Prediction of Inlet Distortion Transfer Through the Blade Rows in a Transonic Axial Compressor

by

John F. Ryman

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Mechanical Engineering

Approved:

_____

W. F. O'Brien, Committee Chair

_____ _____

P. S. King C. L. Dancey

June 10, 2003

Blacksburg, VA

Keywords: Total pressure distortion, Inlet distortion, Distortion transfer, High cycle fatigue

Prediction of Inlet Distortion Transfer Through the Blade Rows in a Transonic Axial Compressor

by

John F. Ryman

(ABSTRACT)

Inlet total pressure non-uniformities in axial flow fans and compressors can contribute to the loss of component structural integrity through high cycle fatigue (HCF) induced by the excitation of blade vibratory modes. As previous research has shown total pressure distortion to be the dominant HCF driver in aero engines [Manwaring et al, 1997], an understanding of its transfer through, and impact on, subsequent turbomachine stages and engine components is an important topic for assessment. Since current modeling techniques allow for total pressure distortion magnitudes to be directly related to blade vibratory response, the prediction of downstream distortion patterns from an upstream measurement would allow for the inference of the vibratory response of downstream blade rows to an inlet total pressure distortion.

Nonlinear Volterra theory can be used to model any periodic nonlinear system as an infinite sum of multidimensional convolution integrals. A semi-empirical model has been developed using this theory by assuming that a distortion waveform is a periodic signal that is being presented to a nonlinear system, the compressor being the system. The use of Volterra theory in nonlinear system modeling relies on the proper identification of the Volterra kernels, which make up the transfer function that defines the system's impulse response characteristics. Once the kernels of a system are properly identified, the

system's response can be calculated for any arbitrary input. This model extracts these kernels from upstream and downstream total pressure distortion measurements of a transonic rotor of modern design. The resulting transfer function is then applied to predict distortion transfer at new operating points on the same rotor and compared with the measured data.

The judicious choice of distortion measurement data allows predictions of the downstream distortion content based on a measured non-uniform inlet flow at conditions different from those at which the transfer function was derived. This allows for the determination of downstream total pressure distortion that has the potential to excite blade vibratory modes that could lead to HCF under operating conditions other than those at which the data was taken, such as varying inlet distortion patterns, mass flow settings, rotational speeds, and inlet geometry.

This report presents the creation of a Volterra model in order to predict distortion transfer in axial flow fans and compressors. This model, in three variations, is applied to a variety of distortions and compressor operating conditions as measured in the ADLARF tests at the Compressor Research Facility. Predictions are compared with data from the test and final results are also compared with two previous studies conducted at Virginia Tech using the same experimental data. Using the Volterra model it is shown that, with appropriate limitations, distortion transfer can be predicted for flow conditions different from those used for calibration. The model is considered useful for both performance and HCF investigations.

Acknowledgements

I would like to thank my graduate committee for serving in this capacity. A special thanks goes to Dr. O'Brien for his initial willingness to take on a new graduate student who had been away from academia for almost 15 years and for his continued guidance and support over the past 2 years. Dr. Pete King is owed a special thank you as well for being a willing and available mentor and friend. I would also like to thank Dr. Douglas Rabe and the Compressor Research Facility at Wright Patterson Air Force Base for funding this project and providing the experimental data for the study. Dr. Patrick Reisenthel, who was kind enough to send me a copy of what ended up being one of my primary reference documents, is owed a very large thank you. This document helped tremendously with the calculation of the higher-order responses of the system.

Thank you to Jon Luedke and Matt Small. Jon, my predecessor on this project, for first introducing me to Volterra theory and then patiently answering all of my questions even while he was busy starting a new job and preparing to be married. Matt Small, even though we have never met, performed the majority of the initial data conditioning saving me a great deal of time and effort.

The friendship of everyone in the Turbolab has provided a constant source of support and camaraderie. Thank you to Rob, Mac, Jon, Kevin, Joe, Melissa, Darius, Mano, Matthew, Katie, and Mike.

I would like to thank Sandra for a number of things. First, for introducing me to the sport of mountain biking, which has provided a fun and often-painful release for the last year. Second, for being either a source of encouragement or prudent judgment…whichever was more appropriate; and finally, for her friendship and companionship. I would also like to thank Scott, Carrie, Larry, and Judy for their kindness over the past year.

My parents, Franklin and Mary Ryman, have been a constant source of guidance, support, and love for all of my life. Even though most people my age

are settled down with children and are looking forward to their next promotion, my parents enthusiastically supported my decision to start over and return to school. I owe more than I could ever express to them.

Table of Contents

Table of Tables

Table of Figures

Table of Equations

1 Introduction

Historically, the assumption has been made that the airflow through axial compressors in aircraft engines is uniform. As the performance continues to increase in modern aero engines, the capabilities of the aircraft that they are put in also increase. It is these capabilities that cause the inlet flow to the engines of these aircraft to often be non-uniform. Extreme flight maneuvers, operating within the wake of other aircraft, and firing wing-mounted weapons are some of the operational causes of non-uniform inlet flow (see Figure 1.1).

Figure 1-1: Capabilities of modern day aircraft increase the potential that the inlet flow to their engines is non-uniform [www.airforce.com].

Another possible cause of non-uniform inlet flow to modern aircraft engines is advanced inlet configurations such as the serpentine inlets used on stealth aircraft. Additionally, when vertical take-off and landing aircraft (VTOL) are hovering, the ingestion and mixing of hot exhaust gases with the cooler atmospheric air could also cause the inlet pressure profile to the plane's engines to be non-uniform (see Figures 1.2 and 1.3).

Figure 1-2: F117 Nighthawk has an 'S' shaped or stealth inlet, increasing the potential for the inlet flow to its engines to be non-uniform [www.aladdinsoft.com].

Figure 1-3: The AV-8B Harrier. The ingestion and subsequent mixing of the hot exhaust gases with the cooler atmospheric air could cause the inlet pressure profile to the aircraft's engine to be non-uniform [www.globalsecurity.org].

Non-uniform inlet flow, also known as inlet distortion, has negative short term and long-term effects on a compressor. The short-term effects are most easily seen on a compressor map as a loss in pressure rise, reduction in mass flow, and/or a reduction of stall margin (see Figure 1.4). A great deal has been done and is known about the effects of non-uniform inlet flow on performance in order to prevent compressor stall and surge.

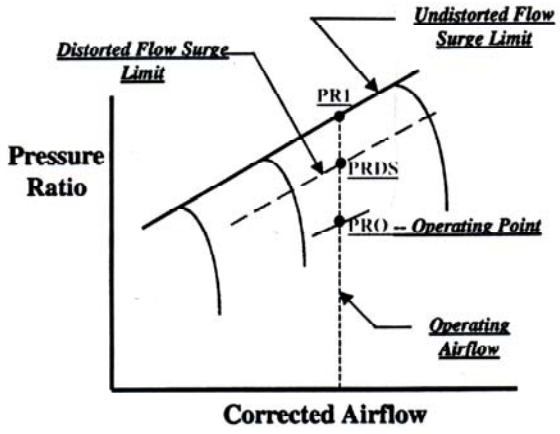

Figure 1-4: Compressor map showing the loss in surge limit caused by a distorted inlet flow [ARP-1420, 2002].

The long-term effects associated with inlet distortion are aeromechanical in nature and could result in the excitation of blade vibratory modes leading to high cycle fatigue. High cycle fatigue (HCF) is defined as "metal fatigue that results in cracking or fracture from a large number of stress cycles well below the yield strength of the material". Several hundred incidents of varying severity have occurred in U.S. Air Force aircraft engines since the late 1960s [Air Force Scientific Advisory Board, 1992]. Most of these incidents have been attributed to HCF. Due to the high number of HCF related incidents, the Air Force declared the elimination of HCF as their number one priority in the 1980s. Previous research has shown total pressure distortion to be the dominant HCF driver in aero engines [Manwaring et al, 1997]. Since current modeling techniques allow for total pressure distortion magnitudes to be directly related to blade vibratory

response, the prediction of downstream distortion patterns from an upstream measurement would allow for the inference of the vibratory response of downstream blade rows to an inlet total pressure distortion. This provides the motivation for my research.

The goals of my research are as follows: (1) create a semi-empirical model in order to predict total pressure distortion transfer in fans and compressors; (2) extract the frequency components of the non-uniform pressure profiles to enable the prediction of blade response; and (3) use the combined pressure and blade response predictions as a tool to identify potential HCF problem areas. This work presents the development and subsequent testing of a modeling technique that is based on nonlinear Volterra theory. The emphasis of this research will be placed on the accurate prediction of the magnitude of the dominant frequency content associated with total pressure distortion transfer. The goal is to be able to make this prediction to within a 20% error at any point along the blade of the compressor. This error rate was chosen to coincide with the Integrated High Performance Turbine Engine Technology (IHPTET) program's goal of blade vibratory stress prediction to within 20% error anywhere on a blade.

2 Literature Review

The history of experimental and analytical research devoted to better understanding compressor performance to varying inlet conditions is extensive, dating back to the late 1950s. As will be shown in the subsequent section, most of this work has focused on the change in compressor performance as caused by non-uniform inlet flow, specifically the reduction in stall margin and pressure rise. More recently, the aeromechanical effects (blade vibratory response) of a compressor rotor to non-uniform flow have received a growing amount of attention.

Despite significant gains made to date, there still exists the need to further our understanding of compressor aeromechanical response to a non-uniform inlet flow. Many different modeling techniques have been developed in this area, with several of them discussed in this chapter. All techniques discussed in this chapter have had varied success coupled with certain limitations or assumptions that must be satisfied. Moreover, the application of nonlinear Volterra theory as an aerodynamic response modeling method will be introduced and reviewed.

2.1 Inlet Distortion Effects on Compressor Performance and Blade Response

In the mid 1970's Danforth [1975] realized that the effects caused by an inlet distortion on compressor stall margin and blade response can be dramatically different. He examined several mechanisms that have the potential to excite blade vibration including circumferential total pressure distortion. Through this examination, he discovered that a high-intensity narrow-sector distortion, despite being too small to affect surge margin, was capable of generating severe resonant response in rotor blades and stator vanes. Danforth also stated that the

primary mechanism to instigate blade vibration was the coupling of an aerodynamic forcing function – circumferential total pressure distortion being the primary one considered – with a natural mode of the blade. The severity of the response depends upon the effectiveness of energy input to the blades. These findings led Danforth to define a distortion index for blade vibration to be used as a design alert to potential vibratory problems.

Following the same school of thought as Danforth [1975], Longley and Greitzer [1992] summarized two different types of distortion indices, the "K" series and the $DC(\theta_{crit})$. Both of these classify the severity of the distortion as a function of the size of the total pressure region that is below the average inlet value. The two indices differ in that the "K" series defines the distortion using all of the distorted sectors while the $DC(\theta_{crit})$ indices uses only the sector with the strongest intensity.

Reid [1969] performed a series of experiments where he examined how a compressor's performance varied with different inlet distortions. Here he defined the angle of spoiling, which is the angular width of the low total pressure region. Figure 2-1 illustrates how the surge delivery static pressure varies as a function of the angle of spoiling. It also shows that there is little change in the surge margin beyond a certain "critical angle". Figure 2-1 also shows the effect of sub-dividing the total angle of spoiling into different numbers of equal sections. This figure showing the effect of sub-division also demonstrates that the greatest effect on peak pressure loss is when there is only one distorted section.

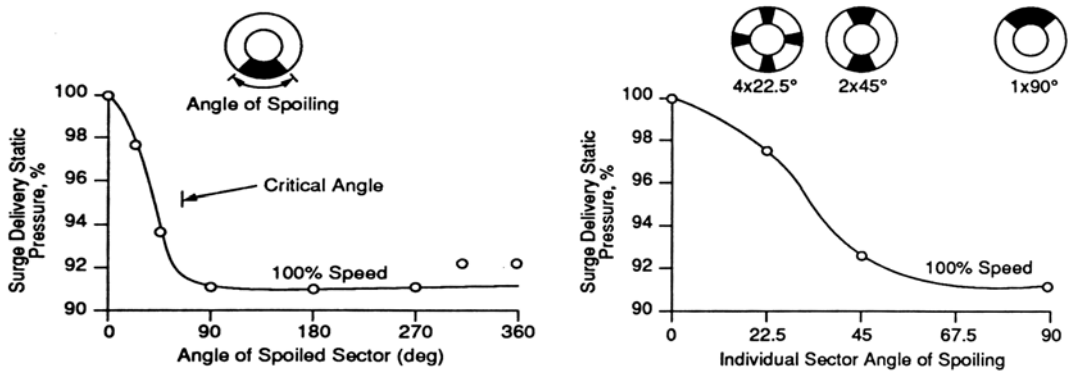

Figure 2-1: The effect on surge delivery static pressure of spoiled sector width and of sub-dividing the spoiled area [Reid, 1968].

The Campbell diagram, named for Wilfred Campbell [1924], is the primary tool used to indicate areas where resonance (blade vibration at its natural frequency) may occur. The Campbell diagram graphically displays the natural frequency of various modes of vibration of a blade overlaid on the forcing frequency or "# per-rev" lines of that compressor. The crossing of a natural frequency line by a "# per-rev" line indicates a critical speed for a compressor. The stiffening effect of centrifugal loading can significantly increase the natural frequencies of a blade as the rotational speed increases. Figure 2-2 from Manwaring et al. [1997] shows a Campbell diagram for a typical blade. Four natural frequency lines are shown: 1st Flexural, 2nd Flexural, 1st Torsional, and 3rd Flexural. The forcing frequency lines are the straight slanted lines. They are an indication of the number of distortions or defects per revolution of a blade, hence the name "# per rev", and increase linearly with rotational speed. Figure 2-2 from Danforth [1975] shows that vibratory stress magnitude is directly related to the intensity of the distortion.

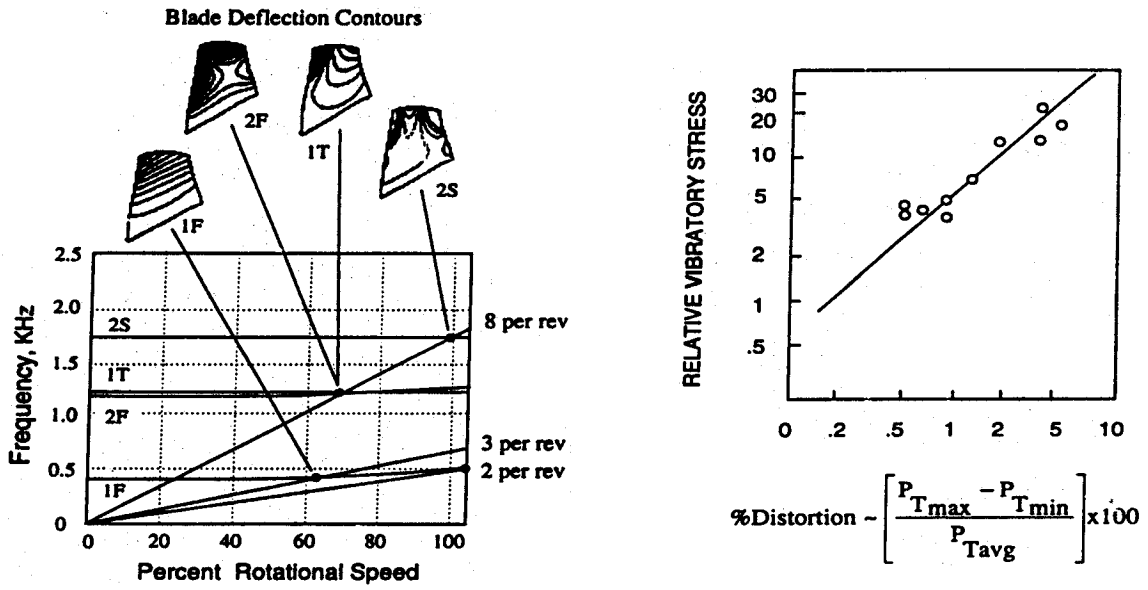

Figure 2-2: Campbell Diagram [Manwaring et al., 1997] and the variation of blade vibration amplitude with the intensity of the distortion [Danforth, 1975].

Peacock and Overli [1975] measured the unsteady normal force at mid span of a lightly loaded low speed compressor as the rotor passed through a steady circumferential distortion. Two different types of distortions were used: a square wave distortion created by a screen of uniform porosity, and a sine wave distortion that was created by a screen of graduated porosity. It was shown in the results that the blade had a more violent response to the square wave than to the sine wave distortion. Suggested was a correlation between the time rate of change of the axial velocity and the normal blade coefficient.

Lecht and Weyer [1976] performed a similar experiment on a subsonic (LP) and a transonic (HP) compressor. The blade force measurements were taken at various locations around the circumference of the compressor relative to the spoiled, or low total pressure, area. Results indicated that the maximum blade loads were reached upon first exiting the low pressure or spoiled area. Minimum loads were experienced when the blade first entered the distorted section.

Datko and O'Hara [1987] tested a transonic compressor's response to seven different inlet total pressure non-uniformities created with distortion screens. The measured response to the various distortions was compared with two "clean" inlet configurations that were also part of the test. The first stage of the compressor was fitted with an integrally bladed disk (blisk). Of primary interest was the blisk's vibratory response to the various inlet distortions. The results indicated that the blisk did experience severe resonant stresses when exposed to inlet distortion. All major resonances were in the first three natural modes of vibration. A blisk is one solid piece of material that is formed into a complete compressor stage, as opposed to a conventional stage where the blades are individually attached to a solid central piece of metal. Because of the way it is made, a blisk has low mechanical damping, which contributes to its high level of vibratory response. The first stage of the compressor used for my research also contains a blisk.

Manwaring and Fleeter [1989, 1990] investigated the rotor blade row unsteady aerodynamics caused by an inlet distortion. In this study, they included the effects of steady loading and various aerodynamic forcing functions on the blade row. Gust amplified two-per-revolution inlet distortions were generated as the aerodynamic forcing function and the resultant blade surface pressures and the aerodynamic response of the rotor were examined. All measurements were transformed into the frequency domain by means of Fourier decomposition. As is to be expected, the second harmonic was the dominant harmonic in all tests, with the amplitude of higher harmonics increasing with gust amplification.

Manwaring et al. [1997] performed a series of experiments on the low-aspect-ratio, transonic, first-stage blade of a two-stage fan. Two types of inlet flow distortions, a 3/rev and an 8/rev, were created with screens. The resultant unsteady blade loading was measured with on-blade pressure transducers and the vibratory response recorded with blade-mounted strain gages. This was the

first time that all aspects of the forced response of a compressor stage were incorporated in an experiment. Blade resonance, the crossing of a natural frequency line by a "# per rev" line, was reached at three points. Results showed the distortions to be strongly vortical and while the steady flow matched modeling methods well, more understanding is needed of the unsteady flow to enable accurate prediction of the unsteady blade stresses associated with it.

Greitzer et al. [1994] examined the effect that a non-uniform inlet flow has on the aeromechanical excitation of a compressor and the stability of the overall compression system. Flutter, a self-excited aeromechanical instability, and forced response were investigated in the area of aeromechanical excitation. In Greitzer's study, it was noted that several hundred incidents of varying severity had occurred in U.S. Air Force aircraft engines since the late 1960's [Air Force Scientific Advisory Board, 1992]. Most of these incidents were attributed to high cycle fatigue (HCF). Greitzer et al. also stated that more work is needed in two areas: the prediction of aerodynamic forcing functions that can cause blade excitation possibly leading to HCF, and a better understanding of the stage—to—stage transfer of inlet distortions in multistage machines.

2.2 Aerodynamic Performance Modeling Techniques

The ability to accurately model a compressor's performance degradation caused by distorted inlet flow is very important. An accurate model, when used in lieu of engine testing, is capable of reducing development time and hardware costs, resulting in large cost savings. Computational fluid dynamics (CFD) is capable of providing such an ability at the expense of being very time consuming and computationally intensive. A simpler and faster modeling method capable of accurately predicting compressor performance would be of great value. Many modeling methods exist that meet the first criterion, most of them having been developed before the advent of CFD. However, as compressor performance

margins increase, so do their expected life spans. For that reason, the second criterion is becoming more and more elusive. This section discusses various techniques, starting with parallel compressor theory, that have been developed to model a compressor's aerodynamic performance.

Pearson and McKenzie [1959] originally developed the parallel compressor theory. The parallel compressor model is based upon dividing the distorted compressor into two or more hypothetical compressors, or flow fields, operating in parallel. The operating point of the distorted compressor is equal to the averaged operating points of each of the individual hypothetical compressors. Parallel compressor theory relies on five critical assumptions, or restrictions: (1) the distorted compressor consists of two or more independently operating sub-compressors; (2) all sub-compressors have individually uniform inlet conditions and operate on the undistorted compressor characteristics; (3) no circumferential cross-flow exists between sub-compressors; (4) exit static pressure of all sub-compressors is equal; and (5) the entire compressor stalls when an individual compressor reaches the undistorted stall pressure ratio. Figure 2-3 provides an illustration depicting a parallel compressor model using two hypothetical flow fields.

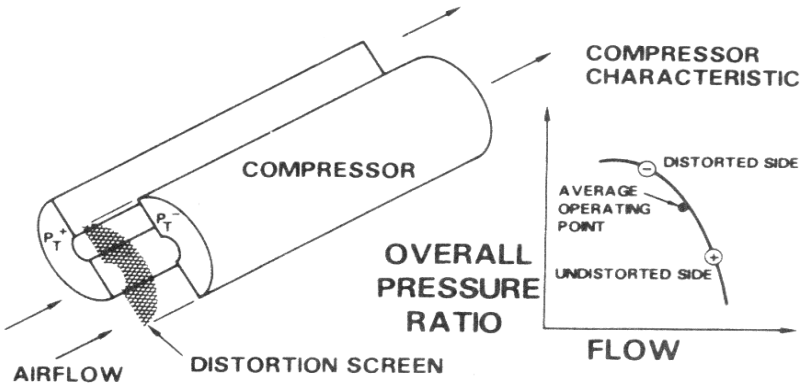

Figure 2-3: Pictorial representation of parallel compressor theory [Reid, 1969].

Many modeling techniques have been developed that concentrate on either reducing the number of assumptions required or improving the results obtained from the parallel compressor model. Roberts et al. [1968] found that the assumption of no circumferential cross-flow caused the predicted waveform to have a square-wave shape, which, especially in downstream stages, is not the case. As can be seen in Figure 2-4, the actual downstream flows vary appreciably from a square-wave.

Roberts et al. [1968] also varied rotor and stator chord lengths in order to investigate blade response time to a disturbance. These tests revealed that compressor rotor blades with longer chord lengths are more tolerant to distortion, suggesting that blade response to an inlet distortion is not immediate. This time lag in blade response to a distortion is more than likely due to downstream effects and the fact that a pressure disturbance takes a certain amount of time to travel through the blade passage. Roberts et al. introduced a reduced frequency parameter in order to define this time dependency.

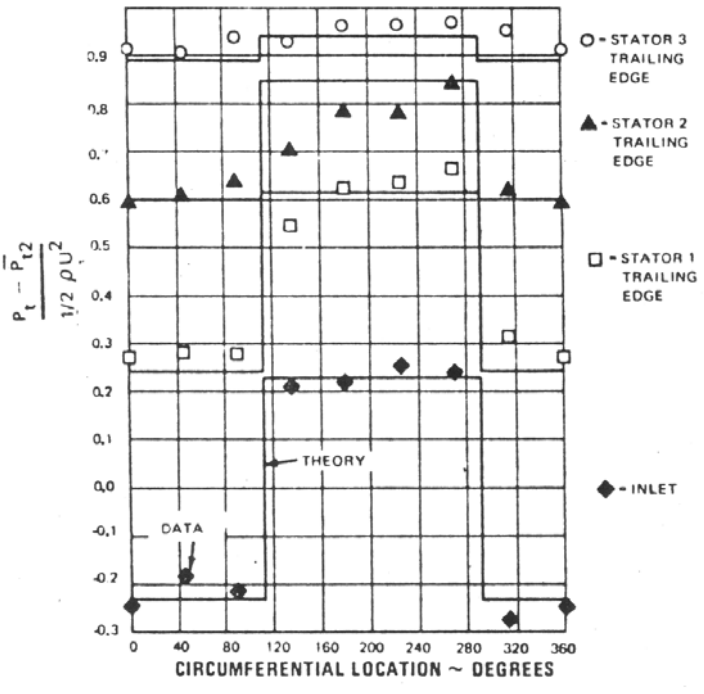

Figure 2-4: Deviation of flow from predicted square-wave pressure pattern [Roberts et al., 1968].

Adamczyk [1974] developed a model that improved upon the original parallel compressor theory. His model did not require exit circumferential static pressure to be constant and also allowed for circumferential cross-flow to exist. In addition, Adamczyk's model could also handle large amplitude distortions, which the original parallel compressor model was unable to do. Adamczyk's problem was that there was no experimental data available to make a direct comparison with his model's predictions.

Significant improvements to parallel compressor theory were made by Mazzawy [1977] by replacing the original sub-compressors with pseudo-streamtubes. This method increased the circumferential resolution as well as reduced the number of restrictions from five to one. The only restriction required in Mazzawy's model was that the distortion's circumferential extent covered several blade passages. Mazzawy used a first-order decay equation with an empirically derived time constant to model blade response to distortion whereas original parallel compressor theory assumed the blade responded instantaneously. Figure 2-5 shows prediction results from Mazzawy's model and the original parallel compressor model as compared with experimental data. Mazzawy's model, represented by the solid line, does a better job of predicting compressor response than the original parallel compressor theory.

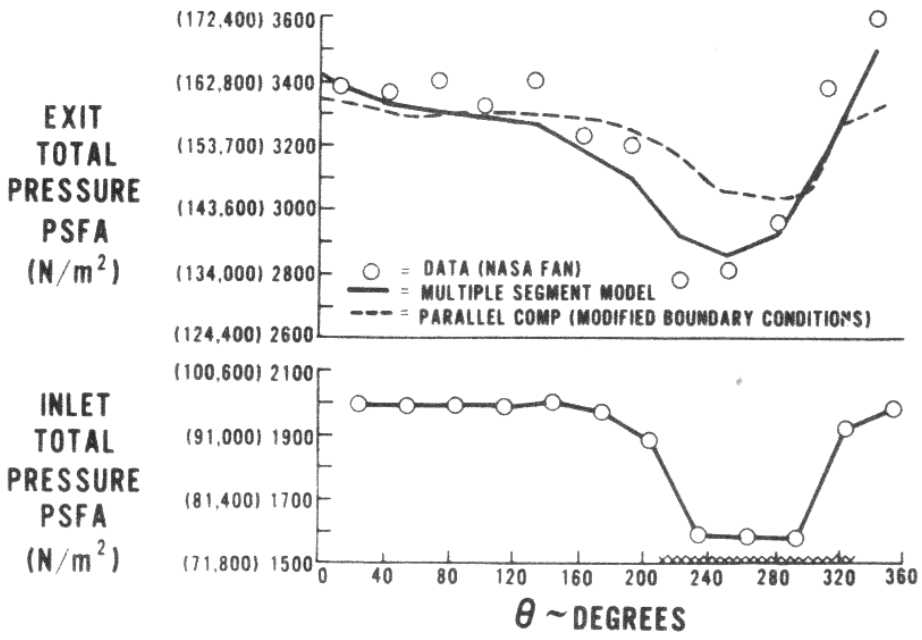

Figure 2-5: Multiple stream-tube model versus parallel compressor theory [Mazzawy, 1977].

Melick [1973] developed yet another modification to parallel compressor theory by combining it with a time-dependent lift coefficient. His model was similar to a spring-mass-damper system with a second order equation describing the lift coefficient, with the two time constants being determined empirically. Lecht [1986] improved this approach when he added the concept of lift overshoot to the model. Lift overshoot occurs when an airfoil momentarily exceeds its maximum lift coefficient without stalling. One result of the approach used by Melick and Lecht was the definition of a time domain transfer function of exponential decay between steady state and the effective angle of attack. Figure 2-6 illustrates the comparison of the exponential decay function to the actual step input.

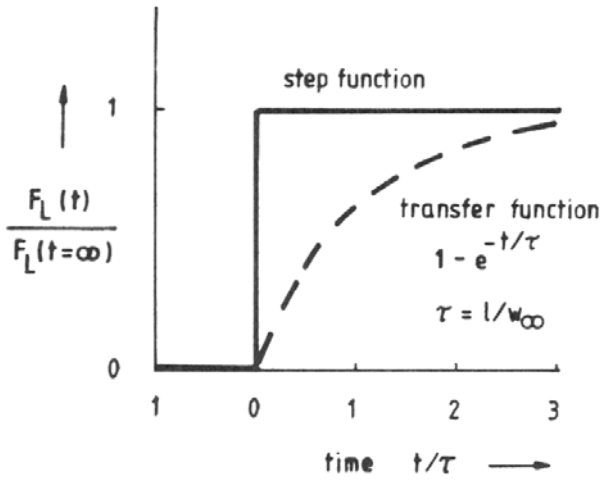

Figure 2-6: Exponential decay transfer function [Lecht, 1986].

Another technique used for compressor modeling uses an actuator disk in place of a compressor's blade row. Greitzer [1973] was the first to use this method and with it was able to include the circumferential velocities and the pressure rise caused on the flow by the compressor stage in the evaluation. With this method, Greitzer was the first to model compressor rotating stall and surge and therefore was able to mathematically describe an experimentally observed problem in turbomachinery.

Barr [1978], Henderson and Shen [1981], and Colpin and Kool [1978] also used the actuator disk modeling method. Barr and then later Henderson and Shen examined the effect of sinusoidal disturbances and found that the maximum distortion attenuation occurred when the ratio of blade spacing to distortion wavelength was one half. Colpin and Kool derived a transfer function to model total enthalpy change over the disk. This derivation enabled the prediction of the exit total pressure, which was compared to experimental measurements, as shown in Figure 2-7.

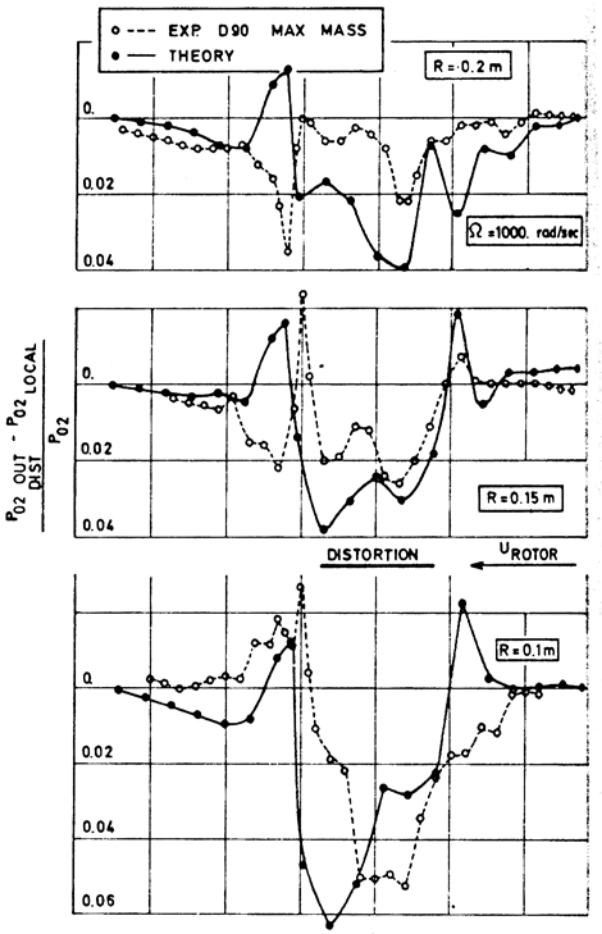

Figure 2-7: Actuator disk total exit pressure predictions compared with experimental measurements [Colpin and Kool, 1978].

The first nonlinear stall model using an actuator disk was proposed by Nagano and Takata [1970]. In their model, they used the non-linear equations of motion coupled with the blade row characteristics to obtain a finite difference solution using a digital computer. Sexton and O'Brien [1981] improved this method by introducing frequency response functions that were used with a semi-actuator disk model to predict the rotating stall behavior of an experimental compressor. The development of the frequency response functions used the quasi-steady total pressure loss as the forcing function and the dynamic total pressure loss as the response function. Cousins and O'Brien [1985] used the same approach in the development of a post-stall compressor model. In this

model an experimentally derived transfer function driven by the quasi-steady total pressure loss variations was used to calculate dynamic loss response.

Rabe et al. [1995] examined the effects on a transonic fan blisk as caused by various circumferential distortion patterns. Investigated were circumferential distortions of 2, 3, and 8 per revolution. Blade response was measured using high frequency pressure transducers mounted in the blade, enabling the investigation of the distribution along the chord of the blade's response. Manwaring et al. [1997] used this same test data to characterize the unsteady aerodynamic forcing function and the blade response as discussed in section 2.1. Hah et al. [1998] modeled the same experiments as Rabe and Manwaring by solving the Reynolds-averaged Navier-Stokes equations with a numerical solution. The predictions of Hah's model were then compared with the experimental data with good prediction results for uniform flow. There was a problem, however, predicting the unsteady blade pressure distributions that resulted from a non-uniform inlet flow. Rabe et al. [1999] went on to extend Hah's numerical procedure to generate a full annular solution with similar results.

Small [2001], Luedke [2001], and my work also use the data from Rabe et al. [1995]. Small [2001] used a frequency response function modeling technique entitled the "Tuned FRF Method" in which frequency response functions are "tuned" to the operating conditions and properties of the inlet flow field. Luedke [2001] created a model based on nonlinear Volterra theory, which is discussed more in depth in the next chapter. Small's, Luedke's, and my work are all designed to predict the transfer of an inlet pressure circumferential distortion. In Chapter 7, the results of Small's and Luedke's models will be compared with the results of my study.

Hale and O'Brien [1998] developed the Turbine Engine Analysis Compressor Code (TEACC). This method uses information calculated by a streamline curvature code regarding mass flow, blade forces, and shaft work.

TEACC uses this information to solve the conservation equations, building a three-dimensional grid representing the full annulus of the compressor as shown in Figure 2-8. In this grid, the three-dimensional Euler equations allow the interaction of circumferential and radial control volumes with one another. TEACC provides the ability to model distortion that is moving; however, TEACC's capabilities are limited to steady-state conditions [Davis et al., 2000].

