

Use Authorization

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at Idaho State University, I agree that the Library shall make it freely available for inspection. I further state that permission to download and/or print my thesis for scholarly purposes may be granted by the Dean of the Graduate School, Dean of my academic division, or by the University Librarian. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Signature _____

Date _____

**SYSTEM IDENTIFICATION AND MODELING OF THE DYNAMICS OF A
ONE STAGE AXIAL FLOW COMPRESSOR SYSTEM**

by

Dane Michael Sterbentz

**A thesis submitted in partial fulfillment of the
requirements for the degree of**

MASTER OF SCIENCE

IN

MEASUREMENT AND CONTROL ENGINEERING

IDAHO STATE UNIVERSITY

Spring 2015

Copyright 2015 Dane Michael Sterbentz

To the Graduate Faculty:

The members of the committee appointed to examine the thesis of DANE M. STERBENTZ find it satisfactory and recommend that it be accepted.

Dr. Marco P. Schoen, Major Advisor

Dr. Kenneth W. Bosworth, Co-Advisor

Dr. Bruce Savage, GFR

ACKNOWLEDGEMENTS

I would like to extend my sincerest gratitude to my major advisor, Dr. Marco Schoen, as well as to my co-advisor, Dr. Kenneth Bosworth, who have provided strong support and guidance for my research at every step of the way. The knowledge and expertise that they have provided has been invaluable and highly appreciated.

I would also like to give a special thanks to Dr. Ji-chao Li and Dr. Feng Lin of the Chinese Academy of Sciences, Institute of Engineering Thermophysics, for making this research possible by aiding and guiding our focus as well as allowing us to use their laboratory to perform numerous experiments. Additional thanks are due for hosting me at the Institute for a month, allowing me to personally perform a number of experiments, and making my time in Beijing such a positive and memorable experience.

Thank you to Idaho State University and the College of Science and Engineering for providing an internal research grant, which made my research trip to the Institute of Engineering Thermophysics in Beijing possible. Thanks also to Dr. Brian Williams and the ISU Department of Mechanical Engineering for providing me with a research assistantship as well as a graduate teaching assistantship that provided me funding to complete my degree.

Lastly, I would like to thank the other students at Idaho State University who also worked on and contributed to this project, namely Thomas Walters, Mary Hofle, and Sujan Prasai.

TABLE OF CONTENTS

NOMENCLATURE	viii
LIST OF FIGURES.....	xi
LIST OF TABLES.....	xiv
ABSTRACT	xv
CHAPTER 1 INTRODUCTION.....	1
1.1 Background and Motivation	1
1.2 Problem Statement	6
1.3 Thesis Goals.....	8
1.4 Research History of Rotating Stall and Surge in Axial Flow Compressors	9
1.5 Introduction to Axial Compressor Slow Dynamics	13
1.5.1 Research Background of Slow Dynamics and Rotating Stall	13
1.5.2 Slow Dynamics Modeling for Compressor Systems	16
1.6 Introduction to Axial Compressor Fast Dynamics.....	19
1.6.1 Research Background of Fast Dynamics	19
1.6.2 Correlation Coefficient for Fast Dynamics Analysis.....	21
1.7 Relation of Slow Dynamics to Fast Dynamics Overview	23
CHAPTER 2 EXPERIMENTAL SETUP AND DATA COLLECTION	26
2.1 Axial Flow Compressor Specifications	26
2.2 Compressor Throttle Actuation.....	28
2.3 Design of High-Frequency Throttle Modules.....	31
2.4 Fast Dynamics Data Collection	35
CHAPTER 3 THEORY ON SYSTEM IDENTIFICATION	38
3.1 Overview of System Identification	38
3.2 System Identification Calculations	40
3.3 Excitation Input for System Identification	42
3.4 Signal Noise in System Identification	43
CHAPTER 4 DEVELOPMENT OF SLOW DYNAMICS MODEL.....	45
4.1 Slow Dynamics Model Development Method.....	45
4.2 Bayesian Filtering of Measured Data Signal	49
4.3 Particle Swarm Optimization for Filter Optimization.....	54

4.4 Slow Dynamics System Identification	58
4.5 Amplitude Spectrums and Bode Plots for Slow Dynamics SI Models	66
CHAPTER 5 DEVELOPMENT OF FAST DYNAMICS MODEL	71
5.1 Dynamic Pressure Sensors and Air Injection	71
5.2 Autocorrelation Coefficient Calculation	74
5.3 Missing Data and Cubic Spline Interpolation.....	77
5.4 Fast Dynamics System Identification.....	79
CHAPTER 6 RESULTS AND DISCUSSION	82
6.1 Slow Dynamics Analysis Results	82
6.1.1 Pole Behavior of Slow Dynamics Transfer Function Models.....	82
6.1.2 Pole Variation Investigation.....	86
6.2 Fast Dynamics Analysis Results	90
6.2.1 Pole Behavior of Fast Dynamics Transfer Function Models	90
CHAPTER 7 CONCLUSIONS AND FUTURE WORK	105
7.1 Summary and Conclusions	105
7.2 Future Work.....	107
REFERENCES	108
APPENDIX A – Matlab™ Files for Slow Dynamics Analysis	113
A1 - Bayesian Filter	113
A2 – Particle Swarm Optimization for Optimizing Bayesian Filter.....	117
A3 – Transfer Function Goodness of Fit Cost Function (Includes Bayesian Filter) for PSO	119
A4 – ARX Goodness of Fit Cost Function (Includes Bayesian Filter) for PSO	123
A5 – Pole Variation Investigation	127
APPENDIX B – Matlab™ Files for Fast Dynamics Analysis	132
B1 – Spline Interpolated Autocorrelation Coefficient Data Set Calculation	132

NOMENCLATURE

AIC: Akaike Information Criterion

a_{i_n} : n th ARX Model Output Coefficient

ARMAX: Autoregressive-Moving-Average with Exogenous Input

ARX: Autoregressive with Exogenous Input

BF: Bayesian Filtered

BPF: Blade Passing Frequency

b_{i_n} : n th ARX Model Input Coefficient

c_1 : Constant that Determines Effect of Local Best on Particle Velocity

c_2 : Constant that Determines Effect of Global Best on Particle Velocity

CAS: Chinese Academy of Sciences

C_x : Axial Flow Velocity

EMG: Electromyogram

e_n : ARX Model Error

f_n : Natural Frequency in Units of Hz

$G(s)$: Transfer Function in Terms of Complex Variable

IET: Institute of Engineering Thermophysics

IGV: Inlet Guide Vanes

$imag$: Imaginary Portion of a Complex Number

L : Number of Data Points Collected in a Single Throttle Oscillation Experiment

M : Number of Throttle Oscillation Experiments Conducted at a Flow Coefficient

N : Gaussian (Normal) Distribution

p : Transfer Function Pole/ARX Model Order

$P[_ | _]$: Mathematical Representation of the Probability Density Function

PDF: Probability Density Function

\hat{P}^{LS} : Covariance of Least-Squares Estimate

PSD: Power Spectral Density

PSO: Particle Swarm Optimization

r : Radius
RBS: Random Binary Sequence
 $real$: Real Portion of a Complex Number
 R_{xx} : Autocorrelation Coefficient
 s : Tip of Rotor Blade to Flow Stream Incidence Point Distance
 $sign$: Signum Function
SIMO: Single Input, Multiple Output
SISO: Single Input, Single Output
 t : Time Variable
TLF: Tip Leakage Flow
 U : Mean Compressor Rotor Velocity
UTLF: Unsteady Tip Leakage Flow
 V_i^j : i Component of the Velocity of Particle j
 x_i^j : i Component of the Position of Particle j
 z : Transfer Function Zero
 α : Bayesian Filter Diffusion Rate
 β : Bayesian Filter Probability of Sudden Jumps in Data
 $\hat{\Gamma}^E$: Empirical Unbiased Covariance Estimate
 ΔP : Change in Pressure from Inlet Atmospheric Pressure to Plenum Pressure
 θ : Flow Stream Angle with Rotor Blade Chord Line
 $\hat{\theta}_n$: Estimate of the n th ARX Model Parameters
 $\hat{\theta}_n^{LS}$: Least-Squares Estimate of the n th ARX Model Parameters
 $\hat{\theta}_0$: Average Least-Squares Estimate of the ARX Model Parameters
 ρ : Density of Fluid Passing through Compressor
 σ^2 : Variance of Normal Gaussian Distribution
 ϕ : Flow Coefficient (Non-Dimensional)
 ψ : Pressure Rise Coefficient (Non-Dimensional)
 ψ : Pressure Disturbance Coefficient (Non-Dimensional)

ω_n : Natural Frequency in Units of rad/sec

ω_s : Flow Stream Oscillation Signature Frequency

LIST OF FIGURES

Figure 1.1: Diagram of a One Stage Axial Flow Compressor System	2
Figure 1.2: Compressor Characteristic Curve for Axial Compressor Undergoing Stall.....	3
Figure 1.3: Traveling Wave Energy versus Time in Rotor Revolutions near Stall Inception	14
Figure 1.4: Relative Locations of Spike and Modal Stall Inception on the Characteristic Curve	15
Figure 1.5: Mass-Spring-Damper System for Grey-Box Modeling Approach	16
Figure 1.6: Bode Plots (Magnitude and Phase) for Transfer Function Model with Spatial Sine Wave Input	18
Figure 1.7: Establishment of the Interface Line for Flow in a Blade Passage	19
Figure 1.8: Routes to Stall in Axial Compressor	20
Figure 1.9: Compressor Characteristic Lines for Several Injection Methods	22
Figure 2.1: Experimental Axial Compressor Setup at the Advanced Energy and Technology Laboratory at the Institute of Engineering Thermophysics, CAS.....	27
Figure 2.2: Compressor with Casing Removed (Left) and Compressor Rotor Blades (Right).....	27
Figure 2.3 : Cone and Ring Throttle Valve Setup.....	28
Figure 2.4: Screw Mechanism for Throttle Actuation Side View (Left) and Frontal View (Right).....	29
Figure 2.5: Stepper Motor for Throttle Ring Actuation (Mounted on Compressor Exterior).....	30
Figure 2.6: Throttle Module Design (Top View).....	32
Figure 2.7: Throttle Module Design (Demonstrative View).....	32
Figure 2.8: Throttle Module Design Sketch with Dimensions in Millimeters	33
Figure 2.9: Ten Kulite Type Pressure Sensors for Fast Dynamics Data Collection (Both Sides).....	35
Figure 2.10: Two Views of Several of the Air Injection Actuators	36
Figure 2.11: Hall Effect Sensor Voltage Signal versus Time for 10 Rotor Revolutions	37
Figure 3.1: Block Diagram Representation of Relation between Input and Output.....	40
Figure 4.1: Flow Chart of Optimization Method for Data Filtering and System Identification	45
Figure 4.2: BF Data Comparison to Measured Data for Pressure Rise Coefficient (ψ) for 5 Cycles.....	52
Figure 4.3: BF Data Comparison to Measured Data for Pressure Rise Coefficient (ψ) for 1 Cycle	52
Figure 4.4: BF Data Comparison to Measured Data for Flow Coefficient (ϕ) for 1 Cycle	53
Figure 4.5: BF Data Comparison to Measured Data for Flow Coefficient (ϕ) for 5 Cycles.....	53
Figure 4.6: PSO Particle Movement from Previous Position to Updated Position	56
Figure 4.7: Transfer Function Model Comparison to the Measured Data, $\phi = 0.60$	60
Figure 4.8: Transfer Function Model Comparison to the Measured Data, $\phi = 0.58$	60
Figure 4.10: Transfer Function Model Comparison to the Measured Data, $\phi = 0.54$	61
Figure 4.9: Transfer Function Model Comparison to the Measured Data, $\phi = 0.56$	61

Figure 4.12: Transfer Function Model Comparison to the Measured Data, $\phi = 0.51$	62
Figure 4.11: Transfer Function Model Comparison to the Measured Data, $\phi = 0.52$	62
Figure 4.14: Transfer Function Model Comparison to the Measured Data, $\phi = 0.58$ for 5 Cycles.....	63
Figure 4.13: Transfer Function Model Comparison to the Measured Data, $\phi = 0.60$ for 5 Cycles.....	63
Figure 4.15: Transfer Function Model Comparison to the Measured Data, $\phi = 0.56$ for 5 Cycles.....	64
Figure 4.16: Transfer Function Model Comparison to the Measured Data, $\phi = 0.54$ for 5 Cycles.....	64
Figure 4.17: Transfer Function Model Comparison to the Measured Data, $\phi = 0.52$ for 5 Cycles.....	65
Figure 4.18: Transfer Function Model Comparison to the Measured Data, $\phi = 0.51$ for 5 Cycles.....	65
Figure 4.19: Frequency Amplitude Spectrum for Measured Pressure Rise Coefficient Data.....	67
Figure 4.20: Frequency Amplitude Spectrum for Measured Flow Coefficient Data.....	67
Figure 4.21: Frequency Amplitude Spectrum for the Bayesian Filtered Pressure Rise Coefficient Data.....	68
Figure 4.22: Frequency Amplitude Spectrum for the Bayesian Filtered Flow Coefficient Data	69
Figure 4.23: Magnitude Bode Plots for the Model Transfer Functions obtained from the BF Data	70
Figure 5.1: Dynamic Pressure Sensor Locations Relative to Rotor Blades	71
Figure 5.2: Power Spectral Density for Dynamic Pressure Sensors Near Stall Point	72
Figure 5.3: Typical Air Injection Pressure Signal versus Time	73
Figure 5.4: Autocorrelation Coefficient versus Time for Sensors CH1, CH2, and CH3 at $\phi = 0.56$	75
Figure 5.5: Autocorrelation Coefficient versus Time for Sensors CH4, CH5, and CH6 at $\phi = 0.56$	76
Figure 5.6: Autocorrelation Coefficient versus Time for Sensors CH7, CH8, and CH9 at $\phi = 0.56$	76
Figure 5.7: Approach for Preparing Fast Dynamics Data for System Identification.....	78
Figure 5.8: Linear Interpolated Autocorrelation Coefficient versus Time for $\phi = 0.56$ at Sensor CH5	78
Figure 5.9: Spline Interpolated Autocorrelation Coefficient versus Time for $\phi = 0.56$ at Sensor CH5	78
Figure 5.10: Block Diagram Representation of the SIMO System	80
Figure 5.11: Transfer Function Model Comparison for $\phi = 0.56$ at Sensor CH5	81
Figure 6.1: Pole Behavior of Poles 1 and 2 for Slow Dynamics Model.....	84
Figure 6.2: Pole Behavior of Poles 3 and 4 for Slow Dynamics Model.....	84
Figure 6.3: Approach for Pole Variation Investigation using Model Parameters of ARX Model	86
Figure 6.4: Pole Variation as Stall is Approached	89
Figure 6.5: Pole Behavior of Poles 1 and 2 for Sensor CH4 (Experiment 1)	94
Figure 6.6: Pole Behavior of Poles 3 and 4 for Sensor CH4 (Experiment 1)	94
Figure 6.8: Pole Behavior of Poles 3 and 4 for Sensor CH5 (Experiment 1)	95
Figure 6.7: Pole Behavior of Poles 1 and 2 for Sensor CH5 (Experiment 1)	95
Figure 6.10: Pole Behavior of Poles 3 and 4 for Sensor CH6 (Experiment 1).....	96

Figure 6.9: Pole Behavior of Poles 1 and 2 for Sensor CH6 (Experiment 1)	96
Figure 6.12: Pole Behavior of Poles 3 and 4 for Sensor CH4 (Experiment 2).....	99
Figure 6.11: Pole Behavior of Poles 1 and 2 for Sensor CH4 (Experiment 2).....	99
Figure 6.13: Pole Behavior of Poles 1 and 2 for Sensor CH5 (Experiment 2).....	100
Figure 6.14: Pole Behavior of Poles 3 and 4 for Sensor CH5 (Experiment 2).....	100
Figure 6.15: Pole Behavior of Poles 1 and 2 for Sensor CH6 (Experiment 2).....	101
Figure 6.16: Pole Behavior of Poles 3 and 4 for Sensor CH6 (Experiment 2).....	101
Figure 6.17: Extrapolated Root Locus Behavior of Poles at Sensor CH5	102
Figure 6.18: Extrapolated Root Locus Behavior of Poles at Sensor CH6	103
Figure 6.19: Magnitude Bode Plots for the Fast Dynamics Model Transfer Functions (CH5).....	104
Figure 6.20: Magnitude Bode Plots for the Fast Dynamics Model Transfer Functions (CH6).....	104

LIST OF TABLES

Table 2.1: Parameters for Experimental Axial Compressor at the IET, CAS.....	26
Table 2.2: Fast Dynamics Pressure Sensor Calibration Factors.....	36
Table 4.1: Transfer Function Model Goodness of Fit at Flow Coefficient Set Points	59
Table 5.1: Dynamic Pressure Sensor Location Distances	71
Table 6.1: Slow Dynamics Transfer Function Model Data	83
Table 6.2: Fast Dynamics Transfer Function Model Data for Sensor CH4 (Experiment 1).....	91
Table 6.3: Fast Dynamics Transfer Function Model Data for Sensor CH5 (Experiment 1).....	92
Table 6.4: Fast Dynamics Transfer Function Model Data for Sensor CH6 (Experiment 1).....	93
Table 6.5: Fast Dynamics Transfer Function Model Data for Sensor CH4 (Experiment 2).....	97
Table 6.6: Fast Dynamics Transfer Function Model Data for Sensor CH5 (Experiment 2).....	97
Table 6.7: Fast Dynamics Transfer Function Model Data for Sensor CH6 (Experiment 2).....	98

ABSTRACT

System Identification and Modeling of the Dynamics of a One Stage Axial Flow Compressor System

Thesis Abstract--Idaho State University (2015)

The research presented in this thesis details the system identification and modeling methods used to characterize the dynamics of a one stage axial flow compressor system. Specifically, the system identification is focused on two primary areas, the dynamic relationship between the compressor flow coefficient and pressure rise coefficient as well as the pressure dynamics at and around the rotor blade passages. A method for optimizing filter parameters when performing system identification is proposed. An autocorrelation coefficient method is also used for processing the dynamic pressure data. Transfer function models are developed using a black-box system identification approach. The results of the system identified models are presented in complex plane pole plots and magnitude Bode plots that demonstrate consequential trend behavior as the compressor stall point is approached.

CHAPTER 1 INTRODUCTION

1.1 Background and Motivation

Optimizing and controlling the flow in fluid machinery can result in vast improvements of its economic and environmental impact. Even small improvements of the efficiency in certain types of fluid machinery can result in significant cost savings. An example of such improvements is given by Przybylko in [1], where one stage of a compressor in an aircraft engine is eliminated through the use of active control of the flow in the engine. This reduction in the number of stages results in a 5 percent increase in the thrust-to-weight ratio and a 1.5 percent reduction in fuel consumption. The resulting improvement in fuel consumption becomes even more impressive if one considers the number of commercial and noncommercial jet aircraft being operated at any given time. In addition to aerospace applications, axial flow compressors are currently used for a number of industrial and research applications [2]. The engineering, economic, and environmental benefits of improving the efficiency of axial flow compressors gives rise to the motivation for researching control strategies for this type of fluid machinery. A necessary first step in implementing a control strategy is the development of a useful dynamic model, the primary objective of this thesis research.

For the research presented in this thesis, the concentration is on axial flow compressor systems, as used in a number of applications, including jet aircraft engines. System identification methods are primarily used for model development. The structure of this thesis is divided into seven chapters. Chapter 1 acts as an introduction with a problem statement and thesis goals as well as background information on axial flow compressor research and dynamics. Chapter 2 gives an overview of the experimental setup and the equipment used for data collection. Chapter 3 briefly discusses the theory behind the implementation of system identification. Chapter 4 and Chapter 5 provide in depth details on the use of system identification for the development of a slow and fast dynamics model, respectively. Chapter 6 summarizes the results of the system identification models and the results of the analysis of these models such as model pole trends. Lastly, Chapter

7 describes the conclusions that can be drawn from the research and future work that will be conducted on this project.

All data that is used in this research is collected from a one stage axial flow compressor. Figure 1.1 depicts a schematic of a one stage axial flow compressor system. The fluid is fed to the compressor stage through the inlet, which may include inlet guide vanes (IGV) for adjusting and guiding the flow. The flow entering from the inlet is then compressed as it passes through the rotor blade passages of the compression stage. After the compression stage, the exit duct guides the flow into the plenum. A throttle is used to manipulate the operating point of the compressor system by adjusting the flow rate of the fluid passing through the compressor system.

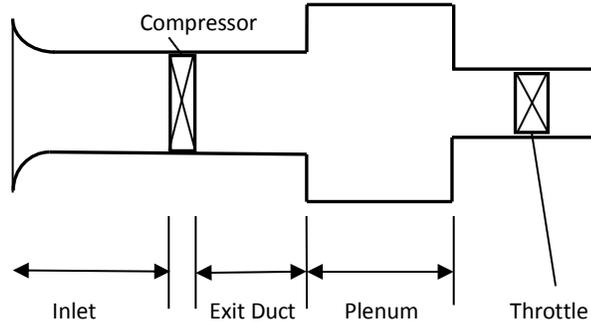


Figure 1.1: Diagram of a One Stage Axial Flow Compressor System

The dynamics of an axial compressor system are generally excited by using a throttle, IGV, or other means to alter the flow through the compressor body. An alteration in the flow causes a corresponding change in the pressure rise from the inlet atmospheric pressure to the compressor plenum pressure. The flow is usually defined using a dimensionless parameter designated as the flow coefficient or the symbol, ϕ [3].

$$\phi = \frac{C_x}{U} \quad (1.1)$$

In this definition of the flow coefficient, C_x is the axial flow velocity and U is the mean rotor velocity. Similarly, the pressure rise is defined using a dimensionless parameter designated as the pressure rise coefficient or the symbol, ψ [3].

$$\psi = \frac{\Delta P}{\frac{1}{2}\rho U^2} \quad (1.2)$$

For this definition of the pressure rise coefficient, ΔP is the change in pressure from inlet atmospheric pressure to plenum pressure, ρ is the density of the fluid passing through the compressor, and U is once again the mean rotor velocity. For this research, the fluid is assumed to be incompressible. Thus, the density ρ of the fluid is taken as a constant value for the pressure rise coefficient calculations. Most modern research into what is defined here as the slow dynamics of an axial compressor system consists of developing a model that relates the behavior of the flow coefficient ϕ to the corresponding behavior in the pressure rise coefficient ψ . The fast dynamics, as opposed to the slow dynamics, are defined as the pressure flow dynamics in and around a rotor blade passage and are also studied.

The general relation between the flow coefficient and pressure rise coefficient can be represented by plotting the characteristic curve. Figure 1.2 shows the typical operating point trajectories along a pressure versus mass flow plot for an axial compressor characteristic curve. When an instability phenomenon known as rotating stall occurs, the compressor operating point moves from position (1) into the stall position at (3). When the throttle is completely opened, the operating point of the compressor moves to position (4). The plot in Figure 1.2 also shows the stall curve that is consistent with the cubic axisymmetric compressor characteristic curve as developed in the Moore-Greitzer model [3].

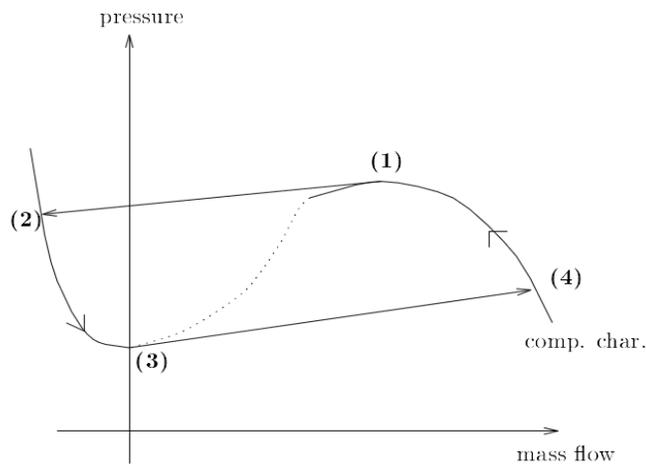


Figure 1.2: Compressor Characteristic Curve for Axial Compressor Undergoing Stall [3]

Under certain operating conditions, the compression systems that are used in aircraft turbine engines and industrial processes may develop severe aerodynamic instabilities. Compression system instabilities are generally caused by one of two phenomena known as rotating stall and surge. The type and magnitude of these instabilities are dependent on the dynamics of the compressor [4-6]. Rotating stall is a phenomenon which causes disturbances in the circumferential flow pattern of the compressor and is usually considered as a two or three dimensional phenomenon [5, 6]. This rotating stall condition is caused by a flow moving slower than the rotors (generally 30 percent to 70 percent of the rotor speed) around the compressor annulus [5-9]. Surge is a large amplitude oscillation which causes an overall reversal of the flow in the compressor [4-9]. The surge phenomenon is dependent on compressor geometry as well as the dynamic properties of the system such as inlet and outlet channels, volume, and throttle resistance [4]. Surge can also be described by fluctuations in flow, rise in pressure, and the rotational speed of the compressor [6].

