oData.ToArray());
}
#endregion
}
}

///// Model coding of the Combustor Component /////

public class Combustor:NaturalEngine
{
    #region Data Members

```

```

//Natural Ports

private IInteractionPoint m_oTemperatureInlet;

private IInteractionPoint m_oTemperatureOutlet;

private IInteractionPoint m_oPressureInlet;

private IInteractionPoint m_oPressureOutlet;

private IInteractionPoint m_oFuelInlet;

//Internal Natural Port

private IInteractionPoint m_oOutletHeatFlowRate;

private IInteractionPoint m_oInletMassFlowRate;

private IInteractionPoint m_oFuelMassFlowRate;

//private IInteractionPoint m_oTemp;

//Parameter

private IInteractionPoint m_oFuelHeatingValue;

private IInteractionPoint m_oPressureLoss;

private IInteractionPoint m_oCombustionChamberEfficiency;

private IInteractionPoint m_oNullMassIn;

private IInteractionPoint m_oNullfuelIn;

private IInteractionPoint m_oNullTin;

private IInteractionPoint m_oNullTout;

private IInteractionPoint m_oPin;

//Viewables

private IInteractionPoint m_oInletTemperature;

private IInteractionPoint m_oInletFlowrate;

```

```

private IInteractionPoint m_oInletPressure;

private IInteractionPoint m_oOutletFlowrate;

private IInteractionPoint m_oOutletPressure;

private IInteractionPoint m_oOutletTemperature;

private IInteractionPoint m_oFuelFlowRate;

private double m_dFuelHeatingValue;

private double m_dCombustionChamberEfficiency;

private double m_dPressureLoss;

private double Tref = 288.15;

#endregion

#region Constructor

public Cb2(IEntity oEntity)

    : base(oEntity)

{

    //Natural Ports

    m_oTemperatureInlet = GetIP("Temperature2");

    m_oTemperatureOutlet = GetIP("Temperature3");

    m_oPressureInlet = GetIP("MassInlet2");

    m_oPressureOutlet = GetIP("MassOutlet3");

    m_oFuelInlet = GetIP("FuelInlet");

    //Internal natural Port

    m_oOutletHeatFlowRate = GetIP("OutletHeatFlowRate");

    m_oInletMassFlowRate = GetIP("InletMassFlowRate");

```

```

m_oFuelMassFlowRate = GetIP("FuelMassFlowRate");

//m_oTemp = GetIP("Temp");

//Parameters

m_oFuelHeatingValue = GetIP("FuelHeatingValue");

m_oPressureLoss = GetIP("PressureLoss");

m_oCombustionChamberEfficiency = GetIP("CombustionChamberEfficiency");

m_oNullMassIn = GetIP("NullMassIn");

m_oNullfuelIn = GetIP("NullfuelIn");

m_oNullTin = GetIP("NullTin");

m_oNullTout = GetIP("NullTout");

m_oPin = GetIP("Pin");

//Viewables

m_oInletTemperature = GetIP("InletTemperature");

m_oInletFlowrate = GetIP("InletFlowrate");

m_oInletPressure = GetIP("InletPressure");

m_oOutletFlowrate = GetIP("OutletFlowrate");

m_oOutletPressure = GetIP("OutletPressure");

m_oOutletTemperature = GetIP("OutletTemperature");

m_oFuelFlowRate = GetIP("FuelFlowRate");

Linear = false;

}

#endregion

public override void OnSimulationStart()

```

```

{
    base.OnSimulationStart();