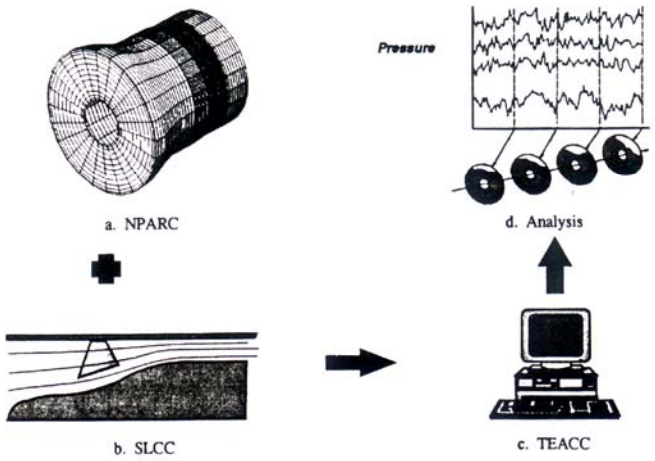

Figure 2-8: TEACC methodology [Davis et al., 2000]: (a) three-dimensional grid built from the conservation equations (b) streamline curvature code calculates mass flow, blade forces, and shaft work (c) digital computer combines a & b to solve the conservation equations and (d) model distortion in steady-state conditions.

2.3 Conclusions from Literature Review

Over the last 40 years there has been a great deal of work devoted to the understanding and modeling of compressor response to non-uniform flow. One has only to do a search for inlet distortion or high cycle fatigue in an Aerospace and High Technology database to see that there is a wealth of information about the two related subjects. While an understanding exists of a compressor's basic performance response to inlet distortion, more knowledge is needed in the areas of distortion transfer and aeromechanical response as it is related to high cycle fatigue. The preferred method to obtain this information is through numerical

modeling, often in the area of Computational Fluid Dynamics (CFD) that is capable of providing accurate information in less time at a lesser cost than an actual test of the compressor in question.

CFD harnesses the calculating power of computers to solve complex three-dimensional equations describing a fluid flow. While CFD is capable of accurately predicting compressor response in uniform flow and advances continue to be made in the area of non-uniform flow response prediction, more development is needed before it can be relied upon as a prediction tool in the design process. Drawbacks of CFD are its computational complexity and the large amount of time required to solve for each inlet condition. Solving for just one inlet condition with CFD could take hours to days. Solving for a number of inlet conditions, as would be required in the design process, could take months and cause significant delays in production.

The option most commonly exercised when developing a new compressor design uses the ever-growing database of knowledge gained through experience and a comprehensive testing program. Every company has its own database of information pertaining to every type of compressor rotor they have ever built. Subsequently, a new design dramatically different or outside of a statistical "comfort zone" offers quite a few unknowns and requires rigorous testing throughout the entire design process. Part of the testing procedure of a new rotor design involves creating various types of inlet pressure distortions with screens and measuring inlet and outlet total stage pressure as well as blade response, as was done in many of the experiments discussed in the previous section. Figure 2-9 shows some typical screen designs used as distortion generators. As can be imagined, the testing process can be time consuming and costly.

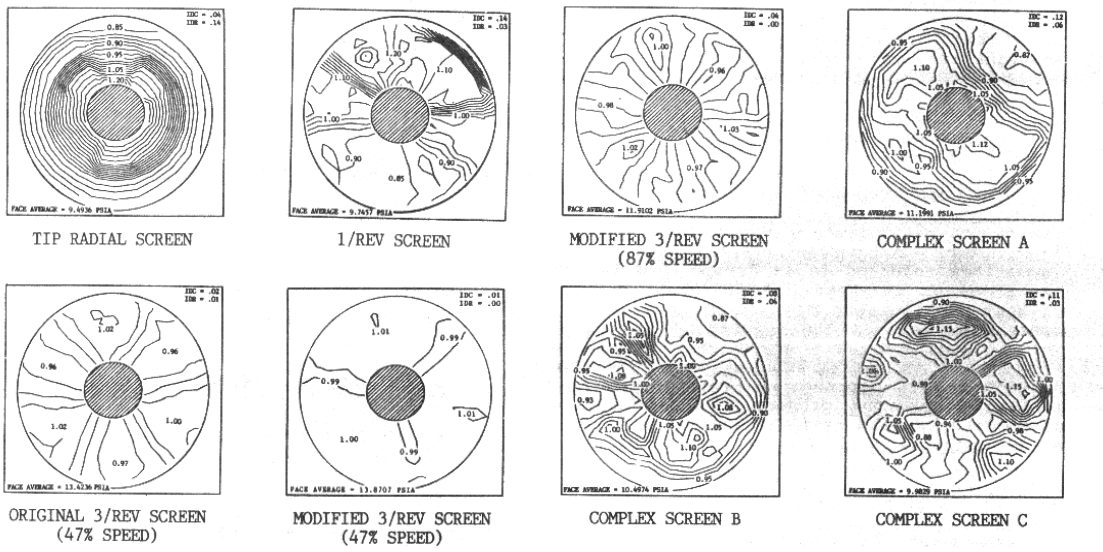

Figure 2-9: Distortion screen designs to generate total pressure non-uniformities [Datko et al., 1987].

In the rapidly changing world of high technology and high competition, every dollar saved without the sacrifice of quality or safety in the production of a better rotor is important and needed. For this reason, a modeling method that would be capable of accurately predicting the response of a rotor to varying inlet conditions in uniform and non-uniform flow would be of great benefit. It would provide an even greater benefit if this modeling method was simple to use and able to make predictions in minutes to a few hours as opposed to hours to a few days. While these goals are ambitious, it is not unrealistic to believe that as the power of computers continues to increase rapidly, such a modeling method or capability will one day exist. The use of nonlinear Volterra theory in aerodynamic response modeling offers some of these desired abilities. The next chapter discusses nonlinear Volterra theory, its development, and its history as an aerodynamic response modeling method.

3 Nonlinear Volterra Theory Modeling

The first section of this chapter will start with the development of nonlinear Volterra theory in the early 1900's and then jump to more recent times when it has started to be used in aerodynamic modeling. The second section explains in more detail the heart of Volterra theory, its kernels, and the chapter ends with a discussion of the system requirements necessary in order to use Volterra theory as a modeling method.

3.1 Nonlinear Volterra Theory

Vito Volterra, an Italian mathematician who lived from 1860 until 1931, developed nonlinear Volterra theory in the early 1900's [Volterra, V., 1930]. During that time he published papers on partial differential equations, but his most famous work was done on integral equations, which became known as integral equations of the Volterra type. Nonlinear Volterra theory states that any time-invariant nonlinear system can be modeled as an infinite sum of multidimensional convolution integrals of increasing order. The series is represented symbolically as shown in Equation 3.1.

$$y(t) = \int_0^\infty h_1(\tau_1)x(t-\tau_1)d\tau_1 + \int_0^\infty\int_0^\infty h_2(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2)d\tau_1 d\tau_2 + \ldots$$
$$\int_0^\infty \ldots \int_0^\infty h_n(\tau_1,\ldots,\tau_n)x(t-\tau_1)\ldots x(t-\tau_n)d\tau_1\ldots d\tau_n$$

Equation 3-1

In the equation, x(t) represents the mean-zero input into the system while y(t) is the output, or system response. As can be seen in the above equation, both the input and output of the system are functions of time (t) and time lag (τ). Another way of describing this type of system is as a system "with memory".

The memory of a system is a measure of the time required for the system to completely recover from an impulse. Convolution is used to predict the exact response of the system to an input.

Each term in the Volterra series contains a kernel, denoted by the *h* terms in the equation, also known as the impulse response function of the system. The first term in the series represents the linear response of the system. Each additional term represents the next successive response of the system, or the number of impulses that act upon the system before it recovers. The second term represents the way the system responds to two inputs, the only time requirement between these inputs being that the second one happens before the system has recovered from the first. Once the kernels of a system are known, the response of the system can be calculated for any arbitrary input.

A third order system is pictorially represented in Figure 3.1. Each of the first three terms in a Volterra series representation of this system would account for each order response within that system. The linear or first order term represents the mean response of the system. Each higher order term accounts for variations about that mean with the addition of all three responses being the system output.

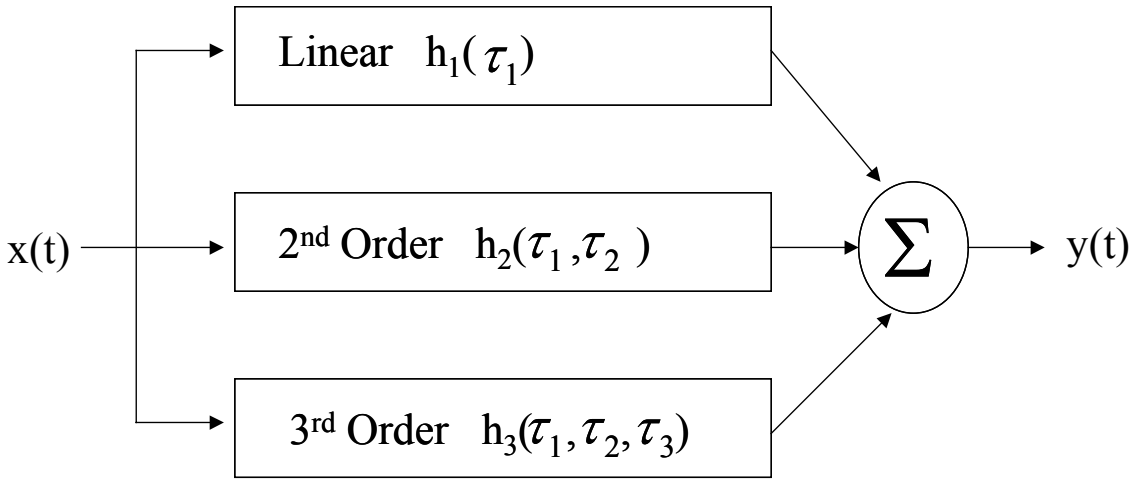

Figure 3-1: Third order system representation [Bendat, 1998].

In order to successfully use the Volterra series to model a nonlinear system, the system needs to be "weakly" nonlinear. This means that the higher order terms within the series quickly drop off to approximately zero so that their effects are negligible. Most causal systems fit within this description [Tromp and Jenkins, 1990]. A system is considered to be causal if the response of the system to any impulse does not depend on the future impulses. Restated, the response of the system at time (t) to a signal applied at time (τ) depends only on the amount of time between the two, (t - τ). All known physical systems are causal.

Volterra theory has seen little use outside of the Electrical Engineering and Biological fields. Not until the last 15 years has Volterra theory started to be used in aerodynamic modeling. This is primarily due to the fact that in the past linear modeling methods provided acceptable accuracy and were much easier to use than nonlinear ones. As the performance of modern day compressors continues to increase as well as their expected life spans, linear methods are no longer adequate. In addition, with the advancement of computers, methods that were once too complex and time consuming to consider are now much more feasible.

Boyd et al. [1983] developed a method to measure the second-order Volterra kernel of a weakly nonlinear system. Boyd et al. defined a weakly nonlinear system similarly to what has been previously discussed, with the additional requirement that the nonlinearities that do exist must be subtle, less than 40 dB. Examples of this type of system are simple communication systems, some high quality transformers, and electro acoustic transducers. This method involves applying a multitone-probing signal to the system being analyzed and uses an interpolation method to estimate the second order response of the system, thereby providing a measurement of the second order kernel. While this is not an example of aerodynamic modeling, this work is referred to in future aerodynamic applications of Volterra theory.

A Volterra kernel identification method originally developed for use in electrical circuits was employed by Tromp and Jenkins [1990] to model nonlinear aerodynamic behavior over a 2-D airfoil. The procedure involved exciting the system with very specific inputs, recording the system response, and then calculating the kernel association with that response. Initially, an input was used employing a small angle of attack to insure linear system response. A pencil of functions (POF) method was then utilized to identify the system's linear kernel. After the linear kernel was known, the system was excited with another input in such a way that third order and above effects were negligible. Since the linear impulse response had just been solved for, the difference between the system's response to the second input and the previously calculated linear response was the second order kernel of the system. This method, despite being successful, is extremely limited by the input types that can be used and, obviously, could not be used with wind-tunnel models or flights tests where the order of the system response is not known.

Silva [1993] identified the linear and second-order kernels of a NACA 64A010 rectangular wing undergoing pitch using a numerical kernel identification technique based on impulse responses. Airflow over the wing was modeled using a finite-difference computer program; the output from this program due to specific unit impulses was used to calculate the two kernels of the system. Silva [1997] continued his research by using digital filter techniques to identify linear and nonlinear aerodynamic impulse responses. In this paper Silva states that the aerodynamic impulse response function is the most fundamental and computationally efficient aerodynamic function that can be extracted from an aerodynamic system. He goes on to state that once the unit impulse response is identified, the output of the system to any input can be calculated via convolution. He also shows how the first two terms in a Volterra series can be used to calculate the system output using the earlier identified linear and second-order impulse response functions. As is the case in the other

methods discussed so far, despite encouraging results, neither of these methods would be applicable with wind tunnel or flight tests.

In 1999 Patrick Reisenthel developed a method to identify Volterra kernels which uses "physically realizable inputs, is robust with respect to noise, and minimizes or eliminates the need for analytical assumptions". Reisenthel [1999] states that the identification of impulse response functions from experimental data is a poorly posed problem. A poorly posed problem cannot be solved using straightforward methods, the result would be that the problem either has no solution whatsoever, or it has more than one. Neither of these alternatives is acceptable in that the one "unique" solution is what is desired. Reisenthel employed regularization methods developed in other fields such as radar detection, mathematical biology, and astronomy in order to overcome this difficulty and successfully used this method to model airflow over a wing. He goes on to show that once the kernels of the system are identified, the prediction of the system's response to any novel maneuver is possible.

3.2 Volterra Kernels

The kernels of a system characterize the system's response to an impulse or a group of impulses. In the Volterra series representation, the *h* terms denote the linear and nonlinear kernels. The first term in the series represents the linear response of the system to an impulse at time τ_1. τ_1 is a "dummy" variable that is used to represent time. The linear response of the system occurs at time t to an impulse that occurred at time τ_1, (t-τ_1) representing the response of the system over time. This is why the Volterra series is sometimes referred to as a way to model a system with "memory".

The second order kernel, $h(\tau_1, \tau_2)$, defines the system's response to two impulses occurring at different times, τ_1 and τ_2. Even though these two impulses have subscripts one and two, that does not imply that there is a definite

order in which they must take place. The only requirement regarding the times that these impulses occur is that the second one happens before the system has recovered from the first.

Taking this one step farther, the system's third order kernel $h(\tau_1, \tau_2, \tau_3)$ characterizes the way the system reacts to three unit impulses, each occurring at a separate time, but all three acting on the system before it has recovered from any one of them. The modeling methods used in this research will not go above a third order kernel for two reasons. The amount of data required to accurately define a fourth order kernel is not available and, the basis of this method is that the system is a weakly nonlinear system, meaning that the higher order kernels fall off to zero very quickly.

3.3 System Requirements for Volterra Series Modeling

The first requirement of a system to be able to use a Volterra series technique to model it is that the system must be physically realizable. A physically realizable system's input, output, and impulse response functions are all real valued functions occurring over time. Causality is also a requirement. To define a system as causal simply means that the system cannot respond before the input occurs, therefore, the impulse response functions, or kernels, of the system have a value of zero for $\tau < 0$. The last requirement is that the system be time-invariant. At first glance this may seem to be a contradiction to the first requirement of time dependent input, output, and response functions; it's not. A time-invariant system is one whose properties do not change over time. Another way of looking at this requirement is that the basic system properties, the impulse response functions or kernels, do not change, hence the term time-invariant. These three requirements, although very important, are not difficult to meet as most real world systems satisfy all three.

4 Identification of Volterra Kernels

Generally stated, and reviewing from the previous chapter, the Volterra theory of nonlinear systems states that any physically realizable nonlinear system may be represented as a series of multidimensional integrals (see Equation 3.1). In the below equation y(t) represents an aerodynamic quantity of interest (stage exit total pressure), x(t) is the stage inlet total pressure, and h(τ) represents the system's impulse response functions (the kernels of the system).

$$y(t) = \int_0^\infty h_1(\tau_1)x(t-\tau_1)d\tau_1 + \int_0^\infty\int_0^\infty h_2(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2)d\tau_1 d\tau_2 + \ldots$$
$$\int_0^\infty \ldots \int_0^\infty hn(\tau_1,\ldots,\tau_n)x(t-\tau_1)\ldots x(t-\tau_n)d\tau_1\ldots d\tau_n$$

Equation 3.1

In order to facilitate the use of Volterra theory as a modeling method, the assumption is made that the value of the higher order kernels falls to zero very quickly [Tromp and Jenkins, 1990]. This is known as a weakly nonlinear system. Boyd et al. [1983] extends this definition even further by stating that the nonlinearities of the system must be subtle (less than 40 dB distortion). According to Silva [1993], for many aeroelastic and vibration analyses, these assumptions are appropriate.

This chapter starts by discussing the three basic categories of kernel identification methods used before 1999. A fourth category or method of kernel identification will then be explained. This fourth method was developed by Patrick Reisenthel [1999] and is used in my research. This technique is based upon expanding the unknown kernel into basis function space. The various basis functions used will be covered along with a more detailed look at the linear and higher order kernels and their identification using Reisenthel's extraction method.

4.1 Volterra Kernel Identification Methods

In the past, methods used to identify Volterra kernels typically fell into one of three categories: "direct" methods, "optimization" methods, and "active probing" or "measurement" methods. Reisenthel [1999] developed a fourth method, an "indirect" technique designed to identify the kernels of a system from unsteady aerodynamic data. The first three categories will be discussed in this section. The fourth method will be introduced with a more detailed explanation following in the next section.

"Direct" methods are very straightforward, therefore the name "direct", and involve applying pulse or step inputs to the system. The system's response to a single pulse or excitation is used to determine the linear kernel $h_1(\tau_1)$. Similarly, applying two pulses to the system enables the second order kernel, $h_2(\tau_1, \tau_2)$, to be constructed. This is the method used by Silva [1993, 1997] in his research as discussed in the previous chapter.

"Optimization" methods involve assuming the form of the kernel and then identifying the unknown coefficients through parameter identification. Optimization methods are typically done in either the time or Laplace domain. Tromp and Jenkins [1990] pencil of functions method falls into this category. Tromp and Jenkins research was done in the Laplace domain.

The third category of methods, "measurement", requires very specific, predetermined excitation inputs be applied to the system in order to "probe" its structure. These methods can be either statistical or deterministic in nature, and, for the most part, have been developed in the electrical engineering field. One technique that falls into this category is that of kernel separation. Kernel separation can be done in the time and frequency domain and involves making repeated measurements with different excitation amplitudes. Boyd et al.'s [1983] multiple harmonic probing, done in the frequency domain, is an example of this type of method.

Reisenthel [1999] developed the fourth method of "indirect" kernel identification, where the unknown kernels are "extracted" from existing aerodynamic data not necessarily obtained with the goal of kernel identification. The goals for the development of this technique were as follows: the method must use physically realizable inputs, must be applicable to existing data, and it should minimize or eliminate the need for any assumptions about the form of the kernels. Reisenthel successfully applied this technique to wind-tunnel data of a maneuvering aircraft and believes that the technique would also be useful for the identification of Volterra kernels in aeroelastic systems.

Indirect kernel identification is an inverse problem, where the goal is to solve for the transfer function that produces an already known output from the input that was introduced to the system in question. In its simplest form, this type of problem is an easy one and can be solved through basic division. However, when dealing with aerodynamic data, there are many more factors and unknowns to deal with and one finds that there is either no exact solution for the data in question or that there are multiple solutions which will produce the already known output. Neither one of these options is desirable, as the goal is to solve for the one unique solution to the problem in question. This is known as an ill-posed problem, and is a result of having to solve integral equations similar to those in inverse scattering problems. A straightforward attempt to solve these equations would result in nonsense; however, in recent years regularization methods have been invented that enable this difficulty to be overcome. These methods have been used successfully in radar detection, medical imaging, mathematical biology, astronomy, optics, geophysics and tomography [Hansen, 1998].

4.2 Indirect Extraction Technique

As previously mentioned, this kernel "extraction" technique is designed to identify the Volterra kernels of a system from arbitrary experimental data that was not necessarily obtained with the goal of kernel identification. The technique is based upon expanding the unknown kernel into basis function space.

As an example, consider the identification of the Volterra kernel of a linear system. Since this system is linear, it can be exactly described by the first term in a truncated Volterra series (see equation 4-1). Since this is a physically realizable system, the limits of integration are from t = 0 until t = T, the length of the sampling period.

$$y(t) = \int_0^T h_1(t-\tau_1)x(\tau_1)d\tau_1$$

Equation 4-1

The unknown kernel, $h_1(t-\tau_1)$, is next expanded in some basis function space, $\varepsilon_j(t)$. The kernel now becomes the summation of unknown coefficients multiplied by the basis function (see Equation 4-2), and the original equation representing the system now becomes Equation 4-3.

$$h_1(t) = \sum_{j=1}^{N} c_j \varepsilon_j(t)$$

Equation 4-2

$$y(t) = \sum_{j=1}^{N} c_j \int_0^T \varepsilon_j(t-\tau_1)x(\tau_1)d\tau_1$$

Equation 4-3

In Equation 4-2 and Equation 4-3 the subscript *j* varies from *j*=1 to *j*=N with N representing the number of basis functions and, therefore, the number of coefficients that need to be solved for. The unknown coefficients, c_j, must satisfy Equation 4-3 at all times. Since there are N coefficients to solve for, there must be N realizations of Equation 4-3. Each realization will look like Equation 4-4.

$$y(t_i) = \sum_{j=1}^{N} c_j \int_{i=0}^{i=T} \varepsilon_j(t_i - \tau_1) x(\tau_1) d\tau_1$$

Equation 4-4

In Equation 4-4, all terms behind the integration sign, the basis function and the input, are known. Therefore, the only unknowns are the coefficients, c_j. Each realization of Equation 4-4 results in one row of a matrix. The matrix will be [*i* x *j*] in size, with *i* representing the number of data points and *j* the number of basis functions. Equation 4-4 can also be representing equivalently as Equation 4-5, with a_{ij} as the matrix formed through the integration process. The final result of this process is the formation of a matrix problem similar to Equation 4-6.

$$y(t_i) = \sum_{j=1}^{N} c_j a_{ij}$$

Equation 4-5

$$[A][C] = [Y]$$

Equation 4-6

In Equation 4-6, $A = [a_{ij}]$ and is known as the motion matrix, $Y = [y(t_1), y(t_2), \ldots, y(t_n)]^T$ is the output total pressure, and $C = [c_1, c_2, \ldots, c_j]^N$ is the solution vector. This same type of equation is also obtained in the nonlinear case as well. In the case of a second-order system, two terms of a truncated Volterra series would be used as in Equation 4-7. Using a similar expansion technique for the

second-order kernel would produce Equation 4-8, which, when substituted back into the original equation, would yield Equation 4-9.

$$y(t) = \int_0^T h_1(t-\tau_1)x(\tau_1)d\tau_1 + \int_0^T\int_0^T h_2(t-\tau_1, t-\tau_2)x(\tau_1)x(\tau_2)d\tau_1 d\tau_2$$

Equation 4-7

$$h_2(t,t') = \sum_{k=1}^{N_2} d_k \mu_k(t,t')$$

Equation 4-8

$$y(t_i) = \sum_{j=1}^{N} c_j \int_{i=0}^{i=T} \varepsilon_j(t_i-\tau_1)x(\tau_1)d\tau_1 + \sum_{k=1}^{N_2} d_k \int_{i=0}^{i=T}\int_{i=0}^{i=T} \mu_k(t-\tau_1, t-\tau_2)x(\tau_1)x(\tau_2)d\tau_1 d\tau_2$$

Equation 4-9

Following the same school of thought as in the linear case, Equation 4-9 could be rewritten in the form of Equation 4-10, with a_{ij} and a_{ik} representing, respectively, the results of the integrations as shown in Equation 4-9. These integrals form the first and second-order motion matrices of the problem. The unknown solution vectors [cj] and [d_k] are next in-lined, forming a linear system of equations, again similar to Equation 4-6. This is a key point in the solution process; even though the original problem is nonlinear, its solution reduces to a linear matrix problem.

$$y(t_i) = \sum_{j=1}^{N} c_j a_{ij} + \sum_{k=1}^{N_2} d_k a_{ik}$$

Equation 4-10

Despite the fact that a nonlinear problem has now been reduced to a much simpler linear matrix problem, its solution is still not straightforward. As mentioned previously, the inverse problem of solving for the unknown

coefficients is often an ill-posed problem. This is a result of the motion matrix, [A] in Equation 4-6, being singular. Since the data used to solve for the kernels is obtained through experimentation, it contains a lot of noise. Solving for the kernels from the output measurements, y(t), tends to amplify the noise. This results in either more than one set of kernels being able to satisfy the equation, or none at all. Neither result is better than the other, as the goal through all of this is to find the one unique solution to the problem at hand.

Both of these difficulties (no unique solution or no solution at all) are handled by considering the solution to Equation 4-5 in the least squares sense. The requirement being that the solution vector [C] is able to solve the matrix equation approximately, as shown in Equation 4-11.

$$\sum_{n=1}^{Nds}[A^{T(n)}][A^{(n)}][C] = \sum_{n=1}^{Nds}[A^{T(n)}][Y^{(n)}]$$

Equation 4-11

Another way to write Equation 4-11 is [M][C]=[B], where [M] is equivalent to the motion matrix multiplied by its transpose, [C] is the solution vector, and [B] is the transpose of the motion matrix multiplied by the output total pressure measurements. *Nds* in Equation 4-11 represents the total number of data sets that are being used to calculate the kernel. Equation 4-11 still cannot be solved in a straightforward manner. The goal is to find the best solution possible for the equation in question. This is done by means of singular value decomposition (SVD).

The SVD procedure works in the following manner. If [M] is nonsingular, it can be broken down as the product of an orthogonal matrix [U], a square diagonal matrix [S], and the transpose of an orthogonal matrix [V] as shown in Equation 4-12. Because of this, the vector [C] can be solved for as indicated in Equation 4-13. Now, if [M] is singular, the values of the diagonal matrix [S] are initially large, but drop off to zero at some point with no evident pattern or way

to calculate when. Since solving for [C] involves taking the inverse of [S], the equation becomes unsolvable once the values of the diagonals fall to zero. Thus, in order to find a "good" solution to the problem, only the nonzero values of [S] can be used. The key to all of this is to decide at what point the diagonal values of [S] are considered to be zero, represented in Equation 4-13 by N. In other words, one has to decide on a noise floor. This is often done as a percentage of the largest value of [S]; one percent is a typical choice.

$$[M] = [U][S][V^T]$$

Equation 4-12

$$[C] = \sum_{i=1}^{N} \frac{1}{[S_i]} ([U_i{}^T][B])[V_i]$$

Equation 4-13

After the solution vector [C] is calculated, it's simply a matter of multiplying it by the basis function(s) to solve for the kernel(s) of the system. Providing that there was enough information in the data used to calculate the kernel(s), the response of the system to any arbitrary input can now be accurately calculated using Equation 3-1.

4.3 Choice of Basis Functions

The choice of basis functions is based on the characteristics of the data that is being analyzed. According to Reisenthel [1999], any type of basis functions can be chosen as long as a complete set of values can be formed from them. In his research, Reisenthel chose an exponential decay function as the basis function. Exponential decay functions are a good representation of the way a physically realizable system responds to an input, with the response of the system gradually recovering to zero output as time passes.

For my research, I also chose basis functions of exponential decay. Again, the linear kernel was assumed to take on the form of Equation 4.2. The linear basis function used by Reisenthel [1999] and also in my research is shown in Equation 4-14.

$$h_1(t) = \sum_{j=1}^{N} c_j \varepsilon_j(t)$$

Equation 4-2

$$\varepsilon_j(t) = e^{-t/\tau_1}$$

Equation 4-14

Following this same line of thinking, the second-order kernel is also one of exponential decay. However, it involves the response of the system to two impulses occurring at times τ_1 and τ_2. Two different forms of the second order basis function were used in my research as shown in Equations 4-15 and 4-16. Equation 4-15 shows the initial form that was considered with Equation 4-16 being a more complicated form in which $H(t_n\text{-}t_m)$ represents the Heaviside step function. The Heaviside step function is basically an on/off switch. If the operation within the parenthesis yields a positive number, the value of the Heaviside step function is one and if that same operation is negative, the step function yields a value of zero. The two forms considered for the third-order basis function are the same as that of the second-order with the addition of one more time variable, t '' or t_3. In all, three different models were created. A listing of each model is included in the Appendix. The higher-order basis functions of Model 1 are in the form of Equation 4-15 and Model 2 uses Equation 4.16. Both Model 1 and 2 utilize the built-in Matlab convolution function as an integration method. Model 3 also uses the Heaviside step function, Equation 16, for its higher-order basis function, but utilizes numerical integration rather than the built-in Matlab convolution function. Models 1 and 2 both are third order, meaning that they calculate the first three kernels of the system being analyzed.

Model 3, due to the dramatic increase in time required to perform numerical integration, is a second order model, calculating the first two kernels from the experimental data.

$$\mu_2(t,t') = e^{-\Delta t/\tau_1} e^{-t/\tau_2}$$

Equation 4-15

$$\mu_2(t_1,t_2) = H(t_1 - t_2) e^{-(t_1-t_2)/\tau_1} e^{-t_2/\tau_2} + H(t_2 - t_1) e^{-(t_2-t_1)/\tau_1} e^{-t_1/\tau_2}$$

$$t_1 = t - \tau_1$$

$$t_2 = t - \tau_2$$

Equation 4-16

At this point, the question may have arisen as to what exactly does a kernel look like? Each kernel characterizes the way the system responds to a certain number of impulses occurring over time, so there is always at least one axis that is or is directly relatable to time. Figure 4-1 shows an example of a linear kernel of the test compressor as calculated from total pressure measurements taken at 9500rpm, RI3.

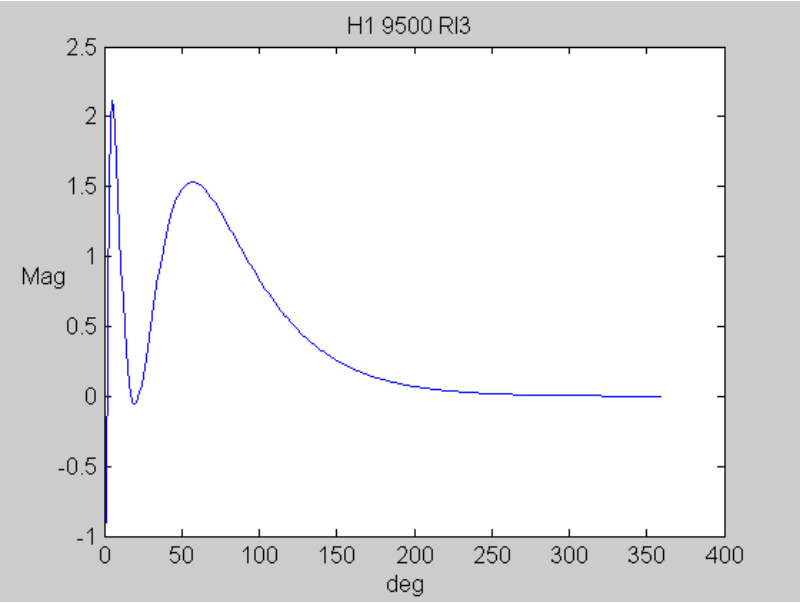

Figure 4-1: Linear kernel calculated from experimental data obtained at 9500rpm, RI3.

The x-axis in Figure 4-1 represents the circumferential location in degrees, which is directly relatable to time. The y-axis is the magnitude of the system response to an input. Notice that the system response is not unlike that of a spring-mass-damper, with the initial value being comparably large and then oscillating around zero until full recovery (zero response) at approximately 180 degrees later.

The second-order kernel is similar with the addition of another axis representing time, as shown in the contour plot in Figure 4-2. The particular kernel shown in Figure 4-2 was calculated under the same operating conditions as the linear kernel shown in Figure 4-1, 9500rpm RI3.

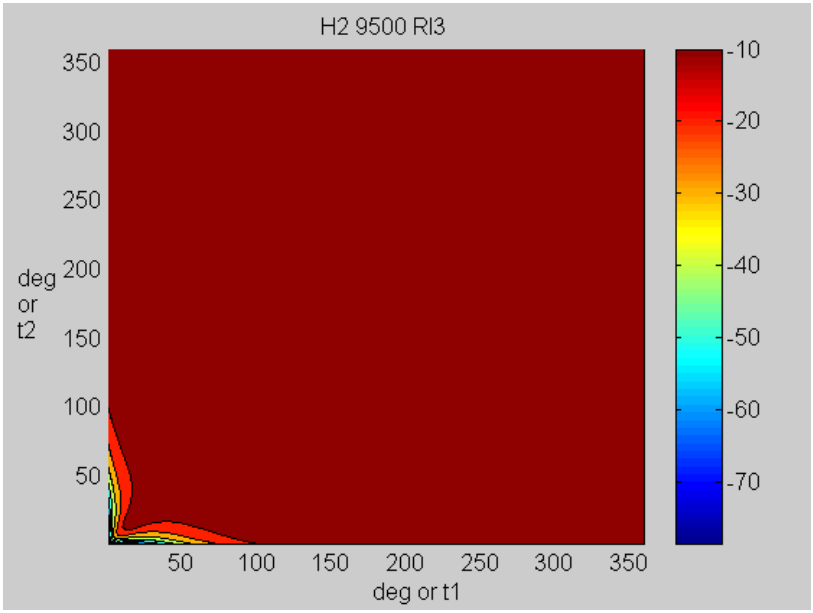

Figure 4-2: Second-order kernel calculated from experimental data obtained at 9500rpm, RI3.

The x and y-axes both represent circumferential location in relation to present location and are directly relatable to time. Notice that the second-order kernel is symmetric. This is due to the fact that the system does not differentiate between which impulse occurs first, just that there are two. For an impulse occurring at the time lag corresponding to one degree, there is a second impulse that could occur anywhere between 1 and 360 degrees. This is true regardless of whether the initial impulse occurs as t1 on the x-axis or t2 on the y-axis; the

magnitude of the system response is the same. A three-dimensional view of the second-order kernel is shown in Figure 4-3.