In the design of axial compressors and other types of turbomachinery, it is of primary importance to consider the prevention of rotating stall and surge. During the normal operation of the system, such instabilities can generally not be endured due to the large mechanical loads imparted on the structure [4]. Rotating stall causes severe uneven loading on the compressor blades. This uneven loading can lead to extreme blade vibration, a decrease in pressure rise, decrease in compressor efficiency, overheating in the burner, and sometimes surge [5, 9, 10]. Surge is a highly undesirable phenomenon due to the severe damage it causes to the compressor as well as the rest of the flow system. Damage to a jet engine caused by compressor surge includes overstress on the compressor blades and a lack of air provided to the jet engine combustor which may lead to flame-out [5, 6, 11]. The traditional way to avoid surge is to run compressors at an operating level away from the surge line [6]. However, this limits the operating range and achievable efficiency of the machine [6]. For obtaining extreme pressure ratios in every stage of the compressor, aircraft engines these days oftentimes use transonic axial flow compressors. Engine weight and size can be reduced if high pressure ratios can be obtained from each stage of the compressor, which in turn reduces the cost of operation and investment [12].

For the reasons stated above, axial compressor systems must be operated at a safe region away from the peak of their performance curves to accommodate flow disturbances during operation. A number of different control strategies have been proposed to mitigate the instabilities and increase the efficiency without jeopardizing the operational safety of the system. The proposed control strategies to limit the instabilities are based on active control schemes, where actuation helps to safely move the operating point further up along the performance curve. In order to achieve active control, it is important to have both a working knowledge of the compressor dynamics and generally a dynamic model of the compressor system characteristics.

1.2 Problem Statement

The fluid dynamics of an axial flow compressor are complicated and in some regards not well understood. A model of some sort, whether it involves the fast or slow compressor dynamics, is necessary for developing an efficient means of controlling the stall behavior dynamics and avoiding the onset of rotating stall and surge. Rather than approaching this problem from a fluid dynamics perspective and obtaining an analytical model, a system identification approach is used. This system identification approach is based on input and output data and minimal knowledge of the compressor dynamics. However, the development of this model has some inherent complications. Research into modeling the fluid dynamics of an axial compressor is a relatively open field and consequently, new methods and combinations of existing methods must be created to analyze data for system identification.

Specific problems that must be dealt with include reducing random noise in measurement signals, creating a satisfactory input excitation, and analyzing the developed models to determine if they adequately model the system dynamics. The effects of signal noise can be reduced using techniques such as signal filtering and correlation coefficient calculations. Different excitation inputs can be applied to the system with limitation to inputs that can be physically produced at the experimental facility at the IET. Lastly, any developed model must be examined to understand whether this model can be appropriately used in a control system application. This model examination can be conducted based on some prior knowledge of the system dynamics.

If a satisfactory system identification model can be developed, the ability to then create a control system using this model would have a multitude of benefits for axial compressor technology in jet engines. For instance, a control system would allow for a compressor being throttled to the stall point to operate efficiently beyond this stall point. Operating beyond the stall point would allow for a higher pressure rise across the compressor which would in turn allow the jet aircraft engine to create a greater thrust force. Air injection actuators can be used to inject air into the tip gaps to prevent stall. By having a model to detect the approach of stall, the air jets could be actuated more efficiently to inject air for stall margin improvement, without expending unnecessary energy for air pressurization. In theory, these injectors can continuously inject air to

improve the stall margin, but an accurate model would allow a control system to determine when injection is necessary. Therefore, this research is conducted in hopes of developing a model for usage in subsequent control systems using an empirical system identification approach.

1.3 Thesis Goals

- Develop a process for creating a model to characterize the slow dynamics (dynamic relation between the flow coefficient and pressure rise coefficient) of a one stage axial flow compressor system through the use of data filtering, filter optimization, and system identification.
- Develop a model through the use of system identification to characterize the fast dynamics (dynamic flow behavior in and around the rotor blade passages) of a one stage axial flow compressor system using dynamic pressure data.
- Relate developed models to a form that can be used for active flow control for an axial compressor.

1.4 Research History of Rotating Stall and Surge in Axial Flow Compressors

Research on rotating stall behavior in axial flow compressors has been a high priority research topic since the 1950s. As axial flow compressors came to be prominently used for jet engine applications, the necessity for research into stall propagation as well as the surge phenomenon in axial flow compressors became increasingly evident. Initial research into stall behavior was conducted at the California Institute of Technology among other research institutions. This research consisted primarily of collecting experimental data and observations to link stall propagation with specific operating conditions of the axial-flow compressors, such as the pressure drop across the compressor blade rows [13, 14].

As the mechanism for the propagation of rotating stall and surge became better understood, research shifted to the development of a theoretical model for understanding stall and surge in axial flow compressors. A milestone in developing the model describing rotating stall and axial flow compressors occurred in the mid-1970s at the Massachusetts Institute of Technology. This research was conducted by E. M. Greitzer who published his work in 1976. Greitzer's research consisted of developing a theoretical system model for the axial compressor using non-dimensional parameters on which the system response is dependent. Additionally, Greitzer collected experimental results in order to compare to and validate this theoretical system model. This theoretical model, published as Parts I and II, is now known as the Greitzer model [10, 15].

A subsequent collaboration between Greitzer and F. K. Moore of Cornell University led to further research into refining the theoretical model and developing a set of partial differential equations for describing the compressor system. Moore had previously conducted extensive research on rotating stall theory with a focus on the many blade row stages of axial compressors [16]. The Moore and Greitzer collaborative research centered on the development of equations that could express the growth and dissolution of stall cells in a mass flow compressor system over time. An important aspect of this research was demonstrating how these equations were coupled for rotating stall and surge motions within the compressor. The results of this research model were published in 1986 [3]. Greitzer and Moore's collaborative work, known as the Moore-Greitzer model, proved to

be the basis for a mathematical model describing rotating stall and surge in axial flow compressor systems. This research has allowed for the development of a variety of methods for controlling and mitigating stall and surge in axial-flow compressors.

Verification and refinement of the Greitzer and Moore-Greitzer models as well as developing further models for flow perturbations in axial compressors soon became the next research priority in the late 1980s and early 1990s. Bifurcation theory was used by a number of researchers in order to determine whether the Greitzer model sufficiently models the physical phenomena of rotating stall and surge. An early research publication into bifurcation analysis is included in Brons [17]. Additionally, during this time the Moore-Greitzer model, which does not account for blade row time lags in flow dynamics, was refined to account for these time lags by Haynes et al. [18]. Other work on the modeling of rotating stall and surge was conducted by Paduano et al. [19]. Their research primarily focused on developing an analytical model based on spatial Fourier analysis of flow perturbations within axial compressors to be used for active control purposes. Research during the late 1980s and early 1990s had shown that proof of concept compressor stabilization could be achieved through the use of active control methods involving sensors and actuators by Simon et al. [20]. More practical techniques for active control were still in development.

During the mid to late 1990s, several major advances were made in active control and stabilization of surge and rotating stall in axial compressors. The research of Hendricks et al. on the inception of instability in axial compressors was useful in subsequent development of active control methods [21]. Another control research area that was developing at around this same time period was the use of air injection actuators as a means of actively stabilizing rotating stall and surge in order to increase the operating range of the compressor. This control scheme was based on obtaining data on the upstream static pressure of the flow entering the compressor and feeding this data to the air injection actuators, which in turn would be used for flow stabilization [22]. Weigl et al. [22] and D'Andrea et al. [23] present examples of the use of air injection for compressor stabilization. Alternative control methods being researched in the late 1990s are given by Humbert et al. [24], Haddad et al. [25], and Kang et al. [26]. Humbert et al. [24] put forth a simplified numerical analysis for the purpose of assessing the dynamics

and use of feedback controllers in axial compressors for stall and surge minimization. Haddad et al. [25] develop and compare a robustly stabilizing nonlinear control technique and compare this technique with existing controllers based on bifurcation stabilization. Lastly, Kang et al. [26] further develop a bifurcation stabilization method with a linear feedback control design and demonstrate the results through numerical simulation.

One of the leading approaches proposed for controlling surge and stall in the early 2000s was put forward by Paduano et al. in 2001 [4]. Their research presented two approaches for use in feedback control methods. The first approach involved developing a way of actively damping the compressor system. The second approach utilizes system manipulation in order to keep the operating point of the compressor system near to the range of instability without become unstable. In addition to Paduano's research, a variety of approaches for controlling stall and surge have been investigated and presented. More recently, methods have been researched to actively control surge of the compression system using compressor drive torque rather than mass flow measurements by Bohagen and Gravdahl [27]. Several other research publications that deal with the development and study of control systems for controlling stall and surge are given in Nayfeh and Eyad [28], Chaturvedi and Bhat [29], and Uddin and Gravdahl [6]. Nayfeh and Eyad [28] use a feedback control method to prevent stall by maintaining a steady-state pressure rise within the compressor. Similarly, Chaturvedi and Bhat [29] make use of an output-feedback control for the prevention of hysteresis and surge in axial compressors. Uddin and Gravdahl [6] present a linear as well as a nonlinear method of feedback control that actuates pistons to control the system and stabilize surge. Feedback is obtained from values such as the pressure within the plenum and piston velocity.

In addition to control and stabilization research, recent research by Sun et al. [30] also encompasses the determination of stall inception points and the onset of instability. Techniques are being developed by Tahara et al. [31] for creating warning signals before a spike in the stall inception occurs to aid in control methods. Furthermore, to better understand the effects of the aforementioned control approaches, high performance computing and computational fluid dynamics (CFD) are now being used to simulate fluid flow through compressor systems. These computational simulations can then be used to study the effects of different types of control methods and understand their effect on stall

and surge prevention and mitigation. Chen et al. [32] document an example of research using high performance computing codes for flow simulation with the implementation of a tip injection control system. CFD models have been increasingly used in the past few years for the determination of stability across blade rows within compressors as seen in Chima [33] and Cornelius et al. [34].

1.5 Introduction to Axial Compressor Slow Dynamics

1.5.1 Research Background of Slow Dynamics and Rotating Stall

An early model of rotating stall dynamics was developed by Emmons et al. [35] in 1955 to explain the mechanism that induces rotating stall. The model included a study of the separation and blockage that occurs between blades and how the incidence of the incoming air affects these properties. Subsequently, the definitive Moore-Greitzer theory model was developed for two dimensional incompressible flow in 1986 [3]. Two dimensional flow is a flow in which the velocity directions at all points within the flow are parallel to a specified plane. The Moore-Greitzer model also included circumferential short-length disturbances. Long-length scale disturbances and perturbation were later introduced to the model by McDougall et al. [36] and Garnier et al. [37]. Long-length scale perturbations are flow disturbances of low frequency and low amplitude.

In 1994, Paduano et al. [19] developed an analytical state-space model for the system and conducted research where the blade stagger was modified. More recent research began to focus on active control for the purpose of increasing the stall margin. For instance, Weigl et al. [22] enhanced the stall margin using actuators which inject air axially into the gap spacing between blade tips and casing. Specifically, they excited the compressor system using sinusoidal actuated air injection. The sinusoidal air injection was superimposed on a constant air injection of 3.6 percent of the main air flow that was also being injected.

Although accepted as the definitive model for axial compressor slow dynamics, the Moore-Greitzer model has several limitations that are addressed in this section. These limitations include the fact that the model does not take into account phenomena such as tip leakage and rotor-stator flow interaction. Additionally, the Moore-Greitzer model only accounts for modal oscillation stall inception and does not include spike detection for stall warning. In addition, there are other more general limitations and restrictions to developing active flow control based on the Moore-Greitzer model for an axial compressor. These limitations include the fact that the Moore-Greitzer model is complex and requires a thorough understanding of the mechanisms involved to achieve active control. A high frequency actuator is also a necessity for control of the overall system in

order to prevent stall inception. Several other improvements and additions to the Moore-Greitzer model and the research conducted to address these improvements are discussed in the following paragraphs.

One addition to the Moore-Greitzer model includes the use of traveling wave energy to provide a warning that precedes stall inception. Traveling wave energy can be used as a measure of compressor stability according to the research of Tryfonidis et al. [38]. This research uses a spatial Fourier transform to analyze the dynamics of prestall traveling wave structures. The circumferential traveling wave energy is used to predict stall precursors based on the propagation velocity of the previous order spatial harmonic. Figure 1.3 [38] shows the relation between traveling wave energy and time in rotor revolutions prior to stall, as determined by Tryfonidis et al.

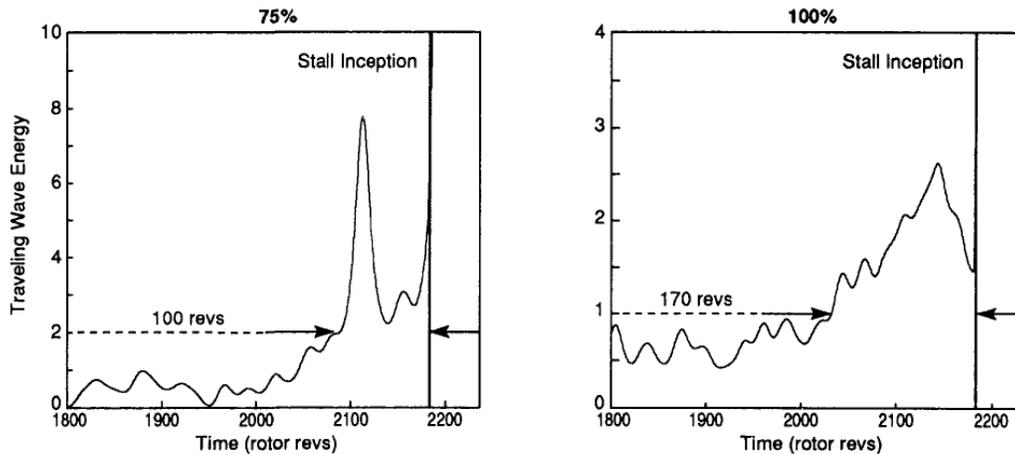


Figure 1.3: Traveling Wave Energy versus Time in Rotor Revolutions near Stall Inception [38]

The data in Figure 1.3 is for a test compressor operating at 75 percent of the maximum rotor speed on the left and at 100 percent of the rotor speed on the right. As can be clearly seen in Figure 1.3, prior to stall inception there are significant increases in the magnitude of the traveling wave energy. These types of traveling wave energy plots allow for the observation and detection of small amplitude perturbations before stall occurs.

The Moore-Greitzer model also tends to use a lumped representation approach and consequently ignores the interaction between certain components within the compressor. For instance, the effect of the interaction between individual blade-rows is not considered. According to Spakovszky [39], these interactions become important to

the overall system dynamics for compressors with significant dynamic contributions from the inter-blade-row ducts. These interactions were addressed by Spakovszky specifically for centrifugal compressors and were used to improve the model and explain the mechanism for backward traveling rotating stall waves.

Additionally, short length scale disturbances were not considered in the Moore-Greitzer model. Studies of these short length scale disturbances found that these types of disturbances have an effect on stall precursors and the inception of stall. Day [40] found that the spreading of these small scale disturbances often leads to fully developed stall in the compressor system. Day's experimental research also concluded that modal oscillation and stall cell formation (also known as spike stall inception) were two physically separate events. More specifically, modal oscillation is generally an axisymmetric circumferential disturbance, whereas spike inception is a localized disturbance that leads to the destruction of the symmetry of the flow field. Camp and Day [41] found that the inception of stall through spike and modal oscillation has a strong correlation to the critical incidence of the rotor blades. Figure 1.4 [41] demonstrates the relative location of the stall point for spike stall inception and modal oscillation stall inception events on the pressure rise coefficient versus flow coefficient characteristic curve.

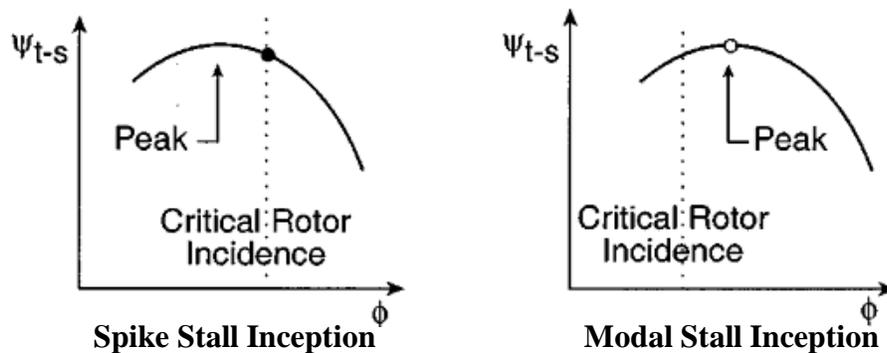


Figure 1.4: Relative Locations of Spike and Modal Stall Inception on the Characteristic Curve [41]

As Figure 1.4 shows, stall tends to occur prior to the characteristic curve peak, when approaching the stall point, for spike stall inception. Conversely, stall tends to occur very near to or at the peak for modal oscillation stall inception.

1.5.2 Slow Dynamics Modeling for Compressor Systems

A variety of methods have been developed for modeling the slow dynamics of compressor systems. Several of these methods, including a grey-box approach as well as a state-space model, are described in this section. A grey-box approach developed by Greitzer [42] is described in the following paragraphs.

This grey-box approach makes use of a mass-spring-damper system model developed and adapted by Greitzer, among others, to represent the compressor system including the plenum and throttle components [42]. In this model, a mass block represents the mass of the air in the compressor duct. Similarly, another mass block is used to represent the air mass in the throttle. A spring can be used to model the interaction between the two masses that is proportional to the compressibility of the air in the plenum. A damping factor can be added to the mass representing the energy being added to the system from the compressor rotation. This damping factor is negative because energy is being added to the system rather than dissipated. Additionally, another damping factor can be added to the air mass in the throttle.

Figure 1.5 [42] below shows the compressor or pumping system setup as well as the mass-spring-damper model representation for the grey-box approach.

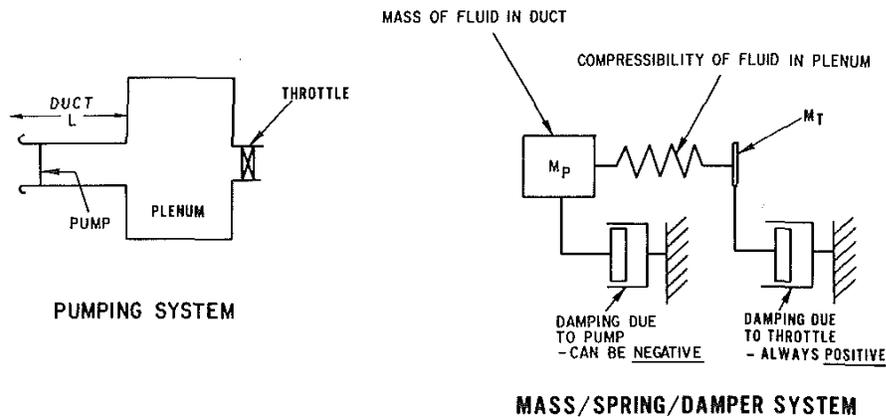


Figure 1.5: Mass-Spring-Damper System for Grey-Box Modeling Approach [42]

In Figure 1.5, M_p is the mass of the air in the pump or compressor and M_t is the mass of the air in the throttle. The mass of the air in the throttle is assumed to be negligible and is consequently approximated as zero. This grey-box model is an idealized system and

consequently a more detailed model is necessary for accounting for certain phenomena in the slow dynamics data.

In addition to grey-box modeling, comprehensive state-space models have also been developed to model the slow dynamics of compressor systems. Paduano et al. [19] developed an analytic state-space model based on the Moore-Greitzer model to describe the dynamic behavior of the axial compressor system. The experiments for this research were conducted using inlet guide vanes in order to excite the system dynamics. 12 inlet guide vanes (IGVs) were positioned upstream of the compressor rotor at angular locations around the compressor annulus and were actuated independently using DC motors. Although there are several alternative methods for affecting the fluid flow through a compressor, IGVs were chosen for the ease with which a variable input for modeling could be created, as well as their ability to be actuated using relatively inexpensive off-the-shelf hardware. The bandwidth of actuation was 80 Hz, which is significantly higher than the process bandwidth at approximately 10 to 14 Hz. These actuated IGVs synthesized a traveling wave to add an exogenous input to the system with a gain that must be quantified. This gain quantification is conducted using system identification. In these experiments, the blade stagger was also modified to determine its effect on the model.

In the research of Paduano et al. [19], the state-space models were converted into a transfer function form. The conversion of the state-space model to a transfer function form was done to put the model in a form more suitable for frequency domain analysis. These transfer function models were subsequently used to study the frequency domain behavior of the state-space model that was originally obtained. Sinusoid inputs as well as a pseudo-random binary input were both used to excite the transfer function modeled system. The sinusoid inputs were used to excite the transfer function models in order to determine the values of the transfer functions for a specified excitation frequency. Figure 1.6 [19] shows the Bode plots (magnitude and phase) of a transfer function model with a spatial sinusoidal wave with oscillating amplitude.

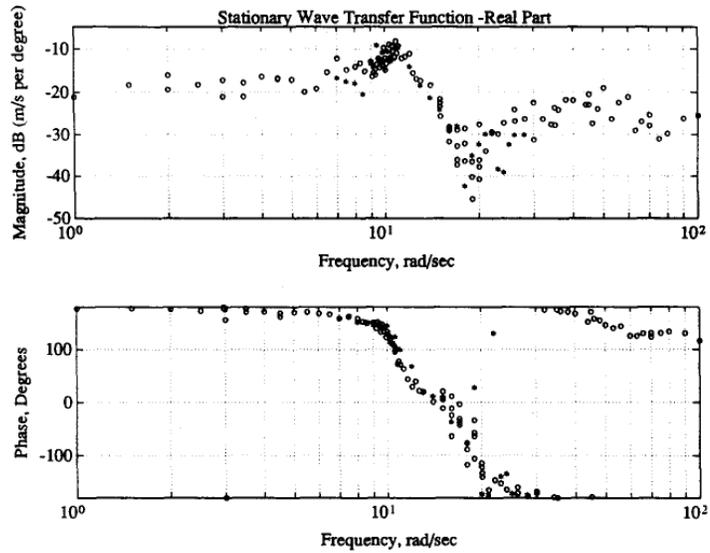


Figure 1.6: Bode Plots (Magnitude and Phase) for Transfer Function Model with Spatial Sine Wave Input [19]

Further research into developing an analytical model for the slow dynamics was carried out by Feulner et al. [43]. This research yielded an analytical model, based on the Moore-Greitzer model, in a form that was more suitable for implementing a control system involving an input and output in the Laplace domain. The experimental compressor system used jet actuators to perturb the system. The Laplace domain model was also found to be an accurate representation of the system behavior near the instability region. Additional research was conducted by Frechette in reference [44].

1.6 Introduction to Axial Compressor Fast Dynamics

1.6.1 Research Background of Fast Dynamics

As opposed to the slow dynamics which relate the overall compressor pressure rise coefficient to the flow coefficient, the fast dynamics of an axial compressor system are focused on the dynamic behavior of the fluid flow at and around the rotor blades and through the blade passages. Very little research has been conducted using system identification to characterize the behavior of an axial compressor's fast dynamics. Consequently, the prospect of using system identification to model the fast dynamics of an axial compressor is a relatively open area of study. The details of fast dynamic behavior for axial compressor flow are explained in the following paragraphs.

As stall is approached, tip leakage flow (TLF) begins to have a significant influence on the behavior of the main incoming flow through the blade passages. The research of Khalid et al. [45] and Vo [46] demonstrates that the flow within these blade passages can be characterized by an interface line created between the TLF and the main incoming flow. Figure 1.7 shows a graphical illustration of how TLF or tip clearance flow interacts with the incoming flow through the blade passage to establish a distinct interface line [46].

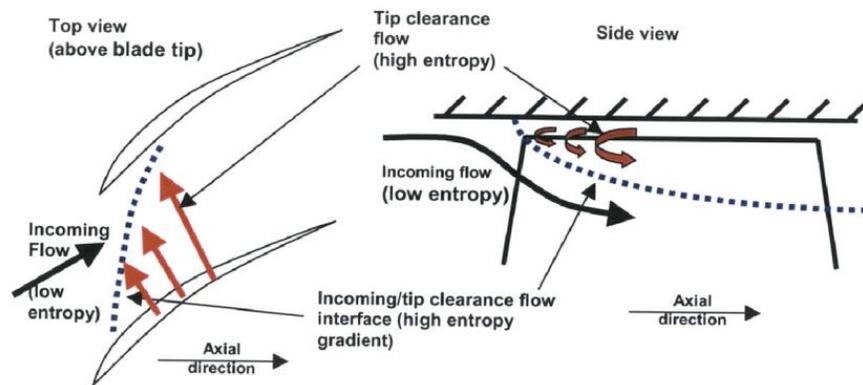


Figure 1.7: Establishment of the Interface Line for Flow in a Blade Passage [46]

This interface line generally extends from the low pressure side of the rotor blade near the leading edge. Although there is an alternative hypothesis for the occurrence of stall, this interface line is hypothesized to be linked to the onset of stall and is assumed to be true for the development of a model in the research described in this document.

A detailed analysis into the relation between the momentum ratio, tip gap, and how these factors affect the route to stall was conducted by Lin et al. [47]. A graphical depiction of the routes to stall for an axial flow compressor is shown in Figure 1.8 [47]. As the compressor is throttled, the momentum ratio, represented by the vertical axis, increases. Figure 1.8 also shows that the tip gap between the rotor blade tip and the casing has a significant effect on how an axial compressor approaches the stall point. For a small enough tip gap, separation of blockage causes the compressor to stall as the compressor flow is throttled, without entering a stable yet unsteady flow region. However, for a tip gap larger than the empirically determined critical tip gap, the TLF has a much larger effect and can cause unsteady compressor flow as the compressor is throttled.