    SetJacobian(m_oPressureInlet, m_oInletMassFlowRate, 1); //set m2 = M
    SetJacobian(m_oPressureOutlet, m_oInletMassFlowRate, -1); //set m3 = -M-mf
    SetJacobian(m_oPressureOutlet, m_oFuelMassFlowRate, -1); //set m3 = -M-Mf
    SetJacobian(m_oFuelInlet, m_oFuelMassFlowRate, 1); //set mf = Mf
    SetJacobian(m_oTemperatureInlet, m_oOutletHeatFlowRate, 1); //set q2 = q3 - mf *
(HV*eta cp3*T3)
    SetJacobian(m_oTemperatureOutlet, m_oOutletHeatFlowRate, -1); //set q2 = Q
    SetJacobian(m_oInletMassFlowRate, m_oPressureOutlet, -1); //set P3 = 0.95 * P2
    SetJacobian(m_oFuelMassFlowRate, m_oFuelInlet, 1); //set Pf = 101325
}

public override void OnRunTimeChange(bool bParameterChanged)
{
    if (bParameterChanged)
    {
        m_dFuelHeatingValue = GetIPValue(m_oFuelHeatingValue);
        m_dPressureLoss = GetIPValue(m_oPressureLoss) * PERCENT;
        m_dCombustionChamberEfficiency =
GetIPValue(m_oCombustionChamberEfficiency) * PERCENT;
    }
}

public override void Step() or NullStep()

```

```

{
    double dTemperatureInlet = GetIPValue(m_oNullTin);
    double dTemperatureOutlet = GetIPValue(m_oNullTout);
    double dMassFlowInlet = GetIPValue(m_oNullMassIn);
    double dFuelInlet = GetIPValue(m_oNullfuelIn);
    double dPin = GetIPValue(m_oPin);
    double dFAR = dFuelInlet / dMassFlowInlet;

    ///Cp data taken from curve fit relation available gas turb details [16]
    double Cp3 = Cp2 + (dFAR / (dFAR + 1)) * CpB;

    // Jacobian changed for including Tref
    double dJacobianT2 = dMassFlowInlet * (Cp2);
    double dJacobianT3 = (dMassFlowInlet + dFuelInlet) * (Cp3);
    double dJacobianM = Cp3 * Tref - Cp2 * Tref;
    double dJacobianmf = m_dFuelHeatingValue *
m_dCombustionChamberEfficiency + Cp3 * Tref;

    SetJacobian(m_oInletMassFlowRate, m_oPressureInlet, 1 - m_dPressureLoss);
    SetJacobian(m_oOutletHeatFlowRate, m_oTemperatureInlet, dJacobianT2);
    SetJacobian(m_oOutletHeatFlowRate, m_oTemperatureOutlet, -dJacobianT3);
    SetJacobian(m_oOutletHeatFlowRate, m_oInletMassFlowRate, dJacobianM);
    SetJacobian(m_oOutletHeatFlowRate, m_oFuelMassFlowRate, dJacobianmf);
    SetBEquivalent(m_oFuelMassFlowRate, 101325);
}

```

```

public override void PostStep()
{
    double dInletTemperature = GetAcross(m_oTemperatureInlet);
    double dInletFlowrate = GetThrough(m_oPressureInlet);
    double dInletPressure = GetAcross(m_oPressureInlet);
    double dOutletFlowrate = GetThrough(m_oPressureOutlet);
    double dOutletPressure = GetAcross(m_oPressureOutlet);
    double dOutletTemperature = GetAcross(m_oTemperatureOutlet);
    double dFuelFlowRate = GetThrough(m_oFuelInlet);
    SetIPValue(m_oInletTemperature, dInletTemperature);
    SetIPValue(m_oInletFlowrate, dInletFlowrate);
    SetIPValue(m_oInletPressure, dInletPressure);
    SetIPValue(m_oOutletFlowrate, dOutletFlowrate);
    SetIPValue(m_oOutletPressure, dOutletPressure);
    SetIPValue(m_oOutletTemperature, dOutletTemperature);
    SetIPValue(m_oFuelFlowRate, dFuelFlowRate);
}
}
}

///// Model coding of the Turbine Component /////

public class Turbine : NaturalSignalEngine
{
    #region Data Members

```



```

//Natural Ports:

private IInteractionPoint m_oAirInlet;

private IInteractionPoint m_oAirOutlet;

private IInteractionPoint m_oTemperatureInlet;

private IInteractionPoint m_oTemperatureOutlet;

private IInteractionPoint m_oShaft;