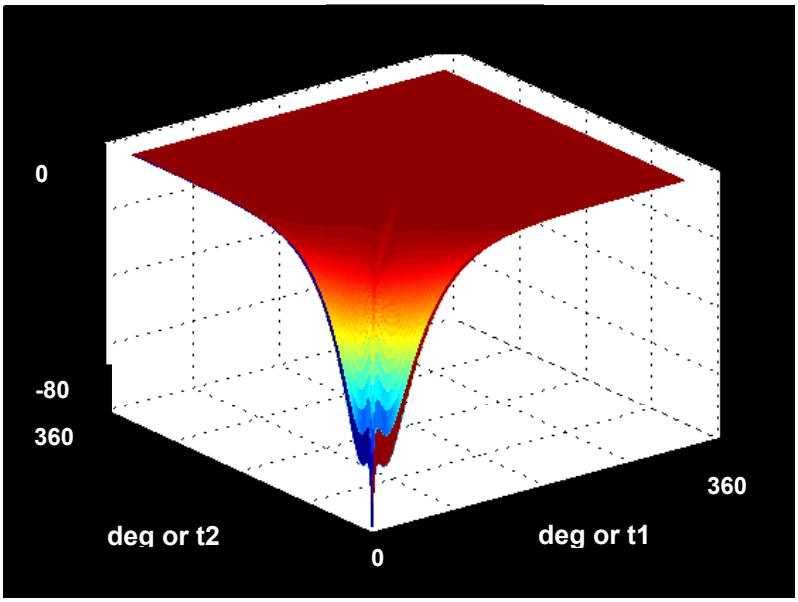

Figure 4-3: Three-dimensional view of the second-order kernel calculated from experimental data obtained at 9500rpm, RI3.

A plot of the third-order kernel is not possible as, similarly to the second-order kernel, it involves another axis representing time. The third-order kernel therefore has four dimensions, three corresponding to time and one to response magnitude. The third-order kernel represents the response of the system to three separate impulses occurring over time, with a certain magnitude associated with the response for every combination of times associated with each of the three impulses.

4.4 Number of Basis Functions

There is no exact way to calculate the number of basis functions that will be needed to adequately identify the kernels of a system. The only method is by trial and error. As the number of basis functions increases, the kernel will converge to its appropriate value. This is shown in Figure 4-4 where the linear kernel is shown with a varying number of basis functions used to calculate it. It

can be seen that after 8 basis functions, the amount the kernel changes is not significant. Because of this, 10 basis functions were used to calculate the linear kernel.

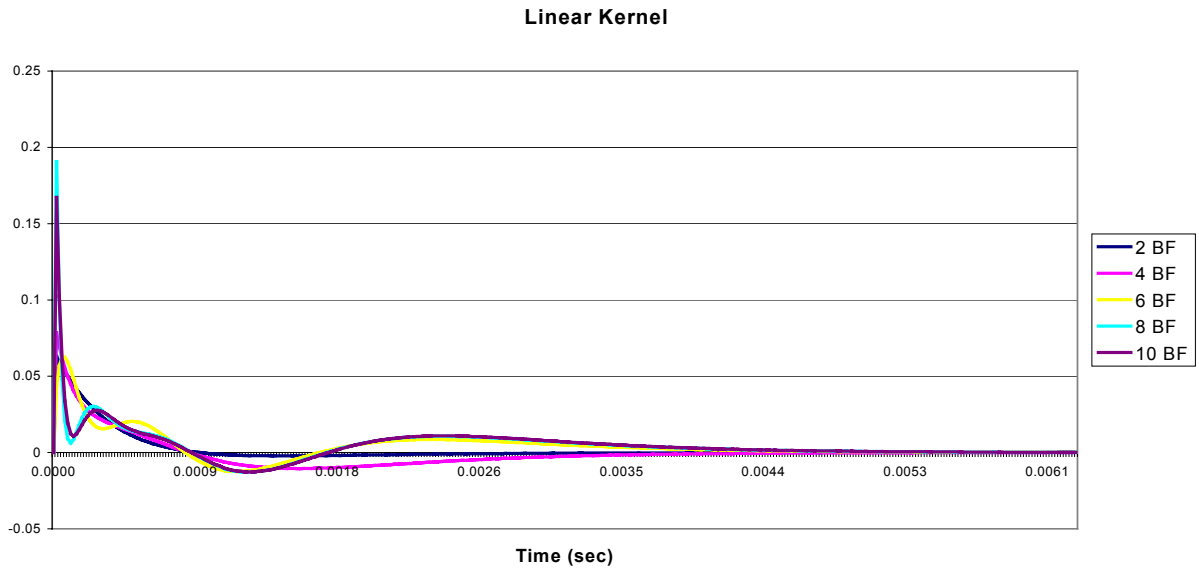

Figure 4-4: Convergence of the linear kernel as the number of basis functions increases.

Another thing to consider when choosing the number of basis functions is the amount of data that is available and will be needed to solve for the coefficients that make up the kernel. In the case of the linear kernel of the system, increasing the number of basis functions by 2 increases the number of equations to solve by 2 also. However, for higher-order kernels, the increase in the number of equations to solve is exponential. For example, 10 basis functions for a linear kernel would result in 10 equations to solve for 10 unknown coefficients. For a second-order kernel, 10 basis functions would result in 10 x 10, or 100 equations and 100 unknowns while in the instance of a third-order kernel, one would generate 10 x 10 x 10, or 1000 equations and 1000 unknowns by choosing 10 basis functions. This exponential increase in basis functions and resulting coefficients to solve for causes the amount of data available to rapidly become an issue. Therefore, both for the sake of time and especially the amount of available data, one does not want to use more basis functions than needed to

solve for each kernel. These facts also prompted the decision to truncate my model at the third term, therefore assuming all fourth and higher-order terms to be insignificant.

5 Experimental Data

The data for this research was acquired from tests that were run on a two-stage transonic, low aspect ratio compressor at the Compressor Research Facility (CRF) at Wright-Patterson Air Force Base, OH. The compressor was operated at certain speeds under specific conditions known to cause rotor resonance in order to acquire aerodynamic and aeromechanical data. This chapter provides a summary of the "Experimental Procedure and Data Acquisition" of Matt Small's thesis [Small, 2001] as an overview of the experimental methods applied.

5.1 Experimental Facility

The CRF is made up of three buildings: the test building, the electrical building, and the operations building. Office space, data acquisition, and control computers are housed in the test building. The electrical building contains transformers and power supplies for the drive motors, and the test building houses the test chamber, flow conditioners, and electrical drive motors.

Figure 5-1 shows a schematic of the test section where the two-stage compressor was operated. The test section draws atmospheric air into the facility via the test compressor. The air is then filtered and run through throttle valves in order to regulate the upstream pressure for the compressor being tested. Flow conditioners are also utilized in order to straighten the flow before it is introduced to the compressor bellmouth. The test chamber is 65 feet long and 20 feet in diameter. After exiting the test chamber, the flow passes through throttle valves used to regulate downstream pressure and control the compressor pressure ratio. The flow is then discharged to the atmosphere via a vertical flue after passing through venturis, which regulate the mass flow. The test article is capable of mass flows from 0 – 500 lbm/s.

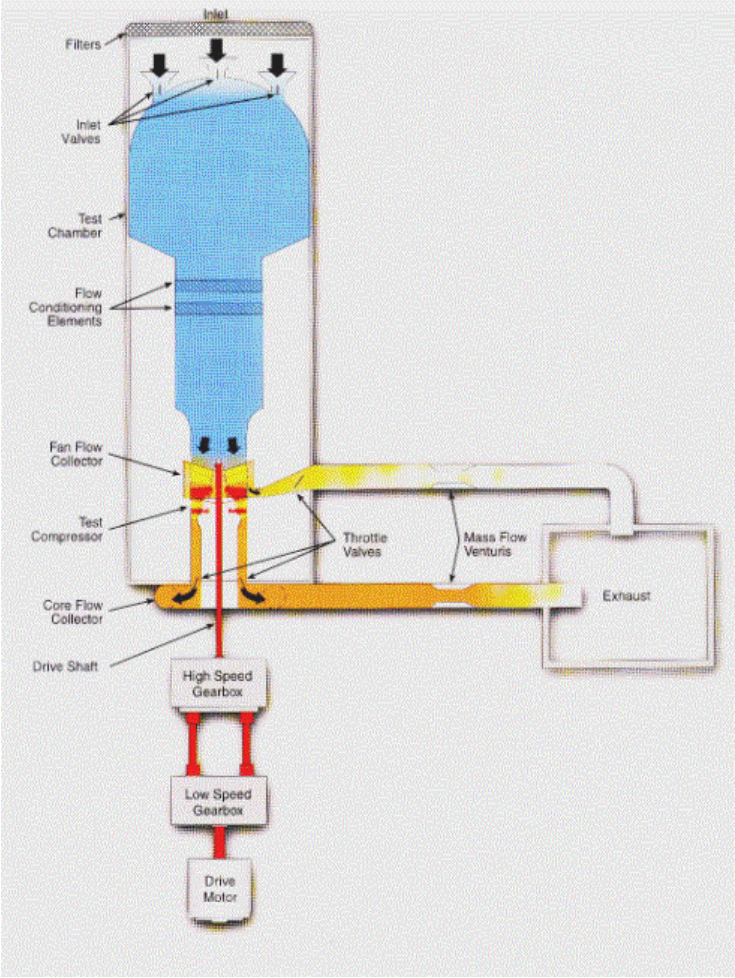

Figure 5-1: Test facility at the Compressor Research Facility, Wright-Patterson AFB [Morrow, 1993].

The speed of the test compressor is determined by the motor-gearbox combination. There are two 30,000 horsepower electric motors to choose from, as well as a high and low-speed gearbox. The CRF Variable Speed Drive control system sets test speed, controls any changes in speed, and also provides automatic safety coast-down modes.

The first stage rotor blisk (modern integrally bladed disk) used for this particular study is shown in Figure 5-2. The blisk consists of 16 low aspect ratio blades of modern design. At the design operating point, rotor relative velocities for inlet temperatures near 300K are supersonic above 45% span. Table 5-1 shows the geometric characteristics of this rotor. The second stage rotor consists

of 40 blades mounted on a disk using a conventional dovetail arrangement (see Figure 5-3).

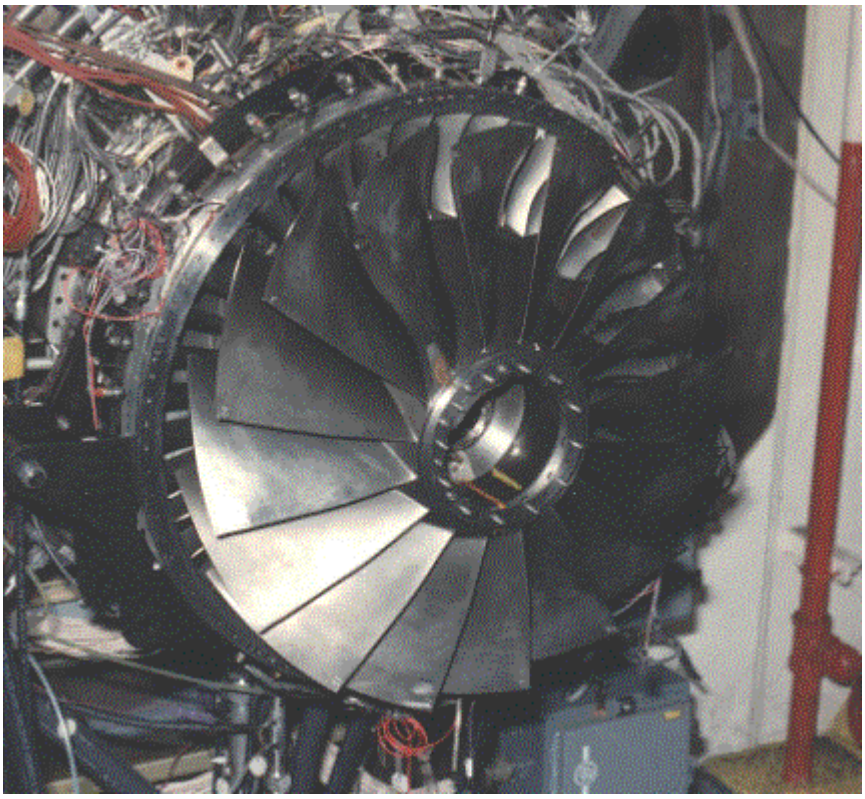

Figure 5-2: First stage blisk mounted on test rig [Morrow, 1993].

Parameter	Value
Average Aspect Ratio	1.22
Rotor Tip Radius (in.)	13.87
Inlet Radius Ratio	0.33
Average Radius Ratio	0.47
Average Tip Solidity	1.50
Maximum Thickness/Chord	0.028

Table 5-1: First stage blisk geometry characteristics [Rabe et al., 1999].

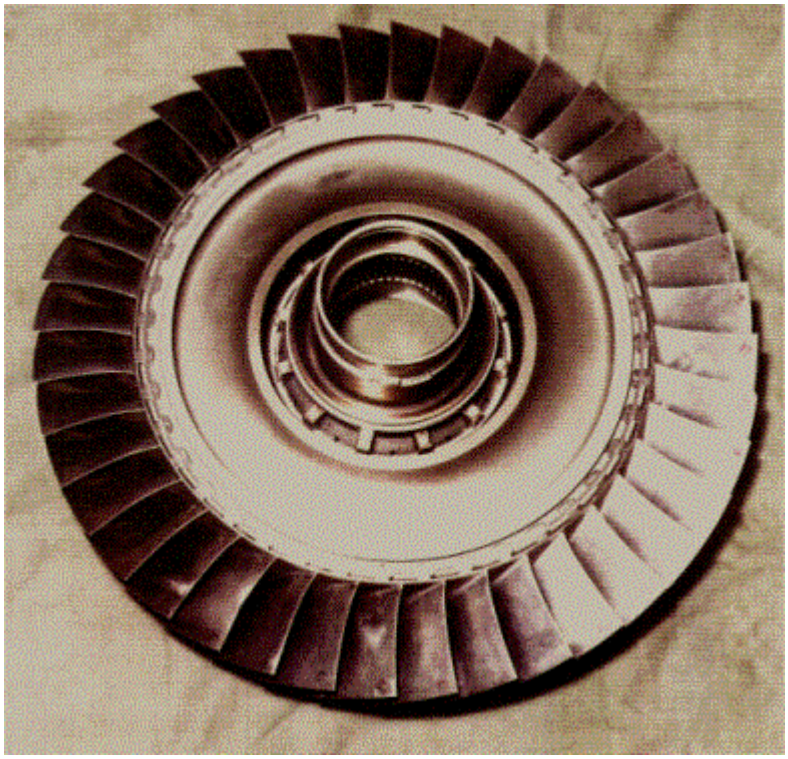

Figure 5-3: Second stage rotor [Morrow, 1993].

5.2 Experimental Procedure

The test setup used for these experiments is shown in Figure 5-4. A distortion generating device was installed downstream of the bellmouth and approximately 1.5 diameters in front of the leading edge of the first stage rotor. The entrance to the compressor was divided into five circumferential tubes of equal mass flow. Radial immersion probes, each one at the center of a circumferential tube and all of them approximately ½ diameter in front of the first stage rotor, measured inlet total pressure. There were a total of eight groups or rakes of inlet total pressure probes equally spaced around the circumference. The radial position of each probe within a rake is shown in Table 5-2.

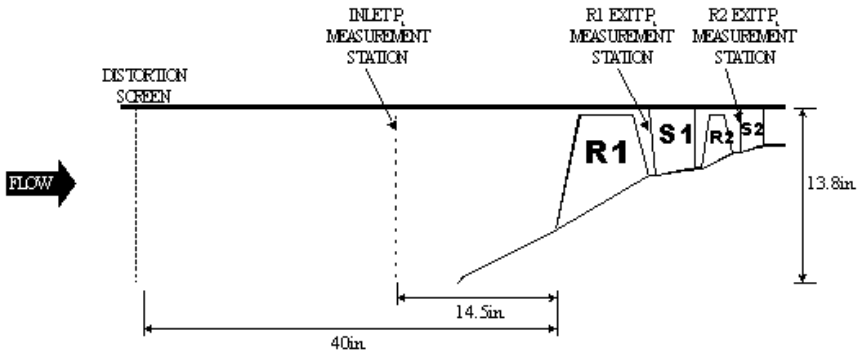

Figure 5-4: Experimental setup [Morrow, 1993].

Immersion	Radial Distance from O.D. (in).
1	0.74
2	2.31
3	4.11
4	6.33
5	9.57

Table 5-2: Radial locations of inlet total pressure probes [Morrow, 1993].

Total pressure was also measured between each stage via seven probes that were imbedded in three equally spaced stator vanes after each of the two stages (see Figure 5-5). The first and second stages probes' positions are shown in Tables 5-3 and 5-4.

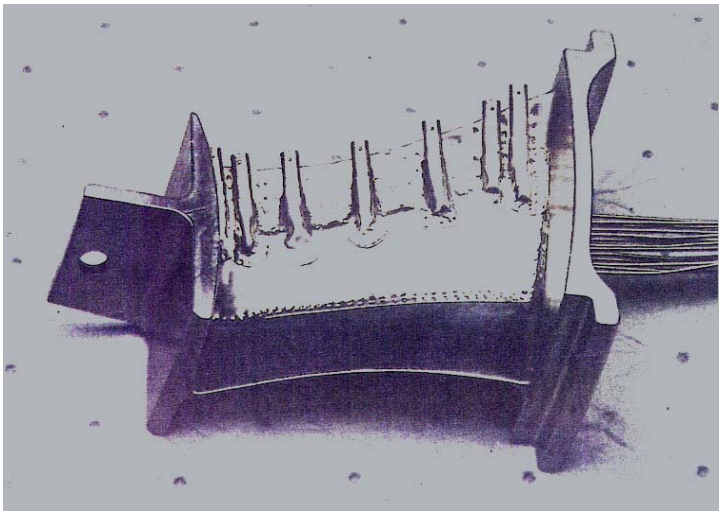

Figure 5-5: Instrumented stator vane [Morrow, 1993].

Immersion	Radial Distance from O.D. (in.)	Axial Distance from R1 LE (in.)
1	0.315	6.60
2	0.662	6.80
3	1.577	7.05
4	2.577	7.27
5	3.573	7.29
6	4.241	7.16
7	4.595	7.10

Table 5-3: First stage stator probe locations [Morrow, 1993].

Immersion	Radial Distance from O.D. (in.)	Axial Distance from R2 LE (in.)
1	0.271	13.07
2	0.479	13.12
3	0.903	13.22
4	1.479	13.30
5	2.011	13.30
6	2.430	13.26
7	2.650	13.22

Table 5-4: Second stage stator probe locations [Morrow, 1993].

The distortion-generating device referenced earlier in this section and depicted in Figure 5-4 that is mounted in front of the compressor inlet is a wire mesh screen of varying porosity. The distortion-generating screen is used to create downstream areas of reduced axial velocity. These areas of reduced axial velocity are also areas of reduced total pressure. The distortion screens used in these experiments were created to cause a circumferentially sinusoidal, steady state distortion. Specifically, 3 per rev and an 8 per rev distortion screens were used. Figure 5-6 depicts a 3 per rev distortion screen.

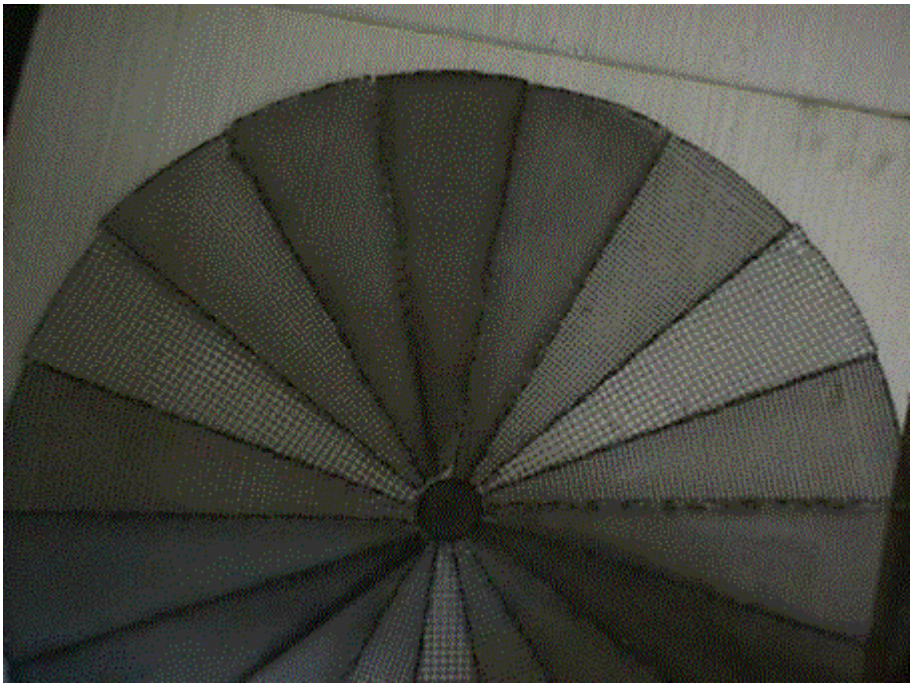

Figure 5-6: Example of a 3 per rev distortion screen [Morrow, 1993].

The distortion generating screens were mounted to a screen rotator for these experiments. The screen rotator consists of a backup screen supported by radial struts. A drive motor, mounted outside the flow field, causes the screen to rotate around the compressor centerline. By rotating the screen, high spatial circumferential data resolution is obtained with a relatively small number of probes. If a non-rotating screen were used, a probe would be required for each spatial position at which a measurement is desired. In other words, if a measurement was desired for each degree around the circumference, 360 probes would need to be mounted, one at each degree, around the compressor's inlet circumference. In order to have good spatial resolution circumferentially and radially, an unreasonably large number of probes would be required, which would complicate data acquisition and produce an unacceptable flow blockage.

With a rotating distortion screen, a single probe's data contains as much or more spatial resolution than would multiple probes used in conjunction with a static screen. Using a high sampling rate with a fixed probe accomplishes this by relating the time of each sample to the position of the screen at that point in

time. In order to use this technique the assumption needs to be made that the screen is rotated slowly enough in order for the data to be considered quasi-steady. Williams [1999] found that a rotation rate of 2 degrees per second is sufficient to make this assumption.

5.3 Data Reduction

The raw data reduction was carried out by Small as part of his M.S. thesis [Small, 2001]. This section summarizes the data reduction procedures from Small's thesis. For further details, the reader is referred to the original document.

Approximately 900 circumferential measurements or one every 0.35 – 0.45 degrees at each probe location were taken for this study. The raw data was then reduced to 360 equally spaced measurements by rounding each data point to the nearest degree (in spatial coordinates) and then averaging all the measurements at the same location or degree. Averaging the probe measurements actually served as a high frequency filter eliminating some of the high frequency noise content.

Streamline position, relative blade velocities, and shock Mach number were calculated by Small [2001] using a streamline curvature (SLC) method. SLC is an iterative solution technique that assumes steady, adiabatic, inviscid, and axisymmetric flow with negligible body forces. Small used SLC code written by Boyer [2001], who had made improvements to the code for transonic applications. Boyer's code calculated 19 axial streamline positions and followed them through each stage of the compressor. The five streamlines that correspond to the five inlet total pressure probes were chosen for this study. Figure 5-7 shows the streamline positions as they travel through the compressor.

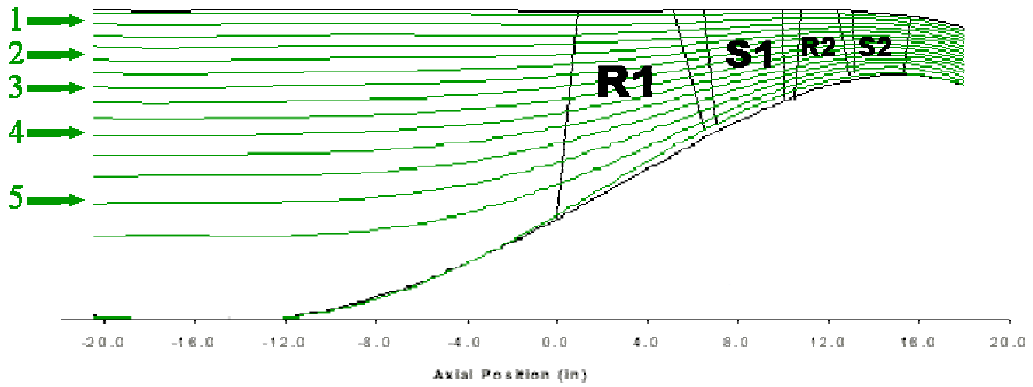

Figure 5-7: SLC computed streamline positions though the compressor [Small, 2001].

Volterra theory deals only with the dynamic component of data. Therefore, it was necessary to subtract the mean from each data set for my study. This has no negative impact on the outcome, as the variation in total pressure from the mean is what creates the distortion and, therefore, causes blade response, which can lead to HCF.

5.4 Data Characteristics

Circumferential total pressure measurements were collected for 11 different operating points of the test compressor. These 11 different operating points are made up of measurements taken at 3 different rotational speeds and 4 mass flow settings. The three speeds are 9100 rpm, 9500 rpm, and 13200 rpm, corresponding to 68.5%, 71.5%, and 99.3% respectively, of design speed. The four mass flow settings at each speed were normal operating line (NOL), near stall (NS), peak efficiency (PE), and wide-open discharge (WOD). The reason that there are 11 data sets as opposed to 12 is that at 13200 rpm, the data sets for mass flow settings of NS and PE were coincident.

A compressor characteristic map showing pressure ratio plotted as a function of corrected mass flow is shown for clean and distorted inlet conditions in Figure 5-8. As can be seen in the plot, a distorted inlet results in dramatically

reduced mass flow at a given rotor speed. Also depicted in the plot is the drop in pressure ratio at a given mass flow setting when operating with a distorted inlet, thus resulting in decreased compressor performance.

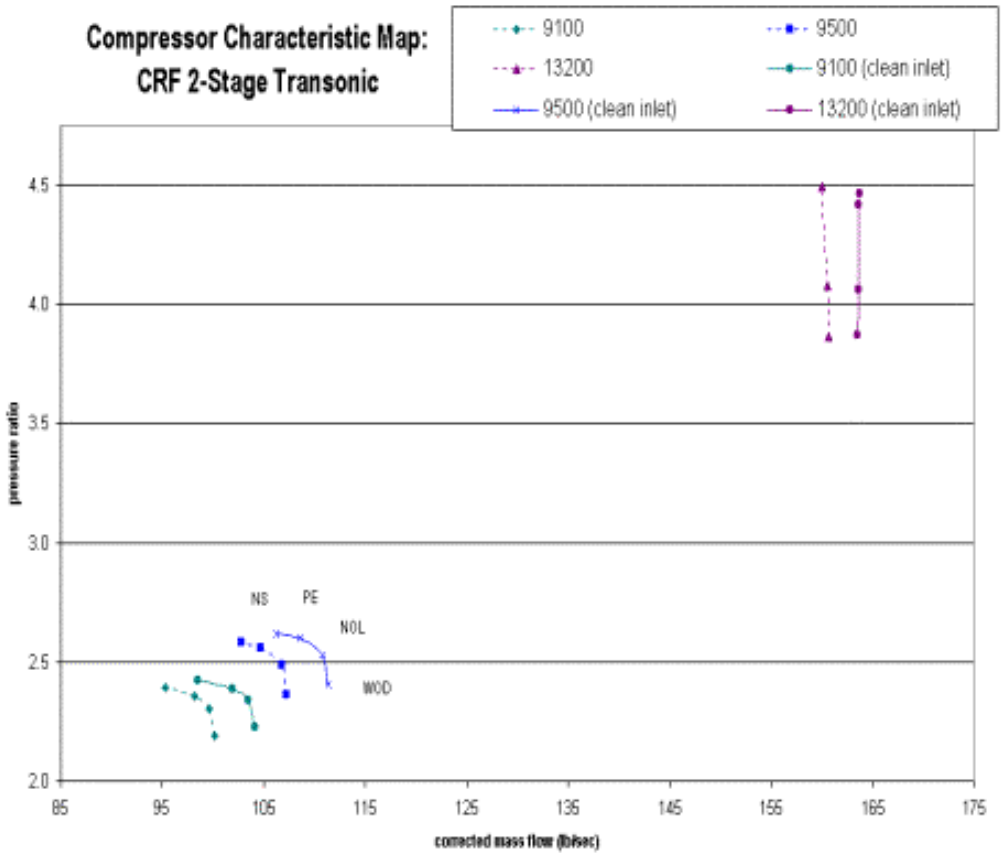

Figure 5-8: Compressor operating map, including clean and distorted inlet conditions [Small, 2001].

As mentioned earlier in this chapter, two different types of distortion-generation screens were used for these experiments, a 3/rev and an 8/rev. The 3/rev screen was used at 9100 rpm and the 8/rev screen was used at 9500 rpm and 13200 rpm. Figure 5-9 shows an example of a typical dataset plotted in the time domain. Since time can be directly related to blade position, the x-axis of this plot represents angular positions around the circumference of the compressor. This particular dataset was measured at 9100rpm, NOL mass flow, and mid-span on the blade. The 3/rev total pressure distortion is easily seen both before and after the first stage. Figure 5-10 portrays the first 15 harmonics

of the frequency domain representation of the same dataset. Since a 3/rev distortion pattern was used in this case, there is dominant third harmonic excitation evident in the magnitude plot. The distortion screens used to generate the total pressure distortions for these experiments are designed to produce a frequency excitation at a specific harmonic with very little secondary harmonic excitation. This can be seen in Figure 5-10 in that there is much less energy at the multiples of the dominant harmonic (the 6th, 9th, or 12th harmonics).

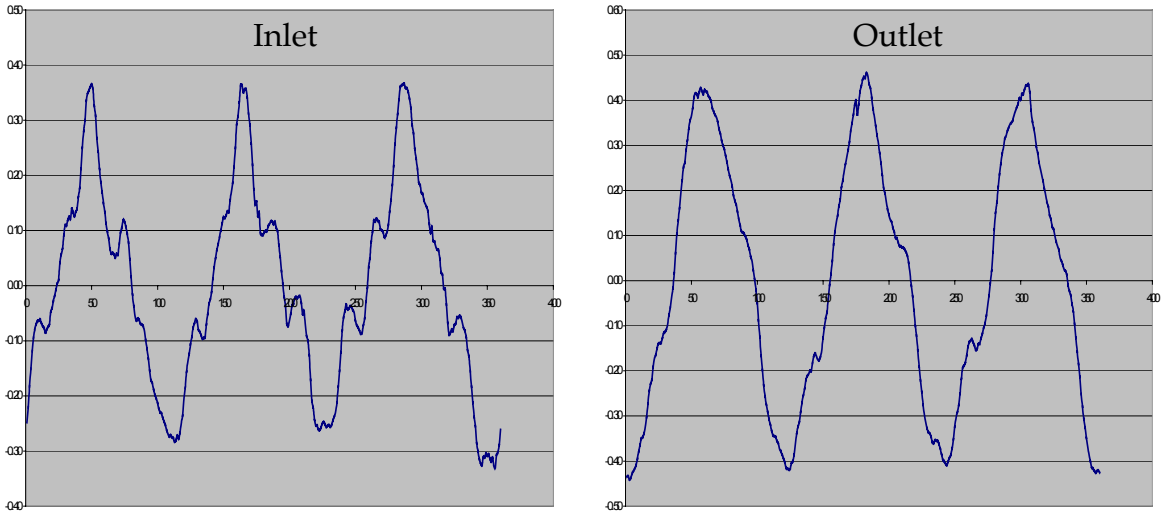

Figure 5-9: Time-series inlet and outlet dynamic total pressure profiles at 9100 NOL RI3.

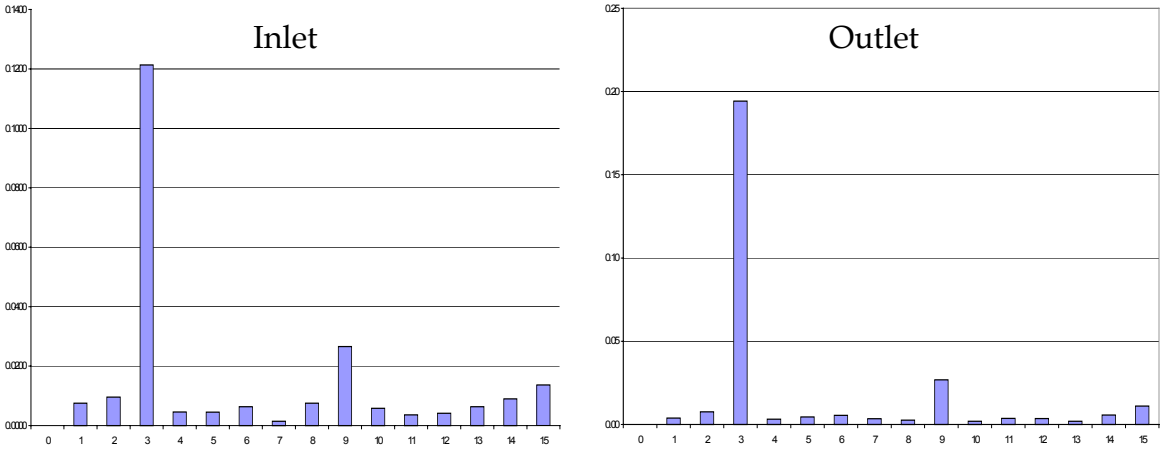

Figure 5-10: Frequency content of 9100 NOL RI3.

A time-series representation of a dataset resulting from an 8/rev distortion screen installed before rotor 1 is shown in Figure 5-11. These plots are of 9500 NOL, RI1 (near tip), before and after the first stage. Figure 5-12 illustrates the first 15 harmonics of the frequency content of the same dataset. The datasets measured at 13200 rpm are similar to the 9500 rpm datasets.