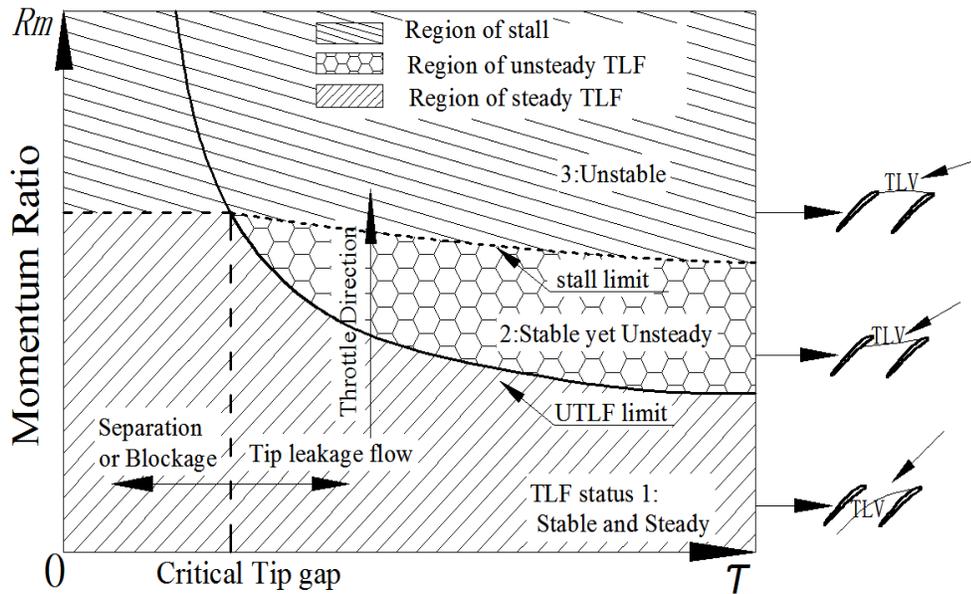


Figure 1.8: Routes to Stall in Axial Compressor [47]

For an axial compressor with a larger than critical tip gap in which TLF is present, the flow interface line undergoes three status regions as stall is approached through flow throttling. The first region is where the flow is stable and steady. In this region of Figure 1.8, the TLF is steady and the interface line does not reach the high pressure side of the adjacent rotor blade. The second region is where the flow is stable yet unsteady. This region of Figure 1.8 is the region of unsteady tip leakage flow (UTLF) and is reached

when the interface line begins to make contact with the high pressure side of the adjacent rotor blade. The final region in the throttle direction is the region of stall. As the stall region is arrived at, the interface line has reached the leading edge of the high pressure side of the adjacent rotor blade and blockages begin to form in the blade passage.

1.6.2 Correlation Coefficient for Fast Dynamics Analysis

A well-established method of analysis for understanding the fast dynamics of an axial compressor is the calculation of the correlation coefficient of the output pressure data across the flow passing through the rotor blade passage. The autocorrelation coefficient is essentially a way of measuring the repeatability of the pressure at a stationary location along the rotor blade and reducing the effect of random noise in the pressure signal measurement. The research of Tahara et al. in [48] and [31], first developed the use of the calculation of the autocorrelation coefficient for usage in stall warning signals for active stall control techniques. Additionally, this research found the mid-chord location on the rotor to approximately be the location of the initial correlation degradation as stall is approached.

Since then, other researchers have also used the correlation coefficient method and further refined this method. Dhingra et al. [49] uses a stochastic model to better understand the stochastic behavior of the correlation coefficient or correlation measure as applied to the management of compressor stability. A primary conclusion of the research of Dhingra et al. was that the dynamic pressure sensor location relative to the rotor has an important effect on the usefulness of the technique. They found that a pressure sensor placed at the mid-chord of the rotor produced the best results for their technique. The work of Christensen et al. [50] builds on the method developed by Dhingra et al. and further expands the method. Christensen et al. developed a robust real-time algorithm for monitoring compressor stability. The results of this research showed that the correlation measure can be used in a control system for effectively managing stall using existing control software.

A more recent use of the correlation coefficient in active stall techniques for stall margin improvement is in the research conducted at the CAS by Li et al. [51]. This work makes use of the cross-correlation coefficient as a stall warning and uses air injection actuators to inject air into the tip gap in order to delay stall. The control scheme involved

a proportional control valve on the air injection jets to proportionally increase air injection as the stall point is approached. The compressor characteristic line as compared with the no injection line as well as a constant steady injection line is shown below in Figure 1.9 [51].

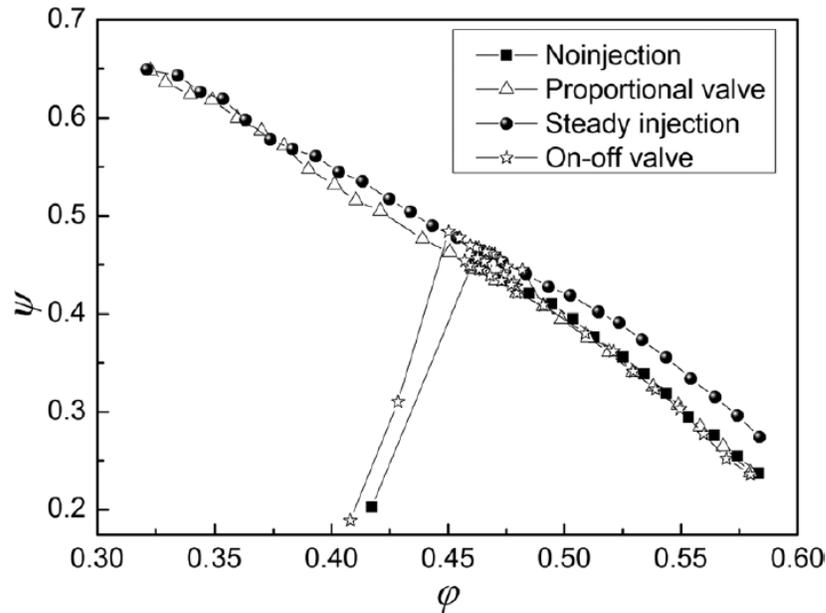


Figure 1.9: Compressor Characteristic Lines for Several Injection Methods [51]

The horizontal axis in Figure 1.9 represents the flow coefficient and the horizontal axis represents the pressure rise coefficient. As can be seen in the above figure, the proportional valve air injection provides a significant stall margin improvement. The research provided in references [31, 48-51] clearly show that usage of the correlation coefficient is a practical means of analyzing and examining the dynamic pressure sensor data across the rotor blades for use in active stability control methods for axial compressors.

1.7 Relation of Slow Dynamics to Fast Dynamics Overview

It is believed that there is a direct link between the slow and fast dynamics of the compressor system. In addition to modeling the behavior of the slow dynamics, observing and linking the behavior of the slow dynamics to the behavior of the fast dynamics is also of key importance. A future objective of this research is to use system identification to obtain a black-box model of the compressor slow dynamics that can be linked through a disturbance variable to the fast dynamics. The relation through a disturbance variable between the compressor's slow and fast dynamics is explained in this section.

There are four cases for the flow through the compressor. These cases include the stable case, unsteady case, the pre-unstable case where spike-stall inception occurs, and the fully unstable case. The dynamics of the system can be represented by the A matrix of a state-space equation, where the elements represent the fast and slow dynamics, as well as how the dynamics and disturbances interact. This dynamic matrix allows for the determination of a system output based on the varying of the slow dynamics (using an oscillating throttle ring) and the varying of the fast dynamics (using air injection actuators). The proposed model structure of the system is as follows,

$$x(k+1) = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} x(k) + Bu(k) \quad (1.3)$$

$$y(k) = Cx(k) + Du(k) \quad (1.4)$$

where $x(k)$ is the vector containing the state variables or set of system parameters and $y(k)$ is the output. $u(k)$ is the input to the system of which there are two types available, throttle oscillation and air injection. The A matrix that is of interest for relating the fast and slow dynamics of the system is of the form:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad (1.5)$$

where the elements of this matrix are each a distinct matrix. Specifically, A_{11} represents the slow dynamics, A_{22} represents the fast dynamics, and A_{12} as well as A_{21} represent the disturbance and link connection between the slow and fast dynamics. B and D are also matrices and may need to be similarly partitioned depending on the selected inputs to the

system. Additionally, C is the observation matrix and can be chosen based on the variables desired for observation and analysis.

In other words, the elements of the matrix A are each matrices that contain the six variables for defining the slow dynamics, pressure disturbance, and fast dynamics. ψ and ϕ are the non-dimensional pressure rise and flow coefficients, respectively, that define the slow dynamics of the system. ψ is the pressure disturbance coefficient. Additionally, θ , ω_s , and s are the variables that define the fast dynamics of the compressor. θ is the angle that the flow streams make with the chord line of the compressor rotor blades. ω_s is the signature frequency of the oscillation movement of the angle of the flow streams θ relative to the compressor blades. Lastly, s represents the distance from the tip of the rotor blade to the incidence point of the flow stream with the blade. Li et al. [52] make use of this s variable in their research. For the stable flow case, the dynamic matrix can be written as:

$$\begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix} \quad (1.6)$$

where A_{12} and A_{21} are the disturbances and are equal to zero for the stable case. For the unsteady case, the matrix contains all of the elements and is written in the following form.

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad (1.7)$$

In the unsteady case, the elements of the matrix are equal to the matrices shown below.

$$A_{11} = \begin{Bmatrix} \psi(t) \\ \phi(t) \\ \psi(t) \\ \theta(t) \\ s(t) \\ \omega_s(t) \end{Bmatrix} \quad A_{12}, A_{21} = \begin{Bmatrix} \psi(t) \\ \phi(\theta, t) \\ \psi(\theta, t) \\ \theta(t) \\ s(t) \\ 0 \end{Bmatrix} \quad A_{22} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ \theta(t) \\ s(t) \\ \omega_s(t) \end{Bmatrix} \quad (1.8)$$

The matrices for the pre-unstable and unstable matrices also contain all of the A elements, but the values of the variables contained in the element vectors have to be

determined. Lastly, the output of the observation equation $y(k)$ contains each of the variables and is of the following form.

$$y(k) = [\psi \quad \phi \quad \psi \quad \theta \quad s \quad \omega_s]^T \quad (1.9)$$

The primary challenge in using this model structure is the determination of the pressure disturbance variable that links the fast and slow dynamics. Two methods are being considered for determining the disturbance variable ψ . A cross-correlation method could be used and additionally a traveling wave energy method could also be used for determining this variable. The traveling wave energy method is used by Tryfonidis et al. [38] and utilizes spatial Fourier transforms to predict stall precursors based on the propagation velocity of previous order harmonics. In order to implement these methods, a slow dynamics model as well as a fast dynamics model must first be obtained.

CHAPTER 2 EXPERIMENTAL SETUP AND DATA

COLLECTION

2.1 Axial Flow Compressor Specifications

Before analysis and modeling can be performed, data must first be collected. The experimental compressor system that is used to obtain data for this research is a one stage axial flow compressor situated at the Advanced Energy and Power Technology Laboratory at the Institute of Engineering Thermophysics (IET), Chinese Academy of Sciences (CAS), located in Beijing, China. The one stage rotor touts a total of 58 blades spaced evenly around the rotor at a fixed angle of attack. A photograph of the axial compressor system at IET is shown in Figure 2.1. Figure 2.2 shows the axial compressor when the casing is removed to expose the rotor blades. The operating rotation frequency of the compressor's rotor is approximately 40 Hz or 2400 revolutions per minute. This research compressor also has several casings that can be switched out in order to alter the tip clearance around the compressor blades. The parameters for the experimental axial compressor are summarized in Table 2.1, below.

Table 2.1: Parameters for Experimental Axial Compressor at the IET, CAS

Parameter	Numeric Value	Units
Design Speed	2400	rpm
Rotor Blade Number	58	-
Outer Casing Diameter	500	mm
Mass Flow Rate	2.9	kg/s
Rotor Tip Chord	36.3	mm
Rotor Tip Stagger Angle	39.2	deg
Hub-tip ratio	0.75	-

The compressor casings that are used for this research are smooth, meaning the casings do not have any type of grooves machined into the inner side of the casing. The typical tip clearance for most tests that have been conducted at IET is around 0.7 mm with the smallest diameter casing able to produce the minimum tip clearance of 0.5 mm. The casing can be switched out in a matter of minutes, which allows for tests to be conducted at a variety of tip clearance values. Although stall is believed to be heavily influenced by the tip clearance, several other factors also affect the occurrence of stall. These factors include flow separation, the inlet velocity of the air flow, the angle of attack of the compressor blades relative to the incoming air flow, as well as the tip

incidence of the blades. Modal inception oftentimes occurs due to the blade angle of attack. However, the research compressor at IET does not allow for the variation of the blade angle of attack and thus the blade angle is not altered in this study. Instead, the research compressor at IET follows spike inception as a precursor to fully developed rotating stall. Therefore, the focus of the research described in this document is focused on spike stall inception rather than modal inception.



Figure 2.1: Experimental Axial Compressor Setup at the Advanced Energy and Technology Laboratory at the Institute of Engineering Thermophysics, CAS

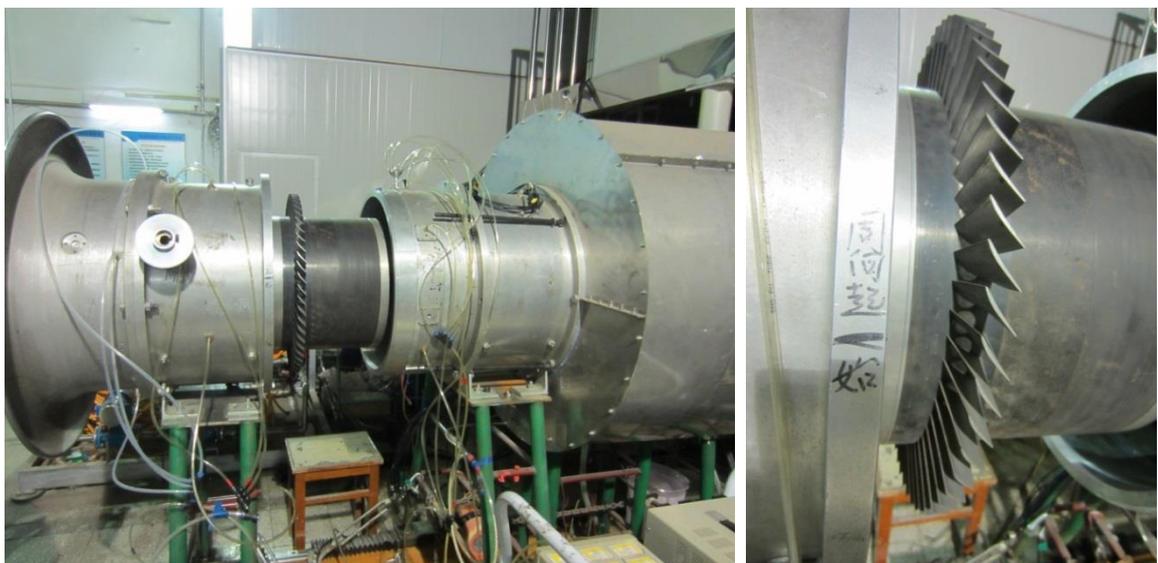


Figure 2.2: Compressor with Casing Removed (Left) and Compressor Rotor Blades (Right)

2.2 Compressor Throttle Actuation

The throttle valve for the experimental compressor setup at the IET consists of two primary pieces. The first of these pieces is the stationary throttle cone. The second piece is the mobile throttle ring. Currently, a stepper motor is used for the actuation of the throttle ring. This stepper motor is capable of moving the throttle ring at a constant speed of 2 mm/sec and producing a frequency oscillation of up to approximately 2 Hz. While this stepper motor is adequate for performing experiments in which a low frequency air flow oscillation is sufficient, this stepper motor cannot perform the wide input frequency range necessary for more useful system identification. The overall setup for the cone throttle valve is shown below in Figure 2.3. Note that the blue motor shown in the left of the picture is used for the compressor blade rotation and not the throttle actuation.

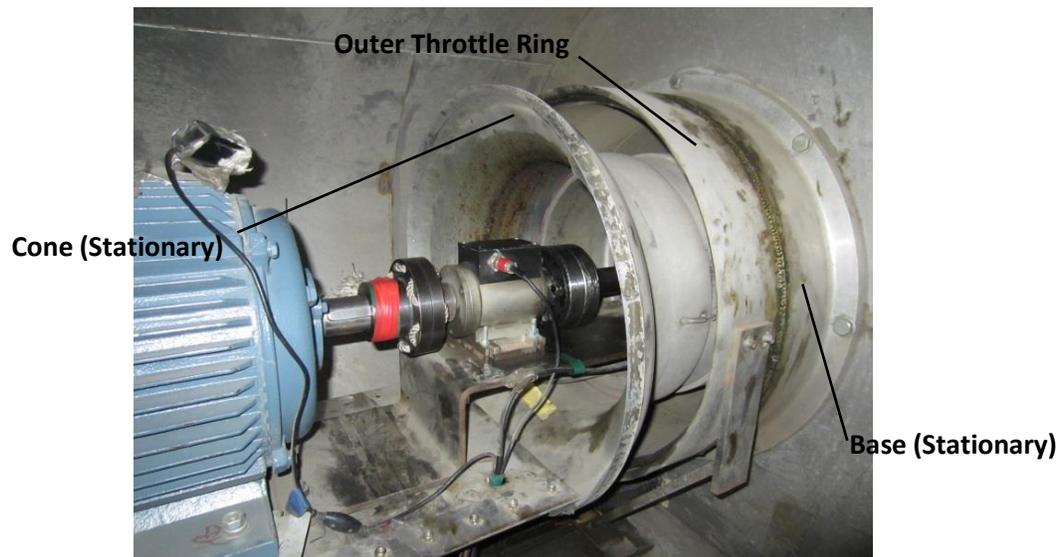


Figure 2.3 : Cone and Ring Throttle Valve Setup

The moveable ring of the throttle is made from aluminum, and is 10 mm in thickness. The outer diameter of this ring is 540 mm and the width of the ring is 118 mm. The edge of the ring is chamfered on the side that is in direct contact with the flow.

In order to move this throttle ring, a screw mechanism is used for movement in the forward and backward directions. The throttle ring is bolted to a support structure that moves along a rail. This structure is directly connected to the actuation screw. The stepper motor rotates this screw and causes the throttle ring and support structure to move along the rail. The clockwise or counter-clockwise rotation of this screw determines

whether the throttle outer ring moves forward or backward. The left side of Figure 2.4 below shows this screw mechanism as seen from the side of the compressor when a section of the casing is removed. Additionally, the right side of Figure 2.4 shows a more frontal view of the same screw mechanism.



Figure 2.4: Screw Mechanism for Throttle Actuation Side View (Left) and Frontal View (Right)

The stepper motor that is used to actuate the screw mechanism for throttle movement is mounted on the exterior of the compressor setup. A small hole was created in the compressor exterior through which the stepper motor could be mounted. A sealing adhesive was used to attach and mount the motor in position. A photograph of the stepper motor used to actuate the throttle ring is pictured in Figure 2.5. The torque shaft from the motor is on the interior of the compressor and attaches to the screw for rotation. A power cord is attached to the exterior side of the stepper motor and is plugged into a power source. The laboratory has outlets for 220 V as well as variable DC power supplies that are being used for purposes such as powering the dynamic pressure sensors.



Figure 2.5: Stepper Motor for Throttle Ring Actuation (Mounted on Compressor Exterior)

2.3 Design of High-Frequency Throttle Modules

Through research into the development of a slow dynamics model, it is found that high frequency throttle actuation may be required to adequately excite the system to obtain a useful dynamic model. From the pole variation investigation in section 6.1.2, it is concluded that the slow dynamics models are unable to adequately capture the signature or natural frequency and show accurate trends as the stall point is approached. The natural frequency of the axial compressor system at the IET is found during the occurrence of rotating stall to be between 16 and 17 Hz. Prior to stall, this frequency may be in the frequency band range of approximately 13 to 19 Hz. Therefore, an actuator that can excite the system above this natural frequency should be able to excite this frequency as well as several modes of this frequency. The slow dynamics data from the system can then be collected and used to create a dynamic model that includes this natural frequency.

The throttle modules are mounted on the outer throttle ring (see Figure 2.3). The outer throttle ring is mobile and is used to control the flow coefficient set point. Rather than oscillating the outer throttle ring, which is only capable of reaching a frequency of approximately 2 Hz, the throttle modules are oscillated using a rack and pinion gear setup actuated using an Ultimag 4EM high frequency rotary solenoid. The solenoid is mounted on a bracket, which is attached to the outer throttle ring. A pinion gear is mounted on the rotary shaft of the solenoid. This pinion gear interfaces with a gear rack that is directly attached to the throttle module. The individual throttle modules are separated by dividers or T-braces. Figure 2.6 and Figure 2.7 demonstrate the throttle module design. In these figures, the bracket is highlighted in green, the throttle modules are highlighted in blue, and the T-braces are highlighted in red. The black arrow in Figure 2.6 represents the oscillatory motion of the throttle module.

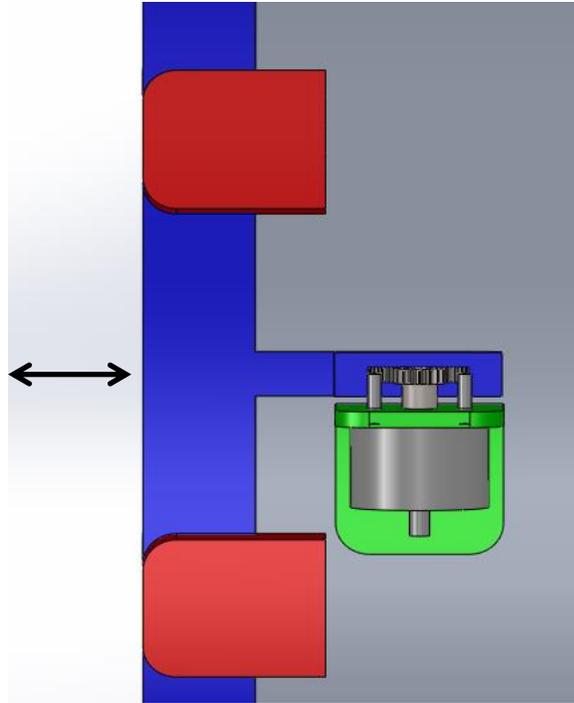


Figure 2.6: Throttle Module Design (Top View)

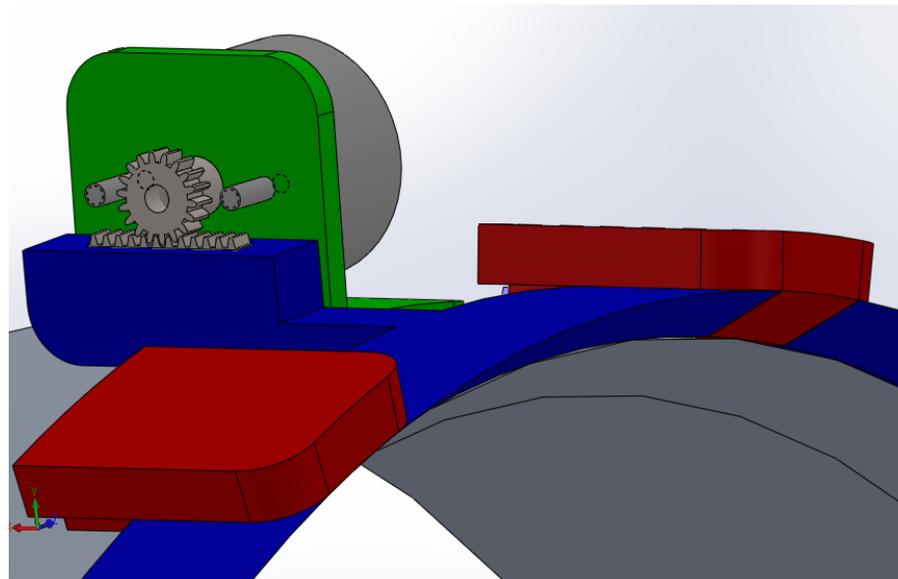


Figure 2.7: Throttle Module Design (Demonstrative View)

The modules are placed side to side around the outer throttle ring annulus. The modules and T-braces are sized so that 12 throttle modules can fit around the annulus. The chamfer angle along the edge of the throttle module is at the same angle as along the edge of the outer throttle ring or approximately 29.05° . The throttle modules are intentionally designed to be low mass in order to reduce the load on the rotary solenoid. See Figure 2.8 for a design sketch of the throttle module with dimensions included in units of millimeters. The speed of the Ultimag 4EM rotary solenoid is greater than 78 Hz with no load applied. However, this speed can decrease depending on a number of factors such as the mass of the throttle module, the coefficient of friction between the throttle module and outer throttle ring, etc. Additionally, the solenoid net starting torque is equal to 2.8 lbF-in and the stroke angle is equal to $\pm 22.5^\circ$. These solenoids are controlled using an H-Bridge setup with a microcontroller.