//Internal natural ports:

private IInteractionPoint m_oOutletHeatFlowRate;

//Signal Input Ports

private IInteractionPoint m_oFAR1; to get the current Fuel added value from the
combustor

//Parameters:

private IInteractionPoint m_oEfficiencyTable;

private IInteractionPoint m_oMassFlowTable;

private IInteractionPoint m_oPressureRatioTable;

private IInteractionPoint m_oRatedTemperature;

private IInteractionPoint m_oRatedSpeed;

private IInteractionPoint m_oFAR;

private IInteractionPoint m_oStdTemp;

private IInteractionPoint m_oNullSpeed;

private IInteractionPoint m_oNullBeta;

private IInteractionPoint m_oNullPin;

private IInteractionPoint m_oNullRelSpeed;

```

```

private InteractionPoint m_oNullTin;

private InteractionPoint m_oBetaOffset;

private InteractionPoint m_oEnthalpyTable;

private InteractionPoint m_oEntropyTable;

//States

private InteractionPoint m_oLastBeta;

//Viewables

private InteractionPoint m_oWork;

private InteractionPoint m_oBeta;

private InteractionPoint m_oEfficiency;

private InteractionPoint m_oInletPressure;

private InteractionPoint m_oOutletPressure;

private InteractionPoint m_oMassFlow;

private InteractionPoint m_oInletTemperature;

private InteractionPoint m_oOutletTemperature;

private InteractionPoint m_oPressureRatio;

private InteractionPoint m_oSpeed;

private InteractionPoint m_oTorque;

private InteractionPoint m_orelN;

private BilinearInterpolation2 m_oEfficiencyInterpolation;

private BilinearInterpolation2 m_oMassFlowInterpolation;

private BilinearInterpolation2 m_oPressureRatioInterpolation;

private BilinearInterpolation2 m_oBetaInterpolation;

```

```

private BilinearInterpolation2 m_oEnthalpyInterpolation;

private BilinearInterpolation2 m_oReverseEnthalpy;

private BilinearInterpolation2 m_oEntropyInterpolation;

private BilinearInterpolation2 m_oReverseEntropy;

private double m_dRelativeSpeedFactor;

private double m_dCp;

private double m_dmass;

#endregion

#region Constructor

public TurbineVertical1(IEntity oEntity)

    : base(oEntity)
{

    //Natural Ports:

    m_oAirInlet = GetIP("AirInlet");

    m_oAirOutlet = GetIP("AirOutlet");

    m_oTemperatureInlet = GetIP("TemperatureInlet");

    m_oTemperatureOutlet = GetIP("TemperatureOutlet");

    m_oShaft = GetIP("Shaft");

    //Internal natural ports:

    m_oOutletHeatFlowRate = GetIP("OutletHeatFlowRate");

    //Signal Input Ports

    m_oFAR1 = GetIP("FAR1");

```

```

//Parameters:

m_oEfficiencyTable = GetIP("EfficiencyTable");

m_oMassFlowTable = GetIP("MassFlowTable");

m_oPressureRatioTable = GetIP("PressureRatioTable");

m_oRatedTemperature = GetIP("RatedTemperature");

m_oRatedSpeed = GetIP("RatedSpeed");

m_oFAR = GetIP("FAR");

m_oStdTemp = GetIP("StdTemp");

m_oNullSpeed = GetIP("NullSpeed");

m_oNullBeta = GetIP("NullBeta");

m_oNullPin = GetIP("NullPin");

m_oNullRelSpeed = GetIP("NullRelSpeed");

m_oNullTin = GetIP("NullTin");

m_oBetaOffset = GetIP("BetaOffset");

m_oEnthalpyTable = GetIP("EnthalpyTable");

m_oEntropyTable = GetIP("EntropyTable");

//States

m_oLastBeta = GetIP("LastBeta");