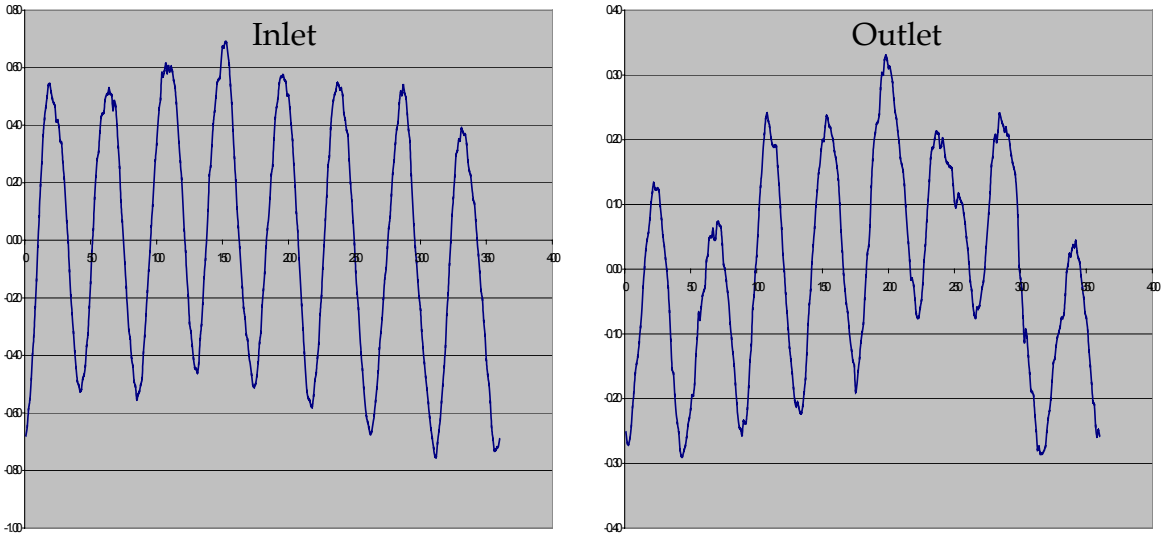

Figure 5-11: Time-series inlet and outlet dynamic total pressure profiles at 9500 NOL RI1.

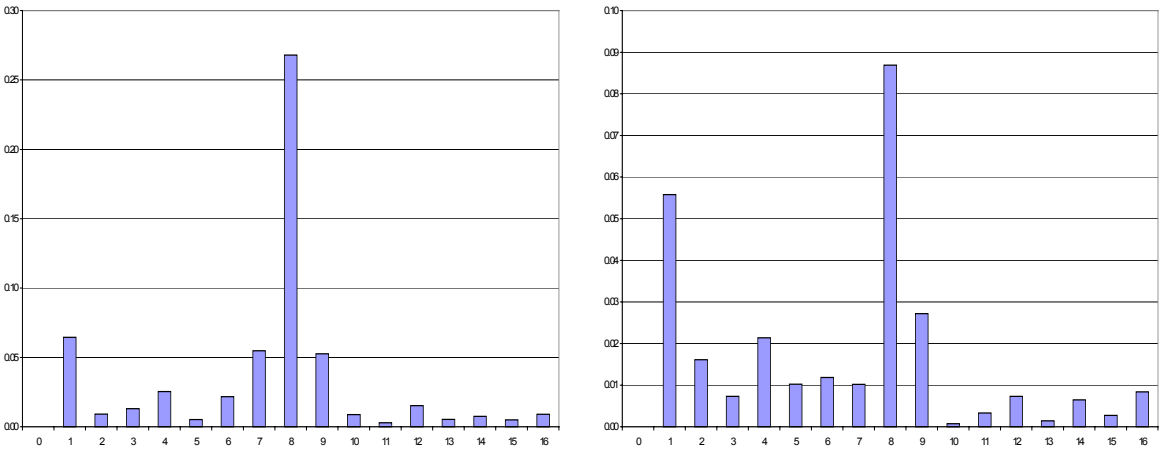

Figure 5-12: Frequency content of 9500 NOL RI1.

Also mentioned earlier in this chapter was that mean-zero data sets were used in my study. The mean value of each separate data set was subtracted out before any analysis was done in order to obtain the dataset's dynamic component. Figure 5-13 shows a plot of dynamic total pressure as a function of

circumferential location for all four mass flow settings at 9100 rpm mid-span. Perhaps surprisingly, mass flow setting has no impact whatsoever on dynamic pressure, as can be seen by the fact that the four plots are practically identical. With the exception of one mass flow setting each at 9500 rpm and 13200 rpm, this was true for all remaining data sets.

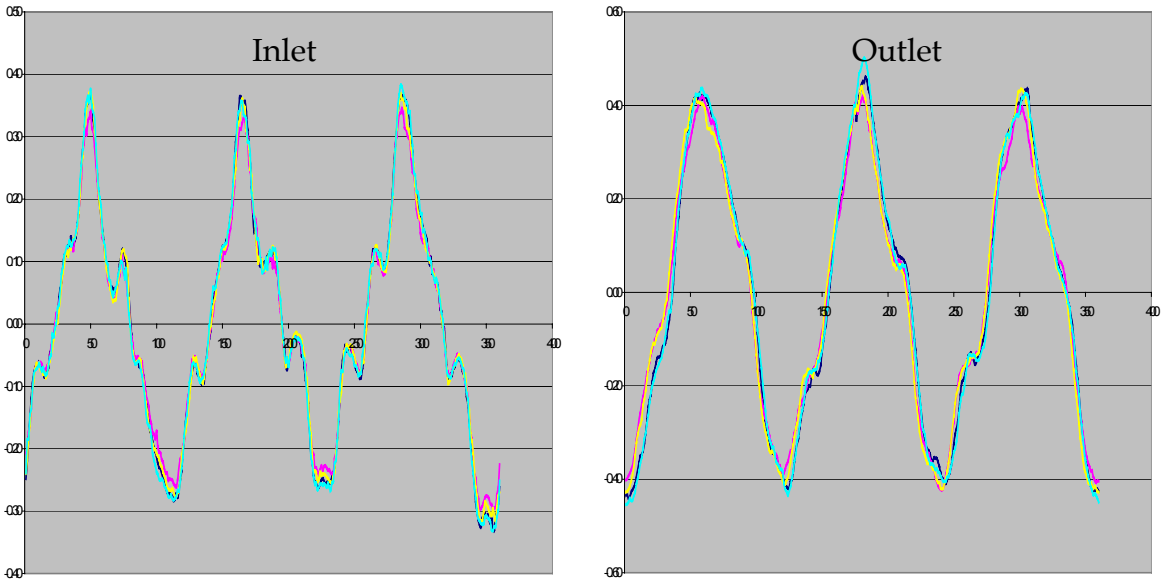

Figure 5-13: First stage dynamic total pressure at 9100 rpm, mid-span, inlet, and outlet.

It is also interesting to look at exactly what happens to the magnitude of the pressure distortion as it travels through the first stage in the test compressor. Figure 5-14 depicts the gain associated with the magnitude of the distortion (maximum total pressure – minimum total pressure) as it travels through the first stage as a function of position on the blade. Therefore, in order for the distortion to travel directly through the first stage with no change whatsoever to its magnitude, it would have a gain of 1, which is depicted by the red horizontal line.

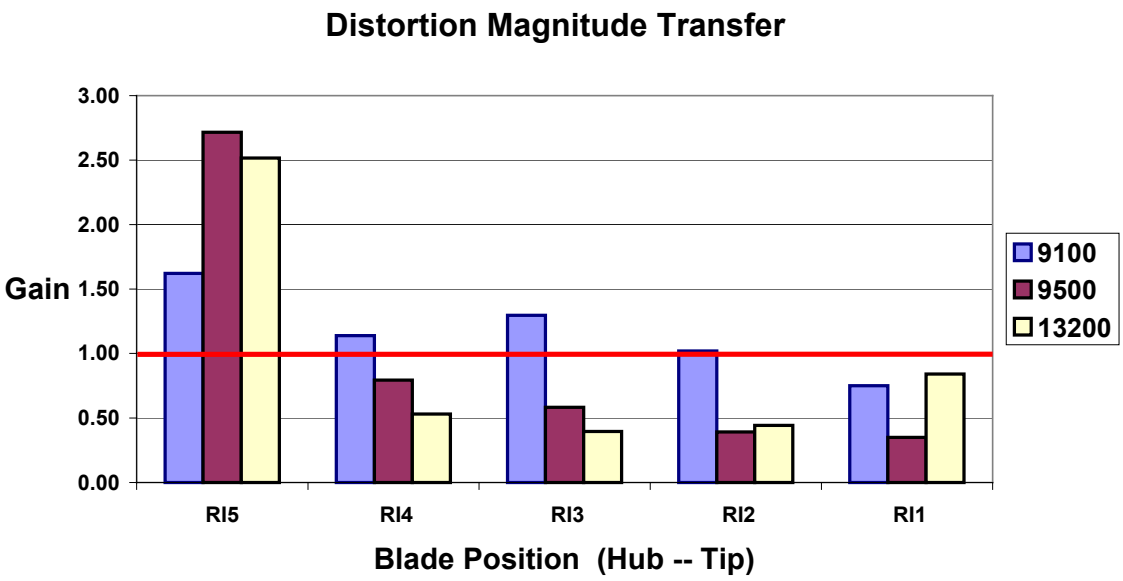

Figure 5-14: First stage distortion magnitude gain as a function of blade position and rotational speed.

As can be seen in Figure 5-14, the transfer of distortion is not only affected by rotational speed, but also radial blade position, with the distortion magnitude actually more than doubling at the hub at 9500 rpm and 13200 rpm. Also remember that each blade position corresponds to its own streamline as a result of the use of the SLC code. Due to the radial dependence of distortion, I chose to apply my compressor model along each individual streamline, calculating 5 separate transfer functions along the span of the blade.

6 Solution Technique

As opposed to airflow over a wing, where the wing in the relative plane is immobile with the air rushing over it, a compressor stage is spinning at a high and varied rate of speed with air rushing through it. This added another dimension and a definite challenge to creating a model. System identification, as well as the choice of non-dimensionalization and normalization parameters, was critical.

This chapter discusses just how the kernel indirect extraction technique as explained in Chapter 4 is used in conjunction with the experimental data as explained in Chapter 5 to create a model of the first stage of the ADLARF test compressor. Included are discussions of further data conditioning, the use of the resultant data sets to ensure that there is enough input to model the system and still allow an accurate assessment of its performance, and different model variations. The first section explains the system definition in order to remove radial dependence from the problem.

6.1 System Identification

An advantage of using Volterra theory to identify the impulse response function of a system is that the Volterra model essentially turns the system in question into a black box. Since the model is purely mathematical, all that it cares about is the inlet and outlet data. One could model any stage or even group of stages in a compressor without making any changes to the model other than altering the input statements to reflect the appropriate system identity.

In my research, I defined each stream tube as it travels through the first stage of the compressor as its own system. This system definition required that the modeling code be run five times – one for each stream tube – in order to predict the first stage exit conditions. The first stage was chosen to model

because there is more data for the first stage and since the inlet air is drawn directly from the atmosphere, there were fewer unknowns and, therefore, assumptions to be made regarding inlet conditions. Additionally, if the model is capable of accurately modeling the first stage, the next stage could be modeled just as accurately by using the outlet predictions from the first stage as the input data for the second.

6.2 Data Set Expansion

Reviewing, the data for my research consists of measurements at 3 rotational speeds, 4 compressor mass flows per rotational speed, 5 streamlines—or stream tubes—per mass flow setting, and 360 measurements—one per degree at the inlet and outlet—per stream tube. By deciding to treat each stream tube of data through the compressor as its own system, I reduced the amount of data available to solve for each kernel by a factor of 5. This, coupled with the fact that the amount of basis functions and resulting coefficients to solve for increase exponentially as the order of the equation increases, caused some concern that the amount of data available would be insufficient to adequately solve for the first three kernels of the system.

In order to ensure that there would be enough information to solve for the first three kernels of the system and still be able to test the model by predicting at an operating condition different from those used to solve for the kernels, more data was needed. There were three options available in order to obtain more data: (1) the original, unconditioned data sets could be used (2) more data could be collected either by myself or by CRF, or (3) data could be "created" by fitting a spline to the existing, conditioned data sets. Since the original data sets contained a measurement approximately every .35 degrees, the amount of information would be almost tripled by using the first option. However, this would also be adding more noise to the problem, which would make the answer

less accurate. Collecting more measurements at other operating conditions was not feasible due to cost and time. Therefore, the third option was the one chosen.

A spline is a flexible ruler that is used to connect plotted data points. Once the spline is shaped so that it lines up with every data point, it is used to draw a line connecting them. Now, new data points can be read from the line that was just drawn. Rather than plotting points on graph paper, computer code was written that accomplished the same task. The type of spline code used was a cubic spline. This code works in essentially the same way as the flexible ruler, only more accurately. Rather than physically drawing a line between all the data points, the cubic spline code works with two consecutive data points at a time. The code considers each group of two data points as a separate function and calculates an equation that will fit those two points. Since each function connects two adjacent data points, the end points of each line already match. By stopping here however, the result would be a jagged plot of little straight lines – not desirable. Since this is a cubic spline code, it is capable of doing three things. Therefore, not only are the end points of each line matched, the first and second derivatives of the end points are forced to match as well. This ensures that the final composite function of all the individual functions provides a smooth fit to all the data points. After calculating this composite function, any output can be calculated for any point between the initial and final data point. The number of data points per stream tube was doubled in this way, resulting in a data point every ½ degree around the circumference of the compressor. Figure 6-1 shows an initial data set of 360 points as well as the data set after running it through the spline code, now consisting of 720 points. As can be seen, the final data set directly overlays the original, proving that the spline code worked properly.

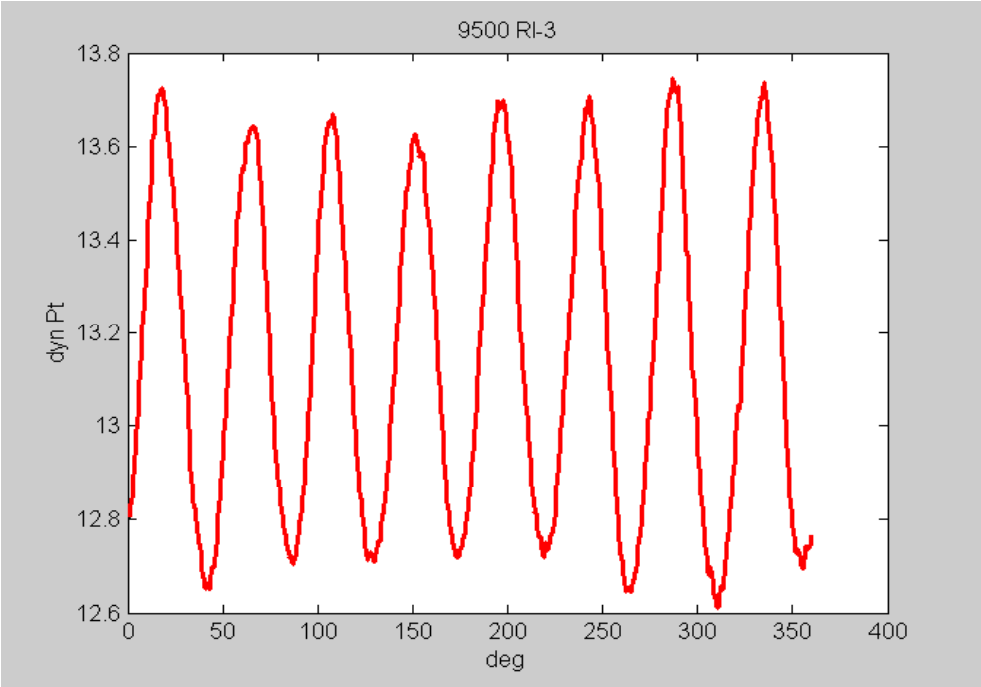

Figure 6-1: Data set before and after being run through spline code. Final data set of 720 points directly overlays the original, which consisted of 360 data points.

6.3 Data Conditioning

Since the Volterra theory is a strictly mathematical theory, it does not account for changes in compressor rotational speed. Therefore, without any type of data conditioning, the Volterra model would have very little hope of accurately predicting compressor response at different inlet conditions than those used to calculate the impulse response function. Due to this, physics-based parameters were used to non-dimensionalize the problem and also to deal with the matter of rotational speed dependence.

The time non-dimensionalization parameter was calculated by dividing the chord length by the inlet relative velocity to the blade (see Equation 6-1). The units of t_{nd} are (s); therefore, time was divided by this parameter in order to obtain a dimensionless time variable. Inlet relative velocity is used because it contains information pertaining to fluid axial and radial velocities at the specified inlet conditions. Chord length is also important since convection speed

-- the speed at which total pressure distortions travel through the rotor – is a function of this variable. Due to the fact that the relative blade velocities are quite large in relation to the chord length – on the order of four magnitudes larger – t_{nd} is a very small number. Dividing by t_{nd} stretches out the time variable, which is desirable due to the rapid changes in inlet and outlet total pressure that occur over small increments in time.

$$t_{nd} = \frac{Chord}{W_1}$$

Equation 6-1

Inlet and outlet total pressure measurements were also non-dimensionalized by dividing by p_{nd} (see Equation 6-2). This variable also includes information pertaining to fluid axial and radial velocities due to the inclusion of inlet relative velocity. In addition, the density of the air at the inlet, ρ_1, is included and multiplication by ½ is done due to convention.

$$p_{nd} = \frac{1}{2}\rho_1 W_1^2$$

Equation 6-2

Some of the Volterra models were also normalized with a non-dimensional work term (see Equation 6-3) [Cousins, 2002]. It was found that some of the models did not adequately account for rotational speed dependence. Multiplication by the non-dimensional work variable defined in Equation 6-3 helped to correct this deficiency. Non-dimensional work was calculated by dividing the stage work by the rotor speed squared, U^2 ($U = r\omega$). Normalizing the data sets with stage work was not necessary in all cases.

$$work_{nd} = \frac{work_{stage}}{U^2}$$

Equation 6-3

All non-dimensional and normalization parameters were calculated with respect to stage inlet conditions and location on the blade. This was necessary because of the system definition. Since there were five separate stream tubes (systems) that made up the first stage, it was required to calculate non-dimensionalization and normalization terms for each system.

6.4 Training and Prediction Data Sets

The data sets used in the indirect extraction technique to calculate the kernels of the system are called the training data sets. This is due to the fact that from these data sets, the model is "taught" the characteristics of the system. The fact that multiple data sets are used in order to calculate the kernels of the system is important since it is doubtful that just one data set could contain all the information necessary to accurately characterize the system. Another important point is that the Volterra model does not simply average all the input and response data sets before extracting the kernels. All the data sets create the boundary conditions that must be satisfied at all times by the basis coefficients, which combined with the basis functions create the system's kernels.

The data sets available consist of measurements taken at three rotational speeds (9100, 9500, and 13200 rpm), four mass flows per speed (NOL, NS, PE, and WOD), and two different distortion patterns (3/rev and 8/rev). The 3/rev distortion pattern was used at 9100 rpm and the 8/rev was used at the other two speeds. None of the speeds contains information of both distortion patterns. In order to capture as much information about the system as possible with these limited data sets, both distortion patterns should be used in kernel training. Also, in order to truly test the model once the kernels were calculated, one of the

three speeds should not be used at all in kernel training. This third speed would be used in the prediction phase of the research as the novel data, meaning that the model would be predicting at a speed it had never seen before. Because of these initial parameters, in the final phase of research, the two speeds used in the kernel extraction phase were 9100 and 13200 rpm. All the data sets at these two speeds were used to calculate the kernels. Predictions were then made using the 9500 rpm data.

As alluded to in the previous paragraph, numerous tests were run through the various Volterra models created in order to gain an understanding of their capabilities and limitations. The next chapter will explain all tests that were run and discuss the results of each in detail.

6.5 Modeling Method Variations

As previous discussed in Chapter 4, more than one Volterra model was created in my research. Different forms of the higher-order basis functions were used, as well as different normalization methods. This was done in an effort to learn as much as possible about the capabilities and limitations of the kernel extraction method used in my research.

In addition to using multiple forms for basis functions, different integrations techniques were also employed. All computer code was written in Matlab, which has numerous built-in functions that were made use of as much as possible. Convolution is one of these functions. Matlab is capable of performing linear convolution in multiple dimensions. However, there is no built-in 2^{nd} or 3^{rd} order convolution function, which is necessary in calculating the higher-order kernels. In two of the methods developed (Model 1 and Model 2), squaring or cubing the input data sets and using the built-in, linear, convolution command addressed the lack of a higher-order convolution function. The problem with this approximation is that all the inputs, or impulses, are forced to occur at the

same time, which isn't necessarily the case. The advantage of this approximation is seen in processing speed. The number of calculations necessary in this method are a number of orders of magnitude less than if a calculation was done for every conceivable combination of impulse times. The cost of this approximation is, obviously, accuracy.

The second integration method utilized was that of numerical integration (Model 3). By using numerical integration rather than the built-in convolution function, every conceivable combination of impulses to the system could be evaluated, making this second integration method the more theoretically correct of the two. Being more theoretically correct should result in an increase in accuracy, with the cost of a dramatic increase in processing time. It is due to this increased processing time as well as a much greater demand for system memory that Model 3 is only a second-order model as opposed to Model's 1 and 2, which are third order. The next chapter discusses the various tests used to assess each model as well as each model's performance in these tests.

7 Prediction Results of Nonlinear Volterra Series Model

All of the discussion and background in the previous chapters leads up to this one final and important question. Does it work? The answer to that question is the focus of this chapter. As will be shown in the subsequent discussions, each model (Model 1, 2, and 3) was evaluated using the same tests. Initially, all three models performed similarly. However, as the tests became more difficult, differences began to materialize. These differences were in the form of prediction accuracy and also analysis speed.

The tests were preformed in such a way that the level of difficulty steadily increased. This was done in an effort to fully understand the capabilities and limitations of this method of analysis as well as each individual model. Before the experimental data from the ADLARF tests were used, data sets were generated from simple functions. In these initial tests, the input waveform was kept constant while the output waveform was varied. In this way, different "systems" were created, each "system" having a different effect on the input. This type of evaluation is similar to the "measurement" kernel identification method as discussed in Chapter 4 and used by Boyd [1983] in his research. Whereas Boyd altered the input to a system in order to only elicit a certain type of response, I altered the "system" itself by changing the output data resulting from a constant input. Section 7.1 discusses these initial tests in more detail.

The remainder of testing done on each model was with the experimental data. Predictions were made in the time domain and then converted to the frequency domain with the output of each run including both. The prediction results were then cut and pasted into Excel spreadsheets, which is the primary method of presentation. Rather than showing all the time domain and frequency domain plots at each radial position for all three models, the following format

will be followed. Plots will be shown of the predictions results for one of the three models in the time domain as well as the magnitude plots in the frequency domain. Results will be shown for each of the five radial positions. Any major differences between the three models will be included in the discussion of the plots and a table summarizing each model's performance will be included at the end of that particular section for the sake of comparison. Since prediction results are shown in the time and frequency domain, the models performance is assessed in both as well. The standard deviation of the prediction as compared with the experimental data is used in the time domain with a perfect prediction having a standard deviation of zero. In the frequency domain, the percent error of the magnitude of the dominant harmonic is used for performance assessment. An error of zero percent would represent a perfect prediction of the magnitude of the dominant harmonic.

7.1 Validation of Model

In order to initially test the model, data sets were generated using sine functions. Sine functions were chosen for two reasons. First, sine waves, being periodic, resemble a waveform caused by a non-uniform inlet pressure profile. Second, by using sine functions to generate data sets, the exact order of operation of the "system" is known. This enabled the calculation of each individual kernel to be evaluated separately based on the data sets that were used as input and output waveforms.

For all of the sine tests performed, the input waveform was sin(x) while the output waveform was some form of sin(x), such as $\sin^2(x)$, cos(x), or the combination of two sine waves. Over 10 different tests were run of varying difficulty. In all of the sine tests, the input and output waveforms were first used to calculate, or extract, the system's kernels. Next, the same input waveform was used with the kernels just calculated to predict the system's output. That

prediction was then compared to the output waveform initially used in kernel extraction to assess how well the model was working. Since these tests were just reproducing the same waveform used in kernel extraction, they are not a true assessment of the model's ability to fully define a system. However, the ability to reproduce the same data used in kernel extraction is an important and necessary step that a model must be able to do before any more difficult scenarios can be attempted. These sine tests were also a good first opportunity to evaluate the different higher-order forms of basis functions as well as methods of integration (Models 1, 2, and 3).

The Volterra models treat the system as if it were a black box, only being concerned with the input and output waveforms as they enter and leave the system. The models are only concerned with what must be done to the input waveform to make it look like the output. Therefore, the models do not recognize these waveforms as sine waves or any type of function whatsoever, they just see an input vector that is changed in some way to end up looking like the output vector.

Results from three of the sine tests are shown in Figures 7-1, 2, and 3. These plots are just of the output waveforms; the input waveform for every sine test was sin(x). The figures shown have outputs in the form of $\sin^2(x)$, $\sin^3(x)$, and sin(x) + sin(2x), respectively. All three models were tested with all of the various data sets generated with sine functions. There was no difference in performance between the models at this level of testing. Each model was able to exactly replicate the output data set that was used in the kernel calculation phase, as can be seen in the following figures.

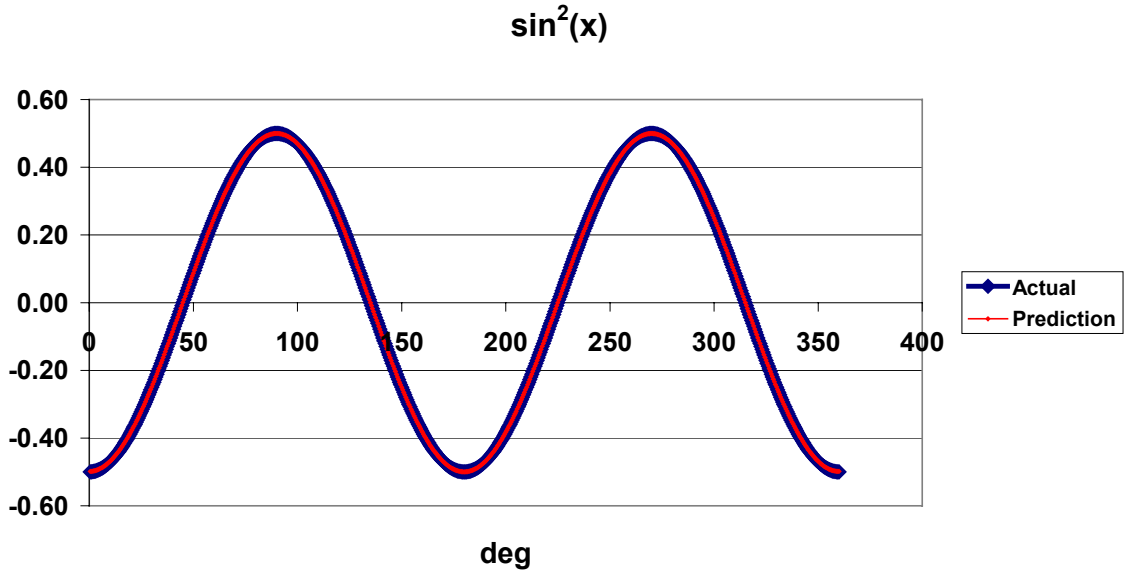

Figure 7-1: Prediction result from kernels calculated with input of sin(x) and output of $\sin^2(x)$. Prediction directly overlays the actual data set analytically calculated from sine squared.

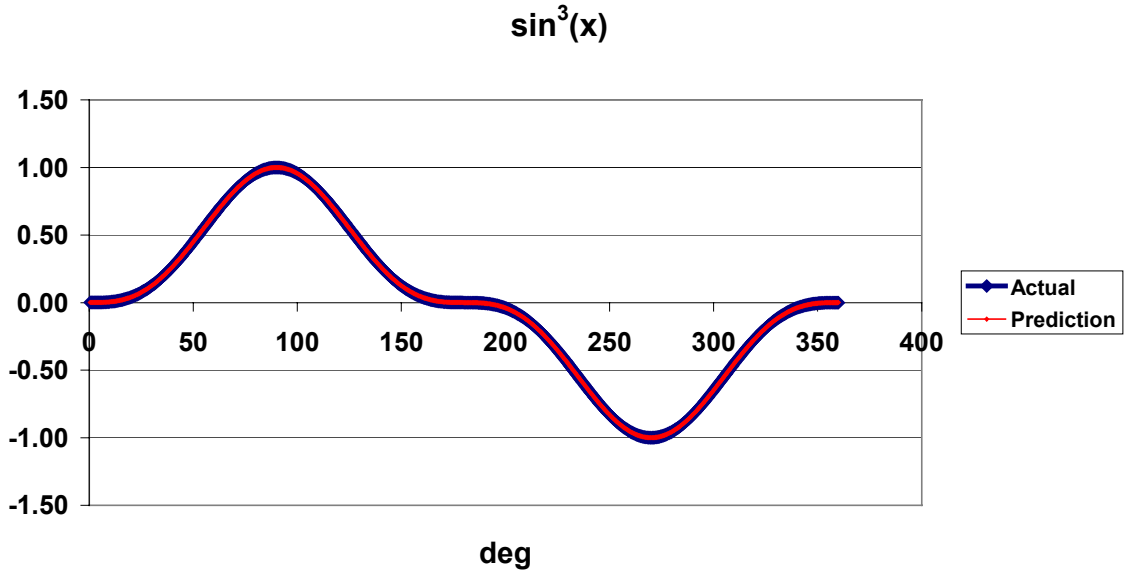

Figure 7-2: Prediction result from kernels calculated with input of sin(x) and output of $\sin^3(x)$. Prediction directly overlays the actual data set analytically calculated from sine cubed.

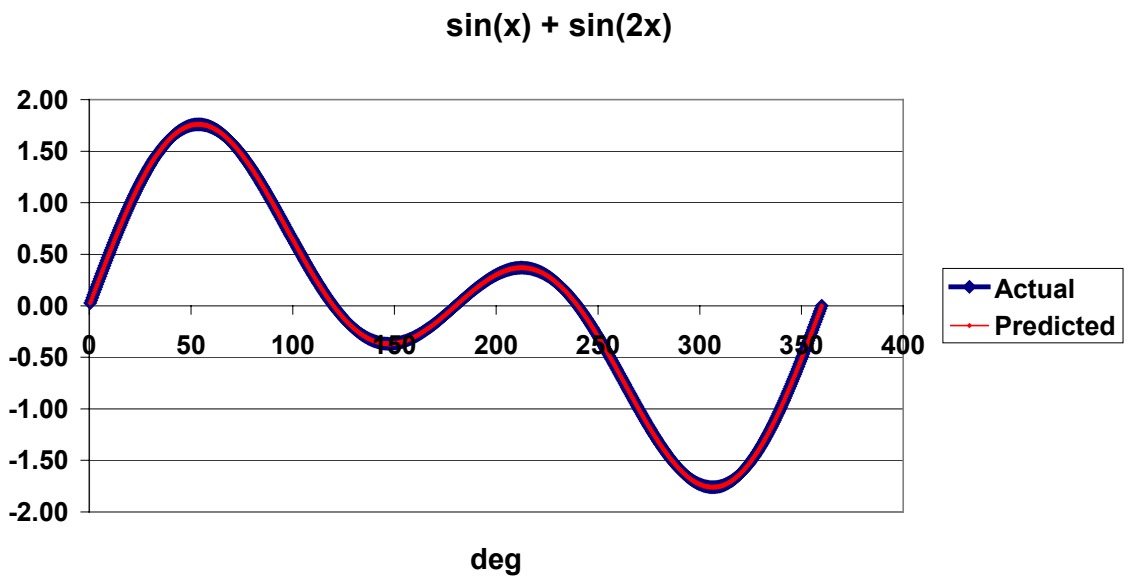

Figure 7-3: Prediction result from kernels calculated with input of sin(x) and output of sin(x) + sin(2x). Prediction directly overlays the actual data set analytically calculated from the addition of sin(x) and sin(2x).

7.2 Same Speed Predictions

The next stage of testing involves predicting the variation in total output pressure of the compressor at the same rotational speed that the kernels were calculated. This particular stage has two parts. The first part is replication of the same data used in kernel calculation, similar to the sine tests. The input and output data sets corresponding to just one operating condition (9100NOL RI3, for example) were used for kernel calculation. The same input data set was then used in conjunction with the previously calculated kernels to predict the compressor's response. Each of the three models was tested under the same operating conditions (9100NOL, all 5 radial positions). Figures 7-4 through 7-8 show the time domain predictions at all five radial positions from Model 3. The corresponding magnitude plots in the frequency domain of those same predictions are shown in Figures 7-9 to 7-13.

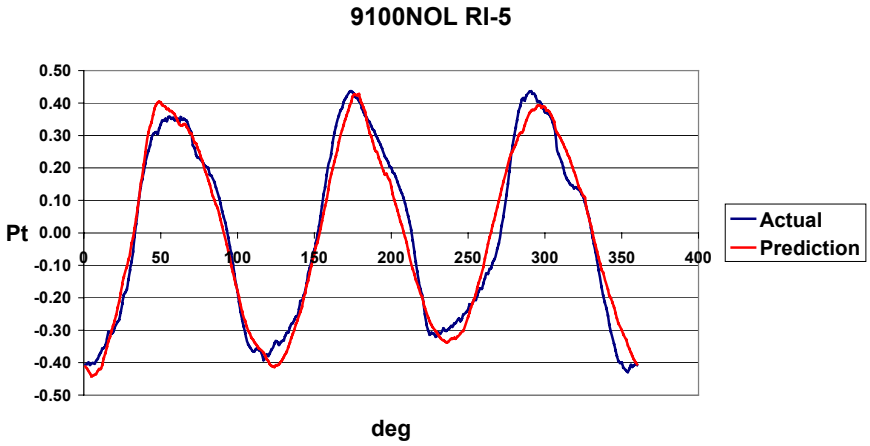

Figure 7-4: Time domain prediction obtained with Model 3 of the output variation in total pressure at 9100NOL, RI5 after calculating the system kernels with the same data.