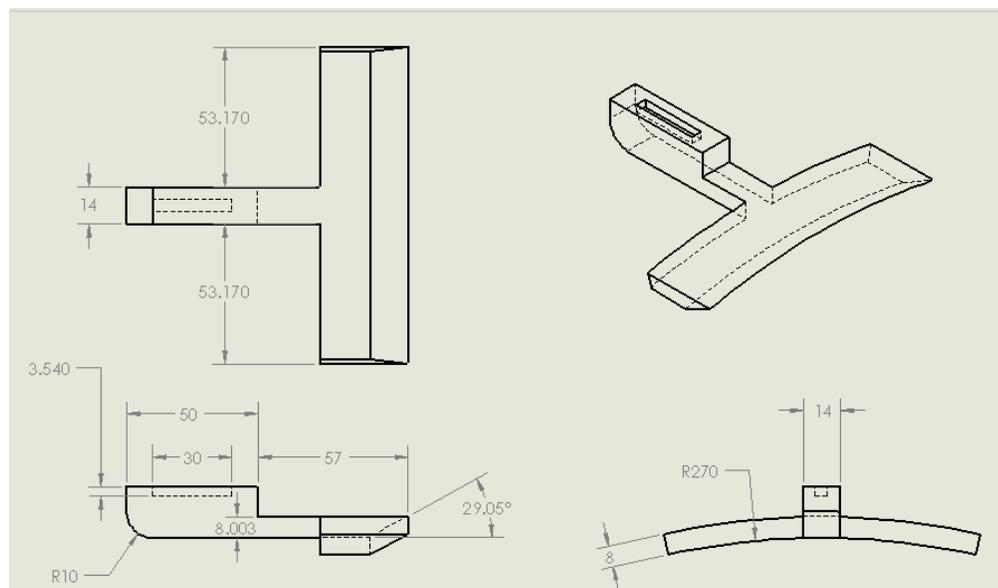


Figure 2.8: Throttle Module Design Sketch with Dimensions in Millimeters

The pinion gears have 18 teeth, a diametral pitch of 32 (0.562 inch or 14.275 mm pitch diameter), and a pressure angle of 14.5° . The racks interface with the pinion gears and therefore also have a diametral pitch of 32 and a pressure angle of 14.5° . With these dimensions and pitch diameters, the throttle modules can oscillate with a maximum amplitude of approximately 2.8 mm. The gears as well as the throttle modules are manufactured from a low density material in order to minimize the load on the solenoid.

The primary candidate materials for this application are aluminum as well as aluminum alloys. Anodizing the surface of the aluminum material for the throttle module has also been proposed to reduce corrosion and wear along the bottom surface of the module during oscillation. The throttle module design parts are currently being manufactured at the machine shop located at the IET.

2.4 Fast Dynamics Data Collection

There are ten dynamic pressure sensors, manufactured by Kulite, located in positions that span the length of a rotor blade along the longitudinal axis of the compressor. The sampling frequency for each individual dynamic pressure sensor is 20 kHz. These sensors are designated CH1 through CH10, with CH1 being the leading sensor ahead of the blade leading edge and CH10 being the trailing sensor just behind the blade trailing edge. Figure 2.9 shows the ten pressure sensors from two side angles to better illustrate the sensor positioning on the compressor casing. For experiments discussed in this thesis, only sensors CH1 through CH9 are used. The calibration factors between the sensor voltage signal and the pressure in Pascals (Pa) are included in Table 2.2. Additional pressure sensors are included within the air injection actuators to monitor the air flow into the tip gap.

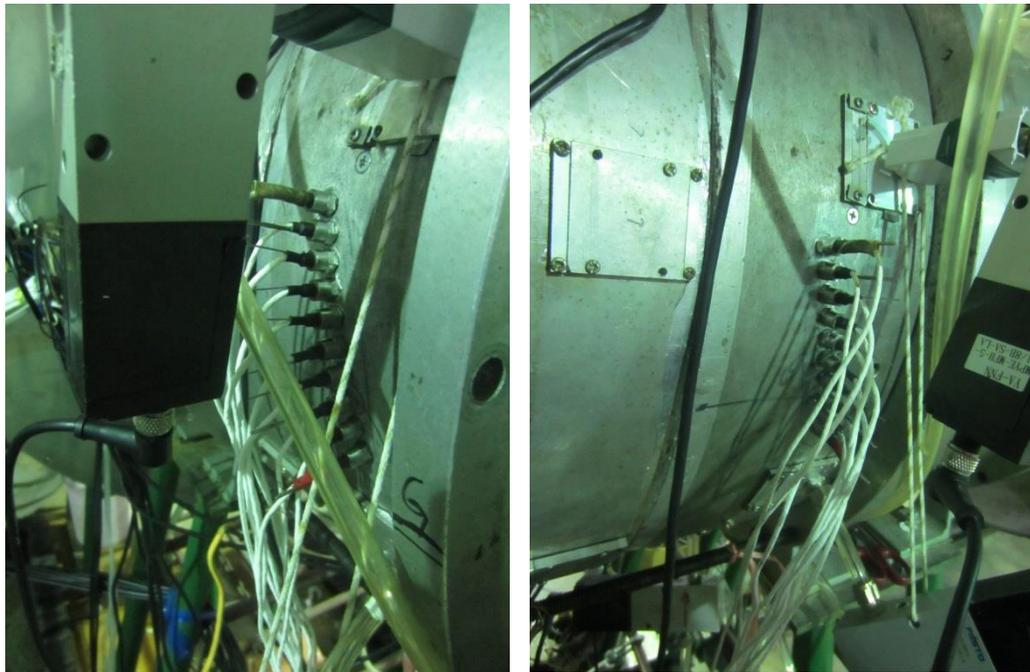


Figure 2.9: Ten Kulite Type Pressure Sensors for Fast Dynamics Data Collection (Both Sides)

Table 2.2: Fast Dynamics Pressure Sensor Calibration Factors

Sensor Number	Sensor Calibration Factor (Pa/V)
Sensor CH1 (Leading)	968.3
Sensor CH2	879.9
Sensor CH3	891.9
Sensor CH4	603.7
Sensor CH5	599.2
Sensor CH6	599.5
Sensor CH7	596.8
Sensor CH8	600.0
Sensor CH9 (Trailing)	917.7

In addition to fast dynamics pressure data collection, the experimental compressor also has the ability to alter the tip leakage flow near the compressor blades. In order to alter the flow through the system, eight air injection actuators are used. These air injection actuators are equally spaced around the compressor casing and are capable of micro-injection or macro-injection depending on the desired experiment. These actuators simultaneously inject air through jets into the tip gap between the top of the rotor blade and the casing. The angle of the jets can be adjusted to angles such as 15° and 45° , in order to alter the angle at which the air is injected into the tip gap. Figure 2.10 shows two different views of several of the air injection actuators. As seen in the figure, a tube is used to carry pressurized air into the actuator from a nearby pressurized storage vessel.

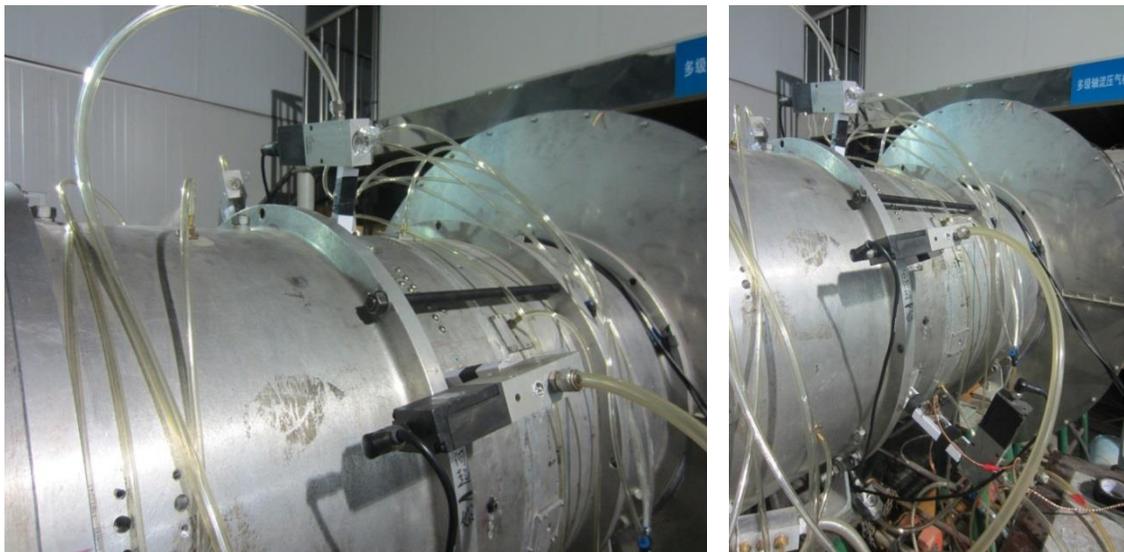


Figure 2.10: Two Views of Several of the Air Injection Actuators

For the purpose of phase locking and average rotor speed calculations, a magnet and Hall effect sensor combination is used. The magnet is embedded within the rotor blade shaft upstream of the rotor. At each revolution of the rotor, the magnetic field of this magnet can be detected by the Hall effect sensor. Figure 2.11 shows the Hall effect sensor voltage signal versus time for approximately 10 rotor revolutions. The signal fluctuates between a voltage value of 5 V and a value of approximately 0 V. When the signal is near 0 V, this indicates that the rotor has completed a revolution and returned to a specific position relative to the dynamic pressure sensors. Data collected when the rotor is at or near this location can subsequently be used for system identification. For the sake of data consistency, the rest of the data between revolutions may have to be interpolated or ignored when performing system identification.

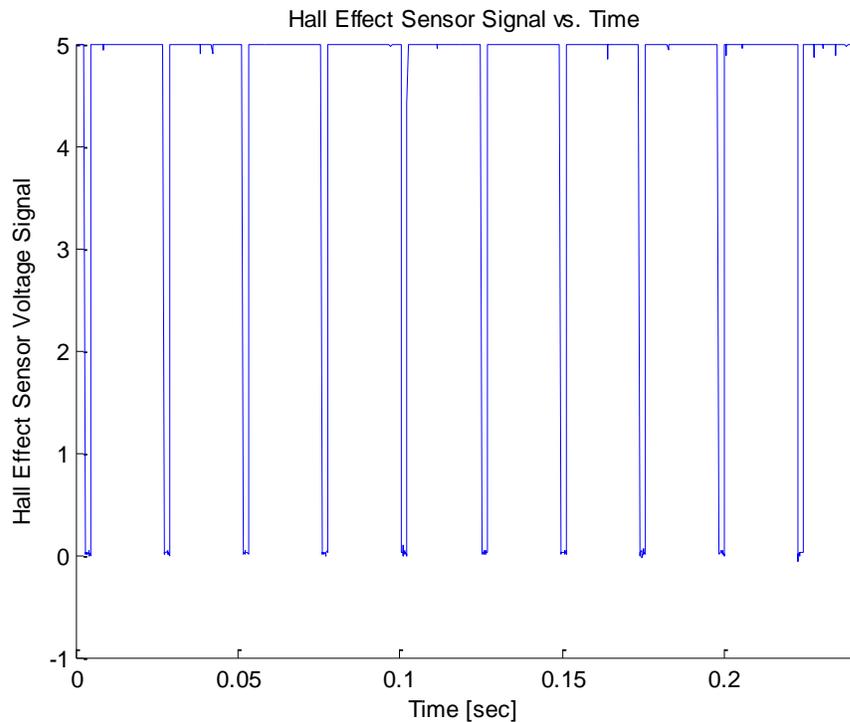


Figure 2.11: Hall Effect Sensor Voltage Signal versus Time for 10 Rotor Revolutions

CHAPTER 3 THEORY ON SYSTEM IDENTIFICATION

3.1 Overview of System Identification

The majority of the research presented in this thesis is based around the application of system identification. Hence, a comprehension of the principles behind system identification is a necessity for understanding the presented research and its implications. System identification is defined as “approximate modeling for a specific application on the basis of observed data and prior system knowledge [53].” In other words, system identification is the development of a model based on data collected from the measurement of specific behavioral parameters of a system. For some systems, a previously developed analytic model of the system can aid in developing the system identified model, but this is not always feasible.

There are two major approaches for using system identification to develop a model. These approaches are designated as grey-box and black-box modeling. Grey-box modeling makes use of a preconceived knowledge of a system such as an analytic model. On the other hand, black-box modeling requires no assumptions about the analytic system behavior and instead relies on data from the inputs and outputs of a system in order to produce a model. For some systems, the analytical behavior may be unknown or not well understood. Often this is due to complicated system behavior and intricate system geometries. For the slow and fast dynamics of an axial compressor system, the analytical system behavior is only marginally understood. Therefore, the black-box system identification approach is used for the research presented in this document.

When using system identification for modeling, it is generally necessary to have data for the system input or inputs as well as data for the system output or outputs. Some systems may have multiple inputs and multiple outputs. More commonly, one input is used with one corresponding output. However, this is not always the case. System identification involves applying a model to these input and output data sets and determining the model parameters that best relate the input and output. Generally, the model parameters are chosen based on some type of fitness criterion such as a goodness of fit test or correlation. The model parameters that produce the best fit between the actual output data and the output data obtained from the model are the ones that should

generally be used. However, other considerations such as whether the model captures the important system frequencies may be more important than a high fitness criterion value.

3.2 System Identification Calculations

The relation between the input and output can be represented by the following convolution integral equation in continuous time [53].

$$y(t) = \int_{-\infty}^t g(t-\tau)u(\tau)d\tau \quad (3.1)$$

In this equation, $u(\tau)$ represents the input, $y(t)$ represents the output, and $g(t-\tau)$ is the impulse response of a system at time t . τ is a “dummy” variable that acts as a place holder for time during integration. This equation is known as the impulse response model representation for a single input, single output (SISO) system. For discrete time systems, which most non-theoretical systems are, Equation (3.1) can be rewritten in the summation form shown below [53],

$$y(t) = \sum_{k=-\infty}^t g(t-k)u(k) \quad (3.2)$$

where k is the time index. The Laplace transform of the impulse response model representation results in:

$$Y(s) = G(s)U(s) \quad (3.3)$$

where $U(s)$ and $Y(s)$ are the Laplace transform as a function of the complex variable of the input $u(t)$ and output $y(t)$, respectively. $G(s)$ is the system model that relates the input and output. Figure 3.1 shows the basic block diagram representation of the relation between input and output data using a system model.

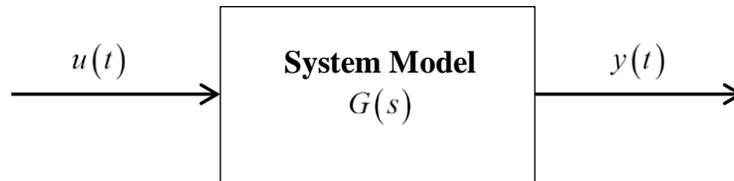


Figure 3.1: Block Diagram Representation of Relation between Input and Output

System identification calculations for producing a system model are carried out using the MatlabTM [54] system identification toolbox. The system identification toolbox uses an iterative numerical approach to match a selected model type to a set or sets of given input and output data based on the goodness of fit test between the model and the

output data. The system identification toolbox allows for several types of linear as well as nonlinear model types. Linear model types that are included in the system identification toolbox include continuous and discrete time transfer function models with variable numbers of zeros and poles, state-space models, and polynomial models including autoregressive with exogenous input (ARX) models. Nonlinear models incorporated into the system identification toolbox include the Hammerstein-Wiener as well as a nonlinear ARX model. In this research, transfer function models and linear ARX models are primarily used. Once a model is identified for the system, this model can ideally be used in control system applications.

3.3 Excitation Input for System Identification

When performing system identification, it is important to excite the system by choosing the proper input. There are several types of input that can be used such as impulse and step inputs. However, the most effective means of exciting a system for system identification is to use a periodic signal that involves one or more frequencies. A sinusoidal input of the form shown below, with frequency ω_0 , can be useful for certain purposes.

$$u(t) = \sin(\omega_0 t) \quad (3.4)$$

The main limitation of using a simple sinusoidal input is that it can only excite the system at a single frequency. A sinusoidal input is used for the slow dynamics research because sinusoidal oscillation of the throttle ring is the only means of exciting the system with the current equipment at the IET. However, multiple excitation frequencies are possible when the throttle modules are put in position around the throttle ring.

An excitation input that is generally the most useful for comprehensive system identification is the random binary sequence (RBS). The RBS alternates between two values, generally -1 and 1 or open and closed, depending on the actuator producing the input. The primary usefulness of using a RBS as an input for system identification comes down to the wide range of frequencies that this type of signal can excite. The frequencies of the RBS are produced randomly and generally encompass a broad range dependent on the pseudo-random algorithm used to produce the RBS sequence. A general equation for the RBS alternating between -1 and 1 is shown in the equation below [53].

$$u(t) = u(t-1) * \text{sign}(w(t) - p_0) \quad (3.5)$$

In this equation, sign is the signum function, $w(t)$ is a uniform random process for time $t = 1, 2, \dots, N$, and p_0 is the switching probability with a value in the range of 0 to 1. The excitation input used in the fast dynamics data research attempts to mimic an RBS by semi-randomly opening and closing the air injection valves. Of course, the actuation air injection is performed by a human operator and is only semi-random with a much smaller range of frequencies than a true RBS.

3.4 Signal Noise in System Identification

When performing system identification, it is important to reduce the amount of random noise in the input and output data signals. A system identified model that primarily models the system noise is essentially useless. Unfortunately, most sensors used to measure the dynamics of a system pick up some amount of noise. This noise may be due to error inherent to the sensor signal or to environmental background noise picked up by the sensor. For the data collected in the presented research, the noise generally manifests approximately as Gaussian distributed white noise with zero-mean and a near uniform power spectral density.

There are several methods for dealing with this noise. A signal filter may be applied in order to reduce the magnitude of specific noise frequencies in the signal. Some types of signal filters include low-pass filters, high-pass filters, band-pass filter, and notch filters. More specialized filters include the Bayesian filter, which is specifically used for filtering the slow dynamics data by setting the probability of random jumps in the data to correspond to that of the signal noise. Averaging windows can also be applied to a data set to average out the noise over a specified time period. However, the window size of the averaging window must be chosen such that the noise is effectively averaged without altering the actual behavior of the signal.

An alternative method of dealing with noise is the use of correlation coefficients. Rather than looking at a signal by its discrete points, correlation coefficients allow for the observation of changes in a signal or between signals over time. An example of an autocorrelation coefficient $R_{XX}(t)$ that is used in the analysis of the fast dynamics data is shown in the equation below.

$$R_{XX}(t) = \frac{\sum_{i=1}^n (X_i(t) - \bar{X}(t))(X_i(t-1) - \bar{X}(t-1))}{\sqrt{\sum_{i=1}^n (X_i(t) - \bar{X}(t))^2} \sqrt{\sum_{i=1}^n (X_i(t-1) - \bar{X}(t-1))^2}} \quad (3.6)$$

In the previous equation, $X_i(t)$ is a data point collected at time t and n is the number of data points collected at this time. $X_i(t-1)$ is a data point collected from the same sensor signal at the previous time step $t-1$. The purpose of the autocorrelation coefficient is to

determine how well the data points correlate between these two time steps. Further explanation of this equation is provided in later sections. Use of the autocorrelation coefficient method is particularly useful for capturing the system frequencies and reducing the effects of the noise for the output data signals obtained from the dynamic pressure sensors.

CHAPTER 4 DEVELOPMENT OF SLOW DYNAMICS

MODEL

4.1 Slow Dynamics Model Development Method

In this section, a method for developing a slow dynamics model for an axial compressor system through the use of Bayesian signal filtering and system identification is proposed. This method also attempts to use an optimization method to optimize the filter parameters to produce a model to better fit or correlate to the filtered data obtained from the raw test data measurements. Figure 4.1 demonstrates in flow chart form the process that can be undertaken to identify a model through system identification as well as to optimize the filter parameters.

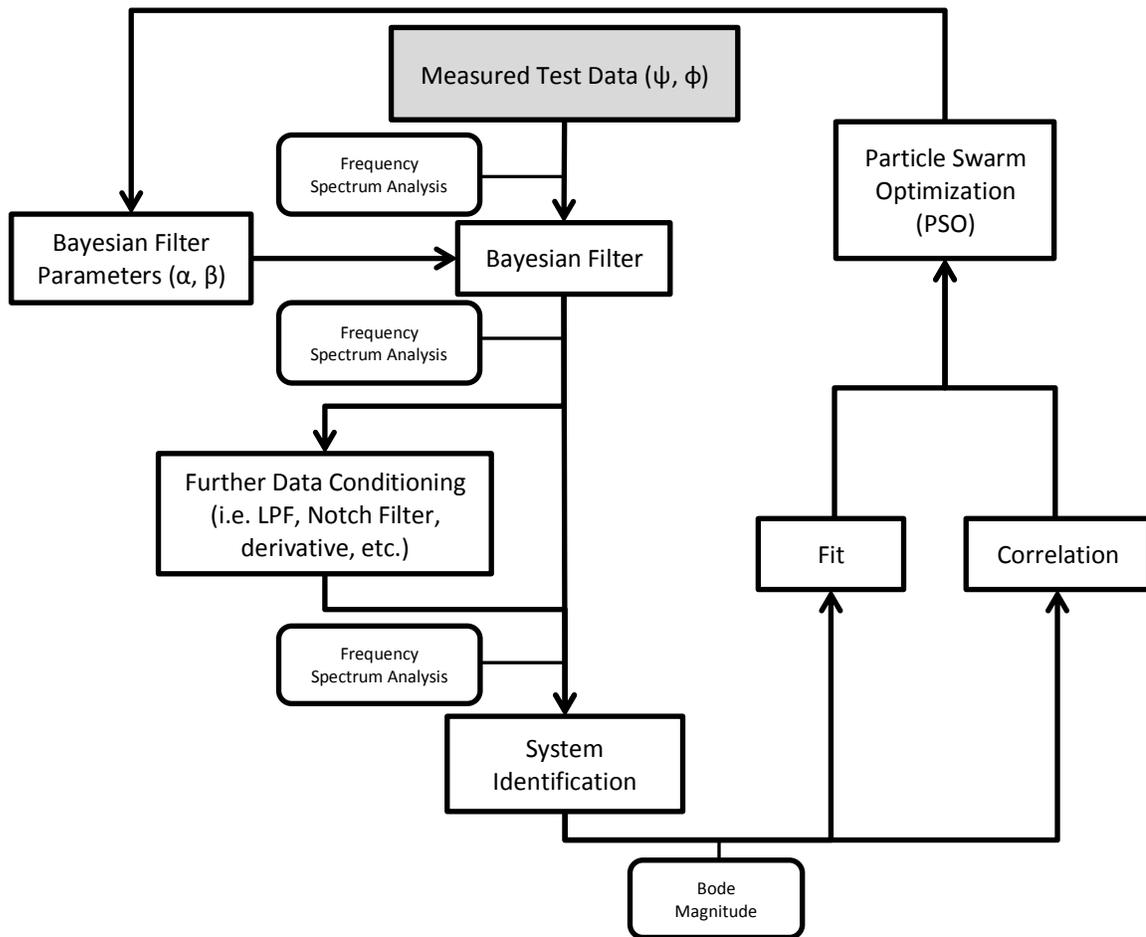


Figure 4.1: Flow Chart of Optimization Method for Data Filtering and System Identification

The first step in the process is to obtain the raw test data from the experimental compressor setup at the IET. Measurements are taken for the axial flow velocity and the change in pressure from inlet atmospheric pressure to plenum pressure and can subsequently be used to calculate the flow coefficient ϕ and the pressure rise coefficient ψ , respectively. At this point a frequency amplitude spectrum analysis is conducted to observe the frequency modes occurring in the non-filtered data. The next step is to reduce noise in the data in order to get a better picture of how the actual system dynamics behave. A Bayesian filter is chosen for this purpose. The Bayesian filter is modified using two parameters designated as α and β . These filter parameters have a physical meaning, where α represents the diffusion rate and β is a measure of the probability of sudden jumps in the data. These two parameters are optimized using a particle swarm optimization (PSO) method, which is discussed below. See Sanger [55] for a detailed description of how the Bayesian filter is used.

After the Bayesian filter is applied to the data, another frequency amplitude spectrum analysis is carried out to determine the effect on the noise frequencies that are filtered. At this point, further data conditioning may be applied. This includes the use of low pass filters to eliminate high-frequency data noise, notch filters to eliminate specified frequencies that do not add to the dynamic model, etc. Other filter types or methods of data conditioning may be applied at this point as necessary. An additional frequency amplitude spectrum analysis can be carried out at this point to observe the effects of the additional data conditioning past the standard Bayesian filtering. After data conditioning, the next step is to use system identification to develop a dynamic model for the conditioned compressor data.

The system identification consists of applying a variety of models including transfer function, state-space, autoregressive with exogenous (ARX), autoregressive-moving-average with exogenous (ARMAX), etc. The continuous time transfer function model tends to have the best goodness of fit for the data of the model types that are attempted. For the system identification, the filtered and conditioned flow coefficient is taken as the input and the filtered and conditioned pressure rise coefficient is taken as the output. After a model is developed, a comparison between the measured output data and the output data obtained using the model simulation is conducted. This comparison can

be performed one of two ways, either a goodness of fit comparison or a linear correlation. The goodness of fit comparison is calculated as a percentage using the equation:

$$Fit = \left[1 - \frac{\|y - \hat{y}\|}{\|y - \bar{y}\|} \right] \cdot 100\% \quad (4.1)$$

where y is the measured output, \hat{y} is the simulated model output, and \bar{y} is the average of the measured output. $\| \cdot \|$ designates the Euclidean norm of the argument. The alternate method of comparing the measured and model simulated output is to calculate the linear correlation between the two data sets. The linear correlation equation used to compare the data sets is equal to:

$$corr(y, \hat{y}) = \frac{E[(y - \mu_y)(\hat{y} - \mu_{\hat{y}})]}{\sigma_y \sigma_{\hat{y}}} \quad (4.2)$$

where μ_y and $\mu_{\hat{y}}$ are the expected values of y and \hat{y} , respectively. The variables σ_y and $\sigma_{\hat{y}}$ represent the standard deviations of the data sets y and \hat{y} , respectively. Lastly, $E[\cdot]$ is the expectation or first moment of the argument within the brackets. The goodness of fit comparison or linear correlation is used to create the cost function that is to be minimized. For these comparisons, the system identification output is the pressure rise coefficient ψ , which is substituted into these equations in place of the output variable y .