//Viewables

m_oWork = GetIP("Work");

m_oBeta = GetIP("Beta");

m_oEfficiency = GetIP("Efficiency");

m_oInletPressure = GetIP("InletPressure");

```

```

    m_oOutletPressure = GetIP("OutletPressure");

    m_oMassFlow = GetIP("MassFlow");

    m_oInletTemperature = GetIP("InletTemperature");

    m_oOutletTemperature = GetIP("OutletTemperature");

    m_oPressureRatio = GetIP("PressureRatio");

    m_oSpeed = GetIP("Speed");

    m_oTorque = GetIP("Torque");

    m_orelN = GetIP("relN");

    Linear = false;
}

#endregion

#region NaturalEngine

public override void OnSimulationStart()
{
    base.OnSimulationStart();

    //Setting up temperature equations

    SetJacobian(m_oTemperatureInlet, m_oOutletHeatFlowRate, 1.0);

    SetJacobian(m_oTemperatureOutlet, m_oOutletHeatFlowRate, -1.0);

    SetJacobian(m_oOutletHeatFlowRate, m_oTemperatureOutlet, -1.0);
}

public override void OnRunTimeChange(bool bParameterChanged)
{
    if (bParameterChanged)

```

```

{
    string sPath;

    GetIPValue(m_oEfficiencyTable, out sPath);
    m_oEfficiencyInterpolation = LoadTable(sPath);
    GetIPValue(m_oMassFlowTable, out sPath);
    m_oMassFlowInterpolation = LoadTable(sPath);
    GetIPValue(m_oPressureRatioTable, out sPath);
    m_oPressureRatioInterpolation = LoadTable(sPath);
    m_oBetaInterpolation = LoadTable(sPath, true);
    GetIPValue(m_oEnthalpyTable, out sPath);
    m_oEnthalpyInterpolation = LoadTable(sPath);
    m_oReverseEnthalpy = LoadTable(sPath, true);
    GetIPValue(m_oEntropyTable, out sPath);
    m_oEntropyInterpolation = LoadTable(sPath);
    m_oReverseEntropy = LoadTable(sPath, true);
    m_dRelativeSpeedFactor =
sqrt(GetIPValue(m_oRatedTemperature)/GetIPValue(m_oStdTemp)) /
(GetIPValue(m_oRatedSpeed));
}
}

public override void SignalStep()
{
}

```

```

public override void NullStep()
{
    double dTemperatureInlet = GetIPValue(m_oNullTin);

    double dInletPressure = GetIPValue(m_oNullPin);

    double dBeta = GetIPValue(m_oNullBeta);

    double dInitialspeed = GetIPValue(m_oNullSpeed);

    double dTheta = dTemperatureInlet / GetIPValue(m_oStdTemp);

    double dDelta = dInletPressure / 101325;

    double FAR = GetIPValue(m_oFAR);

    double dSpeed = (dInitialspeed / sqrt(dTemperatureInlet /
GetIPValue(m_oStdTemp))) * m_dRelativeSpeedFactor);

    m_dCp = Cp2 + (FAR / (FAR + 1)) * CpB;

    double dGamma = m_dCp / (m_dCp - 287);

    double dPressureRatio = m_oPressureRatioInterpolation.getValue(dBeta,
dSpeed);

    double dMassFlow = m_oMassFlowInterpolation.getValue(dBeta, dSpeed) *
dDelta / sqrt(dTheta);

    double dEfficiency = m_oEfficiencyInterpolation.getValue(dBeta, dSpeed);

    double dOutletPressure = dInletPressure / dPressureRatio;

    m_dmass = dMassFlow;

    //////////////////////////////////////

    double dP1 = m_oPressureRatioInterpolation.getValue(dBeta2, dSpeed);

```

```

        double dMass1 = m_oMassFlowInterpolation.getValue(dBeta2, dSpeed) * dDelta
/ sqrt(dTheta);

        double dP2 = m_oPressureRatioInterpolation.getValue(dBeta1, dSpeed);

        double dMass2 = m_oMassFlowInterpolation.getValue(dBeta1, dSpeed) * dDelta
/ sqrt(dTheta);

        double dPr1 = dP1;

        double dPr2 = dP2;

        if (dMass1 == dMass2)