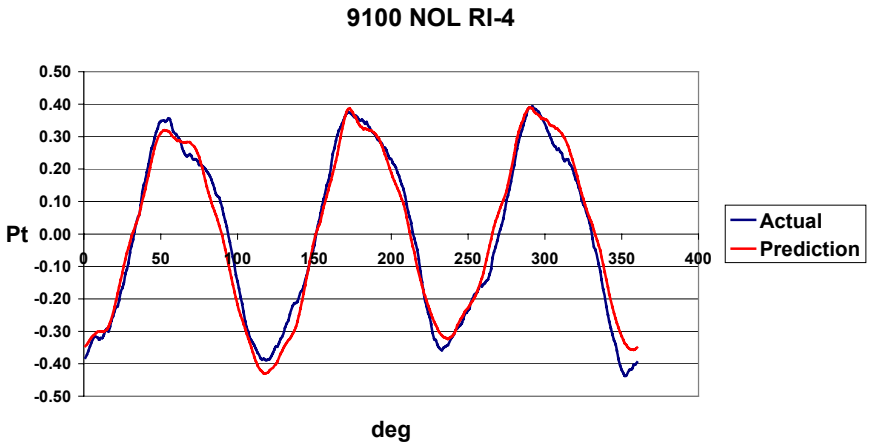

Figure 7-5: Time domain prediction obtained with Model 3 of the output variation in total pressure at 9100NOL, RI4 after calculating the system kernels with the same data.

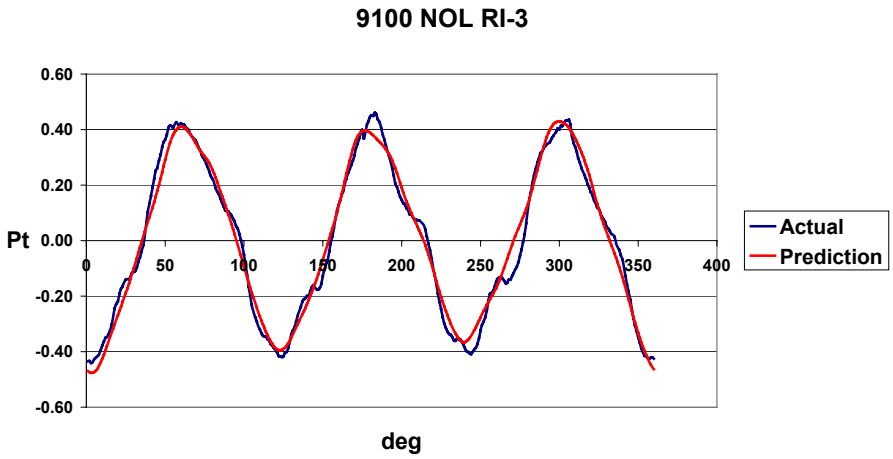

Figure 7-6: Time domain predictions obtained with Model 3 of the output variation in total pressure at 9100NOL, RI3 after calculating the system kernels with the same data.

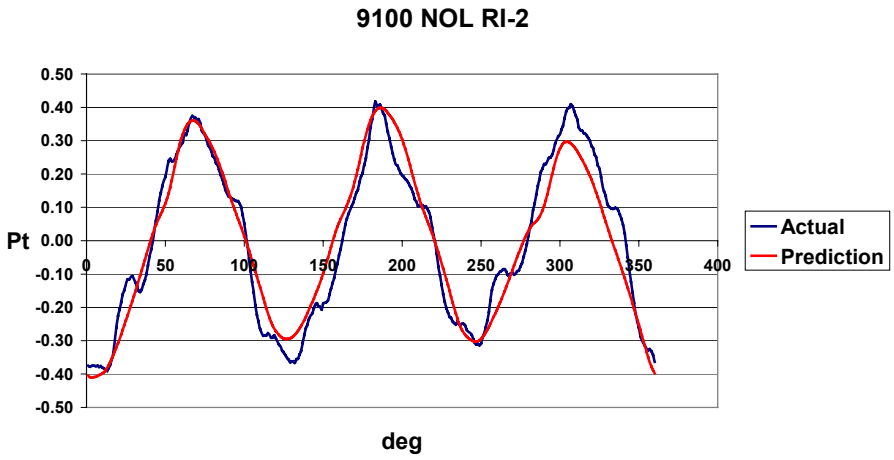

Figure 7-7: Time domain predictions obtained with Model 3 of the output variation in total pressure at 9100NOL, RI2 after calculating the system kernels with the same data.

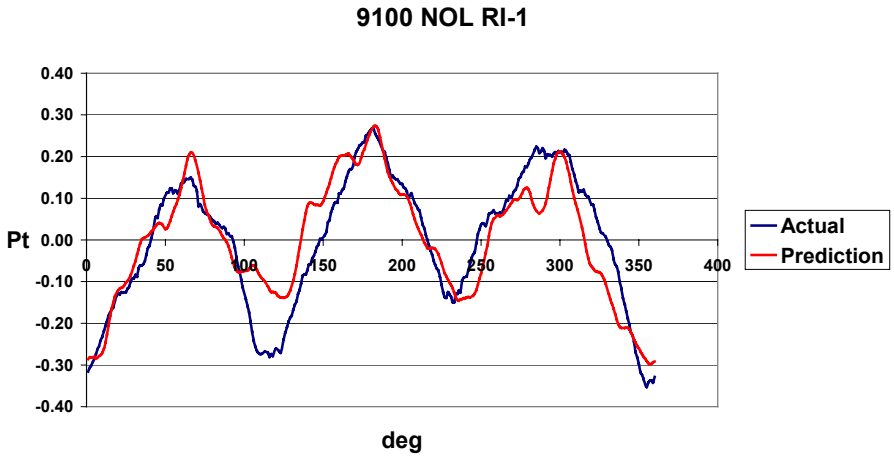

Figure 7-8: Time domain predictions obtained with Model 3 of the output variation in total pressure at 9100NOL, RI1(near tip) after calculating the system kernels with the same data.

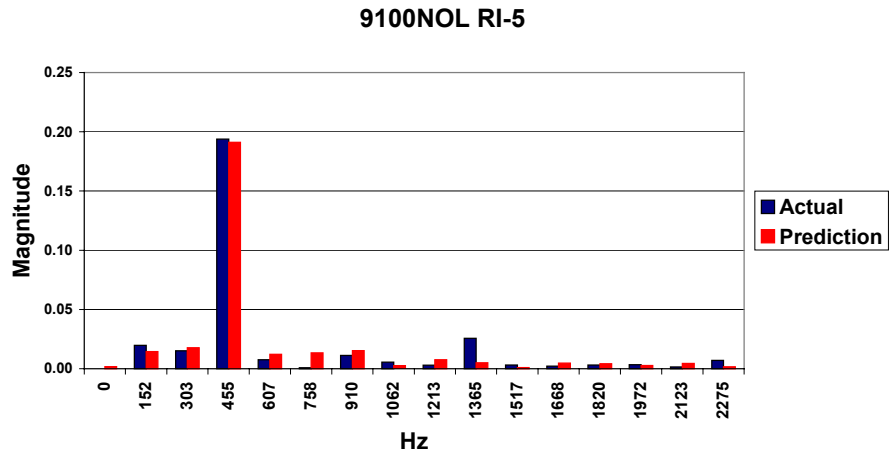

Figure 7-9: Magnitude plot in the frequency domain of the output variation in total pressure corresponding to the prediction shown in Figure 7-4 that was made with Model 3.

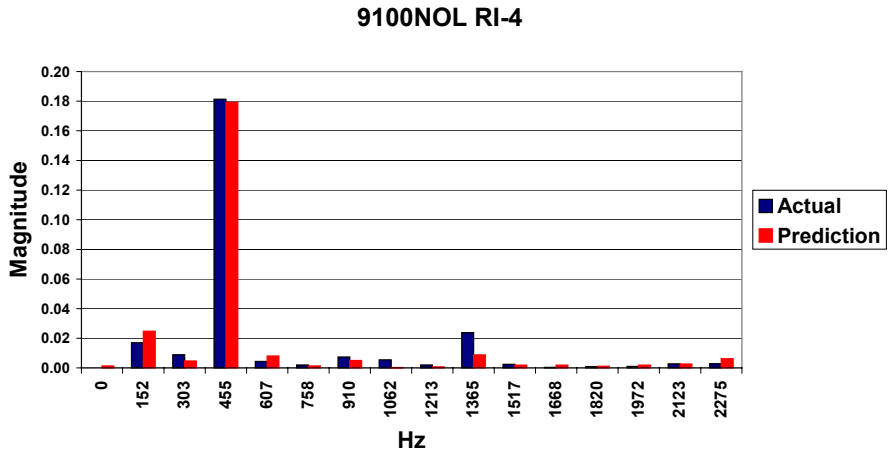

Figure 7-10: Magnitude plot in the frequency domain of the output variation in total pressure corresponding to the prediction shown in Figure 7-5 that was made with Model 3.

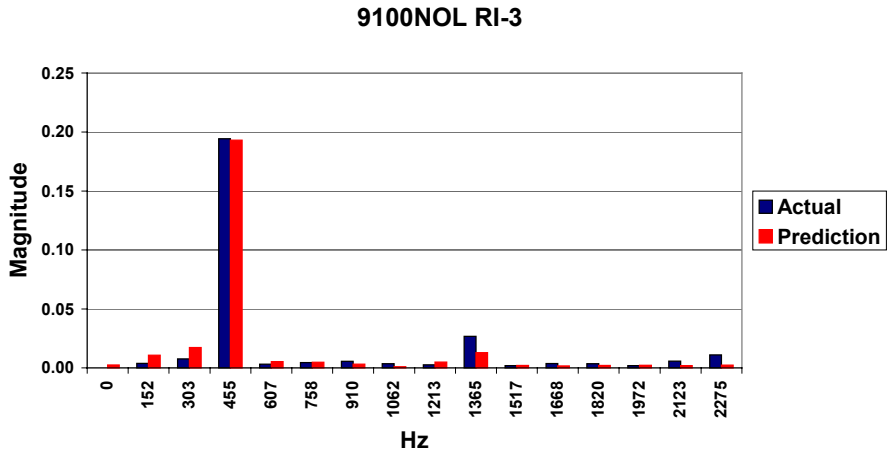

Figure 7-11: Magnitude plot in the frequency domain of the output variation in total pressure corresponding to the prediction shown in Figure 7-6 that was made with Model 3.

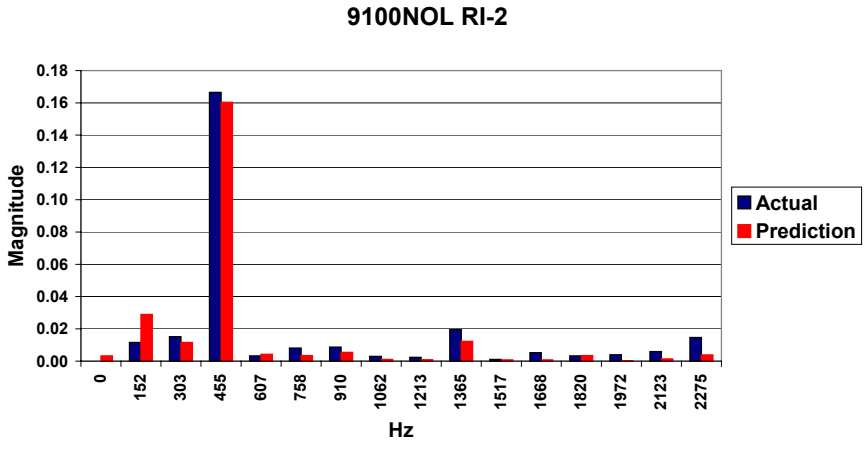

Figure 7-12: Magnitude plot in the frequency domain of the output variation in total pressure corresponding to the predictions shown in Figure 7-7 that was made with Model 3.

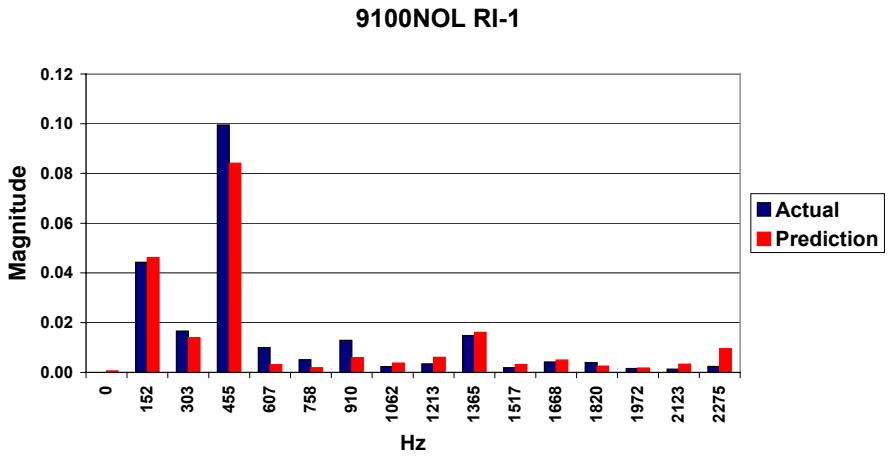

Figure 7-13: Magnitude plot in the frequency domain of the output variation in total pressure corresponding to the prediction shown in Figure 7-8 that was made with Model 3.

The first thing noticed when looking at the time domain plots is that there is an obvious 3 per rev distortion pattern, depicted by the 3 peaks and valleys in total pressure about the mean. There is not a dramatic difference in any one of the peaks or valleys, meaning that the areas of higher/lower density material on the screen are identical. This would cause the sinusoidal total pressure profiles seen in Figures 7-4 through 7-8. The 3 per rev distortion is also plainly seen in the frequency magnitude plots. The third harmonic, at 455 hertz, is obviously the dominant one. In the magnitude plots, the third harmonic is actually the fourth bar. This is because the first bar represents the zeroeth harmonic, or the magnitude associated with the average total pressure. Since all data sets were mean-zeroed, this value is zero. Any magnitude associated with the zeroeth harmonic as predicted by the model is an error.

As seen in Figures 7-4 through 7-8, the model does a good job of replicating the overall shape of the distorted pressure profile in the time domain at all radial locations with the exception of RI1, near tip. In the frequency domain, Figures 7-9 through 7-13, the model seems to do an even better job of accurately predicting the magnitude of the dominant harmonic associated with the distorted pressure profile. At 9100 rpm, a three per rev distortion screen was used to create the total pressure distortions. This is plainly evident when looking

at both the time domain and frequency domain plots. In Figures 7-9 through 7-13, it can be seen that the frequency associated with the third harmonic is approximately 450 hz. It can also be seen in the magnitude plots (Figures 7-9 through 7-13) that the distortion is strongest from the hub to the center of the blade with the magnitude decreasing to half the strength at the tip. In this test, the model is more accurate in that region as well. Despite the fact that this is coincidental, it is encouraging just the same.

Table 7-1 shows how each of the three models performed in this particular test. The best performer at each radial immersion is highlighted in blue in the frequency domain and red in the time domain. Since the goal of predicting total pressure distortion transfer is to help eliminate the incidence of high cycle fatigue, the prediction of the dominant harmonic in the frequency domain was chosen as the more important over the time domain prediction. The best performer in the frequency domain is highlighted in blue for this reason.

	Model 1		Model 2		Model 3	
domain	time	freq	time	freq	time	freq
radial position	std	% error	std	% error	std	% error
RI5 - hub	0.068	**-1.0%**	0.084	-4.6%	**0.054**	-1.5%
RI4	0.051	-1.0%	**0.042**	**-0.6%**	0.044	-1.1%
RI3	0.049	**0.0%**	0.044	**0.0%**	**0.042**	-0.5%
RI2	0.068	**-1.8%**	**0.064**	-2.4%	0.066	-4.2%
RI1 - tip	0.077	**-15.0%**	0.082	-16.0%	**0.074**	-16.0%

Table 7-1: Prediction results when replicating the same data used in kernel calculation (9100NOL) in the time and frequency domain of each model. Best prediction is highlighted in blue in the frequency domain and red in the time domain.

All three models performed very well with there being little difference in the standard deviation of their predictions as well as the percent error of the dominant harmonic magnitude. Model 1, however, did have the lowest error rate. Model 1 also has the least complicated basis function of the three and subsequently, has the shortest run time. Since in this test only one data set was being evaluated at a time, run time wasn't as much of an issue, although there is a definite difference. Model 1 takes approximately 10 minutes to extract the kernels and then replicate or predict the output waveform, Model 3 being the

most complicated needs approximately 2 hours, and Model 2 requires approximately 30 minutes.

The second level of testing done in this phase of analysis involves using three of the four data sets at the same rotational speed (different mass flows) for kernel calculation and then predicting the system response for the fourth mass flow. Remembering that mass flow variation at the same rotational speed has little impact on distortion transfer [Section 5.4]; this is still not a very difficult test. It is, however, a logical progression in assessing each model's capabilities before attempting prediction at an operating condition completely different from those used in kernel calculation. In these tests, data sets from 9500 NOL, NS and PE were used in the kernel calculation stage and then the compressor output was predicted at 9500WOD. The results of the predictions of Model 1 are shown in Figures 7-14 through 7-23 and are in the same format as before, 7-14 through 7-18 are in the time domain with 7-19 through 7-23 depicting the magnitude plots in the frequency domain of those same predictions. Once again, the input waveform is not shown in these plots, just the measured output and the model's prediction.

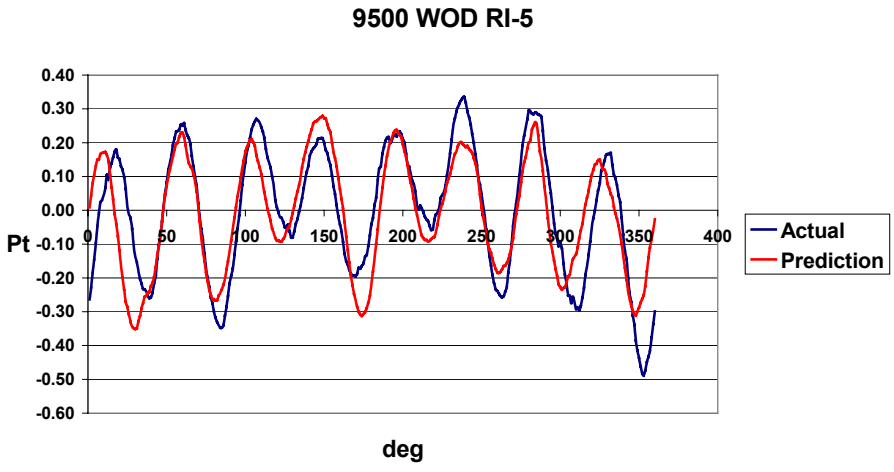

Figure 7-14: Prediction of Model 1 of the output variation in total pressure at 9500WOD, RI5 after calculating the system kernels with the other data sets at that same rotational speed.

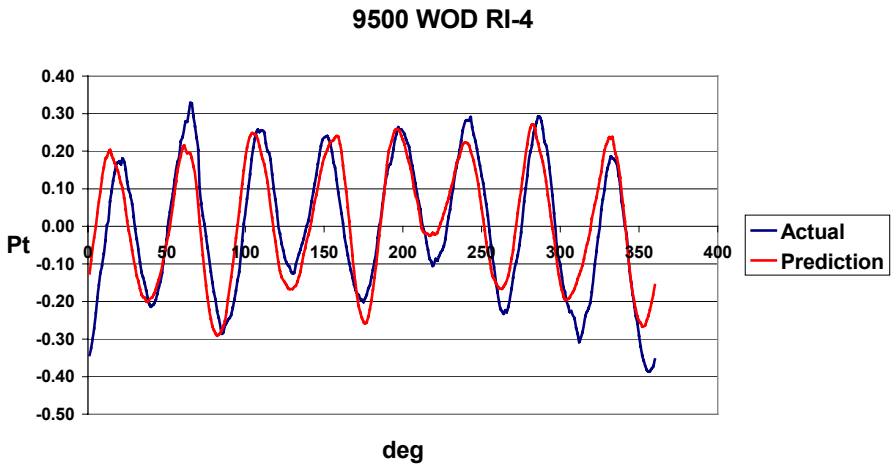

Figure 7-15: Prediction of Model 1 of the output variation in total pressure at 9500WOD, RI4 after calculating the system kernels with the other data sets at that same rotational speed.

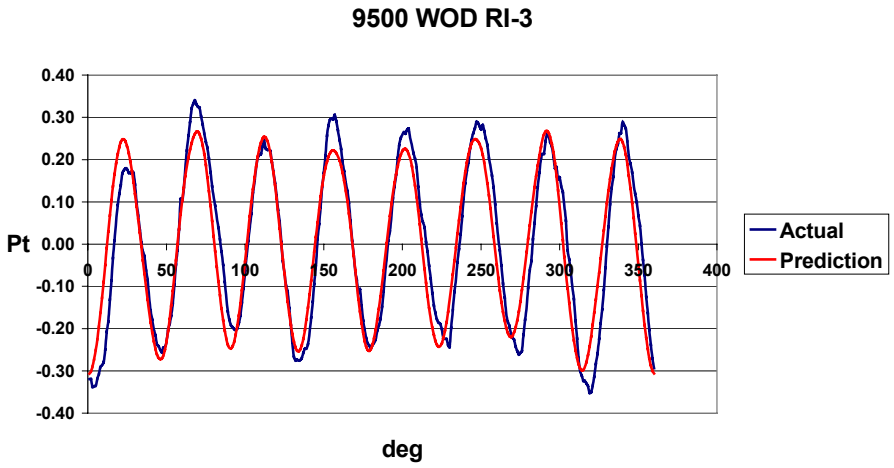

Figure 7-16: Prediction of Model 1 of the output variation in total pressure at 9500WOD, RI3 after calculating the system kernels with the other data sets at that same rotational speed.

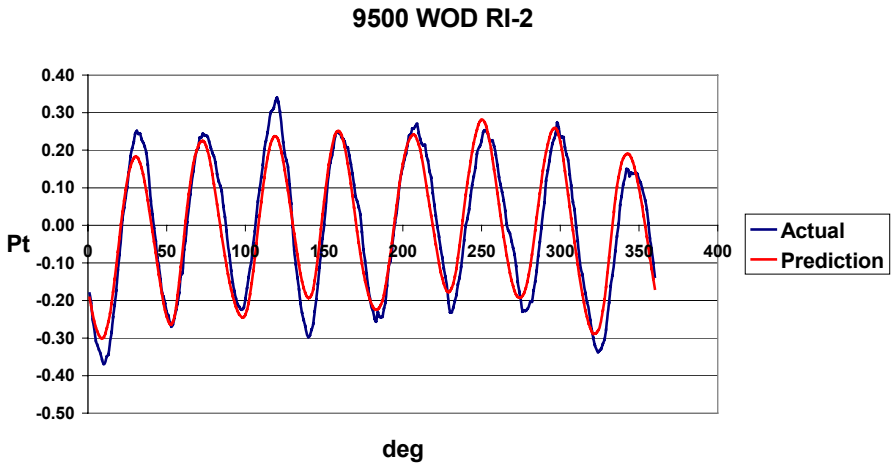

Figure 7-17: Prediction of Model 1 of the output variation in total pressure at 9500WOD, RI2 after calculating the system kernels with the other data sets at that same rotational speed.

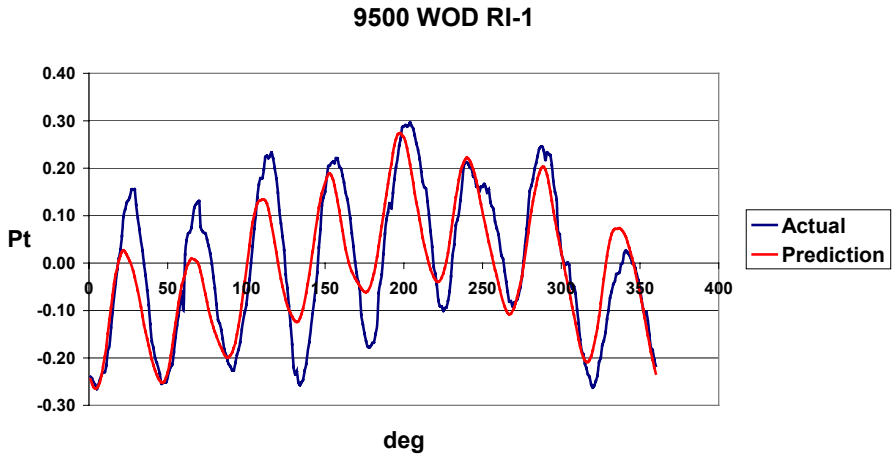

Figure 7-18: Prediction of Model 1 of the output variation in total pressure at 9500WOD, RI1 after calculating the system kernels with the other data sets at that same rotational speed.

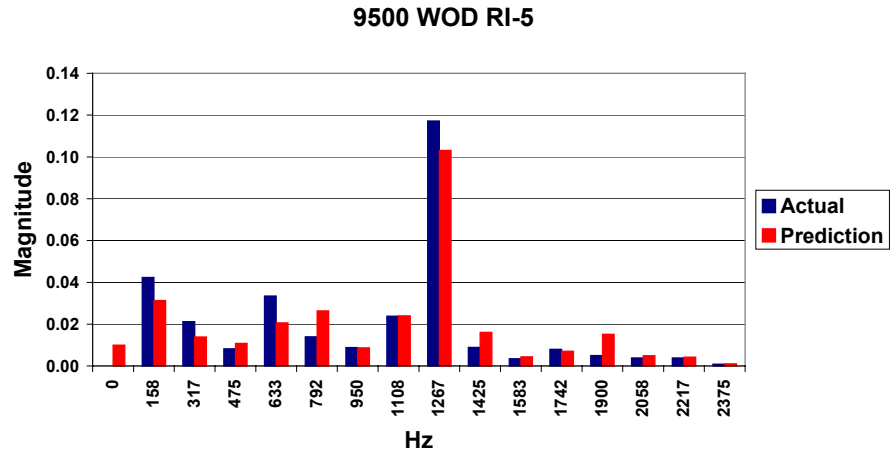

Figure 7-19: Magnitude plot in the frequency domain of the output variation in total pressure corresponding to Figure 7-14 that was made with Model 1.

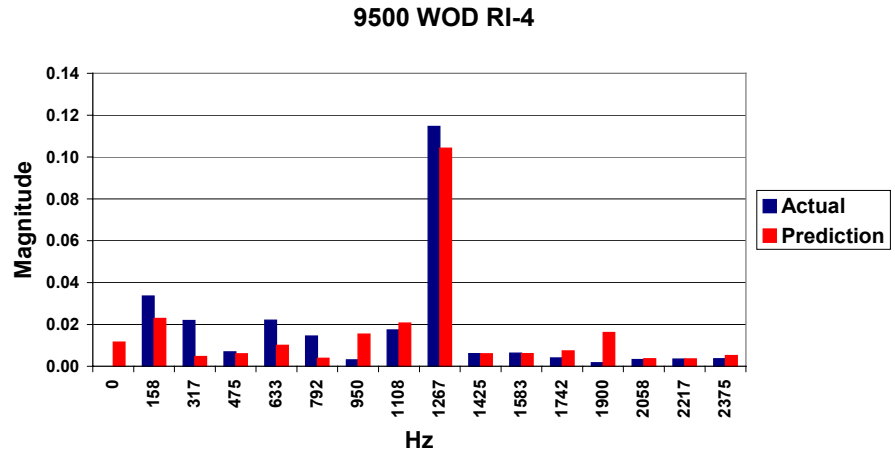

Figure 7-20: Magnitude plot in the frequency domain of the output variation in total pressure corresponding to Figure 7-15 that was made with Model 1.

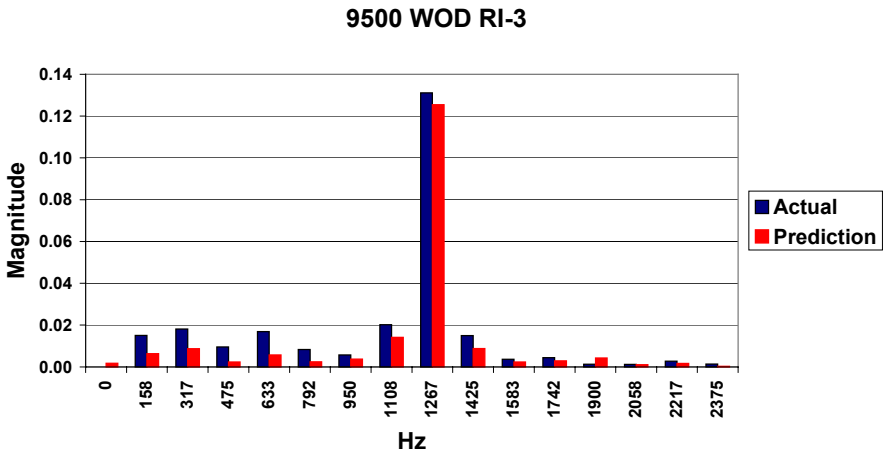

Figure 7-21: Magnitude plot in the frequency domain of the output variations in total pressure corresponding to the prediction shown in Figure 7-16 that was made with Model 1.

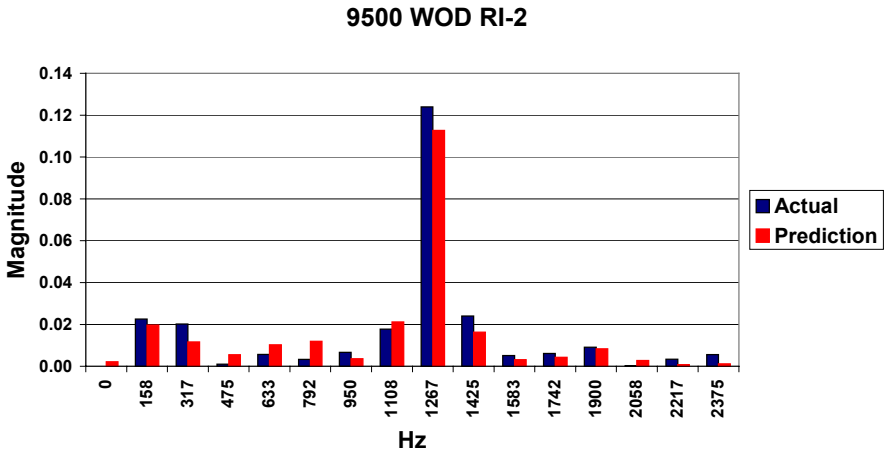

Figure 7-22: Magnitude plot in the frequency domain of the output variations in total pressure corresponding to the prediction shown in Figure 7-17 that was made with Model 1.

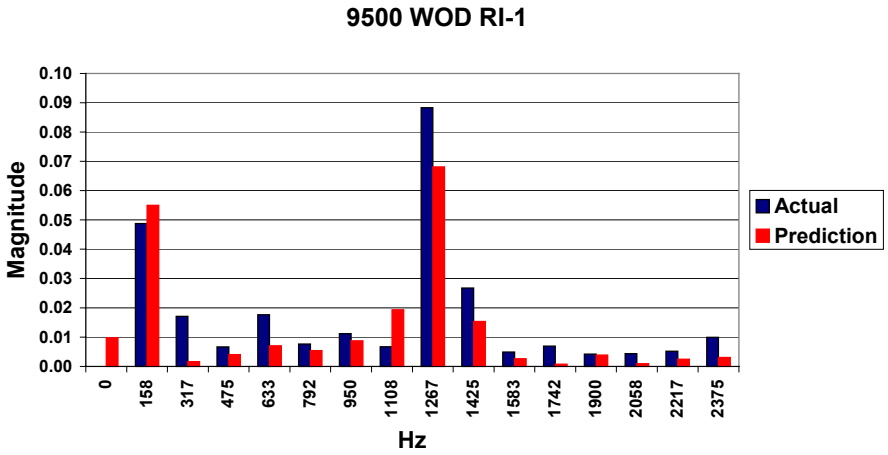

Figure 7-23: Magnitude plot in the frequency domain of the output variations in total pressure corresponding to the prediction shown in Figure 7-18 that was made with Model 1.

One of the first things noticed in these plots is that a different type of distortion screen was used at this rotational speed, an 8 per rev. This can be seen in both the time domain and the frequency domain plots. In the time domain, there are 8 peaks and valleys of total pressure variation about the mean corresponding with the 8 higher density areas in the distortion screen that was used. It can also be seen that there is not a dramatic difference in the magnitude of the peaks, meaning that the screen is made up of 8 identical and repeating sections of higher and lower density material. In the frequency domain plots, the 8 per rev distortion is obvious by the larger magnitude of the 8th harmonic, occurring at just under 1270 hertz.

Since the level of difficulty is not much greater in this test than the first series of tests (replication of the same waveform used in kernel calculation), the results also look quite good in the time domain as well as the frequency domain. It is difficult to see much difference in the time domain plots from the previous tests. The prediction of the magnitude of the dominant frequency is not quite as good as in the previous tests, but it is still very close to the actual data measurements. Table 7-2 shows the standard deviation of the prediction and the percent error of the magnitude of the dominant harmonic for each model. The format of Table 7-2 is identical to that of Table 7-1, with the best prediction highlighted in blue in the frequency domain and red in the time domain.

	Model 1		Model 2		Model 3	
domain	time	freq	time	freq	time	freq
radial position	std	% error	std	% error	std	% error
RI5 - hub	0.115	-11.9%	0.119	**-9.4%**	**0.111**	-12.8%
RI4	0.092	-9.6%	0.094	-8.7%	**0.091**	**-7.8%**
RI3	0.072	**-4.6%**	0.08	-5.3%	**0.071**	-6.1%
RI2	0.064	-8.9%	0.07	-12.9%	**0.06**	**-6.5%**
RI1 - tip	0.079	-22.7%	0.081	**-20.5%**	**0.078**	-23.9%

Table 7-2: Prediction results of compressor output at 9500WOD from 9500NOL, NS, and PE in the time and frequency domain are shown.

Once again, the difference between the three models is not dramatic and any one of them could be used with confidence to predict the output pressure

profile of an alternate mass flow at the same rotational speed. Nevertheless, Model 3 is the clear winner of the three in the time domain. The results are fairly evenly split in the frequency domain, however, which is the more important of the two when dealing with high cycle fatigue. The very slight increase in accuracy in the time domain comes at a fairly high cost in processing time. Model 3 took approximately 4 hours to run, the other two models were finished in less than 45 minutes.

Realizing however, that the impact of mass flow on distortion is minimal, there really is not a need to predict distortion transfer at variable mass flow but constant rotational speed. Once data is measured at one mass flow at any rotational speed, the data in this study indicates that the variation in total pressure about the mean at any other mass flow would be the same [Section 5.4].