The cost function is minimized using a particle swarm optimization (PSO) algorithm. The details of the particle swarm algorithm are outlined by Chen et al. [56]. The PSO algorithm is used specifically for optimizing the α and β parameters of the Bayesian filter. However, PSO can also be used for simultaneously optimizing certain characteristics of the model obtained from system identification. For instance, PSO can be used to determine the optimal number of zeros and poles to be used in a model transfer function. Each iteration of the PSO algorithm produces a new set of optimized parameters, which are fed back to the Bayesian filter and are subsequently used to filter the data to be used in the identification of a new model. The new simulated model is then

compared to the measured data and new optimal parameters are obtained from the PSO, thus completing the cycle of the optimization method shown in Figure 4.1.

4.2 Bayesian Filtering of Measured Data Signal

The slow dynamics signal data collected from the compressor experiments has a significant amount of background noise in the signal. Consequently, this noise has to be filtered out of the signal before a useful model can be developed. A type of filter, known as a Bayesian filter, is used to eliminate a large portion of the background noise in the data sets for this analysis. The method of Bayesian filtering was developed by Sanger [55] primarily for use in the filtration of electromyogram (EMG) signals for biomedical engineering applications. However, the filter has a wide range of applicability and is useful specifically for signal data with amplitude-modulated zero-mean Gaussian noise. The pressure and flow transducer signals are believed to contain this type of noise, hence the Bayesian filtration method is chosen. A brief explanation of the method for Bayesian filtering, based on the work by Sanger [55], is described below.

The relation between the measured pressure rise coefficient signal and the filtered signal can be described by the conditional probability density function (PDF), $P(\psi | x)$. In this PDF, x is the driving signal of the pressure rise coefficient, which cannot be directly measured, and ψ is the measured pressure rise coefficient signal with noise. Assuming the noise in the signal is a result of random events in the signal, the average amplitude in a small time window is proportional to the number of these events during that time period [55]. In the following equation, n represents the number of these random events. The number of these events can be modeled approximately as a Poisson process for the PDF.

$$P(\psi | x) \approx (x^n e^{-x}) / n! \quad (4.3)$$

By empirically observing the behavior of the pressure rise coefficient data, it is believed that the signal noise can be described as amplitude modulated zero-mean Gaussian noise. For values of ψ greater than zero, the PDF can be written as shown in Equation (4.4).

$$P(\psi | x) = 2 \times e^{(-\psi^2/2x^2)} / \sqrt{2\pi x^2} \quad (4.4)$$

For the case of ψ and ϕ , the data is greater than zero for all values prior to filtering. Therefore the mean of the data can be subtracted after the filter is applied. When used with an EMG signal, the signal data must first be rectified and its mean must be

subtracted prior to filtering. The absolute value is then taken to rectify the signal and eliminate any values less than zero. However, for ψ and ϕ , the signal does not need to be rectified because the signal is already greater than zero prior to filtering. Consequently, the mean should be subtracted after the Bayesian filter is applied.

Based on further signal observation, a better approximation of the PDF is given by the Laplacian density function.

$$P(\psi | x) = e^{(-|\psi|/x)} / 2x \quad (4.5)$$

Similarly, for values of ψ greater than zero, the Laplacian density equation is of the following form.

$$P(\psi | x) = e^{(-\psi/x)} / x \quad (4.6)$$

If the pressure rise coefficient is given by $\psi(t)$ at a given time t , then the PDF given by $P[\psi(t) | x(t)]$ specifies the likelihood of each possible value of $x(t)$ given the corresponding measurement. The posterior density is then given by Bayes' theorem applied to the signal.

$$P[x(t) | \psi(t)] = P[\psi(t) | x(t)] \times P[x(t)] / P[\psi(t)] \quad (4.7)$$

In this equation, $P[x(t)]$ is the PDF for $x(t)$ immediately before the value of $\psi(t)$ is measured. The prior $P[x(t)]$ is generally dependent on the entire past history of the $\psi(t)$ measurements. An estimate for $P[x(t)]$ can be obtained by using a recursive discrete time measurement algorithm. This recursive algorithm is obtained from Bayes' theorem and is shown in Equation (4.8) below, where C is a constant that allows the PDF to integrate to one.

$$P[x(t) | \psi(t), \psi(t-1), \dots] = \frac{P[\psi(t) | x(t)] \times P[x(t)]}{P[\psi(t-1), \psi(t-2), \dots]} / C \quad (4.8)$$

In order to implement the recursive algorithm shown in Equation (4.8), the algorithm must first be put into a mathematical form. This can be done by numerical integration of the Fokker-Planck equation, which is a partial differential equation that describes the time evolution of the PDF. This equation is shown below in Equation (4.9).

$$p(x, t-1) \approx \alpha \partial^2 p(x, t-1) / \partial x^2 + \beta + (1-\beta) p(x, t-1) \quad (4.9)$$

Equation (4.10) shows the mathematical form of the probability density algorithm derived in Sanger [55] by numerical integration.

$$p(x, t-1) \approx \alpha p(x-\varepsilon, t-1) + (1-2\alpha) p(x, t-1) + \alpha p(x+\varepsilon, t-1) + \beta + (1-\beta) p(x, t-1) \quad (4.10)$$

Equation (4.10) involves two constant parameters, α and β . These parameters have a physical meaning for the filter, where α represents the diffusion rate and β is a measure of the probability of sudden jumps in the data. According to reference [55], α and β are chosen empirically. However, for this research, a particle swarm optimization algorithm is also used to compute the α and β parameters that produce the best goodness of fit or correlation for the developed model to the measured data. As a side note, these same equations are also applicable to the flow coefficient data when the variable ψ is replaced with the flow coefficient variable ϕ .

A comparison of the Bayesian filtered (BF) data to the sensor measured data for the pressure rise coefficient is shown in Figure 4.2 and Figure 4.3 for 5 throttle oscillation cycles and 1 throttle oscillation cycle, respectively. Similarly, a comparison of the BF data to the sensor measured data for the flow coefficient data is shown in Figure 4.5 and Figure 4.4 for 5 cycles and 1 cycle, respectively. For Figure 4.2 through Figure 4.4, the α parameter is set to 0.1 and the β parameter is set to 150. The most prominent effect of the filter is reducing the effect of high frequency random jumps and noise in the measured data. When choosing the filter parameters empirically, it is found that the α parameter has a relatively small effect on the actual function of the filter when compared to the β parameter. The data for the filter comparison in the following figures has not yet been normalized to a maximum absolute value of one. This normalization is not performed in order to better show the comparison between the measured test data and the BF data. However, the BF data is subsequently normalized to one for the system identification analysis. The MatlabTM [54] code that is used for the Bayesian filter calculations is included in Appendix A1.

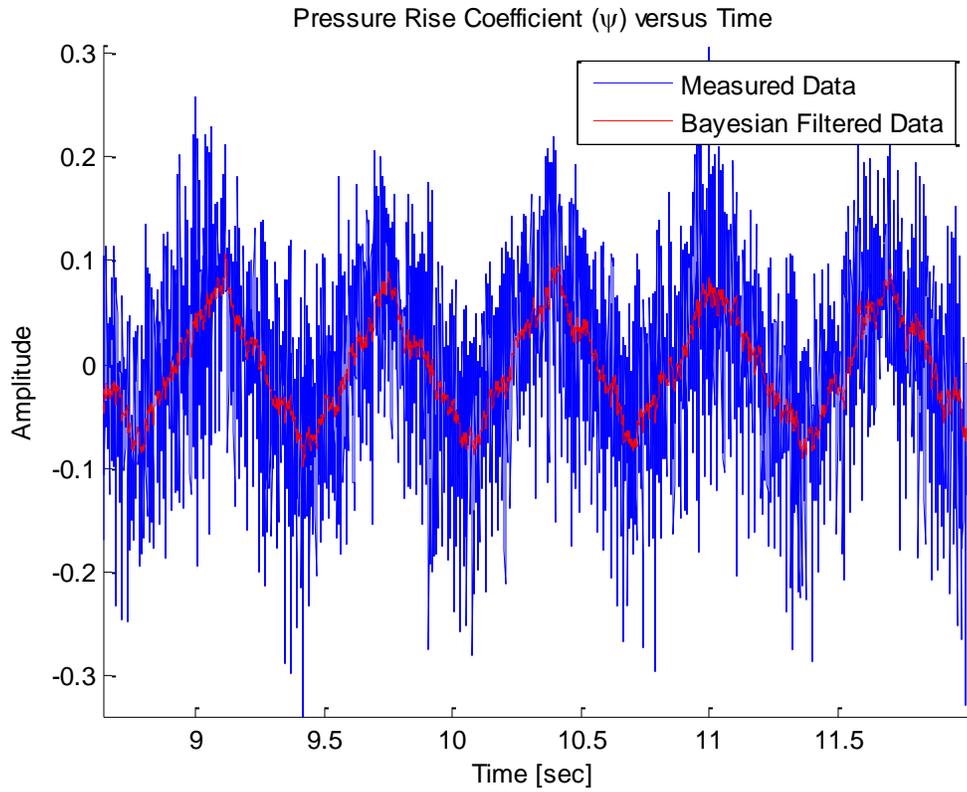


Figure 4.2: BF Data Comparison to Measured Data for Pressure Rise Coefficient (ψ) for 5 Cycles

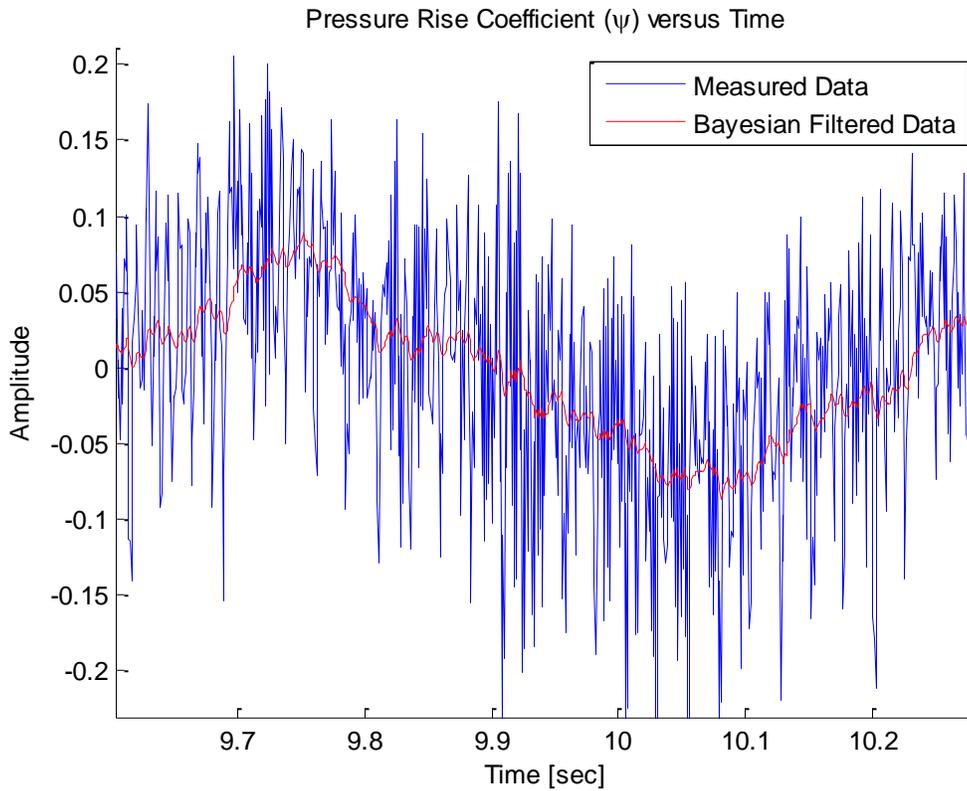


Figure 4.3: BF Data Comparison to Measured Data for Pressure Rise Coefficient (ψ) for 1 Cycle

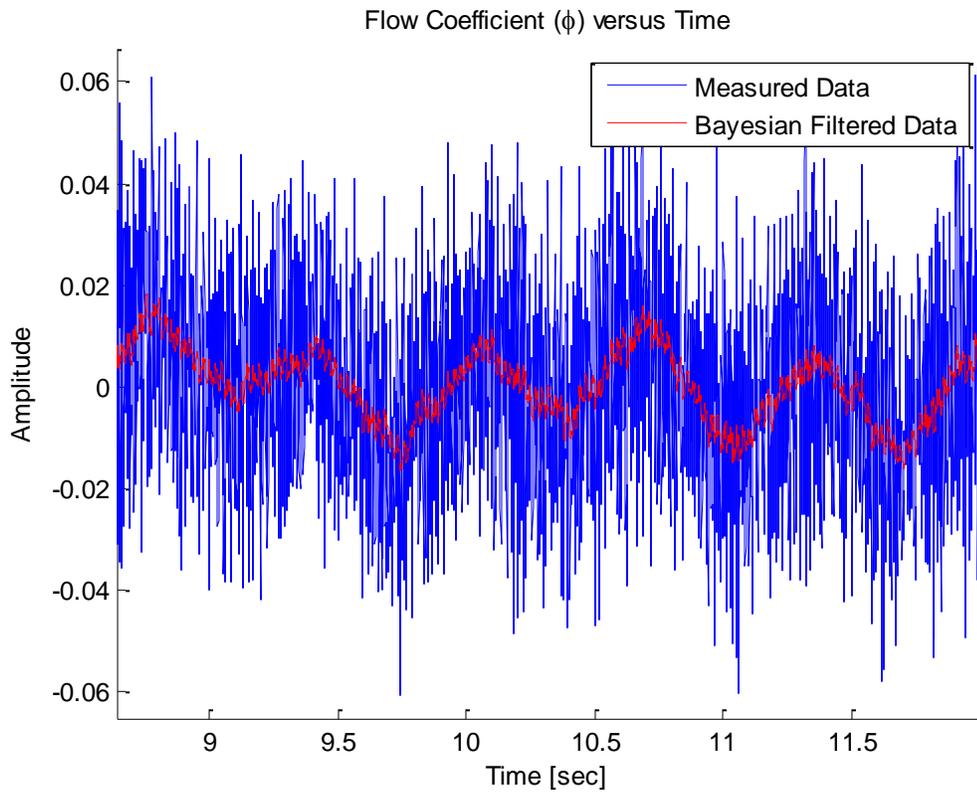


Figure 4.5: BF Data Comparison to Measured Data for Flow Coefficient (ϕ) for 5 Cycles

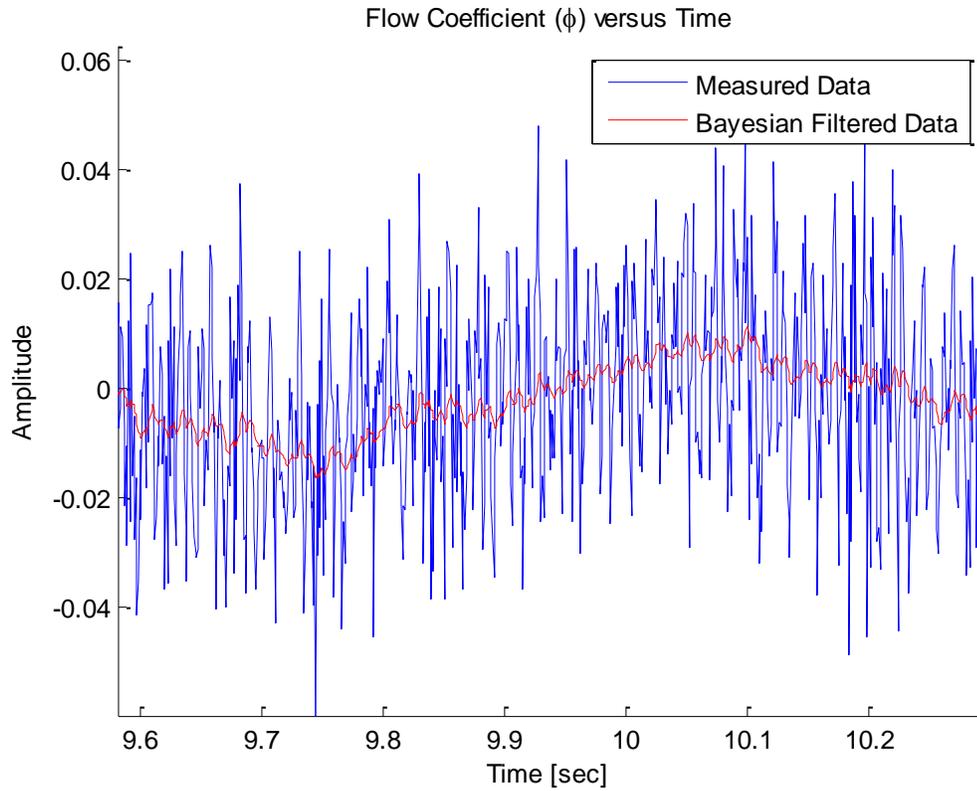


Figure 4.4: BF Data Comparison to Measured Data for Flow Coefficient (ϕ) for 1 Cycle

4.3 Particle Swarm Optimization for Filter Optimization

Particle swarm optimization (PSO) is used to optimize the filter and model parameters in order to best fit or correlate the simulated model to the filtered data. Particle swarm algorithms were originally inspired for use as mathematical operators by the social behaviors of bird flocks and fish schools [56]. This section provides some background information on the dynamics of particle swarm optimization and how this optimization method is applied to the compressor model. The explanation in this section is primarily based on information from Chen et al. [56].

The first step of the PSO is to input the parameters to be optimized. The number of input parameters determines the dimension of the search space. For example, two input parameters creates a search space with two dimensions. A minimum value bound as well as a maximum value bound must be specified for each of these parameters. The minimum and maximum value bounds correspond to the bound dimensions of the search space.

The positions of the particles must then be initialized. Before the first iteration, this is done randomly using the MatlabTM [54] “rand” command to create uniformly distributed random initial positions for the particles. Note that these random values are uniformly distributed between the minimum and maximum bounds specified for each parameter. Once the particles begin to change positions, the particles may exceed the specified bounds. If the particle exceeds the maximum bound of the parameter then the position is set to the maximum value bound. Similarly, if the initial position of a particle is less than the minimum value bound, the particle position is set to the minimum value bound.

The next step after particle position initialization is to substitute the positions of the particles into the cost or fitness function. The cost or fitness function is a function that contains a certain equation or series of calculations that produces a single output. The primary goal of the particle swarm optimization algorithm is to minimize the output of this cost or fitness function and produce the parameter values that create this minimum output. The cost or fitness function is then iteratively evaluated at each of the particle positions. If the new position of the particle minimizes the cost function more than the previous particle position, then the particle assumes the new position.

In the following equations, the variable t is time, the subscript i is the component of particle dimension, and the superscript j is the particle index. For instance, $x_i^j(t)$ is the i position component of particle j at time t . In order for the particle to reach a new position at each iteration, each particle is given a velocity based on the following equation of motion.

$$V_i^j(t) = V_i^j(t-1) + c_1(x_i^{j,lbest}(t-1) - x_i^j(t-1)) + c_2(x_i^{j,gbest}(t-1) - x_i^j(t-1)) \quad (4.11)$$

Once again, the variable t is time, the subscript i is the component of particle dimension, and the superscript j is the particle index. $V_i^j(t)$ and $V_i^j(t-1)$ are the i component of the velocity of particle j , at times t and $t-1$, respectively. The superscript *lbest* denotes the local best for the iteration and *gbest* denotes the global best value. Therefore, $x_i^{j,lbest}(t-1)$ and $x_i^{j,gbest}(t-1)$ represent the particle position of the local best and the global best at time $t-1$, respectively. Randomly specified values, designated by c_1 and c_2 , determine how much the local best and global best component positions affect the particle velocity $V_i^j(t)$ relative to the velocity from the previous iteration $V_i^j(t-1)$. c_1 and c_2 are chosen randomly using either a uniform distribution or normal distribution.

Equation (4.11) is used to update the velocity of each particle at each iteration at time t based on position and velocity values from the previous time step iteration or time $t-1$. The position at time t is then calculated by substituting the velocity $V_i^j(t)$ from Equation (4.11) into the position equation shown below.

$$x_i^j(t) = x_i^j(t-1) + V_i^j(t) \cdot \Delta t \quad (4.12)$$

In this equation, Δt is the increment time between each iteration. For simplicity, Δt is set equal to one. For ease of comparison, the equations of motion listed in Equations (4.11) and (4.12) can be rewritten together in matrix form as shown below.

$$\begin{Bmatrix} V_i^j(t) \\ x_i^j(t) \end{Bmatrix} = \begin{bmatrix} 1 & -c_1 - c_2 \\ 1 & 1 - c_1 - c_2 \end{bmatrix} \begin{Bmatrix} V_i^j(t-1) \\ x_i^j(t-1) \end{Bmatrix} + \begin{bmatrix} c_1 & c_2 \\ c_1 & c_2 \end{bmatrix} \begin{Bmatrix} x_i^{j,lbest}(t-1) \\ x_i^{j,gbest}(t-1) \end{Bmatrix} \quad (4.13)$$

Figure 4.6 [56] shows a graphical depiction of how a single particle would move in a two dimensional search space from the previous position $x_i^j(t-1)$ to the updated position $x_i^j(t)$.

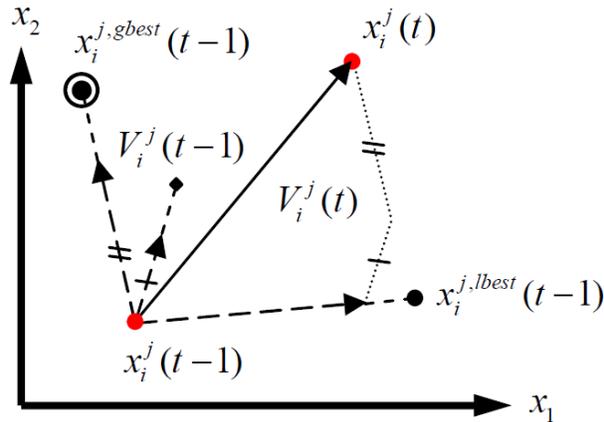


Figure 4.6: PSO Particle Movement from Previous Position to Updated Position [56]

As depicted in Figure 4.6, the updated position and velocity of the particle is influenced simultaneously by the previous particle velocity $V_i^j(t-1)$, the previous particle position $x_i^j(t-1)$, as well as the local best and global best previous positions.

To make the algorithm more efficient, the particles must be prevented from leaving the search space too often. This is done by setting a constraint on the maximum and minimum velocities of the particles. With a maximum velocity constraint designated by V_{\max} , a particle with a velocity magnitude that exceeds this maximum velocity constraint has its velocity readjusted to be equal to V_{\max} . In other words, if $V_i^j(t) > V_{\max}$, then the velocity of particle at the position $x_i^j(t)$ is set to V_{\max} . This prevents the particles from “exploding” outwards and creates a more efficient algorithm with the particles spreading out more evenly within the search space.

For the particle swarm optimization, a specific swarm size as well as a specific number of iterations must be chosen. According to Chen et al. [56], for the search space dimensions and functions used, the “optimal” swarm size was found to be at approximately 40 particles. The “optimal” swarm size may vary depending on the

dimensions of search space. However, a swarm size of 40 particles is deemed a reasonable size for these optimization calculations as well. The number of iterations is set to 100 which appeared to be a suitable number of iterations to produce the optimized parameters. The MatlabTM [54] PSO code that is used is shown in Appendix A2 with a cost function code for transfer function filter optimization in A3 and a cost function code for ARX parameter and filter optimization in A4.

4.4 Slow Dynamics System Identification

The system identification toolbox of MatlabTM [54] is used to perform the system identification analysis for the slow dynamics data. Several model structure types are implemented to determine which model structure has the best goodness of fit for the data. For the system identification analysis, the input is the filtered flow coefficient data and the output is the filtered pressure rise coefficient. Through the use of the system identification toolbox, the best model structure is determined to be a continuous time transfer function model. Equation (4.14) shows the form of the transfer function relative to the input and output in the complex frequency domain.

$$\frac{\psi(s)}{\phi(s)} = G(s) \quad (4.14)$$

In this equation, $\phi(s)$ is the flow coefficient or input in the frequency domain, $\psi(s)$ is the pressure rise coefficient or output in the frequency domain, and $G(s)$ represents the system identified transfer function model that relates the input and output.

Of the transfer functions that are attempted to be fit to the data, it is determined through the use of trial and error that a transfer function with three zeros and four poles would produce a model structure that would best fit the data and produce the highest goodness of fit. The form of this transfer function model is:

$$G(s) = \frac{b_3s^3 + b_2s^2 + b_1s + b_0}{s^4 + a_3s^3 + a_2s^2 + a_1s + a_0} \quad (4.15)$$

where a_0 , a_1 , a_2 , and a_3 represent the denominator coefficients and b_0 , b_1 , b_2 , and b_3 represent the numerator polynomial coefficients. These coefficients are determined iteratively through system identification to optimize the goodness of fitness of the transfer function model to the data. The transfer function of Equation (4.15) can be factored into a form where the zeros and pole values can be observed and the frequency domain behavior can be more clearly observed. The factored form of Equation (4.15) is shown below.

$$G(s) = \frac{b_3(s - z_1)(s - z_2)(s - z_3)}{(s - p_1)(s - p_2)(s - p_3)(s - p_4)} \quad (4.16)$$

A thorough analysis of pole behavior, stability, and other frequency domain analyses can be carried out on the transfer function model using this form of the equation.