        {

            double dMassIn = m_oMassFlowInterpolation.getValue(dBeta2, dSpeed);

            SetJacobian(m_oAirInlet, m_oAirInlet, dMassIn * 1 / (101325 * sqrt(dTheta)));

            SetJacobian(m_oAirOutlet, m_oAirInlet, -dMassIn * 1 / (101325 *
sqrt(dTheta)));

        }

        else

        {

            double dSlope = (dPr1 - dPr2) / (dMass1 - dMass2);

            double dIntercept = dPr1 - dSlope * dMass1;

            //Jacobian for Pressure and Mass Flow

            SetJacobian(m_oAirInlet, m_oAirInlet, 1 / (dSlope * dOutletPressure));

            SetJacobian(m_oAirOutlet, m_oAirInlet, -1 / (dSlope * dOutletPressure));

```



```

//set Bequivalent

SetBEquivalent(m_oAirInlet, dIntercept / dSlope);

SetBEquivalent(m_oAirOutlet, -dIntercept / dSlope);

}

////////////////////////////////////

#region EnthalpyEntropy

//Enthalpy and Entropy

double V1 = m_oEntropyInterpolation.getValue(dTemperatureInlet, FAR);

double V2 = V1 + log(1 / dPressureRatio);

double dT4s = m_oReverseEntropy.getValue(FAR, V2);

//Enthalpy as function temperature and F/A data from literature

////////////////////////////////////

//-----Temperature Equation and speed equation-----

double dTemperatureJacobian = m_oReverseEnthalpy.getValue(FAR, dhT4);

SetBEquivalent(m_oOutletHeatFlowRate, -dTemperatureJacobian);

//Jacobian for torque

double dShaftTemperatureJacobianIN = dMassFlow * dRA / dInitialspeed;

double dShaftTemperatureJacobianOUT = dMassFlow * dRC / dInitialspeed;

SetJacobian(m_oShaft, m_oTemperatureInlet, dShaftTemperatureJacobianIN);

SetJacobian(m_oShaft, m_oTemperatureOutlet, -

dShaftTemperatureJacobianOUT);

double dShaftBE = dMassFlow * (dbA - dbC) / dInitialspeed;

```

```

SetBEquivalent(m_oShaft, -dShaftBE);

SetIPValue(m_oBeta, dBeta);

SetIPValue(m_oEfficiency, dEfficiency);

SetIPValue(m_oLastBeta, dBeta);

SetIPValue(m_orelN, dSpeed);
}

}

public override void PostStep()
{
    double dInletPressure = GetAcross(m_oAirInlet);

    double dOutletPressure = GetAcross(m_oAirOutlet);

    double dInletTemperature = GetAcross(m_oTemperatureInlet);

    double dOutletTemperature = GetAcross(m_oTemperatureOutlet);

    double dMassFlow = GetThrough(m_oAirInlet);

    double dTorque = GetThrough(m_oShaft);

    double dSpeed = GetAcross(m_oShaft);

    SetIPValue(m_oInletPressure, dInletPressure);

    SetIPValue(m_oOutletPressure, dOutletPressure);

    SetIPValue(m_oPressureRatio, dInletPressure / dOutletPressure);

    SetIPValue(m_oInletTemperature, dInletTemperature);

    SetIPValue(m_oOutletTemperature, dOutletTemperature);

    SetIPValue(m_oMassFlow, m_dmss);

    SetIPValue(m_oSpeed, dSpeed);
}

```

```
SetIPValue(m_oTorque, GetThrough(m_oShaft));  
  
SetIPValue(m_oWork, (dTorque*dSpeed));  
  
}  
  
#endregion
```

Region method // same as the compressor component.

//Note: the exhaust/inlet component code is same as the combustor and employs only the pressure equations. The other components such the shaft and pump uses the same procedure of coding as shown earlier.