7.3 Different Speed Predictions

The final and true test of the Volterra models is to predict the compressor response to an input at operating conditions completely different from those that were used during kernel extraction. For these tests and as discussed in section 6.4, all data sets at 9100rpm and 13200rpm were used for kernel extraction. By using these data sets, the kernels will contain information pertaining to the response of the compressor stage to a 3 per rev and an 8 per rev distortion as well as two rotational speeds, 9100 and 13200rpm. Predictions were then made using the input waveforms measured at 9500rpm, NOL mass flow using the previously calculated kernels. Figures 7-24 through 7-33 show the predictions of Model 2 as compared to the measured output data.

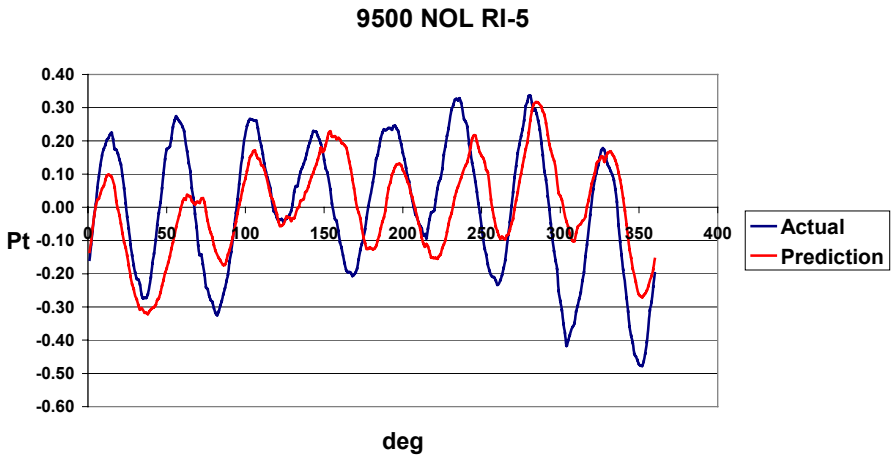

Figure 7-24: Prediction of Model 2 of the output variation in total pressure at 9500NOL, RI5 after calculating the system kernels with the data sets from the other two rotational speeds.

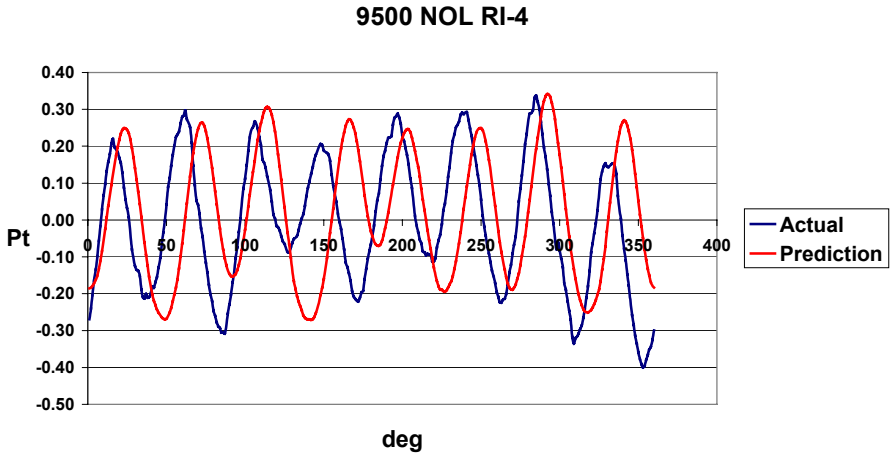

Figure 7-25: Prediction of Model 2 of the output variation in total pressure at 9500NOL, RI4 after calculating the system kernels with the data sets from the other two rotational speeds.

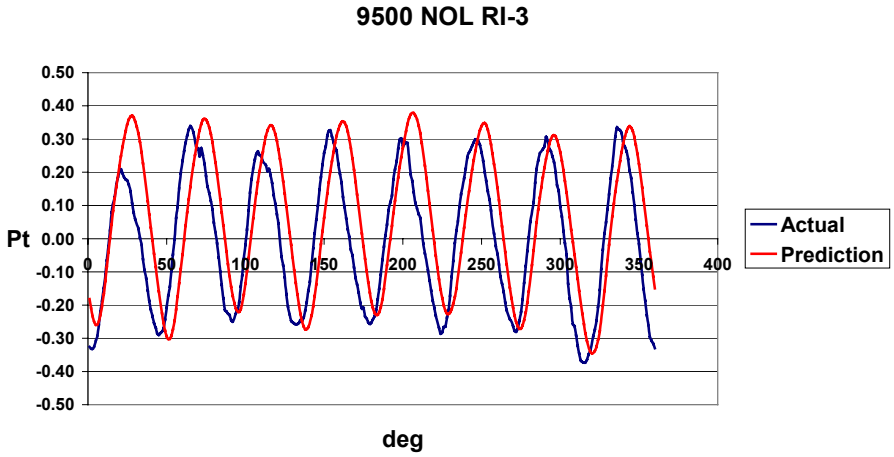

Figure 7-26: Prediction of Model 2 of the output variation in total pressure at 9500NOL, RI3 after calculating the system kernels with the data sets from the other two rotational speeds.

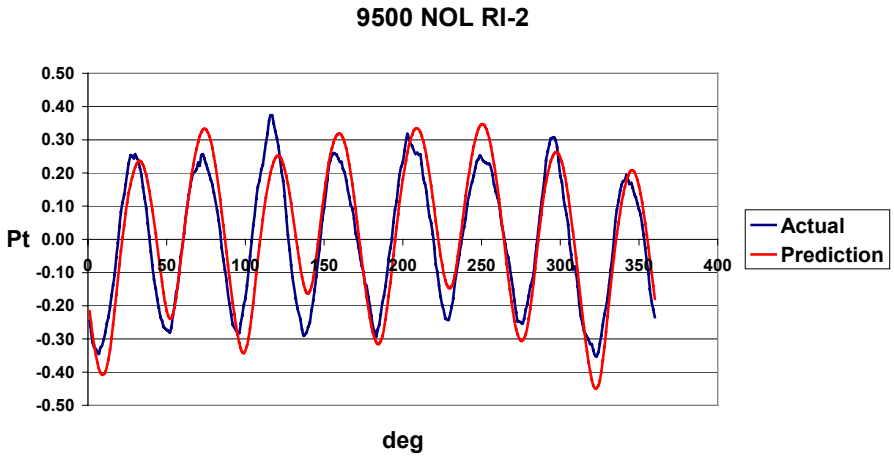

Figure 7-27: Prediction of Model 2 of the output variation in total pressure at 9500NOL, RI2 after calculating the system kernels with the data sets from the other two rotational speeds.

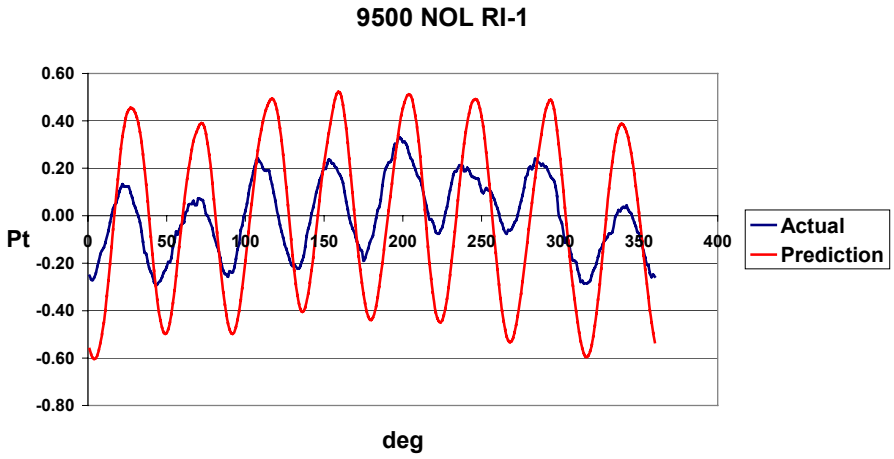

Figure 7-28: Prediction of Model 2 of the output variation in total pressure at 9500NOL, RI2 after calculating the system kernels with the data sets from the other two rotational speeds.

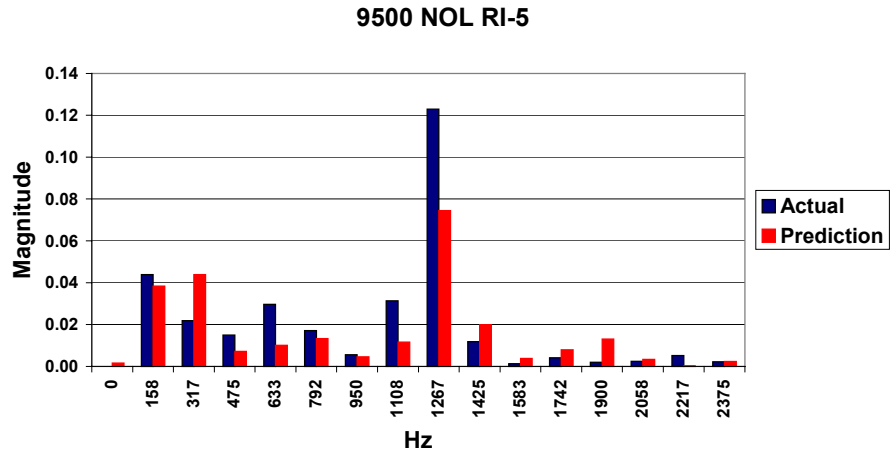

Figure 7-29: Magnitude plot in the frequency domain of the output variation in total pressure corresponding to the prediction shown in Figure 7-24 that was made with Model 2.

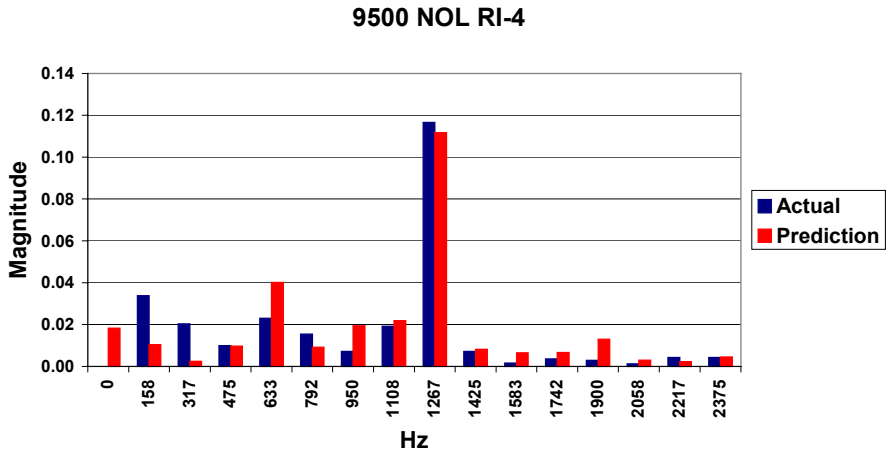

Figure 7-30: Magnitude plot in the frequency domain of the output variation in total pressure corresponding to the prediction shown in Figure 7-25 that was made with Model 2.

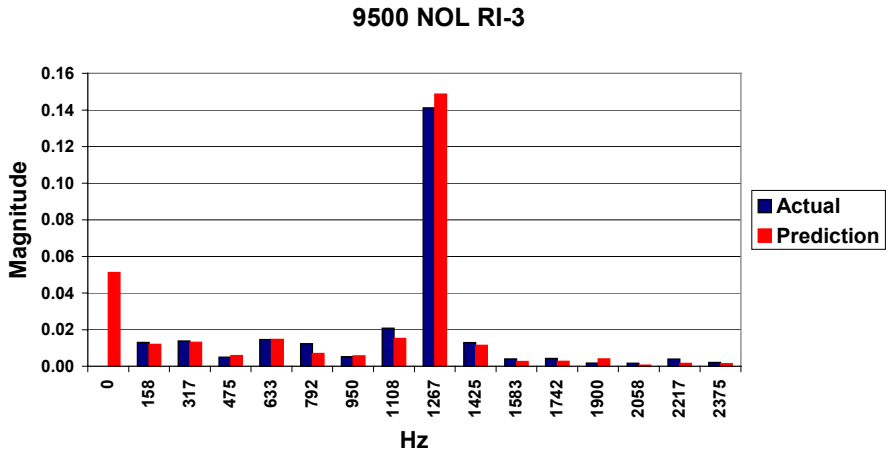

Figure 7-31: Magnitude plot in the frequency domain of the output variation in total pressure corresponding to the prediction shown in Figure 7-25 that was made with Model 2.

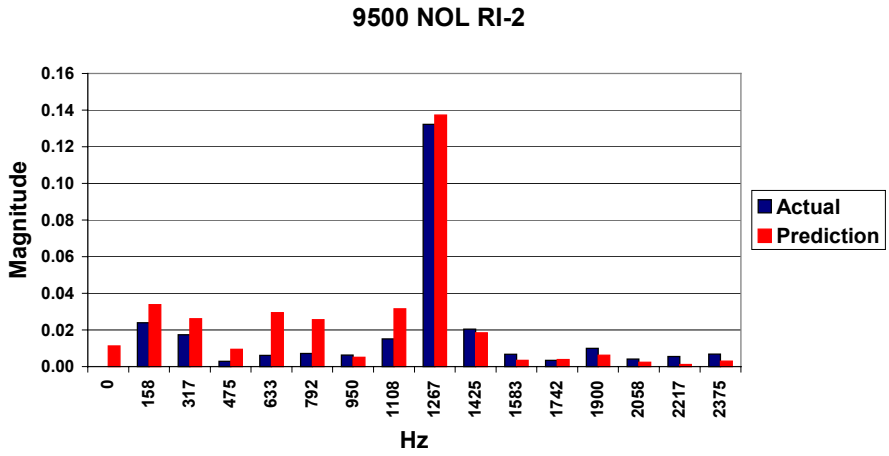

Figure 7-32: Magnitude plot in the frequency domain of the output variation in total pressure corresponding to the prediction shown in Figure 7-26 that was made with Model 2.

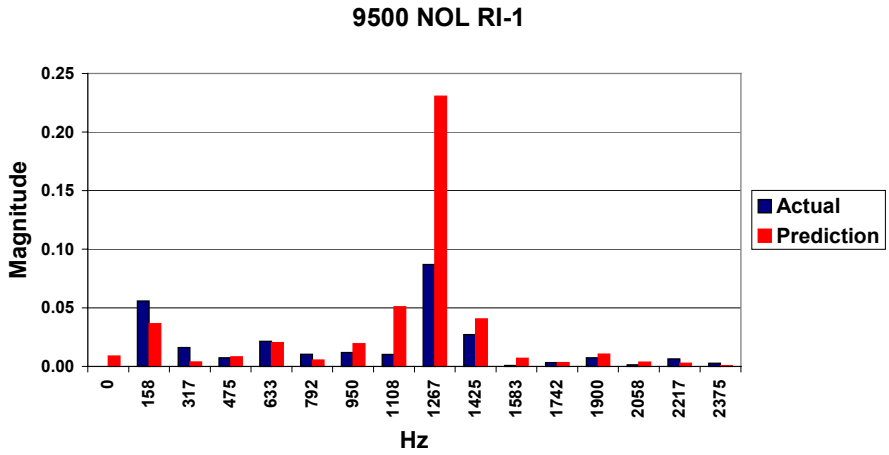

Figure 7-33: Magnitude plot in the frequency domain of the output variation in total pressure corresponding to the prediction shown in Figure 7-27 that was made with Model 2.

The predictions of all three models were not as accurate in these tests, which was suspected due to the increased difficulty. Prediction results also varied dramatically dependent on radial location. All three models were much more accurate at the three interior points of the blade than at the hub and tip where performance was not near as good. It was also revealed in these tests that Models 1 and 3 did not account for the rotational speed dependence adequately. Normalizing the input and output data sets with a non-dimensional work term as discussed in Section 6.3 helped to correct this deficiency [Cousins, 2002].

An example of the impact of normalizing the data sets with non-dimensional stage work within Model 1 is shown in Figures 7-34 through 7-37. Figures 7-34 and 35 show the prediction of output variation in total pressure at 9500rpm, NOL mass flow, RI3 (mid-span) in the time domain and also the magnitude value in the frequency domain. Figures 7-36 and 37 show the prediction results at the same operating condition after normalizing the data sets with non-dimensional stage work.

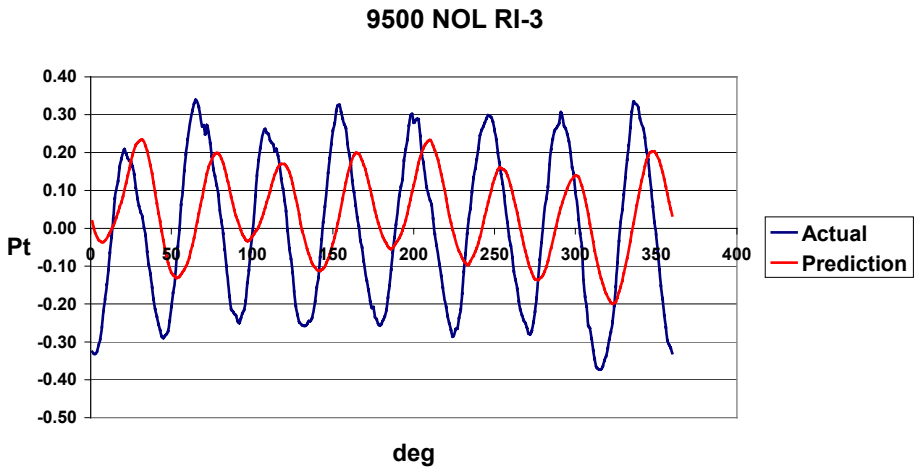

Figure 7-34: Prediction of Model 1 of the output variation in total pressure at 9500NOL RI3, after calculating the system kernels with the data sets measured at 9100 and 13200rpm.

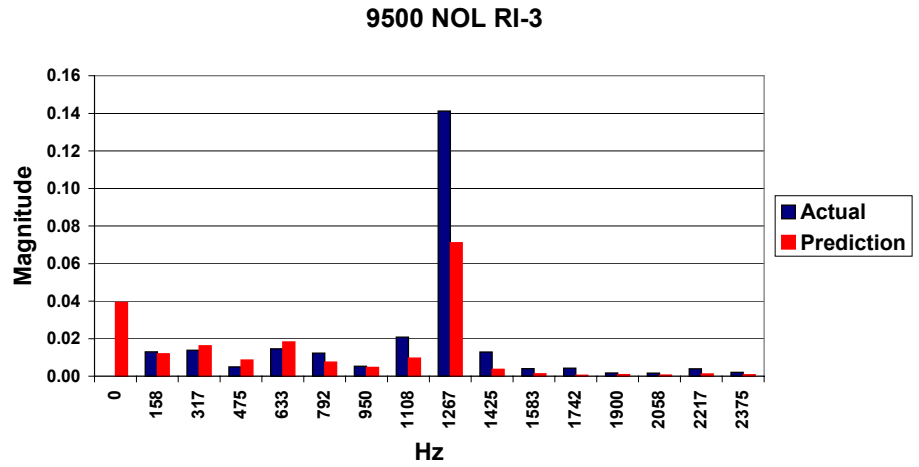

Figure 7-35: Magnitude plot in the frequency domain of the output total pressure prediction shown in Figure 7-34.

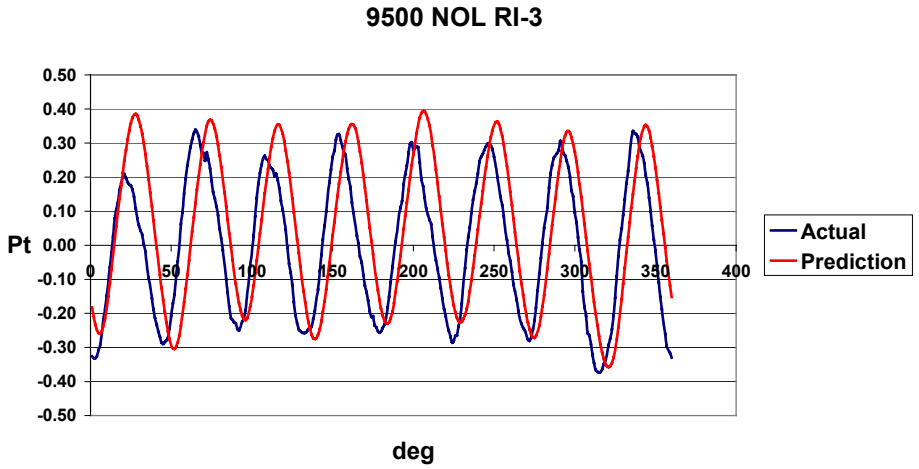

Figure 7-36: Same scenario as in Figure 7-34, only data sets were further normalized with the non-dimensional stage work.

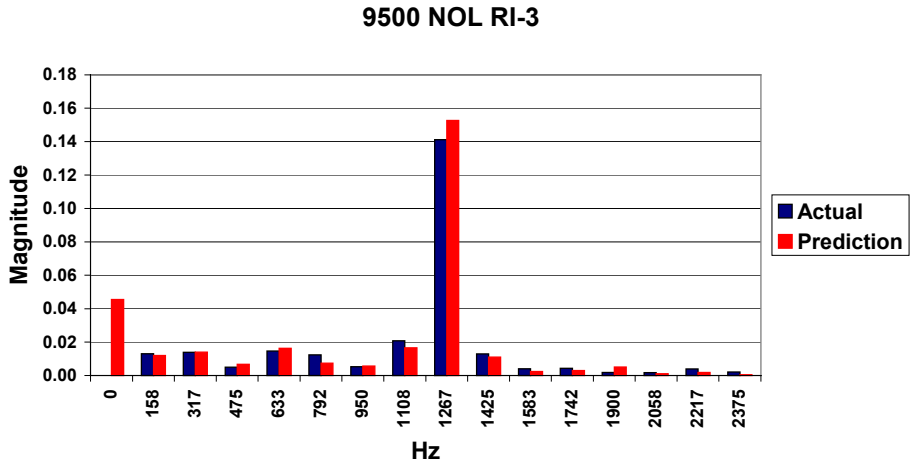

Figure 7-37: Magnitude plot in the frequency domain of the output total pressure prediction shown in Figure 7-36.

As shown in the preceding figures, the prediction accuracy of Model 1 is dramatically improved by taking the stage work into account. When normalizing the data sets, the input and output data sets were divided by the stage work term as defined in Section 6.3 within the code at the same time the data sets were mean-zeroed and non-dimensionalized. This was done to both the input and output waveforms used in the kernel extraction phase and the input waveform used in the prediction phase. The prediction was then multiplied by this work term as well as the non-dimensional term to obtain the final, dimensional, prediction. The improvement results were not as dramatic with Model 3, but the final predictions were still more accurate with work normalization than without. Table 7-3 shows the prediction results numerically of each model and is in the same format as Tables 7-1 and 2. The prediction results of Models 1 and 3 as shown in the table have been work normalized.

	Model 1		Model 2		Model 3	
domain	time	freq	time	freq	time	freq
radial position	std	% error	std	% error	std	% error
RI5 - hub	0.155	-42.3%	0.171	-39.0%	0.201	-90.2%
RI4	0.153	-4.3%	0.227	-4.3%	0.152	-47.9%
RI3	0.176	8.5%	0.167	5.7%	0.241	12.8%
RI2	0.077	3.8%	0.091	3.8%	0.297	25.0%
RI1 - tip	0.289	177.0%	0.253	164.4%	0.429	141.4%

Table 7-3: Prediction results made at 9500NOL after kernel calculation with 9100 and 13200-rpm data.

Despite the fact that the models are evenly split when comparing time domain predictions, Model 2 is the best of the three by a slim margin over Model 1 after considering the frequency domain results. Even though predictions are not very good at the hub and tip, results at the interior points of the blade are a different matter, with less than 6% error of the dominant frequency magnitude as predicted by Models 1 and 2. This low error rate is especially significant since the models were predicting compressor response to an input measured at a rotational speed that was not at all included in kernel calculation.

In order to further assess Volterra modeling, one more test scenario was created. The goal of this test was to see if the Volterra models were capable of predicting distortion transfer at a frequency that was not included in the training data sets. The best option for this type of scenario would be to use data for kernel calculation and prediction that was collected at the same rotational speed, but with two different types of distortions, therefore two different dominant frequencies. One of the distortion types could be used in kernel training and the other in the prediction phase. Unfortunately, none of the rotational speeds included in the data had been exposed to two types of distortions. Two of the speeds, however, were very close to one another, 9100 and 9500rpm. Therefore, the 9100rpm data sets were used for kernel training and compressor response was predicted at 9500rpm. Rather than showing time and frequency domain plots of all the predictions along the span of the blade, Table 7-4 summarizes the standard deviation of the prediction in the time domain and percent error of the magnitude of the dominant frequency harmonic as predicted by Model 2.

	Model 2	
domain	time	freq
radial position	std	% error
RI5 - hub	0.221	-64.2%
RI4	0.229	-0.9%
RI3	0.111	16.3%
RI2	0.882	76.2%
RI1 - tip	0.876	273.6%

Table 7-4: Summary of results obtained with Model 2 when predicting compressor output at 9500NOL after calculating the kernels with 9100rpm data.

The results shown in Table 7-4 indicate that the Volterra model is not able to accurately predict distortion transfer at a frequency not included in the initial kernel calculation phase. While it would be very nice if the predictions obtained in this test were more accurate, it is logical that they are not. This is not the same as predicting at a different rotational speed. Differences in rotational speed are accounted for through the non-dimensional and normalization parameters that are used within the code. There are no such parameters to deal with variations in excitation frequency. Therefore, the frequency of concern must be included in the data sets used for kernel calculation. This is especially challenging when creating distortions with screens, as a screen is designed to contain energy at specific frequencies and have very little energy anywhere else. Because of this, distortion screen selection to generate data sets for kernel training is critical.

All of the models created were coded in Matlab with the final prediction results cut and pasted into Excel. This was done solely due to the ease of data storage and display that Excel offers. The actual output of the Matlab code presented the prediction results in a manner very similar to what has been shown. An example of what the output looked like is shown in Figure 7-38.

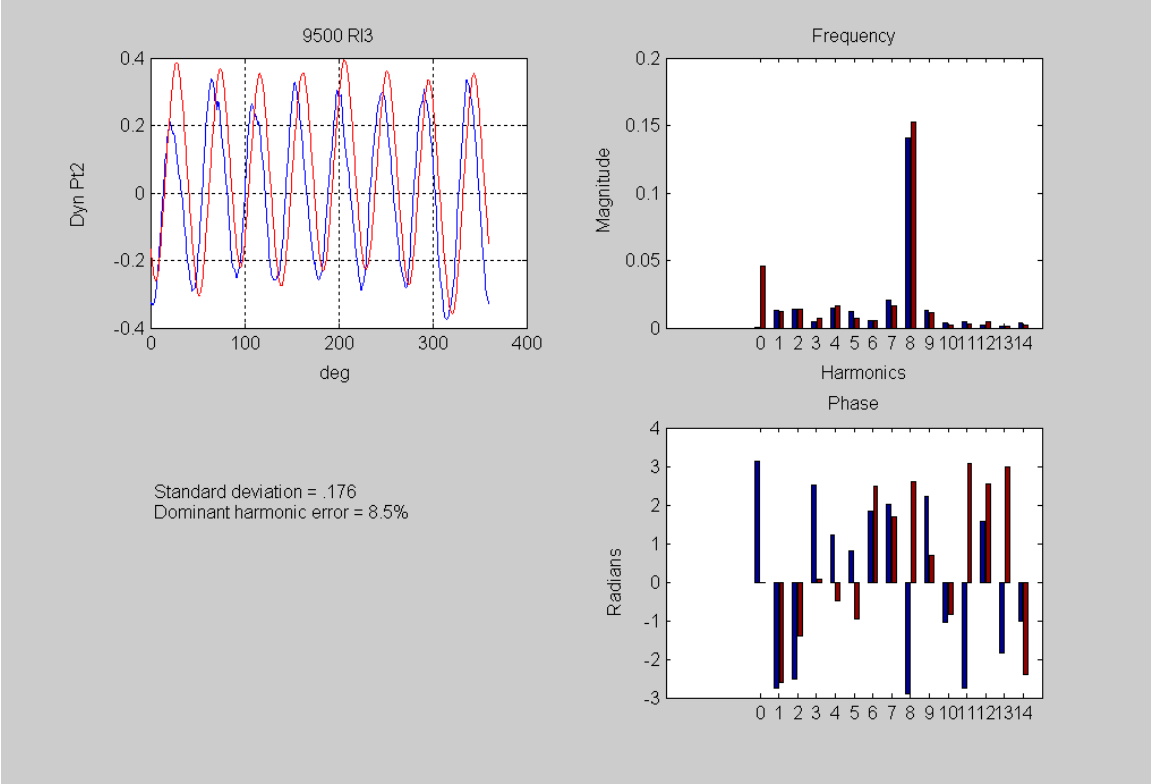

Figure 7-38: Example of the output in Matlab of a prediction of the variation in total pressure at 9500rpm, NOL mass flow, RI3 (mid span). Blue coloring corresponds to the actual experimental data and red the prediction.

7.4 Higher-Order Effects

In this section, the influence that each of the higher order kernels has on the final prediction will be examined. The highest order of operation that any of the three Volterra models is capable of analyzing is third order. One reason for truncating the Volterra series after the third term is the extremely large number of coefficients that would be required to solve for if more terms were used. A fourth order model with 10 basis functions per order would require solving for $10 + 10^2 + 10^3 + 10^4$ (11,110) coefficients. This would more than likely require a great deal more data than is currently available. Besides, the whole idea of using Volterra theory in system modeling is that the system is "weakly" non-linear, meaning that the higher order impulse response functions, or kernels, fall off to zero very rapidly.

Within the code, all of the coefficients that make up each separate kernel are calculated concurrently. However, once the coefficients are known each individual kernel is calculated separately. The first, second, and third-order kernels are then used to separately calculate the first, second, and third-order terms associated with the final response prediction. The final prediction is the summation of these three terms. Figure 7-39 shows each of the three terms plotted separately and the final summed prediction of compressor response at 9500rpm, NOL mass flow. In this particular scenario, the kernels were calculated with the same waveform that is being predicted, part of the same tests that were discussed in the beginning of section 7.2.

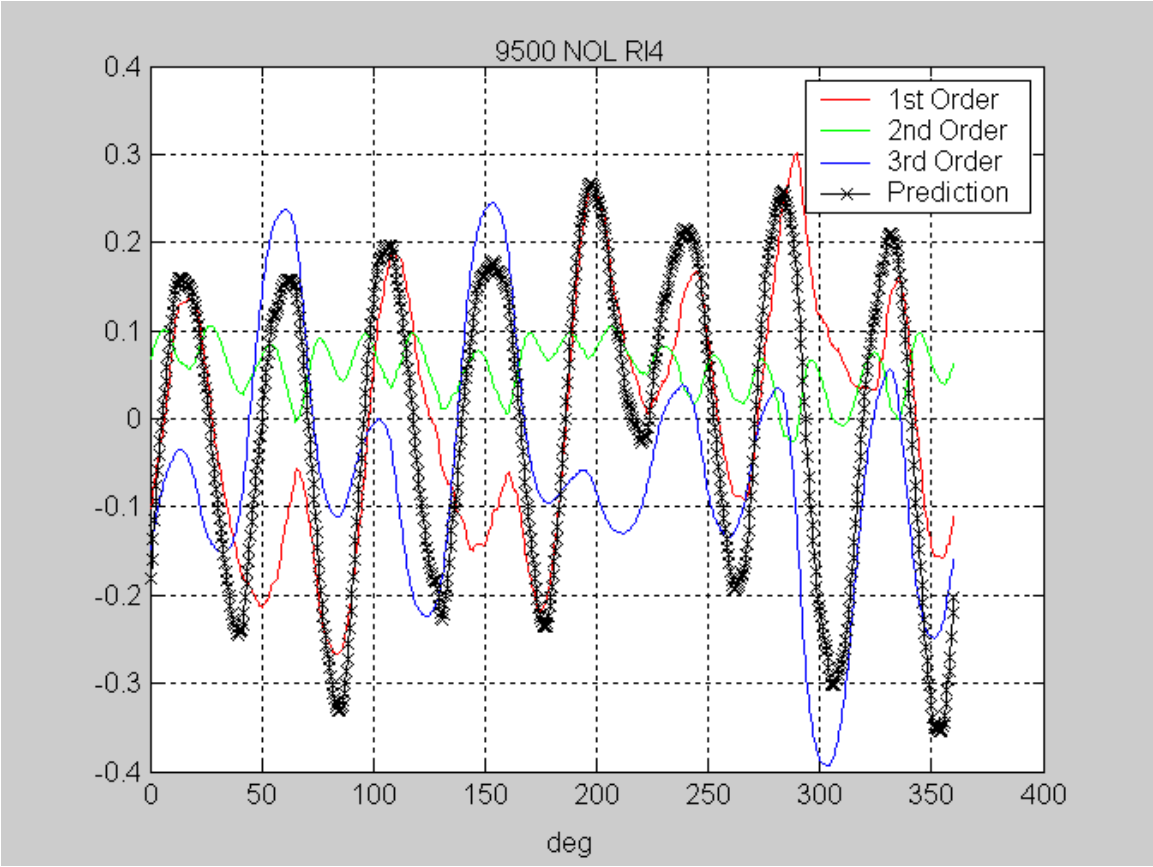

Figure 7-39: Graphical display of the 1st, 2nd, and 3rd order terms and their combination to form the final response prediction using Model 2.

In Figure 7-39 it can be seen that each term or "sub" prediction is independent of the other two and that when added together, their sum forms the final response prediction. The significance of each order of operation is also evident by the influence of each "sub" prediction. This particular case is primarily a 1st and 3rd order operation with the second order effects just wavering between 0 and 0.1.

7.5 Comparison with Previous Models

As discussed in Chapters 2 and 5, my research data has been used for two previous projects at Virginia Tech with the goal of all three being distortion transfer prediction in an axial compressor. Matt Small [2001] used a "tuned" frequency response function (FRF) modeling method and Jon Luedke [2001] created the first Volterra series models, prompting the interest in that same modeling method for my work. An advantage to all of us using the same data is that direct comparison of modeling methods/results is easily accomplished. As Luedke compared his results to Small at the end of his research, I will also compare my results with those of Luedke and Small in this section.