A system identification analysis is performed in Matlab™ [54] in order to develop a transfer function model for the slow dynamics. The data used in this analysis is first passed through a Bayesian filter with α equal to 0.1 and β equal to 150. The sensor sampling frequency for the test to obtain this data is set at 1000 Hz. The compressor throttle ring is oscillated at a rate of approximately 1.53 Hz for approximately 60 seconds at flow coefficient ϕ set points of 0.60, 0.58, 0.56, 0.54, 0.52, and 0.51. Figure 4.7 through Figure 4.11 show the comparison of the BF output data or BF pressure rise coefficient (black) data to the transfer function model data (blue) for each of the set points. These plots show the comparison for approximately 32 seconds of the throttle oscillation. Furthermore, Figure 4.14 through Figure 4.18 show this same comparison for approximately 5 throttle oscillation cycles between the time 8.76 seconds to 12 seconds after the throttle begins oscillation at the ϕ set point. The BF pressure rise coefficient data as well as the BF flow coefficient data are both normalized to have a maximum absolute value of one after being filtered. Lastly, Table 4.1 shows the goodness of fit percentage values for the transfer function models that are developed at each flow coefficient set point.

Table 4.1: Transfer Function Model Goodness of Fit at Flow Coefficient Set Points

Flow Coefficient (ϕ) Set Point	Goodness of Fit: %
0.60	73.38
0.58	69.77
0.56	76.57
0.54	78.81
0.52	74.89
0.51	78.67

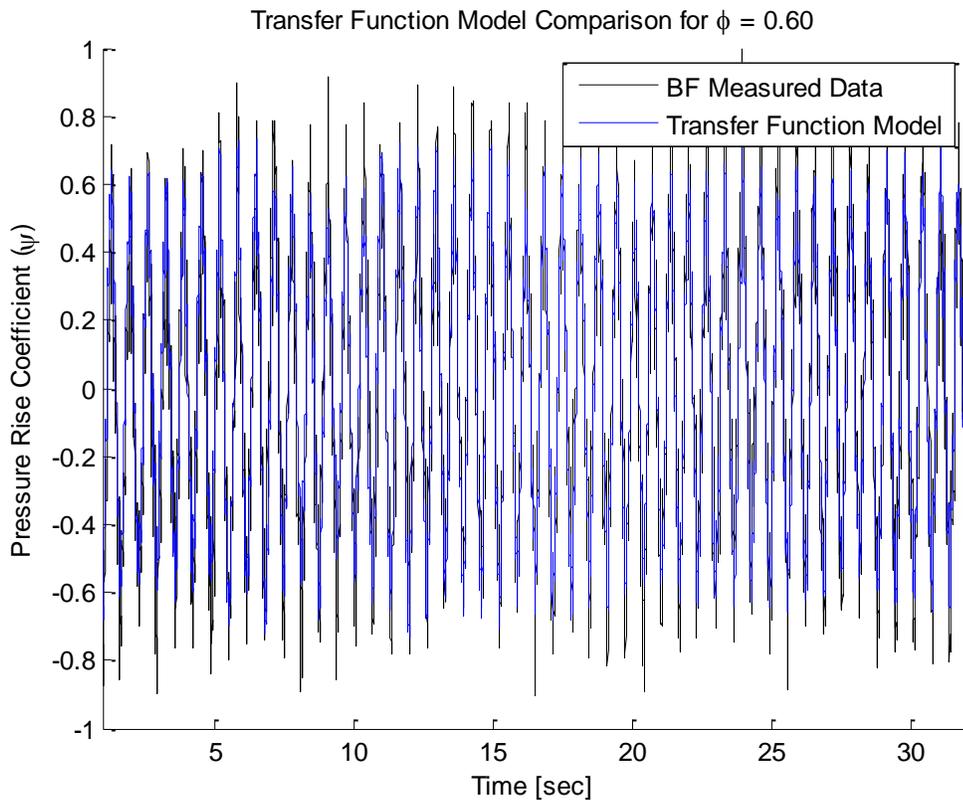


Figure 4.7: Transfer Function Model Comparison to the Measured Data, $\phi = 0.60$

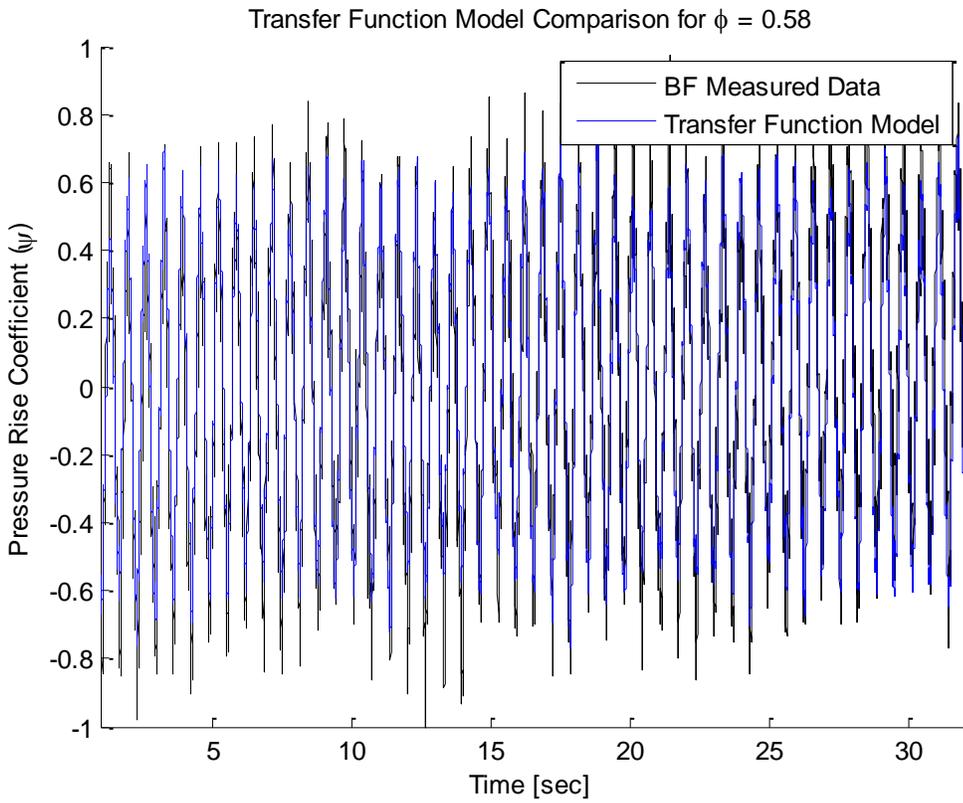


Figure 4.8: Transfer Function Model Comparison to the Measured Data, $\phi = 0.58$

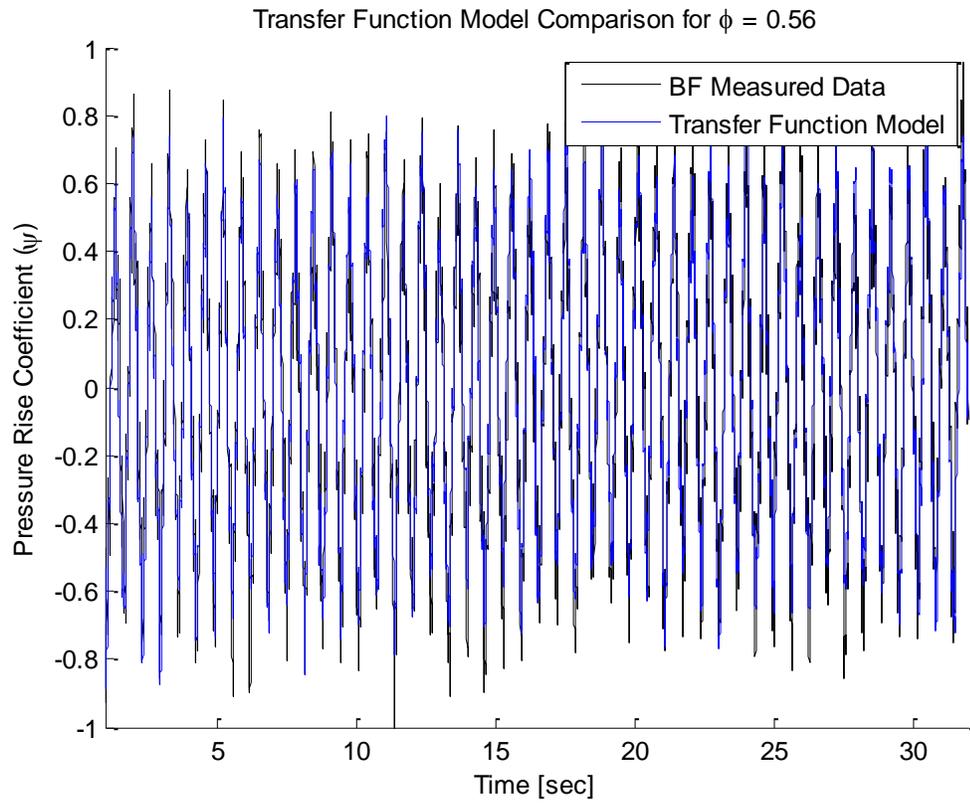


Figure 4.10: Transfer Function Model Comparison to the Measured Data, $\phi = 0.56$

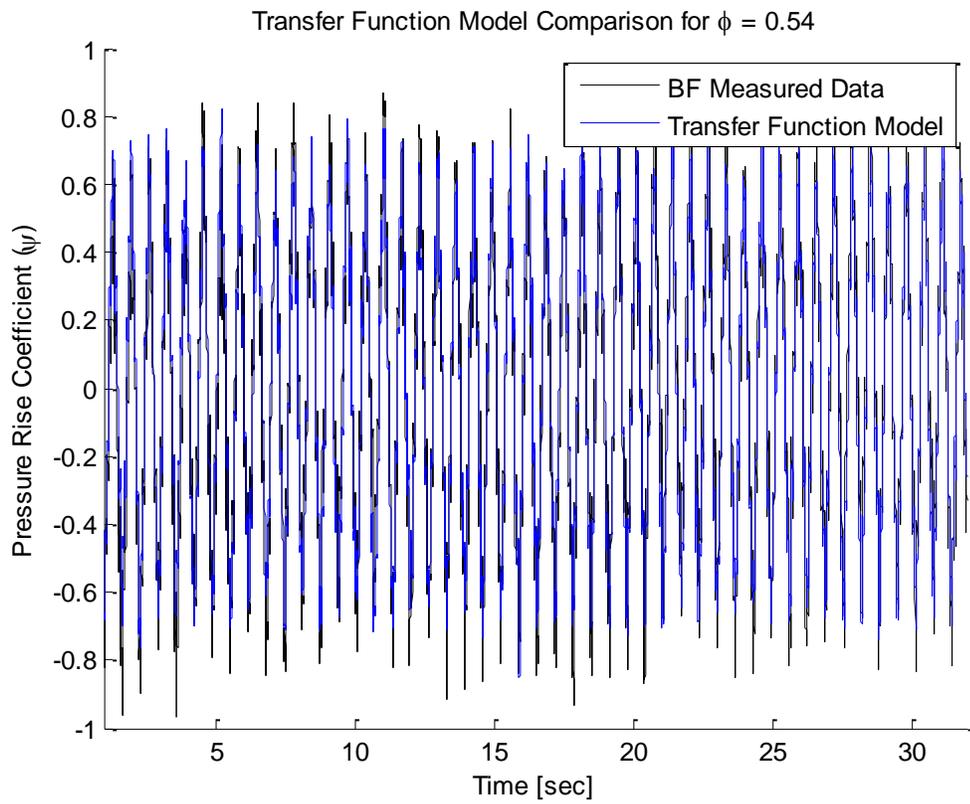


Figure 4.9: Transfer Function Model Comparison to the Measured Data, $\phi = 0.54$

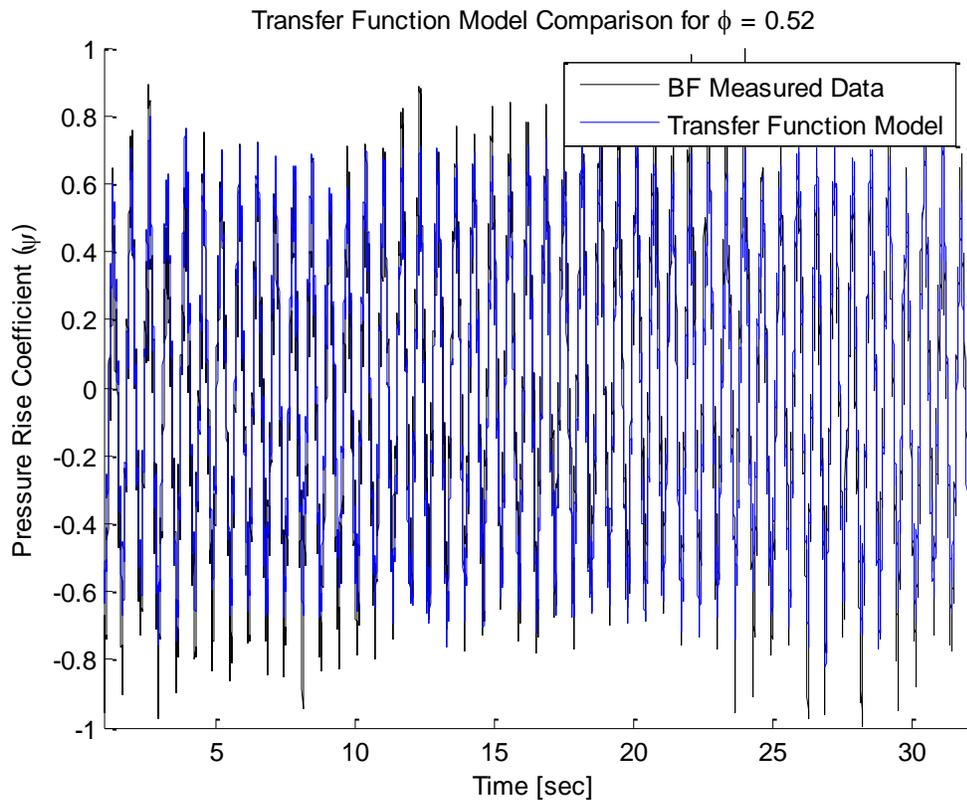


Figure 4.12: Transfer Function Model Comparison to the Measured Data, $\phi = 0.52$

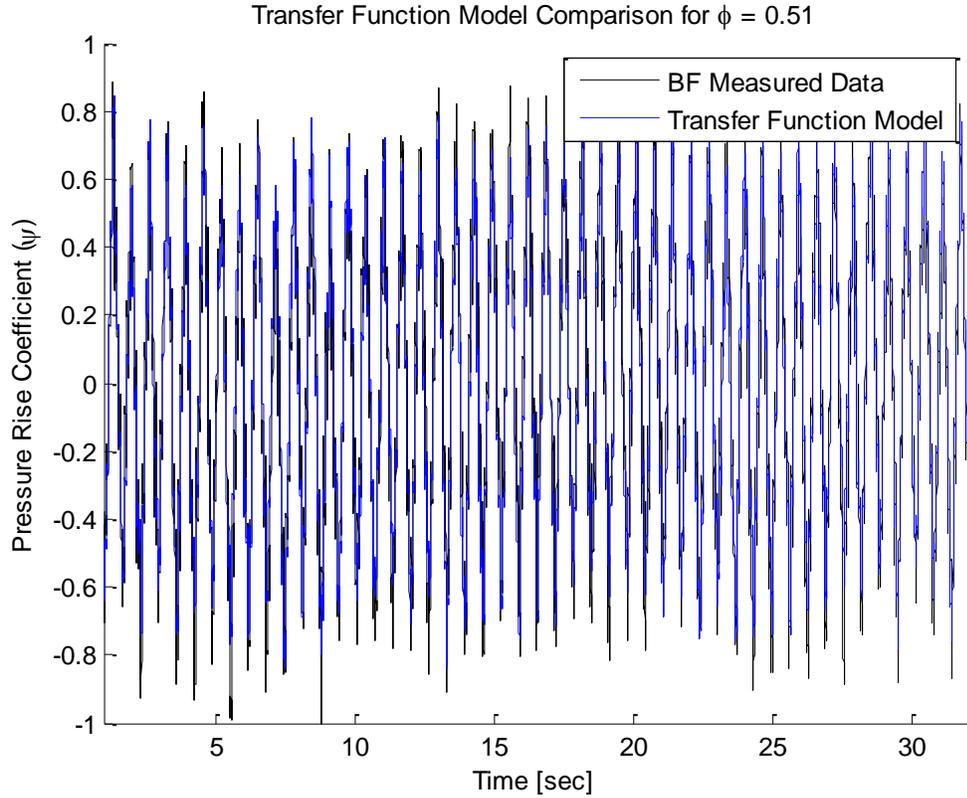


Figure 4.11: Transfer Function Model Comparison to the Measured Data, $\phi = 0.51$

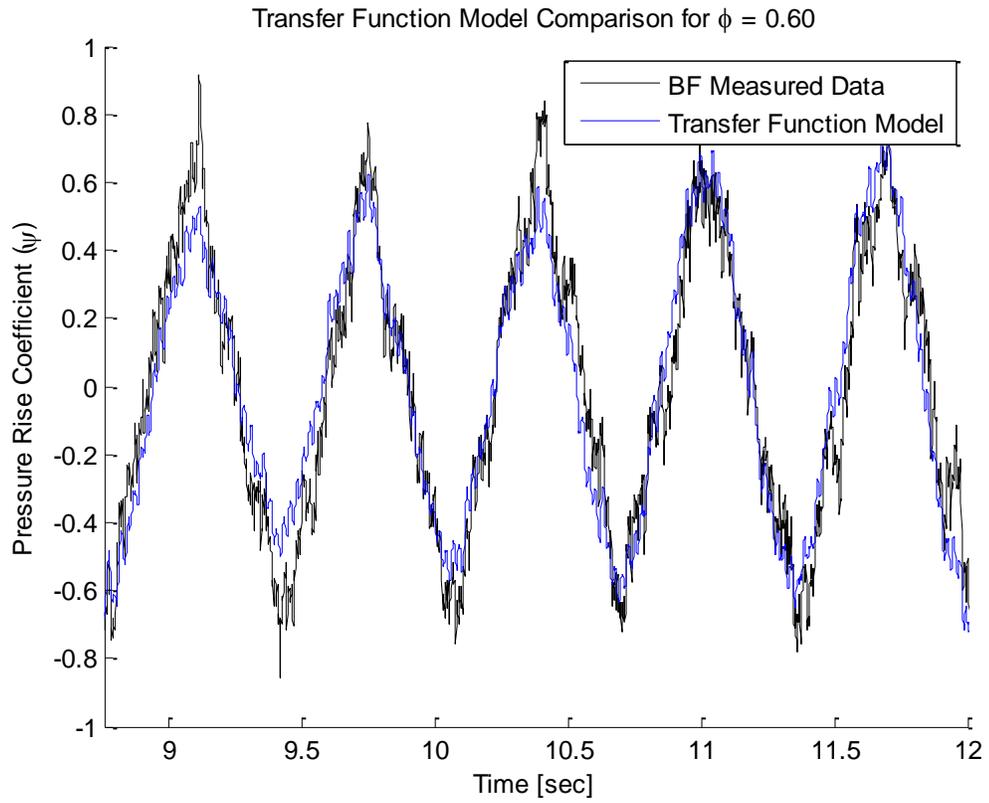


Figure 4.14: Transfer Function Model Comparison to the Measured Data, $\phi = 0.60$ for 5 Cycles

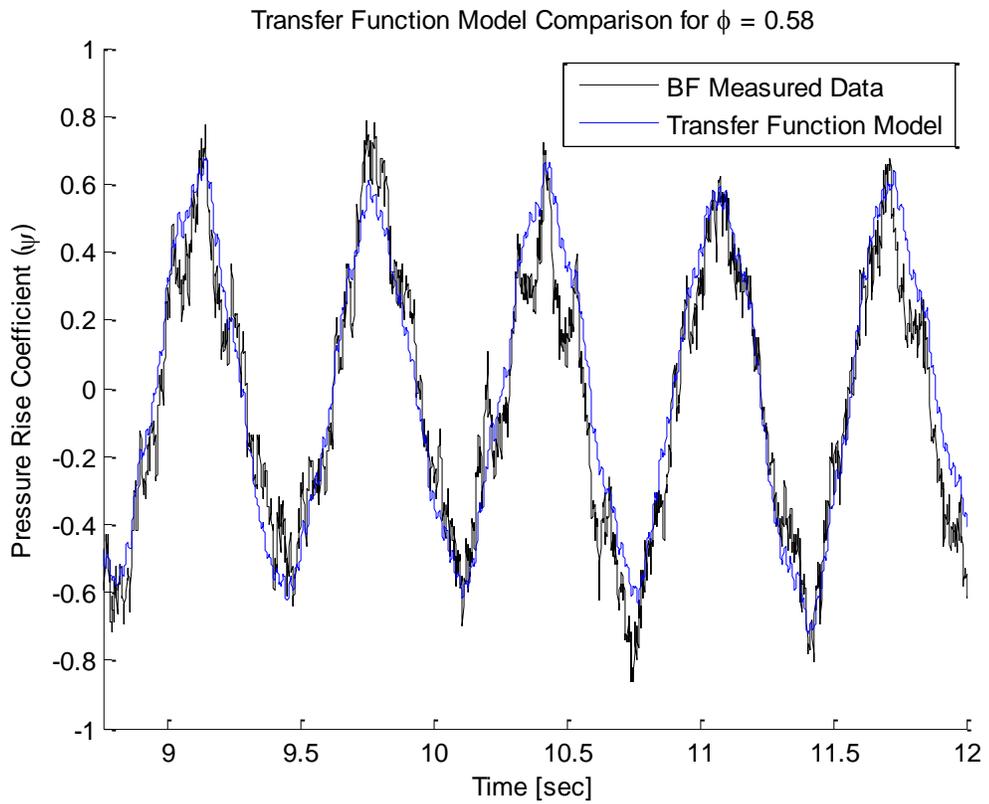


Figure 4.13: Transfer Function Model Comparison to the Measured Data, $\phi = 0.58$ for 5 Cycles

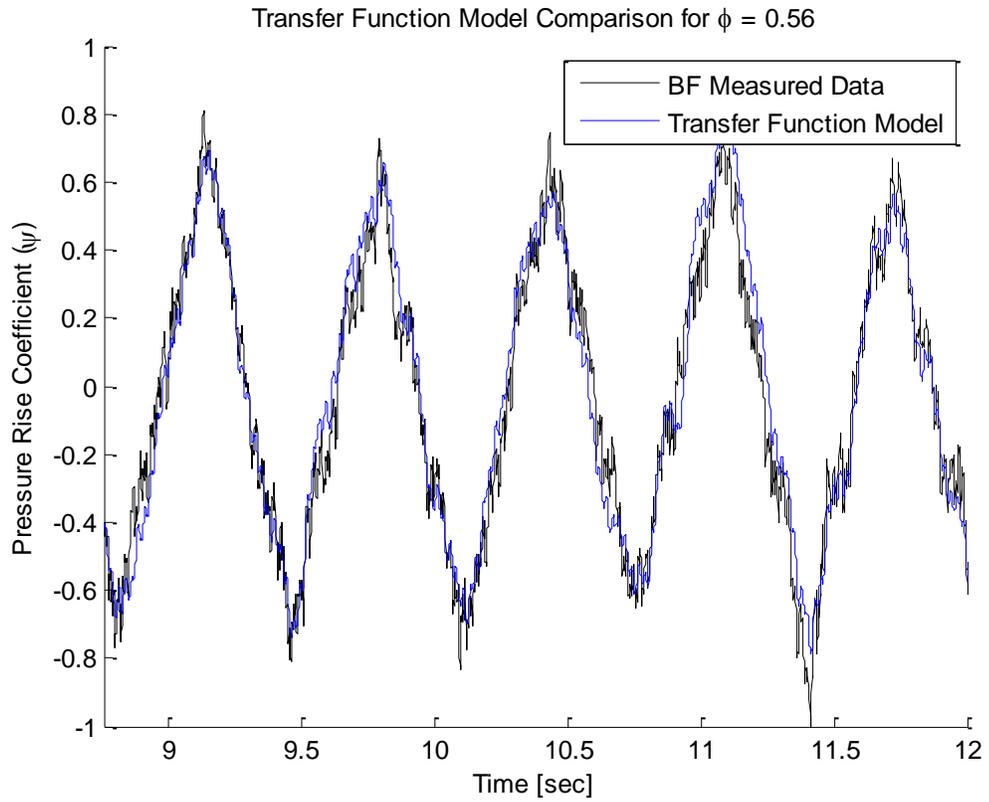


Figure 4.15: Transfer Function Model Comparison to the Measured Data, $\phi = 0.56$ for 5 Cycles