In Small's, Luedke's, and my research, the models all initially "learn" about the system they are representing using experimental data (the kernel training phase in my work). In Table 7-5, prediction results of the tuned FRF [Small, 2001], Volt 33 [Luedke, 2001], Volt 33 Rad [Luedke, 2001], and Model 2 are compared with the percent error of the dominant frequency magnitude as the means of comparison. The operating condition that is being predicted by all three is 9500PE. Model 2 and the FRF model both used all data sets at 9100 and 13200rpm in the initial training phase, while the Volt 33 model used the following: 9100PE, 9100WOD, 9500NOL, and 9500NS. Even though the mass flow is different, it is worth noting that the Volt 33 models in this particular case are predicting compressor response at a rotational speed that is also used in

model training. Model 2 and Volt 33 Rad are streamline specific, the other two use data and make predictions along the entire span of the blade. Model 2 and Volt 33 Rad must be run five separate times in order to predict compressor response along the entire span of the blade.

radial position	FRF	Volt 33	Volt 33 Rad	Model 2
RI5 - hub	-52.7%	-87.9%	**-37.2%**	-39.0%
RI4	-40.1%	-52.4%	-40.6%	**-4.3%**
RI3	**-13.6%**	-34.2%	**-8.7%**	**5.7%**
RI2	-47.6%	**-12.0%**	-25.3%	**3.8%**
RI1 - tip	**-7.6%**	27.6%	161.0%	164.4%
Avg |error|	**32.3%**	42.8%	54.6%	43.4%
w/o hub & tip	33.8%	32.9%	24.9%	**4.6%**

Table 7-5: Comparison of Model 2 with previous modeling methods. The percent error of the dominant frequency magnitude is shown.

In Table 7-5, all predictions at individual radial positions that are within the acceptable error rate of 20% are highlighted in red and the best, or lowest, average percent error is highlighted in blue. Model 2 is more consistently accurate at the interior radial positions on the blade. The tuned FRF method has the lowest average error, although it is still greater than the 20% cut-off with two of its individual predictions within that same cut-off. The final average error was calculated by only considering prediction performance at the three interior blade positions. Model 2 has the best results in this category with less than 5% average error at the three interior points along the blade.

8 Summary and Conclusions

The transfer of circumferential total pressure distortion in a transonic axial compressor has been investigated. A semi-empirical modeling method has been developed, using nonlinear Volterra theory, in order to predict the transfer of circumferential total pressure distortion. This modeling method is based on calculating, or extracting, the 1st, 2nd, and 3rd order impulse response functions (kernels) from experimental data. It has been shown in this study that modeling methods using nonlinear Volterra theory are capable of predicting the transfer of total pressure distortion in axial compressors and fans – with limitations.

Properly identifying the Volterra kernels is essential; once the kernels of a system are known, the output for any arbitrary input can be accurately calculated. The solution method presented is based on expanding the unknown kernels into basis function space. The basis functions are multiplied by the input to the system, forming what is known as the motion matrix. Now a matrix problem has been formed where the system output is equal to the motion matrix multiplied by some unknown coefficients. The coefficients are solved for in a least squares sense using a regularization technique called singular value decomposition.

In the work presented, three different models were created using various forms of higher order basis functions and/or integration methods. The best of the three models was able to predict the dominant frequency magnitude associated with total pressure distortion transfer to within a 5% error rate along the interior points of the blade. These predictions were done at operating conditions that the model had not previously seen before, which is especially significant. Also of importance is the fact that the largest magnitude associated with distortion transfer is at the mid-span of the blade, where the prediction is the most accurate. Prediction results at the hub and tip, however, proved to be

unreliable. This can be explained by referring to the distortion magnitude transfer plot, Figure 5-14, as shown below.

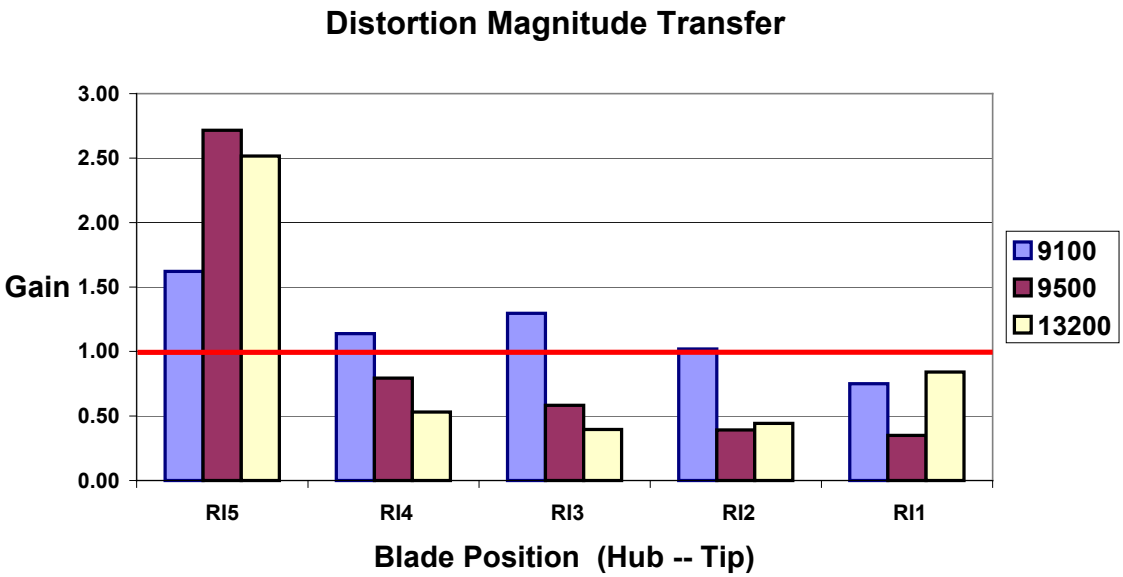

Figure 5-14. First stage distortion magnitude gain as a function of blade position and rotational speed.

In the above plot, distortion transfer at the interior points of the blade (RI4, 3, and 2) follows a trend. The distortion transfer at 9100 is larger than that at 9500, which is larger than the distortion transfer at 13200. This is most certainly true at RI4 and 3 and very close to being true at RI2. This trend, however, is not followed at the hub and tip of the blade, with the distortion transfer either being much larger (hub) or smaller (tip) at 9500rpm than at the other two rotational speeds. Data at two rotational speeds was used in kernel calculation, 9100 and 13200rpm. System response was then predicted at 9500rpm, which at these two points on the blade is very atypical when compared to distortion transfer at the other two speeds. This explains why the model is not capable of accurately predicting distortion transfer at the hub and tip. If Figure 5-14 did not have any information pertaining to distortion transfer at 9500rpm, one would expect the gain associated with 9500rpm to fall somewhere between the gain associated with 9100 and 13200rpm. That is exactly what happens on the interior points of the blade, consequently resulting in good model

performance. At the other two points on the blade this does not happen, resulting in an under-prediction at the hub and an over-prediction at the tip.

The frequency range of interest must be included in the training data sets. In other words, if one is interested in the transfer of a 4 per rev distortion and that coincides with a frequency of 500hz; the data sets used for kernel training must have some frequency content at 500hz. Otherwise, the model will have no idea how the system responds to energy contained at that particular frequency and will, subsequently, be unable to accurately predict system response. As discussed in Chapter 7, this presents a challenge when testing with distortion screens as screens are designed to have significant energy at a specific frequency and very little anywhere else.

Finally, the speed of prediction should be between the speeds used for kernel calculation. The data sets used for kernel training should be considered as the limits of known performance with all predictions done within these limits. Although this is not a linear process, an analogy can be made to interpolation as opposed to extrapolation.

The goals of this particular study were initially discussed in Chapter 1 and will be re-iterated here: (1) create a semi-empirical model capable of accurately predicting total pressure distortion transfer in fans and compressors, (2) extract the frequency components of non-uniform pressure profiles to enable the prediction of blade response, and (3) use the combined pressure and blade response predictions as a tool to identify potential HCF problem areas. The first goal has definitely been attained with the limitations as discussed in this chapter. The frequency components associated with total pressure distortion have also been extracted, although this research did not include actually applying these frequencies to a blade model in order to predict blade response/stress. This work is part of another study at Virginia Tech being carried out by Rob Wallace for his Masters thesis. Part of the third goal has also been attained in that this study has shown that total pressure distortion is definitely transferred through

an axial compressor stage and that the accurate identification of the dominant frequency magnitude associated with that distortion can be accomplished at mid-span conditions via Volterra modeling.

9 Recommendations

This study has shown that Volterra modeling is capable of accurately predicting total pressure distortion transfer in axial compressors and fans within specific limitations at mid-span conditions. While promising, this study was done with limited data and therefore, more research and testing of this modeling method should be done before even considering its use in industry.

The next step in testing should be the use of a total pressure distortion generator that produces wide-band frequency content. Use a distortion generator that has significant frequency content within the first 8 – 12 harmonics to generate data sets when testing a compressor. Collect data at two rotational speeds, the lower and upper limit in the range of operation, using this distortion generator. Use these data sets in the kernel-training phase of the model. After the kernels have been calculated, predict compressor response using data that was collected using a specific per rev distortion screen, such as the 3 or 8 per rev used in this study. If the prediction results are similar to those obtained in this study, the usefulness of Volterra modeling will be even greater.

The models created in this study are semi-empirical models, meaning that rather than being purely mathematical, the physics of the flow was considered and incorporated as non-dimensional parameters. Quantities such as velocity and pressure were non-dimensionalized by relevant physical parameters, such as U^2 and dynamic pressure, $\frac{1}{2}\rho W^2$, in order to bring some physics and fluid dynamics into the math model. Thought should be given to incorporating even more physics-based parameters into the models to better account for phenomena associated with transonic flow, such as shocks.

Also, this study revealed the strong radial dependence of total pressure distortion transfer. As a result of this finding, the models created were streamline specific, calculating the impulse response functions as functions of

time for one point along the span of the blade. The next step to better understand the way in which a compressor stage responds to an impulse is to develop a model that calculates the impulse response functions as a function of time and radial position. In other words, adding a radial dimension to the impulse response functions so that they could be used to calculate the system response along the entire span of the blade, rather than at just one point, for example $y(r,t) = \int h(r,t)x(t-\tau)d\tau + \ldots$ where, perhaps $h(r,t) = f(r)h(t)$.

More testing is needed to further understand compressor response at the hub and tip of the blade. What was observed in this study could have been a result of boundary layer or secondary flow effects. Another thought is that the model at the tip should account for the tip gap between the end of the blade and the outer wall of the compressor. Whatever the cause for the unusual gain associated with distortion transfer at these two blade locations, more data is needed before it can be fully understood.

10 Postscript

While it has been shown that Volterra modeling is capable of predicting total pressure distortion transfer, the dominant driver in HCF, the question remains as to the usefulness of this type of modeling. Does Volterra modeling actually have an application in industry and, if so, what is it? First, before that question can be confidently answered in the positive or negative, more testing must be done utilizing data collected under many different operating conditions. The data used for this study, besides being very informative and useful in the development of this model, is limited to 12 different operating conditions that, after examination, can essentially be reduced down to 3 data sets (one for each rotational speed) at each radial location on the blade. This is not enough information to truly test any modeling method. All of this being said, Volterra modeling has shown definite potential in its ability to accurately predict the magnitude associated with the dominant frequency of a total pressure distortion.

The usefulness of Volterra modeling is not as a replacement for CFD analysis, as CFD gives a much more detailed breakdown of the entire flow field. The potential of Volterra modeling is as a quick assessment tool to ascertain if a more detailed analysis is even necessary. Typically, mid-span conditions are used for quick assessment, which is where the Volterra model is the most accurate. In addition, the largest frequency magnitude associated with total pressure distortion is at the mid-span of the blade, also good for Volterra modeling. An analysis of an axial compressor stage could be done with a Volterra model using a typical desktop computer in as little as 30 minutes to an hour, depending on the number of data sets used in kernel training. Also, after the kernels are calculated defining a certain compressor stage, they could be saved and any future predictions of total pressure distortion transfer could be done just utilizing the prediction phase of the model, further reducing processing

time. The prediction phase of the Volterra model runs in as little as 10 minutes on a 1 Ghz desktop computer. If the Volterra model predicts a substantial transfer of the frequency of interest, then a more detailed analysis could be done with CFD. However, if the energy contained at the frequency of interest were small, further analysis would not be necessary producing a potential savings of days to weeks in analysis time.

11 Works Cited

Adamczyk, J.J., "Unsteady Fluid Dynamic Response of an Isolated Rotor with Distorted Inflow," AIAA Paper No. 74-49.

Barr, L.C., "The Unsteady Response of an Axial Flow Turbomachinery Rotor to Inlet Flow Distortions," The Pennsylvania State University Applied Research Laboratory, TM 78-253.

Bendat, J.S., *Nonlinear Systems Techniques and Applications, John Wiley & Sons*, Inc., New York, 1998.

Boyd, S., Tang, Y.S., Chua, L.O., "Measuring Volterra Kernels", *IEEE Transactions on Circuits and Systems*, vol. 30, no. 8, Aug. 1983, pp. 571-577.

Boyer, K.E., "An Improved Streamline Curvature Approach for Off-Design Analysis of Transonic Compression Systems," Ph.D. Dissertation, Virginia Polytechnic Institute and State University (Blacksburg, VA, 2001).

Campbell, Wilfred, "Protection of Turbine Disk Wheels from Axial Vibration," *Proceedings of the Cleveland Spring Meeting*, ASME, May 1924.

Colpin, J., Kool, P., "Experimental Study of an Axial Compressor Rotor Transfer Function with Non-Uniform Inlet Flow." ASME Paper No. 78-GT-69, Gas Turbine Conference and Products Show, April 9-13, 1978, London, England.

Cousins, W.T., private communication, 2002.

Cousins, W.T., O'Brien, W.F., "Axial-Flow Compressor Stage Post-Stall Analysis," AIAA Paper No. 85-1349, 1985.

Danforth, C.D., "Distortion-Induced Vibration in Fan and Compressor Blading," *Journal of Aircraft*, vol. 12, no. 4, pp. 216-225, April 1975.

Datko Jr., J.T., O'Hara, J.A., "The Aeromechanical Response of an Advanced Transonic Compressor to Inlet Distortion," ASME 87-GT-189, 1987.

Davis, M., Hale, A., Beale, D., "An Argument for Enhancement of the Current Inlet Distortion Ground Test Practice for Aircraft Gas Turbine Engines," ASME 2000-GT-0505, 2000.

Greitzer, E.M., "Upstream Attenuation and Quasi-Steady Rotor Lift Fluctuations in Asymmetric Flows in Axial Compresors," ASME 73-GT-30, 1973.

Greitzer, E.M., Tan, C.S., Wisler, D.C., Adamczyk, J.J., Stazisar, A.J., "Unsteady Flows in Turbomachines: Where's the Beef?," *ASME Unsteady Flows in Aeropropulsions*, 1994, pp. 1-11.

Hah, C., Rabe, D.C., Sullivan, T.J., Wadia, A.R., "Effects of Inlet Distortion on the Flow Field in a Transonic Compressor Rotor," *ASME Journal of Turbomachinery*, Vol. 120, April 1998, pp. 233-246.

Hale, A.A., O'Brien, W.F., *A 3-D Turbine Engine Analysis Compressor Code (TEACC) for Steady-State Inlet Distortion.* Ph.D. Dissertation, Virginia Polytechnic Institute and State University (Blacksburg, Virginia, 1996.).

Hamed, A., Numbers, K., "Inlet Distortion Considerations for High Cycle Fatigue in Gas Turbine Engine," AIAA Paper 97-3364, AIAA 33rd Joint Propulsion Conference and Exhibit, July 6-9, 1997, Seattle, WA.

Hansen, P.C., *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM, Philadelphia, 1998.

Henderson, R.E., Shen, I.C., "The Influence of Unsteady Rotor Response on a Distorted Flow Field," ASME Paper No. 81-GT-185, International Gas Turbine Conference and Products Show, March 9-12, 1981, Houston, Texas.

Jenkins, J., Tromp, J., "A Volterra Kernel Identification Scheme Applied to Aerodynamic Reactions," AIAA 90-2803, August 1990.

Lecht, M., "Improvement of the Parallel Compressor Model by Consideration of Unsteady Blade Aerodynamics," AGARD CP-400, 1986.

Lecht, M., Weyer, H. B., "On the Unsteady Aerodynamic Rotor Blade Loading in a Transonic Axial Flow Compressor with Unsteady State Inlet Distortion," IUTAM Symposium on Aeroelasticity in Turbomachines, October 18-23, 1976, Paris.

Luedke, J., private communication, 2002.

Luedke, J., "Use of Nonlinear Volterra Theory in Predicting the Propagation of Non-uniform Flow Through an Axial Compressor," M.S. Thesis, Virginia Polytechnic Institute and State University (Blacksburg, Virginia, 2001.).

Longley, J.P., Greitzer, E.M., "Inlet Distortion Effects in Aircraft Propulsion System Integration," *Steady and Transient Performance Prediction of Gas Turbine Engines*, AGARD LS-183, May 1992.

Manwaring, S.R., Fleeter, S., "Inlet Distortion Generated Periodic Aerodynamic Rotor Response," ASME 89-GT-299, 1989.

Manwaring, S.R., Fleeter, S., "Forcing Function Effects on Rotor Periodic Aerodynamic Response," *Transactions of the ASME*, vol. 113, April 1991, pp. 312-319.

Manwaring, S.R., Rabe, D.C., Lorence, C.B., Wadia, A.R., "Inlet Distortion Generated Forced Response of a Low Aspect-Ratio Transonic Fan," *ASME Journal of Turbomachinery*, vol. 119, October 1997, pp. 665-676.

Mazzawy, R.S., "Multiple Segment Parallel Compressor Model for Circumferential Flow Distortion," *ASME Journal of Engineering for Power*, April, 1977.

Melick, H.C., "Analysis of Inlet Flow Distortion and Turbulence Effects on Compressor Stability," NASA CR 114577, 1973.

Morrow, P., "Detailed Test Plan for the Augmented Damping of Low Aspect Ratio Fans (ADLARF)," USAF Technology Branch Turbine Engine Division, WPAFB, March 1993.

Nagano, S., Takata, H., "Nonlinear Analysis of Rotating Stall," Institute of Space and Aeronautical Science, University of Tokyo, Report No. 449, 1970.

Peacock, R.E., Overli, J., "Dynamic Internal Flows in Compressors with Pressure Maldistributed Inlet Conditions," AGARD CP-177, 46th P.E.P. Conference of AGARD, 1975, Monterey, CA.

Pearson, H., McKenzie, A., "Wakes in Axial Compressors." *Journal of the Royal Aeronautical Society*, vol. 63, July 1959.

Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T., *Numerical Recipes in C – The Art of Scientific Computing*, Cambridge University Press, Cambridge, 1988.

Rabe, D., Bolcs, A., and Russler, P., "Influence of Inlet Distortion on Transonic Compressor Blade Loading," AIAA 95-2461, Presented at the 31st Joint Propulsion Conference and Exhibit, July 1995, San Diego, CA.

Rabe, D., Williams, C., and Hah, C., "Inlet Flow Distortion and Unsteady Blade Response in a Transonic Axial Compressor Rotor," ISABE 99-7297, 1999.

Reid, C., "The Response of Axial Flow Compressors to Intake Flow Distortion." ASME Paper No. 69-GT-29, 1969.

Reisenthel, P.H., "Application of Nonlinear Indicial Modeling to the Prediction of a Dynamically Stalling Wing," AIAA 96-2493, June 1996.

Reisenthel, P.H., "Development of a Nonlinear Indicial Model for Maneuvering Fighter Aircraft," AIAA 96-0896, 1996.

Reisenthel, P.H., "A Nonlinear Volterra Kernel Identification System for Aeroelastic Applications," NEAR TR 547, June 1999.

Reisenthel, P.H., "Prediction of Unsteady Aerodynamic Forces Via Nonlinear Kernel Identification," Langley International Forum on Aeroelasticity and Structural Dynamics, Williamsburg, VA, June 1999.

Reisenthel, P.H. and Bettencourt, M.T., "Data-Based Aerodynamic Modeling Using Nonlinear Indicial Theory," AIAA 99-0763, Jan. 1999.

Reisenthel, P.H. and Bettencourt, M.T., "Extraction of Nonlinear Indicial and Critical State Responses from Experimental Data," AIAA 99-0761, Jan. 1999.

Roberts, F., Plourde, G.A., Smakula, F. "Insights into Axial compressor Response to Distortion," AIAA Paper No. 68-565, 1968.

Rugh, W.J., *Nonlinear System Theory -- The Volterra/Wiener Approach*, The Johns Hopkins University Press, Baltimore, 1981.

Sexton, M.R., O'Brien, W.F., "A Model for Dynamic Loss Response in Axial-Flow Compressors," ASME 81-GT-154, 1981.

Silva, W.A., "Identification of Linear and Nonlinear Aerodynamic Impulse Responses Using Digital Filter Techniques," AIAA 97-3712, Aug. 1997.

Silva, W.A., "Extension of a Nonlinear Systems Theory to General-Frequency Unsteady Transonic Aerodynamic Responses," AIAA 93-1590, Apr. 1993.

Small, M., "Improved Methods for Predicting the Effects of Inlet Flow Distortion on the Performance of Axial Flow Compressor," M.S. Thesis, Virginia Polytechnic Institute and State University (Blacksburg, Virginia, 2001.).

Tromp, J.C., and Jenkins, J.E., "A Volterra Kernel Identification Scheme Applied to Aerodynamic Reactions," AIAA 90-2803, Aug. 1990.

Volterra, V., *Theory of Functionals and of Integral and Integro-Differential Equations*, Blackie & Sons Ltd., London, 1930.

Williams, C., "Turbine Engine Research Center Inlet Distortion Measurement," 4th National Turbine Engine High Cycle Fatigue Conference, 1999, Monterey, CA.

Wu, C.H., "A General Through-Flow Theory of Fluid Flow with Subsonic or Supersonic Velocity in Turbomachines or Arbitrary Hub and Casing Shapes," NACA TN 2302, 1951.

12 Appendix

12.1 Model 1

```
clear all;
close all;

%%Number of Data Sets
nds = 1;
%%Number of Data Sets Used to Train Kernel
ndsk = 7;
%%Number of Linear Basis Functions
lbf = 10;
%%Number of 2nd Order Basis Functions
qbf = 8;
%%Number of 3rd Order Basis Functions
cbf = 6;
%%Radial Immersion Identifier: 1 = r5, 2 = r4, 3 = r3, 4 = r2, 5 = r1 :: (r5 = hub, r1
= tip) ::
r = 1;

%%Stag Density at Inlet
rhos = .0627;  %9100
rhom = .0614;  %9500
rhof = .0426;  %13200

%%Time ND factors  (chord length/W1)
%9100 Tip ==> Hub
tnd(5) = .000597;  tnd(4) = .000636;  tnd(3) = .000691;  tnd(2) = .000775;  tnd(1) =
.000980;

%%W1 values for Each Rotational Speed
%9100 Tip==>Hub
WS(5) = 1129;  WS(4) = 1033;  WS(3) = 922;  WS(2) = 786;  WS(1) = 590;
%9500 Tip==>Hub
WM(5) = 1185;  WM(4) = 1084;  WM(3) = 969;  WM(2) = 827;  WM(1) = 623;
%13200 Tip==>Hub
WF(5) = 1736;  WF(4) = 1601;  WF(3) = 1439;  WF(2) = 1234;  WF(1) = 944;

%Work
%9100 Tip==>Hub
```

```
wks(5) = 17907.36;  wks(4) = 15996.65;  wks(3) = 16497.95;  wks(2) = 16656.24;
wks(1) = 18065.16;
%9500 Tip==>Hub
wkm(5) = 19860.62;  wkm(4) = 17793.54;  wkm(3) = 18220.27;  wkm(2) = 18431.00;
wkm(1) = 19724.34;
%13200 Tip==>Hub
wkf(5) = 32651.21;  wkf(4) = 34500.59;  wkf(3) = 36231.22;  wkf(2) = 36039.70;
wkf(1) = 37910.55;

%%Time Vector
load ..\datajfr\textdata\time9100.txt

%%Non-Dimensional Time
for i = 1:720
   t(i,1) = [[time9100(i,1)]/tnd(r)];
end
for i = 721:1440
   t(i,1) = t(i-720,1)+t(720,1);
end

deg = .5:.5:360;

%%Input and Response Vectors
load ..\datajfr\textdata\hall9100in.txt
load ..\datajfr\textdata\hall9100out.txt
load ..\datajfr\textdata\hall9500in.txt
load ..\datajfr\textdata\hall9500out.txt
load ..\datajfr\textdata\hall13200in.txt
load ..\datajfr\textdata\hall13200out.txt

%%Training data - 9100 : 13200nol,ns/pe
for p = 1:720
   in(p,1) = [hall9100in(p,r)];
   out(p,1) = [hall9100out(p,r)];
   in(p,2) = [hall9100in(p,5+r)];
   out(p,2) = [hall9100out(p,5+r)];
   in(p,3) = [hall9100in(p,10+r)];
   out(p,3) = [hall9100out(p,10+r)];
   in(p,4) = [hall9100in(p,15+r)];
   out(p,4) = [hall9100out(p,15+r)];
%     in(p,1) = [hall9500in(p,r)];
%     out(p,1) = [hall9500out(p,r)];
%     in(p,2) = [hall9500in(p,5+r)];
```

```
%     out(p,2) = [hall9500out(p,5+r)];
%     in(p,3) = [hall9500in(p,10+r)];
%     out(p,3) = [hall9500out(p,10+r)];
%     in(p,4) = [hall9500in(p,15+r)];
%     out(p,4) = [hall9500out(p,15+r)];
    in(p,5) = [hall13200in(p,r)];
    out(p,5) = [hall13200out(p,r)];
    in(p,6) = [hall13200in(p,5+r)];
    out(p,6) = [hall13200out(p,5+r)];
    in(p,7) = [hall13200in(p,10+r)];
    out(p,7) = [hall13200out(p,10+r)];
end

%%Prediction data - 9500 : 13200wod
for p = 1:720
%     inp(p,1) = [hall9100in(p,r)];
%     outp(p,1) = [hall9100out(p,r)];
%     inp(p,2) = [hall9100in(p,5+r)];
%     outp(p,2) = [hall9100out(p,5+r)];
%     inp(p,3) = [hall9100in(p,10+r)];
%     outp(p,3) = [hall9100out(p,10+r)];
%     inp(p,4) = [hall9100in(p,15+r)];
%     outp(p,4) = [hall9100out(p,15+r)];
    inp(p,1) = [hall9500in(p,r)];
    outp(p,1) = [hall9500out(p,r)];
%     inp(p,2) = [hall9500in(p,5+r)];
%     outp(p,2) = [hall9500out(p,5+r)];
%     inp(p,3) = [hall9500in(p,10+r)];
%     outp(p,3) = [hall9500out(p,10+r)];
%     inp(p,4) = [hall9500in(p,15+r)];
%     outp(p,4) = [hall9500out(p,15+r)];
%     inp(p,1) = [hall13200in(p,r)];
%     outp(p,1) = [hall13200out(p,r)];
%     inp(p,2) = [hall13200in(p,5+r)];
%     outp(p,2) = [hall13200out(p,5+r)];
%     inp(p,3) = [hall13200in(p,10+r)];
%     outp(p,3) = [hall13200out(p,10+r)];
end

%% Mean Pressure Calculation
for n = 1:ndsk
    mn(n) = mean(in(:,n));
    mo(n) = mean(out(:,n));
```

```
end
for n = 1:nds
   mnp(n) = mean(inp(:,n));
   mop(n) = mean(outp(:,n));
end

%% Non-Dimensional Pressure Calculation (1/2*rho*W1^2)
for n = 1:ndsk
   if n < 5
      P(n) =  .5 * rhos * WS(r)^2 / 4636.8;
   else
      P(n) = .5 * rhof * WF(r)^2 / 4636.8;
   end
end
for n = 1:nds
   Pp(n) = .5 * rhom * WM(r)^2 / 4636.8;
end

%% Dynamic Response ND'd by 1/2*rho*w^2 and scaled with work/w1^2
for n=1:ndsk
   if n < 5
      x(:,n)=((in(:,n) - mn(n))/P(n)) / ((wks(r)*32.2)/WS(r)^2);
      dynout(:,n) = ((out(:,n) - mo(n))/P(n)) / ((wks(r)*32.2)/WS(r)^2);
   else
      x(:,n)=((in(:,n) - mn(n))/P(n)) / ((wkf(r)*32.2)/WF(r)^2);
      dynout(:,n) = ((out(:,n) - mo(n))/P(n)) / ((wkf(r)*32.2)/WF(r)^2);
   end
end
for n=1:nds
   xp(:,n)=((inp(:,n) - mnp(n))/Pp(n)) / ((wkm(r)*32.2)/WM(r)^2);
   dynoutp(:,n) = (outp(:,n) - mop(n));
end

%% Pad input to 2 revolutions
for n = 1:ndsk
   for i = 721:1440
      x(i,n) = x(i-720,n);
      dynout(i,n) = dynout(i-720,n);
   end
end
for n = 1:nds
   for i = 721:1440
      xp(i,n) = xp(i-720,n);
```

```
   end
end

%% Linear Basis Functions & Motion Matrix
for q = 1:ndsk
   for i = 1:1440
      for j = 1:lbf
         z(i,j) = exp(-j .* t(i,1));
      end
   end
   aaa = conv2(z, x(:,q));
   for i = 1:1440
      aa(i,:) = aaa(i,:);
   end
   a(:,:,q) = aa;
end

%%2nd Order Time Constants & Basis Function Formulation
for i = 1:1440
   for j = 1:qbf              %% j = Tau1
      for k = 1:qbf           %% k = Tau2
         if i < 2
            mu(i,j,k) = exp(-j * t(i,1));
         else
            mu(i,j,k) = exp(-j*(t(i,1)-t(i-1,1)) - k*t(i-1,1));
         end
      end
   end
end

%%2nd Order Motion Matrix
for q = 1:ndsk
   bbb = convn(mu, x(:,q).^2);
   for i = 1:1440
      bb(i,:,:) = bbb(i,:,:);
   end
   ii = 0;
   for j = 1:qbf
      for k = 1:qbf
         ii = ii+1;
         bi(:,ii,q) = bb(:,j,k);
      end
   end
```

```
end

%%3rd Order Time Constants & Basis Function Formulation
   for i = 1:1440
      for j = 1:cbf              %% j = 1/Tau1
         for k = 1:cbf           %% k = 1/Tau2
            for L = 1:cbf        %% L = 1/Tau3
               if i < 3
                  if i < 2
                     mu3(i,j,k,L) = exp(-j * t(i,1));
                  else
                     mu3(i,j,k,L) = exp(-j*(t(i,1)-t(i-1,1)) - k*t(i-1,1));
                  end
               else
                  mu3(i,j,k,L) = exp(-j*(t(i,1)-t(i-1,1)) - k*(t(i-1,1)-t(i-2,1)) - L*t(i-2,1));
               end
            end
         end
      end
   end

%%3rd Order Motion Matrix
for q = 1:ndsk
   ddd = convn(mu3, x(:,q).^3);
   for i = 1:1440
      dd(i,:,:,:) = ddd(i,:,:,:);
   end
   ii = 0;
   for j = 1:cbf
      for k = 1:cbf
         for L = 1:cbf
            ii = ii+1;
            di(:,ii,q) = dd(:,j,k,L);
         end
      end
   end
end

%%Solution Matrix
p = zeros(1440,lbf+qbf^2+cbf^3,ndsk);
for q=1:ndsk
   for n=1:lbf
      p(:,n,q)=a(:,n,q);
```

```
   end
   for j2 = 1:qbf^2
      p(:,lbf+j2,q) = bi(:,j2,q);
   end
   for j3 = 1:cbf^3
      p(:,lbf+qbf^2+j3,q) = di(:,j3,q);
   end
end

%SVD / Basis Coefficients
for q = 1:ndsk
   ms(:,:,q) = transpose(p(:,:,q))*p(:,:,q);
   bs(:,:,q) = transpose(p(:,:,q))*(dynout(:,q));
end
m = sum(ms,3);
b = sum(bs,3);
cnd = 1;
for jj = 4:12
   if cnd < 4.1e-4
      jj = jj - 1
      [u,s,v] = svds(m,jj);
      si = inv(s);
      break
   end
   [u,s,v] = svds(m,jj);
   si = inv(s);
   cnd = rcond(si)
end
ip = transpose(u)*b;
c = v*si*ip;

%%Linear Kernel H1
for n = 1:lbf
  c1(n,1) = c(n,1);
end
for j = 1:lbf
   hc1(:,j) = c1(j,1) * z(:,j);
end
h1(:,:) = sum(hc1,2);

%%2D Kernel H2
for n=1:qbf^2
  bc2(n,1)=c(n+lbf,1);
```

```
end
for j = 1:qbf
   acc = (j-1)*qbf;
   for k = 1:qbf
      c2(j,k) = bc2(acc+k,1);
   end
end
for j = 1:qbf
   for k = 1:qbf
      hc2(:,j,k) = c2(j,k) * mu(:,j,k);
   end
end
hs2(:,:,:) = sum(hc2,2);
h2(:,:) = sum(hs2,3);

%3D Kernel H3
for n = 1:cbf^3
   dc3(n,1) = c(n+lbf+qbf^2,1);
end
for j = 1:cbf
   for k = 1:cbf
      acc2 = (j-1)*cbf^2 + (k-1)*cbf;
      for L = 1:cbf
         c3(j,k,L) = dc3(acc2+L,1);
      end
   end
end
for j = 1:cbf
   for k = 1:cbf
      for L = 1:cbf
         hc3(:,j,k,L) = c3(j,k,L) * mu3(:,j,k,L);
      end
   end
end
hst3(:,:,:,:) = sum(hc3,2);
ht3(:,:,:,:) = sum(hst3,3);
h3(:,:) = sum(ht3,4);

%% Linear Response
for n = 1:nds
   fpp1 = conv2(hc1, xp(:,n));
   for i = 1:1440
      fp1(i,:) = fpp1(i,:);
```

```
   end
   prr1(:,:,n) = fp1;
   for i = 1:1440
      pr1(i,n) = sum(prr1(i,:,n));
   end
end

%%Quadratic Response
for n = 1:nds
   fpp2 = convn(hc2, xp(:,n).^2);
   for i = 1:1440
      fp2(i,:,:) = fpp2(i,:,:);
   end
   ii = 0;
   for j = 1:qbf
      for k = 1:qbf
         ii = ii+1;
         prr2(:,ii,n) = fp2(:,j,k);
      end
   end
   for i = 1:1440
      pr2(i,n) = sum(prr2(i,:,n));
   end
end

%%Cubic Response
for n = 1:nds
   fpp3 = convn(hc3, xp(:,n).^3);
   for i = 1:1440
      fp3(i,:,:,:) = fpp3(i,:,:,:);
   end
   ii = 0;
   for j = 1:cbf
      for k = 1:cbf
         for L = 1:cbf
            ii = ii+1;
            prr3(:,ii,n) = fp3(:,j,k,L);
         end
      end
   end
   for i = 1:1440
      pr3(i,n) = sum(prr3(i,:,n));
   end
```

```
end


%%Prediction
for q = 1:nds
   maxo(q) = max(dynoutp(:,q));
   mino(q) = min(dynoutp(:,q));
   for i = 1:1440
      fpr(i,q) = (pr1(i,q) + pr2(i,q) + pr3(i,q));
   end
end
for q = 1:nds
   for i = 1:720
      pr(i,q) = (fpr(i+720,q) * Pp(q)) * ((wkm(r)*32.2)/WM(r)^2);
   end
end

%% Error and Std. Deviation
for q = 1:nds
   for i = 1:720
      e(i,q) = abs((dynoutp(i,q) - pr(i,q)) / (maxo(q) - mino(q)));       %percent
error of prediction
      ep(i,q) = (pr(i,q) - dynoutp(i,q)) ./ dynoutp(i,q);
      dev(i,q) = pr(i,q) - dynoutp(i,q);
      devs(i,q) = dev(i,q).^2;
   end
   ss(q) = sum(devs(:,q));
   var(q) = ss(q)/719;
   std(q) = sqrt(var(q));
end

%Frequency Conversion
for n = 1:nds
   step = 0;
   for i = 2:2:720
      step = step+1;
      dynp(step,n) = dynoutp(i,n);
      pro(step,n) = pr(i,n);
   end
   freqout(:,n) = fft(dynp(:,n),360)./360;
   magout(:,n) = abs(freqout(:,n));
   phaseout(:,n) = angle(freqout(:,n));
   freqpr(:,n) = fft(pro(:,n),360)./360;
```

```
   magpr(:,n) = abs(freqpr(:,n));
   phasepr(:,n) = angle(freqpr(:,n));
end
for n = 2:2:nds*2
   magc(:,n-1) = [magout(:,n/2)];
   phasec(:,n-1) = [phaseout(:,n/2)];
   magc(:,n) = [magpr(:,n/2)];
   phasec(:,n) = [phasepr(:,n/2)];
end

%Output
ri = 6-r;
harm = 1:15;
for q = 1:nds
  figure
  subplot(2,2,1);plot(deg,dynoutp(:,q),deg,pr(:,q),'r');grid
  title(sprintf('====== RI%d',ri));xlabel('deg');ylabel('Dyn Pt2');
  subplot(2,2,2);bar(harm-
1,magc(1:15,q:q+1));title('Frequency');xlabel('Harmonics');ylabel('Magnitude');
  subplot(2,2,4);bar(harm-
1,phasec(1:15,q:q+1));title('Phase');xlabel('Harmonics');ylabel('Radians');
  avge(q) = mean(e(:,q));
  avgep(q) = mean(ep(:,q));
end
```