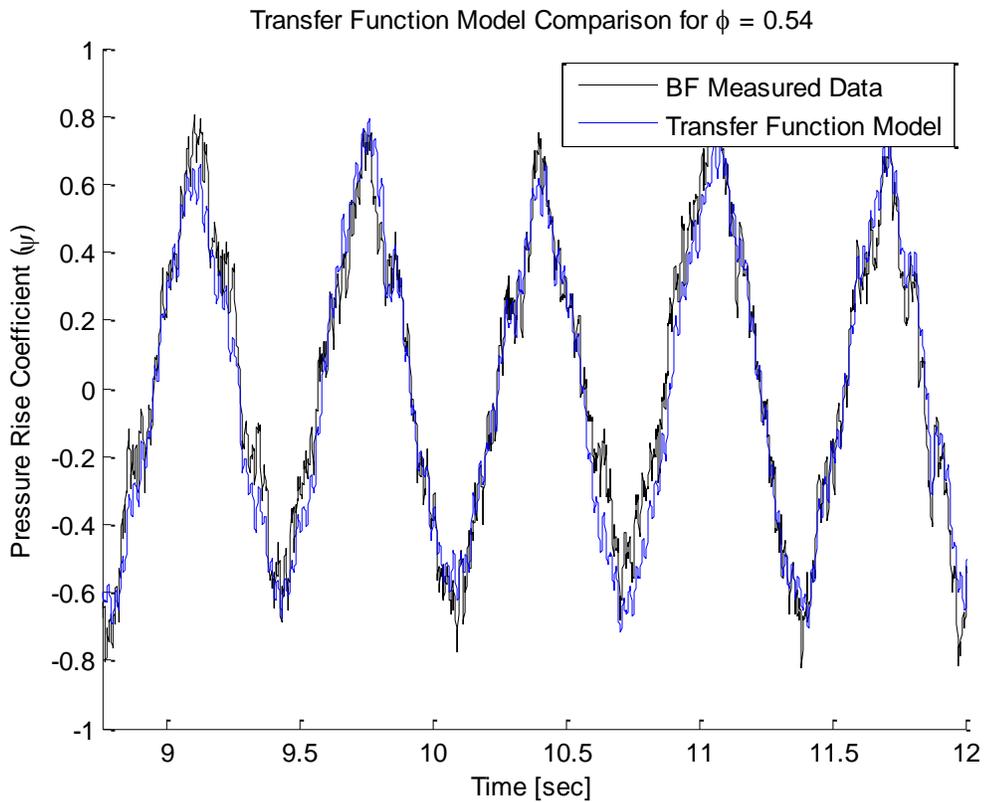


Figure 4.16: Transfer Function Model Comparison to the Measured Data, $\phi = 0.54$ for 5 Cycles

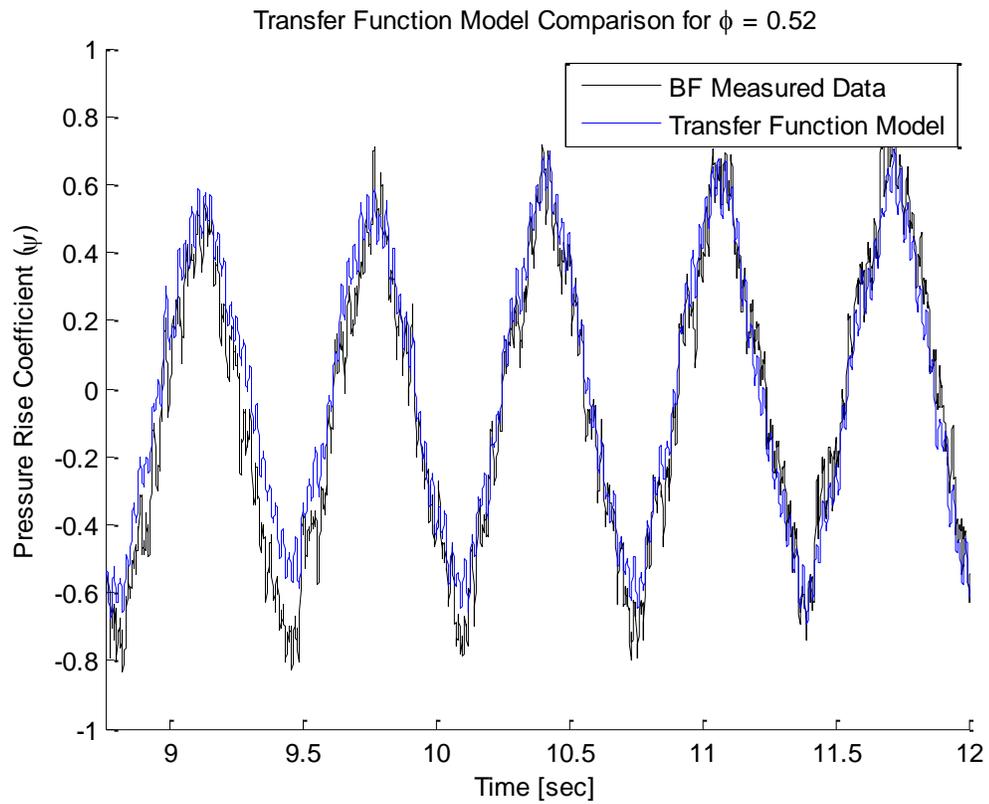


Figure 4.17: Transfer Function Model Comparison to the Measured Data, $\phi = 0.52$ for 5 Cycles

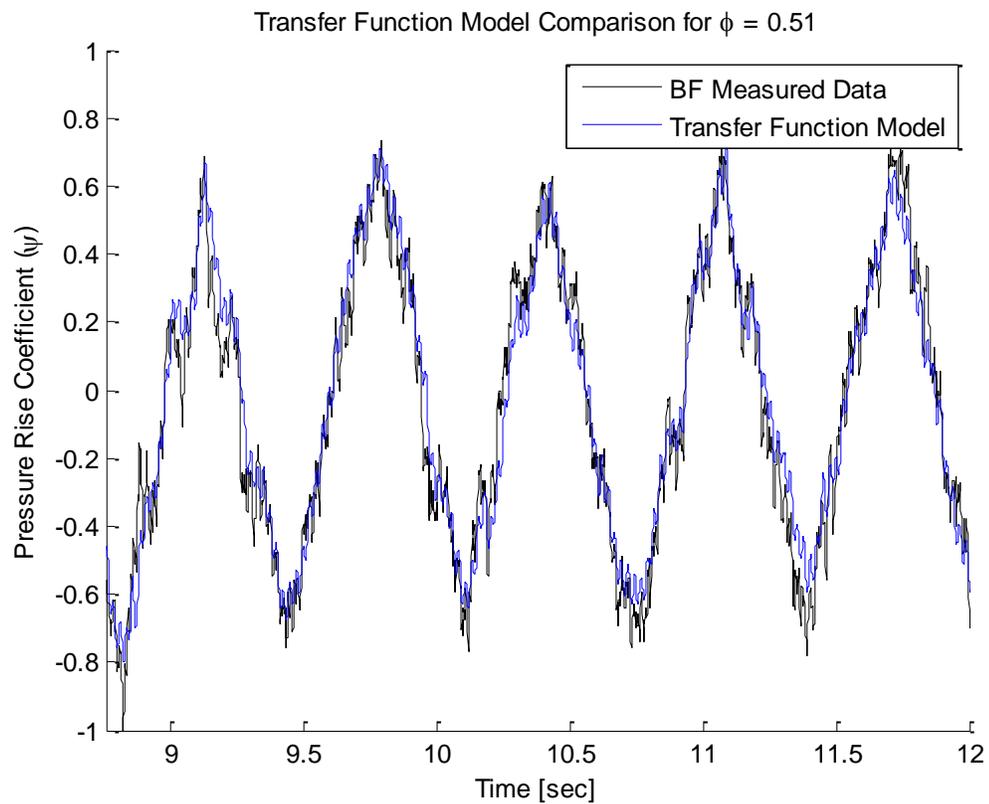


Figure 4.18: Transfer Function Model Comparison to the Measured Data, $\phi = 0.51$ for 5 Cycles

4.5 Amplitude Spectrums and Bode Plots for Slow Dynamics SI Models

The section discusses the effect of signal conditioning and filtering as well as system identification on the frequency modes of pressure rise coefficient data and flow coefficient data. A frequency amplitude spectrum plot is created at each step of signal conditioning. An amplitude spectrum plot is created for the data at each of the following flow coefficient set points: 0.60, 0.58, 0.56, 0.54, 0.52, and 0.51. Additionally, Bode plots are included for the transfer function models at the same flow coefficients. The data in this section is obtained using a 1000 Hz sampling frequency and a flow coefficient that is perturbed by oscillating the compressor throttle ring at approximately 1.53 Hz. The frequency amplitude spectrums for the measured pressure rise coefficient ψ data and the measured flow coefficient ϕ data are shown in Figure 4.19 and Figure 4.20, respectively.

In Figure 4.19 and Figure 4.20, a distinct frequency peak can be observed at the throttle oscillation frequency of 1.53 Hz. Another frequency peak occurs at approximately 40.94 Hz due to the rotor shaft rotation. The rotor shaft rotates at 2400 rpm meaning that the rotor rotates at a frequency speed of approximately 40 Hz. This 40 Hz frequency is known as the blade passing frequency (BPF). Multiples of the BPF, occurring at around 81.9 Hz, 122.8 Hz, 163.8 Hz, etc., can also be observed in this data. Additional frequencies caused either by noise in the data or the natural compressor system dynamics also affect the measured test data signal for the pressure rise coefficient and flow coefficient and should appear in the amplitude spectrum.

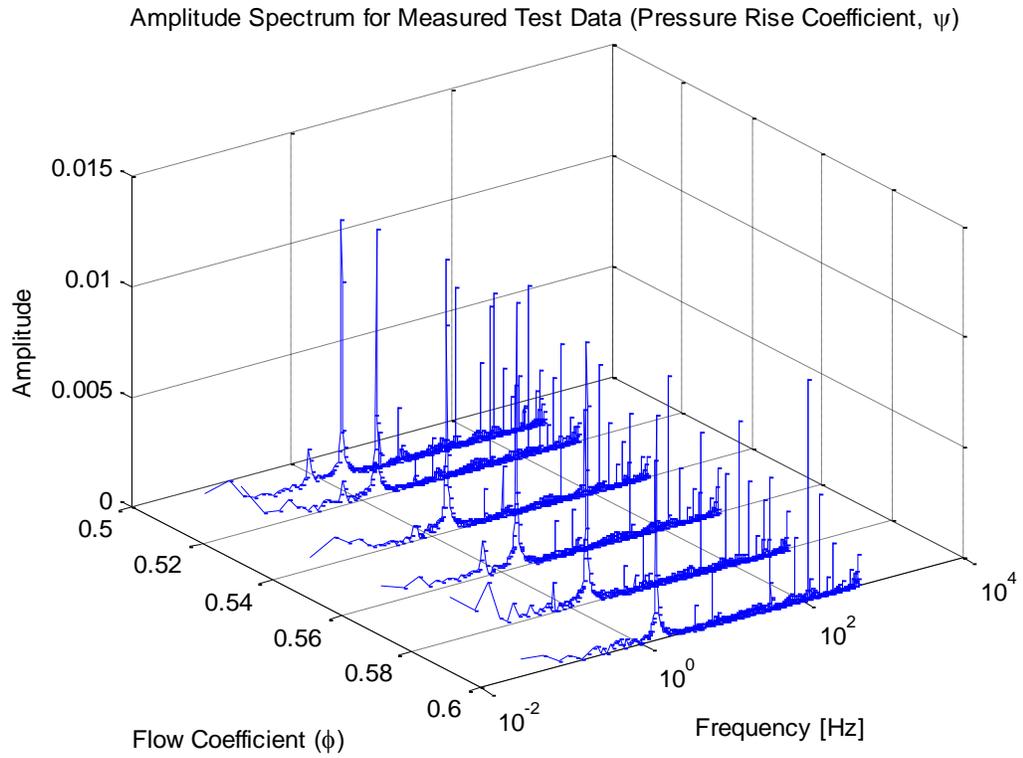


Figure 4.19: Frequency Amplitude Spectrum for Measured Pressure Rise Coefficient Data

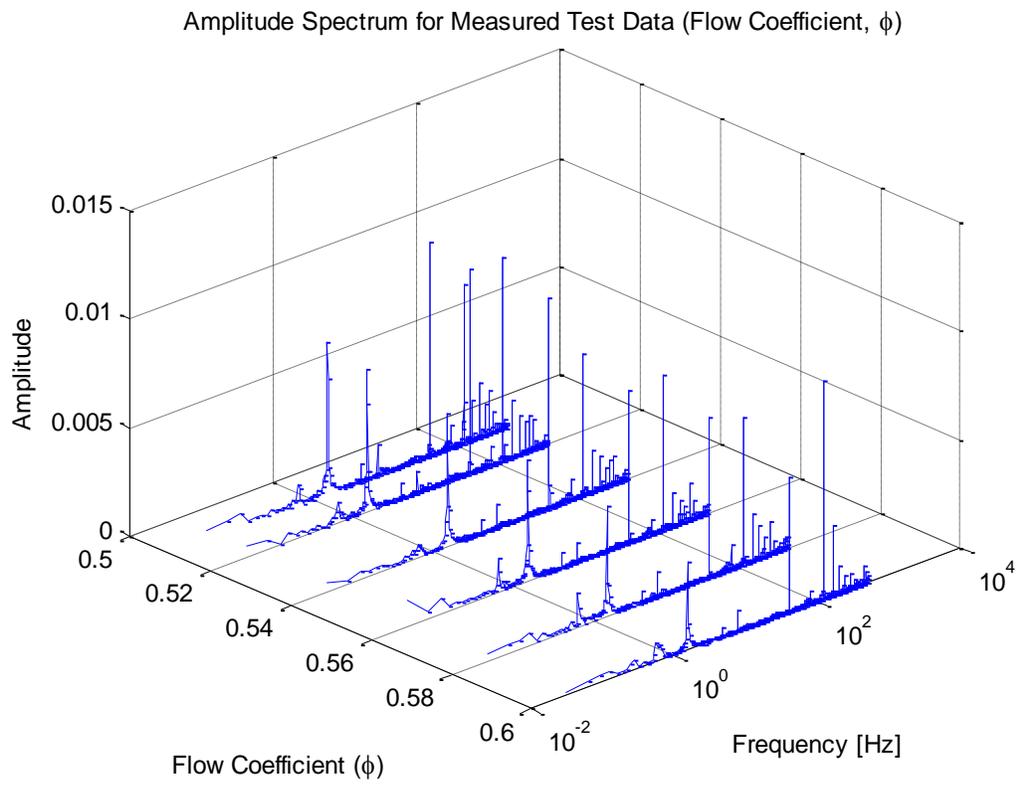


Figure 4.20: Frequency Amplitude Spectrum for Measured Flow Coefficient Data

The next step after acquiring the measured data is to filter the data to eliminate the noise that occurs in the measured signals. The Bayesian filter that is applied to the data has parameter values of α equal to 0.1 and β equal to 150. These values are chosen empirically by testing out different parameters until a data signal with visibly less high frequency noise is obtained. Figure 4.21 and Figure 4.22 show the frequency amplitude spectrums for the BF pressure rise coefficient and BF flow coefficient, respectively. The BF data shows a clear decrease in the amplitude of some of the high frequency noise when compared with the raw measured data in Figure 4.19 and Figure 4.20.

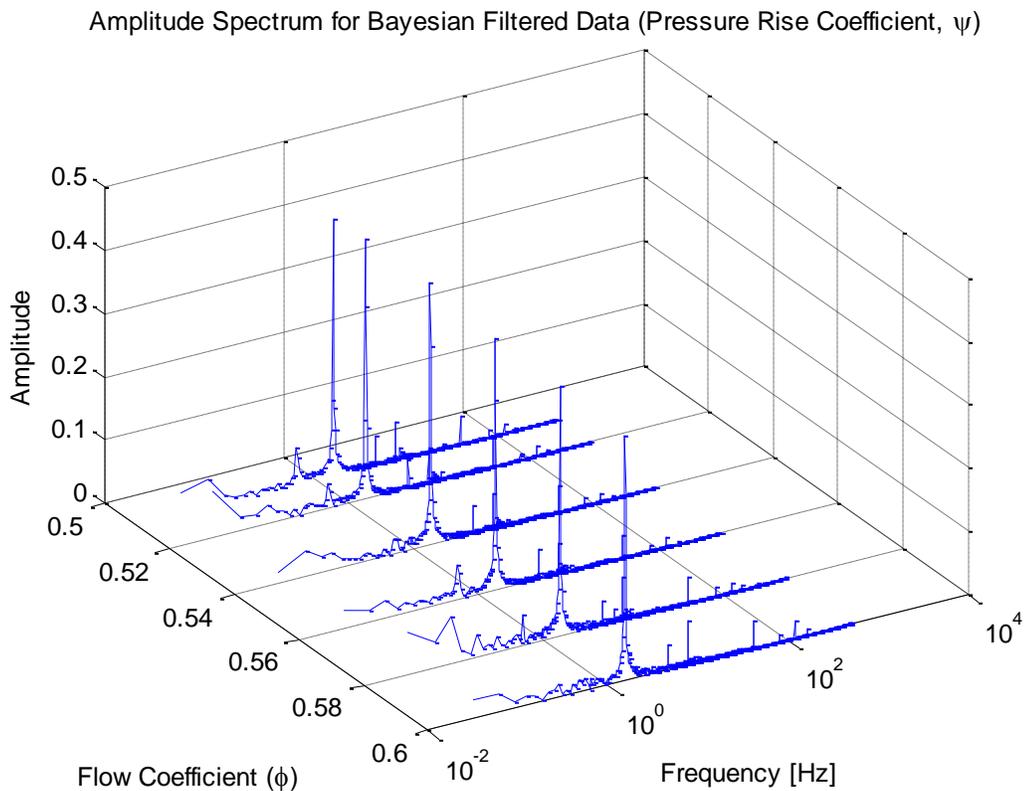


Figure 4.21: Frequency Amplitude Spectrum for the Bayesian Filtered Pressure Rise Coefficient Data

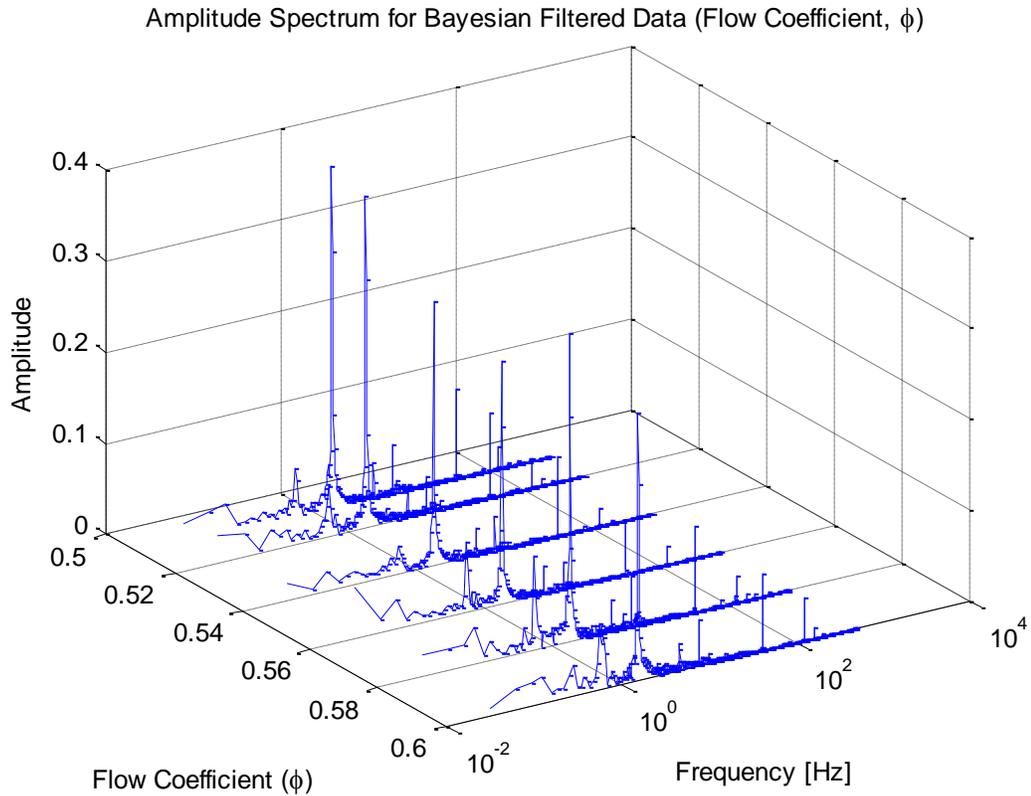


Figure 4.22: Frequency Amplitude Spectrum for the Bayesian Filtered Flow Coefficient Data

The BF data with parameter values of α equal to 0.1 and β equal to 150 is then used to create transfer function models at each of the flow coefficient set points. For these transfer functions, the flow coefficient is used as the input and the pressure rise coefficient is used as the output. The system identification to determine these transfer functions is conducted using the system identification toolbox in MatlabTM [54]. As mentioned before, the sampling frequency of this data is 1000 Hz and is perturbed by a throttle oscillation of 1.53 Hz. Figure 4.23 shows the magnitude Bode plots for the transfer function models obtained at each of the flow coefficient set points.

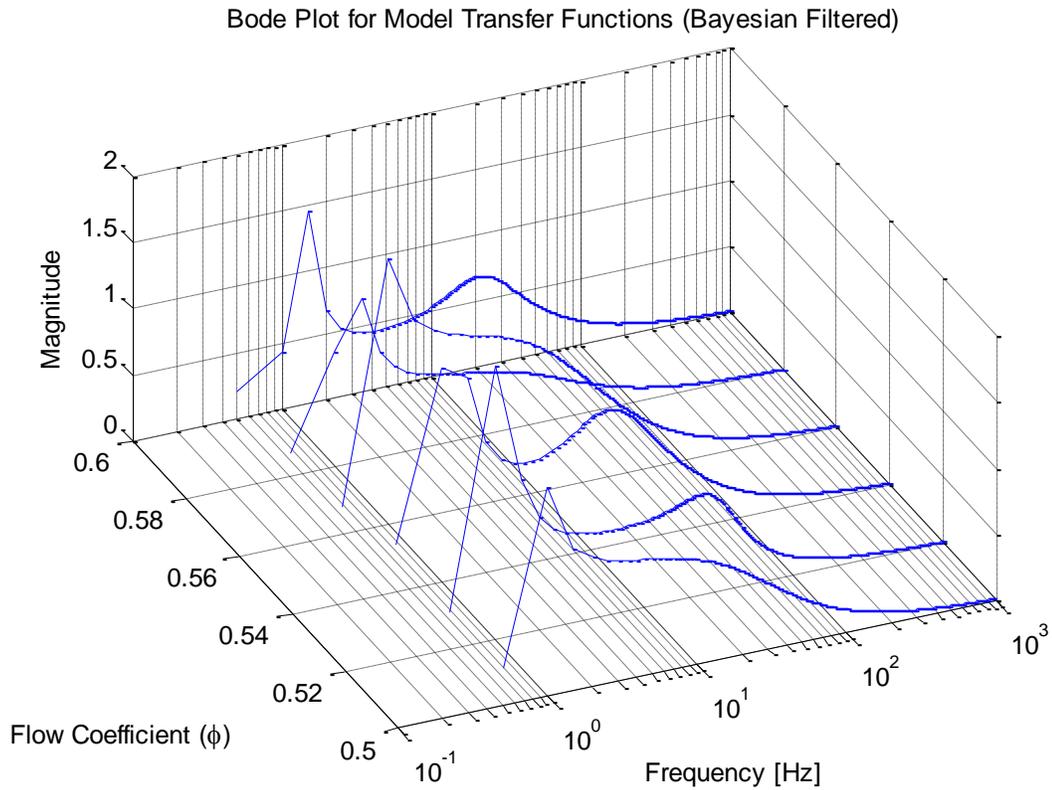


Figure 4.23: Magnitude Bode Plots for the Model Transfer Functions obtained from the BF Data

The natural frequency of the compressor system is believed to be approximately 16 to 17 Hz. As seen in the magnitude Bode plots of Figure 4.23, there does not appear to be a significant peak at this frequency. Based on these Bode plots, it is concluded that the transfer function models for the data are insufficient for producing a useful model of the compressor dynamics because the natural frequency of the system is not excited. The throttle ring in the IET experimental setup is only able to be actuated at 1.53 Hz. Thus, it is necessary to create a throttle capable of being actuated at a frequency above 16 to 17 Hz in order to excite the natural frequency as well as higher frequency modes of the compressor system.

CHAPTER 5 DEVELOPMENT OF FAST DYNAMICS

MODEL

5.1 Dynamic Pressure Sensors and Air Injection

The research of Li et al. [51] provides a basis for which dynamic pressure sensors are the best suited for correlation coefficient calculations. The research in this document uses the same dynamic pressure sensor setup as was used by Li and his colleagues. An explanation of why specific sensors are chosen is described in the following paragraphs. The dynamic pressure sensors as they are positioned in the compressor casing relative to the rotor blades is shown in Figure 5.1. A photograph of these pressure sensors is previously shown in Figure 2.9. The exact axial distances of the pressure sensors relative to the leading edge of the blade are presented in Table 5.1 in units of millimeters (mm) as well as a percentage of the axial rotor length. Negative values in Table 5.1 indicate a distance axially ahead of the leading edge of the rotor blade.