12.2 Model 2

```
clear all;
close all;

%%Number of Data Sets
nds = 1;
%%Number of Data Sets Used to Train Kernel
ndsk = 4;
%%Number of Linear Basis Functions
lbf = 10;
%%Number of 2nd Order Basis Functions
qbf = 8;
%%Number of 3rd Order Basis Functions
cbf = 6;
%%Radial Immersion Identifier: 1 = r5, 2 = r4, 3 = r3, 4 = r2, 5 = r1 :: (r5 = hub, r1
= tip) ::
r = 4;

%%Stag Density at Inlet
rhos = .0627;  %9100
rhom = .0614;  %9500
rhof = .0426;  %13200

%%Time ND factors  (chord length/W1)
%9100 Tip ==> Hub
tnd(5) = .000597;  tnd(4) = .000636;  tnd(3) = .000691;  tnd(2) = .000775;  tnd(1) =
.000980;

%%W1 values for Each Rotational Speed
%9100 Tip==>Hub
WS(5) = 1129;  WS(4) = 1033;  WS(3) = 922;  WS(2) = 786;  WS(1) = 590;
%9500 Tip==>Hub
WM(5) = 1185;  WM(4) = 1084;  WM(3) = 969;  WM(2) = 827;  WM(1) = 623;
%13200 Tip==>Hub
WF(5) = 1736;  WF(4) = 1601;  WF(3) = 1439;  WF(2) = 1234;  WF(1) = 944;

%Work
%9100 Tip==>Hub
wks(5) = 17907.36;  wks(4) = 15996.65;  wks(3) = 16497.95;  wks(2) = 16656.24;
wks(1) = 18065.16;
```

```
%9500 Tip==>Hub
wkm(5) = 19860.62;  wkm(4) = 17793.54;  wkm(3) = 18220.27;  wkm(2) = 18431.00;
wkm(1) = 19724.34;
%13200 Tip==>Hub
wkf(5) = 32651.21;  wkf(4) = 34500.59;  wkf(3) = 36231.22;  wkf(2) = 36039.70;
wkf(1) = 37910.55;

%%Time Vector
load ..\datajfr\textdata\time9100.txt

%%Non-Dimensional Time
for i = 1:720
   t(i,1) = [[time9100(i,1)]/tnd(r)];
end
for i = 721:1440
   t(i,1) = t(i-720,1)+t(720,1);
end

deg = .5:.5:360;

%%Input and Response Vectors
load ..\datajfr\textdata\hall9100in.txt
load ..\datajfr\textdata\hall9100out.txt
load ..\datajfr\textdata\hall9500in.txt
load ..\datajfr\textdata\hall9500out.txt
load ..\datajfr\textdata\hall13200in.txt
load ..\datajfr\textdata\hall13200out.txt

%%Training data - 9100 : 13200nol,ns/pe
for p = 1:720
   in(p,1) = [hall9100in(p,r)];
   out(p,1) = [hall9100out(p,r)];
   in(p,2) = [hall9100in(p,5+r)];
   out(p,2) = [hall9100out(p,5+r)];
   in(p,3) = [hall9100in(p,10+r)];
   out(p,3) = [hall9100out(p,10+r)];
   in(p,4) = [hall9100in(p,15+r)];
   out(p,4) = [hall9100out(p,15+r)];
%     in(p,1) = [hall9500in(p,r)];
%     out(p,1) = [hall9500out(p,r)];
%     in(p,2) = [hall9500in(p,5+r)];
%     out(p,2) = [hall9500out(p,5+r)];
%     in(p,3) = [hall9500in(p,10+r)];
```

```
%     out(p,3) = [hall9500out(p,10+r)];
%     in(p,4) = [hall9500in(p,15+r)];
%     out(p,4) = [hall9500out(p,15+r)];
    in(p,5) = [hall13200in(p,r)];
    out(p,5) = [hall13200out(p,r)];
    in(p,6) = [hall13200in(p,5+r)];
    out(p,6) = [hall13200out(p,5+r)];
    in(p,7) = [hall13200in(p,10+r)];
    out(p,7) = [hall13200out(p,10+r)];
end

%%Prediction data - 9500 : 13200wod
for p = 1:720
%     inp(p,1) = [hall9100in(p,r)];
%     outp(p,1) = [hall9100out(p,r)];
%     inp(p,2) = [hall9100in(p,5+r)];
%     outp(p,2) = [hall9100out(p,5+r)];
%     inp(p,3) = [hall9100in(p,10+r)];
%     outp(p,3) = [hall9100out(p,10+r)];
%     inp(p,4) = [hall9100in(p,15+r)];
%     outp(p,4) = [hall9100out(p,15+r)];
    inp(p,1) = [hall9500in(p,r)];
    outp(p,1) = [hall9500out(p,r)];
%     inp(p,2) = [hall9500in(p,5+r)];
%     outp(p,2) = [hall9500out(p,5+r)];
%     inp(p,1) = [hall9500in(p,10+r)];
%     outp(p,1) = [hall9500out(p,10+r)];
%     inp(p,4) = [hall9500in(p,15+r)];
%     outp(p,4) = [hall9500out(p,15+r)];
%     inp(p,1) = [hall13200in(p,r)];
%     outp(p,1) = [hall13200out(p,r)];
%     inp(p,2) = [hall13200in(p,5+r)];
%     outp(p,2) = [hall13200out(p,5+r)];
%     inp(p,3) = [hall13200in(p,10+r)];
%     outp(p,3) = [hall13200out(p,10+r)];
end

%% Mean Pressure Calculation
for n = 1:ndsk
    mn(n) = mean(in(:,n));
    mo(n) = mean(out(:,n));
end
for n = 1:nds
```

```
   mnp(n) = mean(inp(:,n));
   mop(n) = mean(outp(:,n));
end

%% Non-Dimensional Pressure Calculation (1/2*rho*W1^2)
for n = 1:ndsk
   if n < 5
      P(n) =  .5 * rhos * WS(r)^2 / 4636.8;
   else
      P(n) = .5 * rhof * WF(r)^2 / 4636.8;
   end
end
for n = 1:nds
   Pp(n) = .5 * rhom * WM(r)^2 / 4636.8;
end

%% Dynamic Response ND'd by 1/2*rho*w1^2
for n=1:ndsk
   if n < 5
      x(:,n)=(in(:,n) - mn(n))/P(n);
      dynout(:,n) = (out(:,n) - mo(n))/P(n);
   else
      x(:,n)=(in(:,n) - mn(n))/P(n);
      dynout(:,n) = (out(:,n) - mo(n))/P(n);
   end
end
for n=1:nds
   xp(:,n)=(inp(:,n) - mnp(n))/Pp(n);
   dynoutp(:,n) = outp(:,n) - mop(n);
end

%% Pad input to 2 revolutions
for n = 1:ndsk
   for i = 721:1440
      x(i,n) = x(i-720,n);
      dynout(i,n) = dynout(i-720,n);
   end
end
for n = 1:nds
   for i = 721:1440
      xp(i,n) = xp(i-720,n);
   end
end
```

```
%% Linear Basis Functions & Motion Matrix
for q = 1:ndsk
   for i = 1:1440
      for j = 1:lbf
         t1 = t(i,1) - 1/j;
         z(i,j) = exp(-j * t1);
      end
   end
   aaa = conv2(z, x(:,q));
   for i = 1:1440
      aa(i,:) = aaa(i,:);
   end
   a(:,:,q) = aa;
end

%%2nd Order Time Constants & Basis Function Formulation
for i = 1:1440
   for j = 1:qbf
      t1 = t(i,1) - 1/j;
      for k = 1:qbf
         t2 = t(i,1) - 1/k;
         if (t1-t2) > 0
            mu(i,j,k) = exp(-j*(t1-t2) - k*t2);
         else
            mu(i,j,k) = exp(-j*(t2-t1) - k*t1);
         end
      end
   end
end

%%2nd Order Motion Matrix
for q = 1:ndsk
   bbb = convn(mu, x(:,q).^2);
   for i = 1:1440
      bb(i,:,:) = bbb(i,:,:);
   end
   ii = 0;
   for j = 1:qbf
      for k = 1:qbf
         ii = ii+1;
         bi(:,ii,q) = bb(:,j,k);
      end
```

```
   end
end

%%3rd Order Time Constants & Basis Function Formulation
for i = 1:1440
   for j = 1:cbf              %% j = 1/Tau1
      t1 = t(i,1) - 1/j;
      for k = 1:cbf           %% k = 1/Tau2
         t2 = t(i,1) - 1/k;
         for L = 1:cbf
            t3 = t(i,1) - 1/L;
            if (t1-t2) & (t2-t3) > 0
               mu3(i,j,k,L) = exp(-j*(t1-t2) - k*(t2-t3) - L*t3);
            elseif (t1-t3) & (t3-t2) > 0
               mu3(i,j,k,L) = exp(-j*(t1-t3) - k*(t3-t2) - L*t2);
            elseif (t2-t1) & (t1-t3) > 0
               mu3(i,j,k,L) = exp(-j*(t2-t1) - k*(t1-t3) - L*t3);
            elseif (t2-t3) & (t3-t1) > 0
               mu3(i,j,k,L) = exp(-j*(t2-t3) - k*(t3-t1) - L*t1);
            elseif (t3-t1) & (t1-t2) > 0
               mu3(i,j,k,L) = exp(-j*(t3-t1) - k*(t1-t2) - L*t2);
            else
               mu3(i,j,k,L) = exp(-j*(t3-t2) - k*(t2-t1) - L*t1);
            end
         end
      end
   end
end

%%3rd Order Motion Matrix
for q = 1:ndsk
   ddd = convn(mu3, x(:,q).^3);
   for i = 1:1440
      dd(i,:,:,:) = ddd(i,:,:,:);
   end
   ii = 0;
   for j = 1:cbf
      for k = 1:cbf
         for L = 1:cbf
            ii = ii+1;
            di(:,ii,q) = dd(:,j,k,L);
         end
      end
```

```
   end
end

%%Solution Matrix
p = zeros(1440,lbf+qbf^2+cbf^3,ndsk);
for q=1:ndsk
   for n=1:lbf
      p(:,n,q)=a(:,n,q);
   end
   for j2 = 1:qbf^2
      p(:,lbf+j2,q) = bi(:,j2,q);
   end
   for j3 = 1:cbf^3
      p(:,lbf+qbf^2+j3,q) = di(:,j3,q);
   end
end

%SVD / Basis Coefficients
for q = 1:ndsk
   ms(:,:,q) = transpose(p(:,:,q))*p(:,:,q);
   bs(:,:,q) = transpose(p(:,:,q))*(dynout(:,q));
end
m = sum(ms,3);
b = sum(bs,3);
cnd = 1;
for jj = 4:15
   if cnd == 0
      jj = jj - 2
      [u,s,v] = svds(m,jj);
      si = inv(s);
      cnd = rcond(si)
      break
   end
   [u,s,v] = svds(m,jj);
   si = inv(s);
   cnd = rcond(si)
end
ip = transpose(u)*b;
c = v*si*ip;

%%Linear Kernel H1
for n = 1:lbf
  c1(n,1) = c(n,1);
```

```
end
for j = 1:lbf
   hc1(:,j) = c1(j,1) * z(:,j);
end
h1(:,:) = sum(hc1,2);

%%2D Kernel H2
for n=1:qbf^2
  bc2(n,1)=c(n+lbf,1);
end
for j = 1:qbf
   acc = (j-1)*qbf;
   for k = 1:qbf
      c2(j,k) = bc2(acc+k,1);
   end
end
for j = 1:qbf
   for k = 1:qbf
      hc2(:,j,k) = c2(j,k) * mu(:,j,k);
   end
end
hs2(:,:,:) = sum(hc2,2);
h2(:,:) = sum(hs2,3);

%3D Kernel H3
for n = 1:cbf^3
   dc3(n,1) = c(n+lbf+qbf^2,1);
end
for j = 1:cbf
   for k = 1:cbf
      acc2 = (j-1)*cbf^2 + (k-1)*cbf;
      for L = 1:cbf
         c3(j,k,L) = dc3(acc2+L,1);
      end
   end
end
for j = 1:cbf
   for k = 1:cbf
      for L = 1:cbf
         hc3(:,j,k,L) = c3(j,k,L) * mu3(:,j,k,L);
      end
   end
end
```

```
hst3(:,:,:,:) = sum(hc3,2);
ht3(:,:,:,:) = sum(hst3,3);
h3(:,:) = sum(ht3,4);

%% Linear Response
for n = 1:nds
   fpp1 = conv2(hc1, xp(:,n));
   for i = 1:1440
      fp1(i,:) = fpp1(i,:);
   end
   prr1(:,:,n) = fp1;
   for i = 1:1440
      pr1(i,n) = sum(prr1(i,:,n)) * Pp(n);
   end
end

%%Quadratic Response
for n = 1:nds
   fpp2 = convn(hc2, xp(:,n).^2);
   for i = 1:1440
      fp2(i,:,:) = fpp2(i,:,:);
   end
   ii = 0;
   for j = 1:qbf
      for k = 1:qbf
         ii = ii+1;
         prr2(:,ii,n) = fp2(:,j,k);
      end
   end
   for i = 1:1440
      pr2(i,n) = sum(prr2(i,:,n)) * Pp(n);
   end
end

%%Cubic Response
for n = 1:nds
   fpp3 = convn(hc3, xp(:,n).^3);
   for i = 1:1440
      fp3(i,:,:,:) = fpp3(i,:,:,:);
   end
   ii = 0;
   for j = 1:cbf
      for k = 1:cbf
```

```
            for L = 1:cbf
                ii = ii+1;
                prr3(:,ii,n) = fp3(:,j,k,L);
            end
        end
    end
    for i = 1:1440
        pr3(i,n) = sum(prr3(i,:,n)) * Pp(n);
    end
end


%%Prediction
for q = 1:nds
    maxo(q) = max(dynoutp(:,q));
    mino(q) = min(dynoutp(:,q));
    for i = 1:1440
        fpr(i,q) = pr1(i,q) + pr2(i,q) + pr3(i,q);
    end
end
for q = 1:nds
    for i = 1:720
        pr(i,q) = fpr(i+720,q);
    end
end

%% Error and Std. Deviation
for q = 1:nds
    for i = 1:720
        e(i,q) = abs((dynoutp(i,q) - pr(i,q)) / (maxo(q) - mino(q)));        %percent
error of prediction
        ep(i,q) = (pr(i,q) - dynoutp(i,q)) ./ dynoutp(i,q);
        dev(i,q) = pr(i,q) - dynoutp(i,q);
        devs(i,q) = dev(i,q).^2;
    end
    ss(q) = sum(devs(:,q));
    var(q) = ss(q)/719;
    std(q) = sqrt(var(q));
end

%Frequency Conversion
for n = 1:nds
    step = 0;
```

```
   for i = 2:2:720
      step = step+1;
      dynp(step,n) = dynoutp(i,n);
      pro(step,n) = pr(i,n);
   end
   freqout(:,n) = fft(dynp(:,n),360)./360;
   magout(:,n) = abs(freqout(:,n));
   phaseout(:,n) = angle(freqout(:,n));
   freqpr(:,n) = fft(pro(:,n),360)./360;
   magpr(:,n) = abs(freqpr(:,n));
   phasepr(:,n) = angle(freqpr(:,n));
end
for n = 2:2:nds*2
   magc(:,n-1) = [magout(:,n/2)];
   phasec(:,n-1) = [phaseout(:,n/2)];
   magc(:,n) = [magpr(:,n/2)];
   phasec(:,n) = [phasepr(:,n/2)];
end

%Output
ri = 6-r;
harm = 1:15;
for q = 1:nds
  figure
  plot(deg,pr1(721:1440,q),'r',deg,pr2(721:1440,q),'g',deg,pr3(721:1440,q),'b',...
     deg,pr(:,q),'k-x');legend('1st Order','2nd Order','3rd Order','Prediction');grid
  figure
  subplot(2,2,1);plot(deg,dynoutp(:,q),deg,pr(:,q),'r');grid
  title(sprintf('====== RI%d',ri));xlabel('deg');ylabel('Dyn Pt2');
  subplot(2,2,2);bar(harm-
1,magc(1:15,q:q+1));title('Frequency');xlabel('Harmonics');ylabel('Magnitude');
  subplot(2,2,4);bar(harm-
1,phasec(1:15,q:q+1));title('Phase');xlabel('Harmonics');ylabel('Radians');
  avge(q) = mean(e(:,q));
  avgep(q) = mean(ep(:,q));
end
```

12.3 Model 3

```
clear all;
close all;

%%Number of Data Sets
nds =1;
%%Number of Data Sets Used to Train Kernel
ndsk=7;
%%Number of Linear Basis Functions
lbf=10;
%%Number of 2nd Order Basis Functions
qbf=5;
%%Radial Immersion Identifier: 1 = r5, 2 = r4, 3 = r3, 4 = r2, 5 = r1 :: (r5 = hub, r1
= tip) ::
r = 1;
%%Rotational speed of prediction
speed = 9500;

%%Stag Density at Inlet
rhos = .0627;  %9100
rhom = .0614;  %9500
rhof = .0426;  %13200

%%Time ND factors  (chord length/W1)
%9100 Tip ==> Hub
tnd(5) = .000597;  tnd(4) = .000636;  tnd(3) = .000691;  tnd(2) = .000775;  tnd(1) =
.000980;

%%W1 values for Each Rotational Speed
%9100 Tip==>Hub
WS(5) = 1129;  WS(4) = 1033;  WS(3) = 922;  WS(2) = 786;  WS(1) = 590;
%9500 Tip==>Hub
WM(5) = 1185;  WM(4) = 1084;  WM(3) = 969;  WM(2) = 827;  WM(1) = 623;
%13200 Tip==>Hub
WF(5) = 1736;  WF(4) = 1601;  WF(3) = 1439;  WF(2) = 1234;  WF(1) = 944;

%Work
%9100 Tip==>Hub
wks(5) = 17907.36;  wks(4) = 15996.65;  wks(3) = 16497.95;  wks(2) = 16656.24;
wks(1) = 18065.16;
```

```
%9500 Tip==>Hub
wkm(5) = 19860.62;  wkm(4) = 17793.54;  wkm(3) = 18220.27;  wkm(2) = 18431.00;
wkm(1) = 19724.34;
%13200 Tip==>Hub
wkf(5) = 32651.21;  wkf(4) = 34500.59;  wkf(3) = 36231.22;  wkf(2) = 36039.70;
wkf(1) = 37910.55;

%%Time Vector
load ..\datajfr\textdata\t9100.txt

%%Non-Dimensional Time
for i = 1:360
   t(i,1) = [[t9100(i,1)]/tnd(r)];
end
for i = 361:720
   t(i,1) = t(i-360,1)+t(360,1);
end

deg = 1:360;

%%Input and Response Vectors
load ..\datajfr\textdata\all9100in.txt
load ..\datajfr\textdata\all9100out.txt
load ..\datajfr\textdata\all9500in.txt
load ..\datajfr\textdata\all9500out.txt
load ..\datajfr\textdata\all13200in.txt
load ..\datajfr\textdata\all13200out.txt

%%Training data
for p = 1:360
   in(p,1) = [all9100in(p,r)];
   out(p,1) = [all9100out(p,r)];
   in(p,2) = [all9100in(p,5+r)];
   out(p,2) = [all9100out(p,5+r)];
   in(p,3) = [all9100in(p,10+r)];
   out(p,3) = [all9100out(p,10+r)];
   in(p,4) = [all9100in(p,15+r)];
   out(p,4) = [all9100out(p,15+r)];
%    in(p,1) = [all9500in(p,r)];
%    out(p,1) = [all9500out(p,r)];
%    in(p,2) = [all9500in(p,5+r)];
%    out(p,2) = [all9500out(p,5+r)];
%    in(p,3) = [all9500in(p,10+r)];
```

```
%     out(p,3) = [all9500out(p,10+r)];
   in(p,4) = [all9500in(p,15+r)];
   out(p,4) = [all9500out(p,15+r)];
   in(p,5) = [all13200in(p,r)];
   out(p,5) = [all13200out(p,r)];
   in(p,6) = [all13200in(p,5+r)];
   out(p,6) = [all13200out(p,5+r)];
   in(p,7) = [all13200in(p,10+r)];
   out(p,7) = [all13200out(p,10+r)];
end

%%Prediction data
for p = 1:360
%     inp(p,1) = [all9100in(p,r)];
%     outp(p,1) = [all9100out(p,r)];
%     inp(p,2) = [all9100in(p,5+r)];
%     outp(p,2) = [all9100out(p,5+r)];
%     inp(p,3) = [all9100in(p,10+r)];
%     outp(p,3) = [all9100out(p,10+r)];
%     inp(p,4) = [all9100in(p,15+r)];
%     outp(p,4) = [all9100out(p,15+r)];
   inp(p,1) = [all9500in(p,r)];
   outp(p,1) = [all9500out(p,r)];
%     inp(p,2) = [all9500in(p,5+r)];
%     outp(p,2) = [all9500out(p,5+r)];
%     inp(p,3) = [all9500in(p,10+r)];
%     outp(p,3) = [all9500out(p,10+r)];
%     inp(p,1) = [all9500in(p,15+r)];
%     outp(p,1) = [all9500out(p,15+r)];
%     inp(p,1) = [all13200in(p,r)];
%     outp(p,1) = [all13200out(p,r)];
%     inp(p,2) = [all13200in(p,5+r)];
%     outp(p,2) = [all13200out(p,5+r)];
%     inp(p,3) = [all13200in(p,10+r)];
%     outp(p,3) = [all13200out(p,10+r)];
end

%% Mean Calculation
for n = 1:ndsk
   mn(n) = mean(in(:,n));
   mo(n) = mean(out(:,n));
end
for q = 1:nds
```

```
   mnp(q) = mean(inp(:,q));
   mop(q) = mean(outp(:,q));
end

%% Non-Dimensional Pressure Calculation (1/2*rho*W1^2)
for n = 1:ndsk
   if n < 5
      P(n) =  .5 * rhom * WM(r)^2 / 4636.8;
   else
      P(n) = .5 * rhof * WF(r)^2 / 4636.8;
   end
end
for q = 1:nds
   Pp(q) = .5 * rhom * WM(r)^2 / 4636.8;
end

%% Dynamic Response ND'd by 1/2*rho*w^2
for n=1:ndsk
   if n < 5
      x(:,n)=((in(:,n) - mn(n))/P(n));
      dynout(:,n) = ((out(:,n) - mo(n))/P(n));
   else
      x(:,n)=((in(:,n) - mn(n))/P(n));
      dynout(:,n) = ((out(:,n) - mo(n))/P(n));
   end
end
for q=1:nds
   xp(:,q)=((inp(:,q) - mnp(q))/Pp(q));
   dynoutp(:,q) = (outp(:,q) - mop(q));
end

%% Pad input to 2 revolutions
for n = 1:ndsk
   for i = 361:720
      x(i,n) = x(i-360,n);
   end
end
for q = 1:nds
   for i = 361:720
      xp(i,q) = xp(i-360,q);
   end
end
```

```
%% Linear Basis Function
for i = 1:360
   for j = 1:lbf
      z(i,j) = exp(-j * t(i,1));
   end
end

%% Linear Motion Matrix
dt = t(1,1);
for n = 1:ndsk
   for i = 1:360
      ii = i+360;
      ac1 = 0;
      for i1 = 1:360
         aa(i1,:) = z(i1,:) .* x(ii-ac1,n) .* dt;
         ac1 = ac1 + 1;
      end
      a(i,:,n) = sum(aa,1);
   end
end

%%2nd Order Basis function
t1 = 0:dt:t(359,1); t1 = t1';
t2 = 0:dt:t(359,1); t2 = t2';
for i1= 1:360
   for i2 = 1:360
      for k1 = 1:qbf
         for k2 = 1:qbf
            if (t1(i1,1) - t2(i2,1)) > 0
               mu(i1,i2,k1,k2) = exp(-k1*(t1(i1,1) - t2(i2,1)) - k2*t2(i2,1));
            else
               mu(i1,i2,k1,k2) = exp(-k1*(t2(i2,1) - t1(i1,1)) - k2*t1(i1,1));
            end
         end
      end
   end
end

%%2nd Order Motion Matrix
for n = 1:ndsk
   for i = 1:360
      ii = i+360;
      ac1 = 0;
```

```
      for i1 = 1:360
         ac2 = 0;
         for i2 = 1:360
            mmsub1(i1,i2,:,:) = mu(i1,i2,:,:) .* x(ii-ac1,n) .* x(ii-ac2,n) .* dt^2;
            ac2 = ac2 + 1;
         end
         ac1 = ac1 + 1;
      end
      mmsub2 = sum(mmsub1,2);
      mmsub = sum(mmsub2,1);
      bb(i,:,:,n) = mmsub;
   end
end
acc=0;
for k1 = 1:qbf
   for k2 = 1:qbf
      acc = acc+1;
      bi(:,acc,:) = bb(:,k1,k2,:);
   end
end

%%Solution Matrix
p = zeros(360,lbf+qbf^2,ndsk);
for n=1:ndsk
   for j=1:lbf
      p(:,j,n)=a(:,j,n);
   end
   for k = 1:qbf^2
      p(:,lbf+k,n) = bi(:,k,n);
   end
end

%SVD / Basis Coefficients
for q = 1:ndsk
   ms(:,:,q) = transpose(p(:,:,q))*p(:,:,q);
   bs(:,:,q) = transpose(p(:,:,q))*(dynout(:,q));
end
m = sum(ms,3);
b = sum(bs,3);
cnd = 1;
for jj = 4:12
   if cnd == 0
      jj = jj - 2
```

```
        [u,s,v] = svds(m,jj);
        si = inv(s);
        break
    end
    [u,s,v] = svds(m,jj);
    si = inv(s);
    cnd = rcond(si)
end
ip = transpose(u)*b;
c = v*si*ip;

%%Linear Kernel H1
for n = 1:lbf
  c1(n,1) = c(n,1);
end
for j = 1:lbf
   hc1(:,j) = c1(j,1) * z(:,j);
end
h1(:,:) = sum(hc1,2);

%%2D Kernel H2
for n=1:qbf^2
  bc2(n,1)=c(n+lbf,1);
end
for k1 = 1:qbf
   acc = (k1-1)*qbf;
   for k2 = 1:qbf
      c2(k1,k2) = bc2(acc+k2,1);
   end
end
for k1 = 1:qbf
   for k2 = 1:qbf
      hc2(:,:,k1,k2) = c2(k1,k2) * mu(:,:,k1,k2);
   end
end
hs2(:,:,:,:) = sum(hc2,4);
h2(:,:) = sum(hs2,3);

%% Linear Response
for q = 1:nds
   for i = 1:360
      ii = i+360;
      ac1 = 0;
```

```
      for i1 = 1:360
         pr1sub(i1,1) = h1(i1,1) * x(ii-ac1,q) * dt;
         ac1 = ac1 + 1;
      end
      pr1(i,q) = sum(pr1sub,1);
   end
end

%%Quadratic Response
for q = 1:nds
   for i = 1:360
      ii = i+360;
      ac1 = 0;
      for i1 = 1:360
         ac2 = 0;
         for i2 = 1:360
            pr2sub(i1,i2) = h2(i1,i2) * x(ii-ac1,q) * x(ii-ac2,q) * dt^2;
            ac2 = ac2 + 1;
         end
         ac1 = ac1+1;
      end
      pr2(i,q) = sum(sum(pr2sub,2),1);
   end
end

%%Prediction
for q = 1:nds
   maxo(q) = max(dynoutp(:,q));
   mino(q) = min(dynoutp(:,q));
   for i = 1:360
      pr(i,q) = (pr1(i,q) + pr2(i,q))*Pp(q);
   end
end

%% Error and Std. Deviation
for q = 1:nds
   for i = 1:360
      e(i,q) = abs((dynoutp(i,q) - pr(i,q)) / (maxo(q) - mino(q)));       %percent
error of prediction
      ep(i,q) = (pr(i,q) - dynoutp(i,q)) ./ dynoutp(i,q);
      dev(i,q) = pr(i,q) - dynoutp(i,q);
      devs(i,q) = dev(i,q).^2;
   end
```

```
   ss(q) = sum(devs(:,q));
   var(q) = ss(q)/359;
   std(q) = sqrt(var(q));
end

%Frequency Conversion
for n = 1:nds
   freqout(:,n) = fft(dynoutp(:,n),360)./360;
   magout(:,n) = abs(freqout(:,n));
   phaseout(:,n) = angle(freqout(:,n));
   freqpr(:,n) = fft(pr(:,n),360)./360;
   magpr(:,n) = abs(freqpr(:,n));
   phasepr(:,n) = angle(freqpr(:,n));
end
for n = 2:2:nds*2
   magc(:,n-1) = [magout(:,n/2)];
   phasec(:,n-1) = [phaseout(:,n/2)];
   magc(:,n) = [magpr(:,n/2)];
   phasec(:,n) = [phasepr(:,n/2)];
end

%Output
ri = 6-r;
harm = 1:15;
for q = 1:nds
  figure
  subplot(2,2,1);plot(deg,dynoutp(:,q),deg,pr(:,q),'r');grid
  title(sprintf('%d RI%d',speed,ri));xlabel('deg');ylabel('Dyn Pt2');
  subplot(2,2,2);bar(harm-
1,magc(1:15,q:q+1));title('Frequency');xlabel('Harmonics');ylabel('Magnitude');
  subplot(2,2,4);bar(harm-
1,phasec(1:15,q:q+1));title('Phase');xlabel('Harmonics');ylabel('Radians');
  avge(q) = mean(e(:,q));
  avgep(q) = mean(ep(:,q));
end
```

VITA

John F. Ryman

The author, son of Franklin and Mary Ryman, was born in July 1966 in Gettysburg, PA. He grew up on the family farm, which his father still works as his retirement "hobby" to this day. After attending high school at a small town school, he applied and was accepted to the Virginia Military Institute on a full ROTC scholarship where he studied Mechanical Engineering. Upon graduation with a B.S. degree, he was commissioned a Second Lieutenant in the Marine Corps and served for four years during which time he was deployed to the Persian Gulf for Operation Desert Storm. After his time in the military, John worked in maintenance management with an equipment rental company in California before moving to Massachusetts where he worked in the chemical manufacturing industry as a chemical operator and, later on, as a shift supervisor. Upon realizing that he had reached a stalling point in his career and wanting to return to engineering, he decided to return to school in pursuit of a graduate degree in Mechanical Engineering. Not liking the winters in the North and having enjoyed living in Virginia during his time at VMI, he decided on Virginia Tech and spent two years there working towards a Master's degree in Mechanical Engineering. Upon graduation, he will begin employment at GE Power Systems in Greenville, SC as an engineer in the aerodynamics design group.