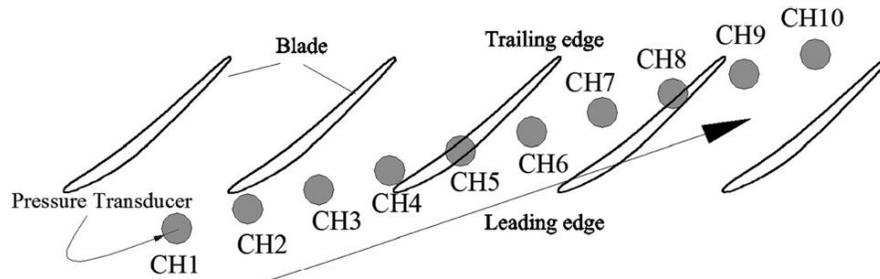


Figure 5.1: Dynamic Pressure Sensor Locations Relative to Rotor Blades

Table 5.1: Dynamic Pressure Sensor Location Distances

Sensor Number	Axial Distance from Leading Edge of Rotor Blade (mm)	Axial Distance from Leading Edge of Rotor Blade (% of Axial Rotor Length)
Sensor CH1 (Leading)	-5.33	-23.71
Sensor CH2	-1.79	-7.96
Sensor CH3	1.14	5.07
Sensor CH4	4.295	19.10
Sensor CH5	7.08	31.49
Sensor CH6	10.99	48.89
Sensor CH7	13.705	60.96
Sensor CH8	17.09	76.02
Sensor CH9 (Trailing)	19.76	87.90

When modeling the compressor dynamics, it is important to ensure that the natural frequency of the compressor system is included in the model. For the axial compressor at IET, the natural frequency lies somewhere in the frequency band range of 13 to 19 Hz. The dynamic pressure sensors in which the natural frequency is at its highest magnitude is consequently the best suited for collecting data for model development and system identification. The power spectral density (PSD) plot for sensors CH1 through CH10 for a point near stall is shown in Figure 5.2 [51].

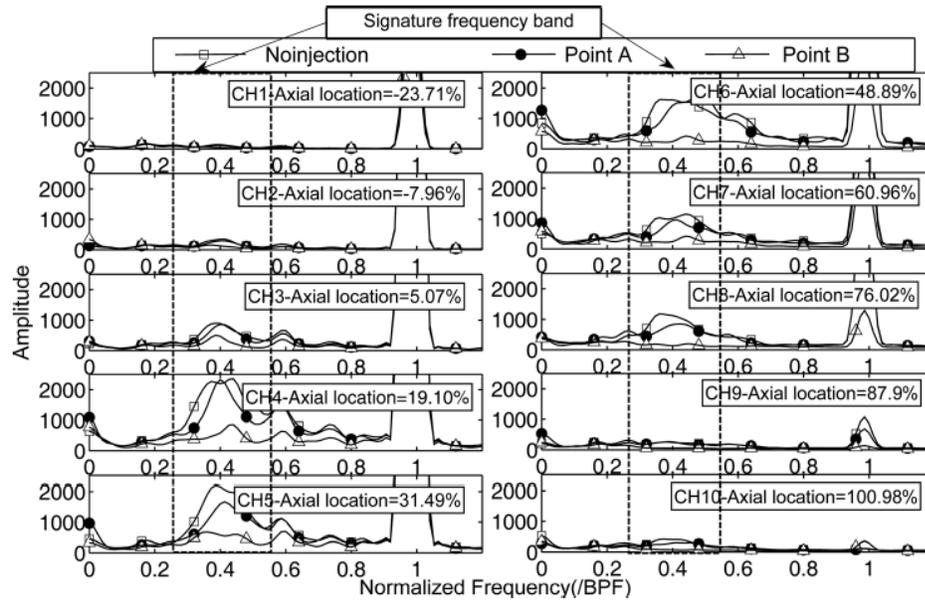


Figure 5.2: Power Spectral Density for Dynamic Pressure Sensors Near Stall Point [51]

In Figure 5.2, the frequency is normalized to the blade passing frequency (BPF). The compressor natural frequency, when normalized to the BPF, is in the vicinity of 0.4. There is a sharp peak at 1 for all the sensors, which is of course the BPF normalized to itself. As can be clearly seen from Figure 5.2, there is also a relatively high peak at the normalized natural frequency near 0.4 for sensors CH4, CH5, and CH6 when no air injection is being used. This indicates that sensors CH4, CH5, and CH6 have the highest magnitude for the compressor natural frequency and are consequently chosen to collect data for system identification.

In order to produce a useful set of pressure data for system identification, the system first has to be excited through the use of an input to the system. This excitation

input is provided by eight air injection actuators located on the casing around the rotor blade section. The air injection actuators are intended to be semi-randomly activated in order to excite a multitude of frequencies in the system. However, the air injection jets are actuated by a human operator rather than a pseudo-random algorithm. This semi-random excitation creates an approximation for a pseudo-random binary input and appears to excite the system frequencies reasonably well. An example of how the average air injection dynamic pressure signal looks for a typical experiment is shown in Figure 5.3. As can be seen in this figure, when the air injection jets are opened, there is a large spike in the dynamic pressure signal. This spike then drops sharply and settles down to a constant value before the air injection jets are closed.

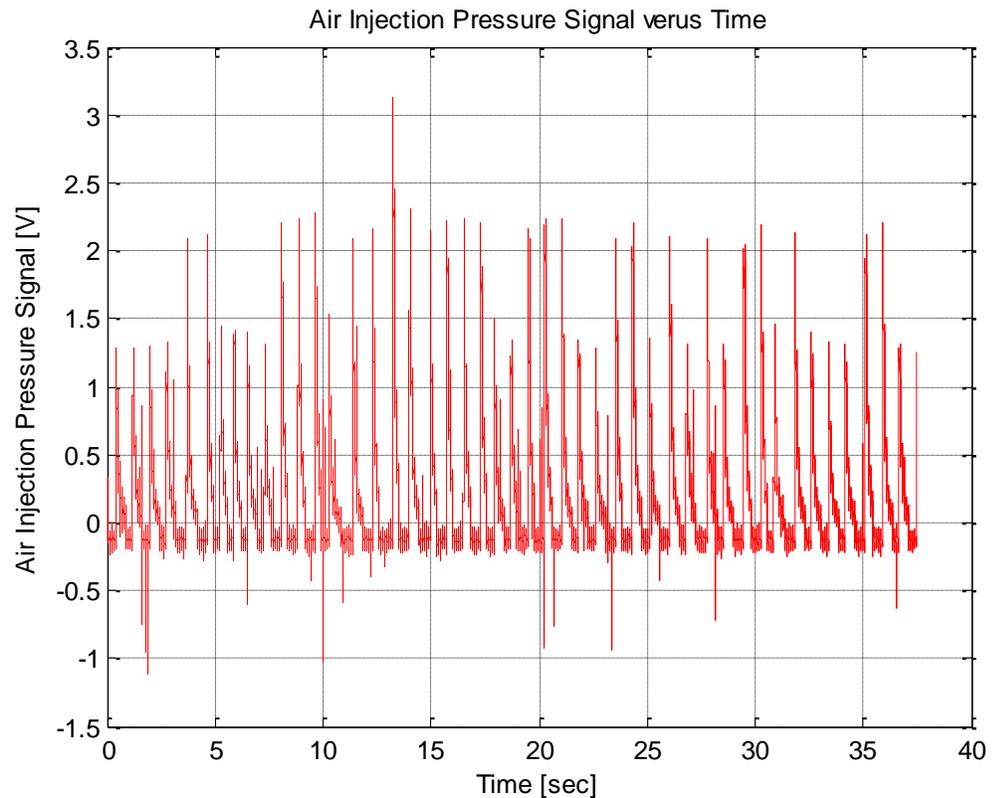


Figure 5.3: Typical Air Injection Pressure Signal versus Time

5.2 Autocorrelation Coefficient Calculation

The next step in the fast dynamics analysis is to calculate the autocorrelation coefficient of the pressure sensor signals. The autocorrelation coefficient is calculated to determine the repeatability between consecutive data point groups in the pressure sensor signals along the rotor blades and is intended to reduce the effect of noise in the signals. In order to do this, the usable data collected from the dynamic pressure sensors must first be extracted. The Hall effect sensor signal, an example of which is shown in Figure 2.11, is used to determine when the rotor has completed a revolution and returned to the same rotor position relative to the pressure sensors. Approximately 490 data points are collected for each individual sensor over one revolution, with the sampling frequency of the dynamic pressure sensors being equal to 20 kHz. However, only 30 of these data points are collected from the specified rotor position in one revolution. These 30 data point groups, which are collected at each revolution, are subsequently used in the autocorrelation coefficient calculation.

The equation for calculating the autocorrelation coefficient is shown below in Equation (5.1).

$$R_{xx}(t) = \frac{\sum_{i=1}^n (X_i(t) - \bar{X}(t))(X_i(t-1) - \bar{X}(t-1))}{\sqrt{\sum_{i=1}^n (X_i(t) - \bar{X}(t))^2} \sqrt{\sum_{i=1}^n (X_i(t-1) - \bar{X}(t-1))^2}} \quad (5.1)$$

In this equation, $R_{xx}(t)$ represents the autocorrelation coefficient as a function of time t between two consecutive data groups at time indices of t and $t-1$, each containing n points. These two consecutive n point data groups are represented by X at a given time t and at the previous time index $t-1$. More specifically, X_i represents the i th data point in each respective data group and \bar{X} represents the average values of their respective data groups. For this analysis, the data groups at each time index contain 30 data points, which implies that n is equal to 30.

The autocorrelation coefficient is calculated for the data collected from sensors CH1 through CH9. Figure 5.4, Figure 5.5, and Figure 5.6 show the autocorrelation coefficient plots for sensors CH1 through CH9 for a flow coefficient equal to 0.56. The data observed in these figures is fairly typical of how the autocorrelation coefficient

behaves at each tested flow coefficient. Sensors CH4, CH5, and CH6 appear to have a mean value closest to unity, implying a high correlation, as well as the lowest variance. This is another reason sensors CH4, CH5, and CH6 are chosen for the system identification analysis. On the other hand, sensors CH8 and CH9 have the lowest mean autocorrelation value and the largest variance. This is most likely due to the fact that sensors CH8 and CH9 are located nearest to the trailing edge of the rotor blades and are consequently more susceptible to turbulence and vortices that may occur in the blade passage flow.

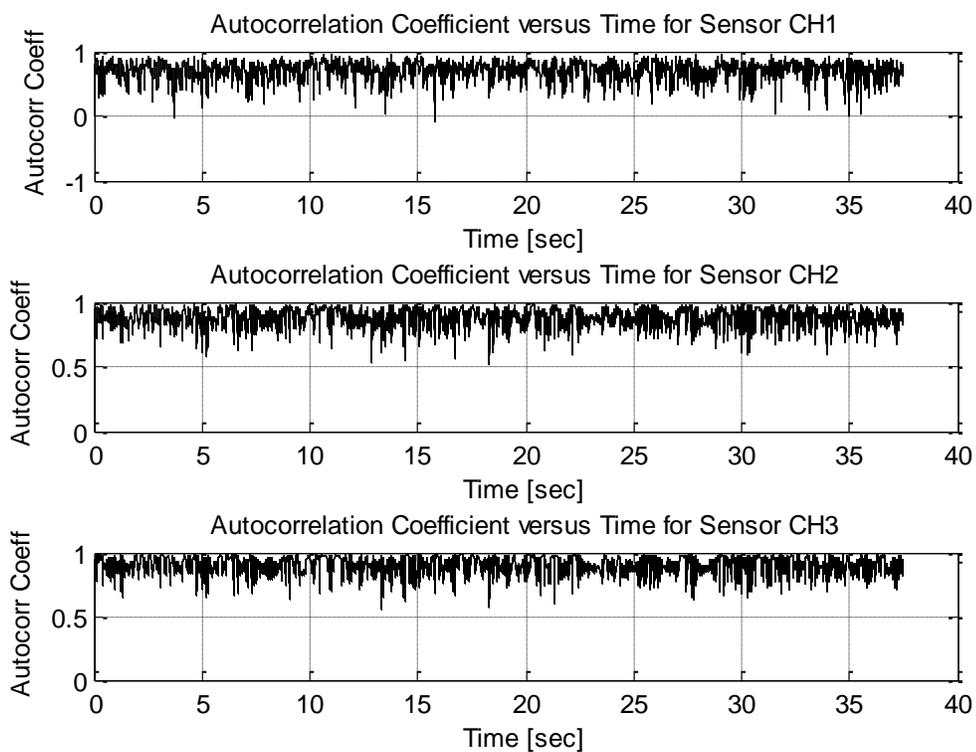


Figure 5.4: Autocorrelation Coefficient versus Time for Sensors CH1, CH2, and CH3 at $\phi = 0.56$

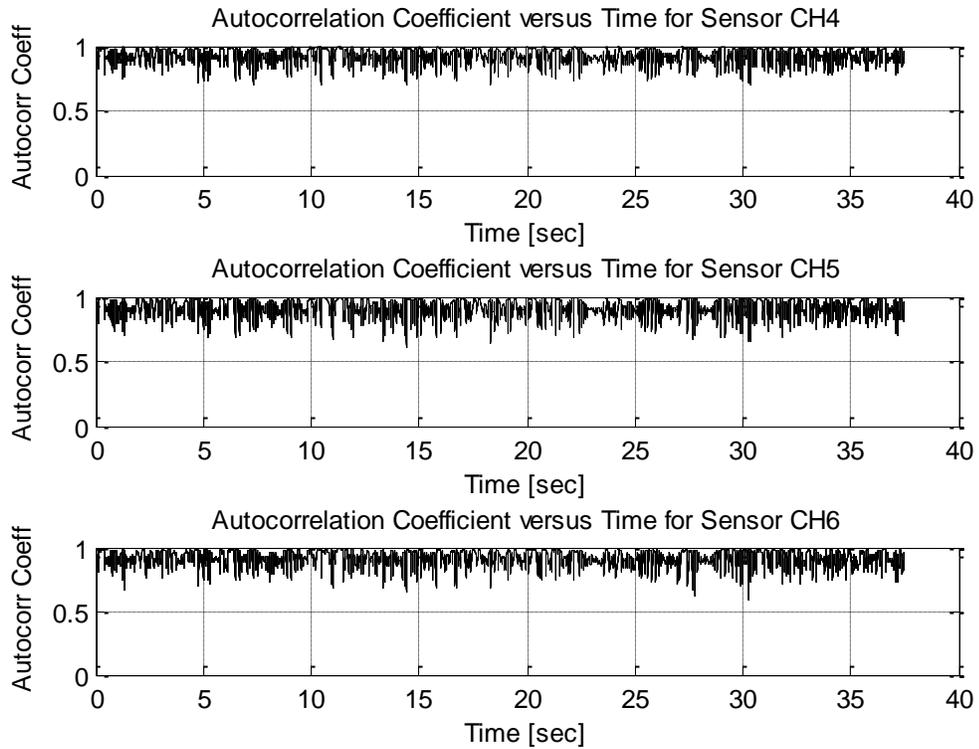


Figure 5.5: Autocorrelation Coefficient versus Time for Sensors CH4, CH5, and CH6 at $\phi = 0.56$

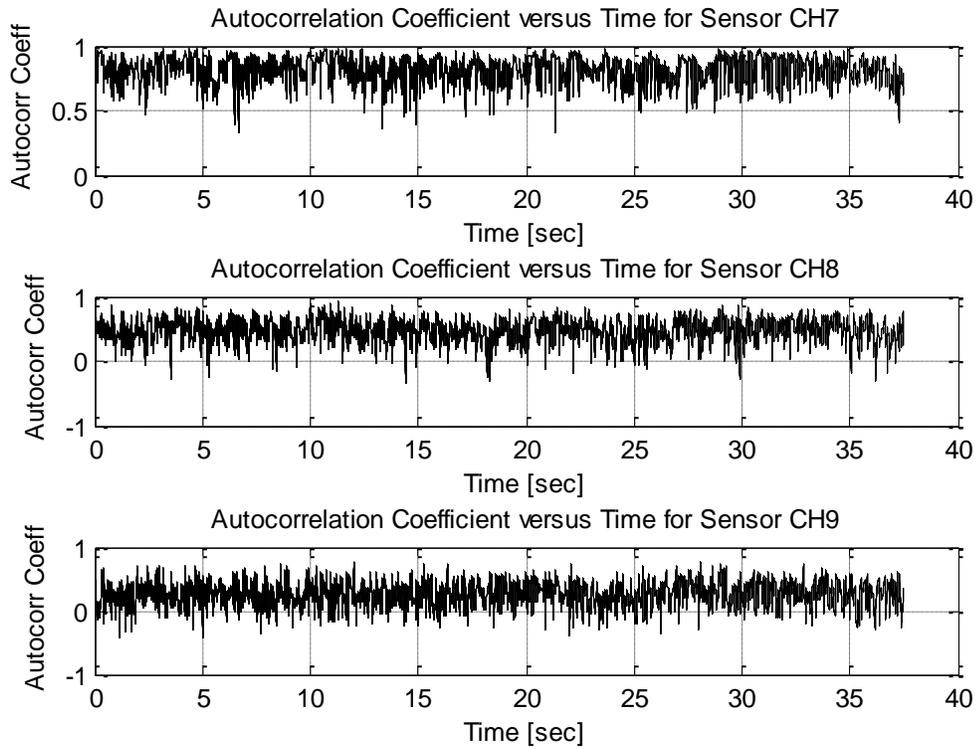


Figure 5.6: Autocorrelation Coefficient versus Time for Sensors CH7, CH8, and CH9 at $\phi = 0.56$

5.3 Missing Data and Cubic Spline Interpolation

Once the autocorrelation coefficient is calculated between each of the 30 data point groups, there is essentially one data point at each time index t for every 490 points that are collected from each pressure sensor during one revolution. This creates a large deficiency in the amount of actual data that is used, with approximately only 0.2 percent of the data actually being used. This deficit leads to the problem of missing data. In order to deal with the missing data problem, several approaches are attempted. An ARX model is used to estimate points in between the autocorrelation data points. Another approach is to use down sampling of the data to only make use of the data points corresponding to the autocorrelation coefficient points and simply ignoring the data points in between. However, the method for dealing with the missing data that appears to be the most effective for system identification involves using a simple cubic spline curve to interpolate the data points in between the autocorrelation data points. The Matlab™ [54] function “spline” is used to calculate this cubic spline interpolation for the missing data.

This new spline curve can be resampled to produce a new data set. The resampled data set is then used in system identification, with the measured pressure signal from the air injectors being the input and the autocorrelation coefficient spline curve being the output. Rather than using the original sampling frequency of 20 kHz, the sampling frequency is reduced to 1 kHz in order to reduce the time required for system identification computations. Figure 5.7 summarizes the approach that is used to prepare the fast dynamics data for system identification. Additionally, the mean of the spline curve is subtracted to more effectively produce a system identified model. Figure 5.9 shows an example of the linear interpolated autocorrelation coefficient data for $\phi = 0.56$ at sensor CH5 to compare with the spline interpolated data shown in Figure 5.8. The interpolated data in these figures is taken over a two second time interval. Lastly, the Matlab™ [54] program that is used to remove unusable data, calculate the autocorrelation coefficient, and create and resample the cubic spline is included in Appendix B1.

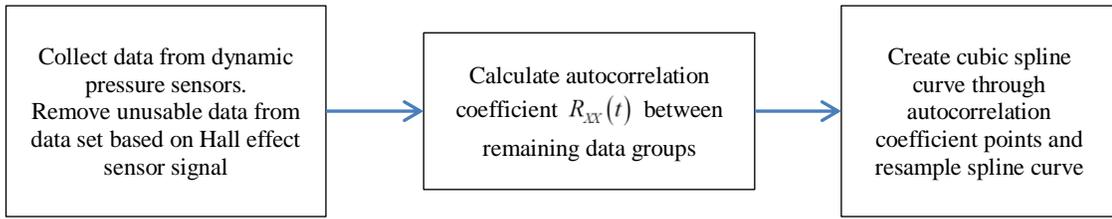


Figure 5.7: Approach for Preparing Fast Dynamics Data for System Identification

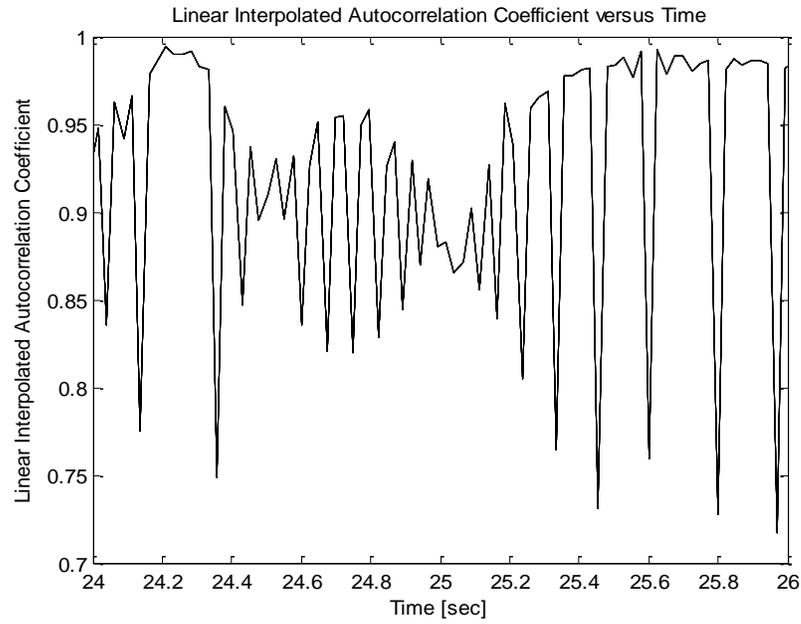


Figure 5.9: Linear Interpolated Autocorrelation Coefficient versus Time for $\phi = 0.56$ at Sensor CH5

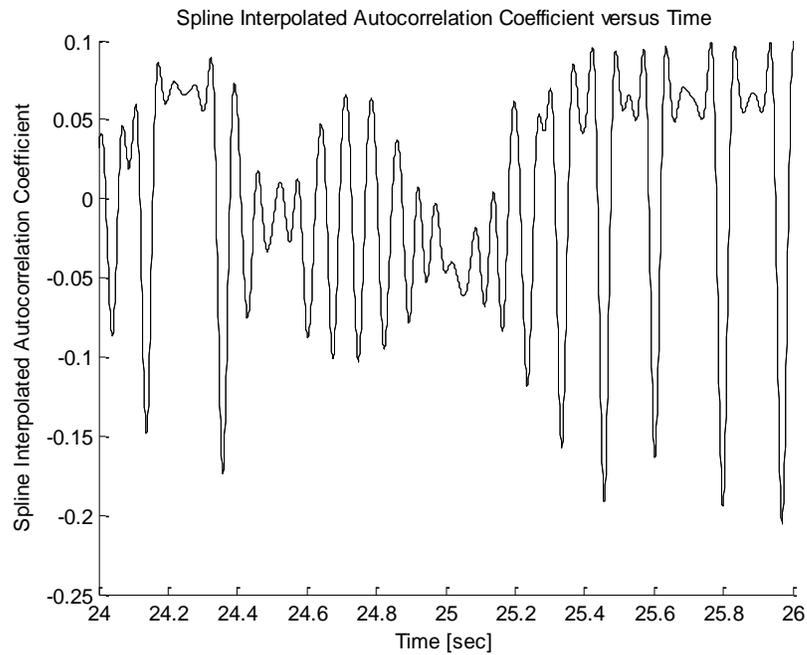


Figure 5.8: Spline Interpolated Autocorrelation Coefficient versus Time for $\phi = 0.56$ at Sensor CH5

5.4 Fast Dynamics System Identification

As in the slow dynamics system identification, the Matlab™ [54] system identification toolbox is used to carry out the system identification analysis. The system identification toolbox uses a numerical algorithm to iteratively optimize the goodness of fit (shown in Equation (4.1)) of a specified model for an input-output data set. For the fast dynamics system identification, the input is the pressure sensor signal from the air injection actuator jet designated by the variable U . The outputs for the system identification are the resampled data sets from the spline curve fitted to the autocorrelation coefficient data points for pressure sensors CH1 to CH9. The outputs for the system identification are designated by the variables Y_n where n is the sensor number and is equal to $1, 2, \dots, 9$. As there is only one input and multiple dynamic pressure sensor outputs, this system can be categorized as a single input, multiple output (SIMO) type system. It is hoped that a single input, single output (SISO) system can eventually be obtained through further research.

Several types of models are attempted to be fitted to the input and outputs. These models included ARX, state-space, and primarily discrete and continuous time transfer function models. Several pole and zero combinations are also attempted for the transfer function models and it is found that a three zero and four pole transfer function produced the best fit for the data, while also including the compressor natural frequency. A transfer function $G_n(s)$ is developed at each of the 9 sensors. The form of these transfer functions, as a function of the complex frequency domain variable s , related to the input and output variables is shown in Equation (5.2).

$$\frac{Y_n(s)}{U(s)} = G_n(s) \quad (5.2)$$

The transfer function equation can also be written in terms of the numerator and denominator polynomials as shown below in Equation (5.3).

$$G_n(s) = \frac{b_{3n}s^3 + b_{2n}s^2 + b_{1n}s + b_{0n}}{s^4 + a_{3n}s^3 + a_{2n}s^2 + a_{1n}s + a_{0n}} \quad (5.3)$$

The numerator is made up of a third order polynomial with constant coefficients of b_{0n} , b_{1n} , b_{2n} , and b_{3n} , and the denominator is made up of a fourth order polynomial with

constant coefficients of a_{0n} , a_{1n} , a_{2n} , and a_{3n} . For these constant coefficients, n is again the sensor number.

The system identification block diagram structure for the SIMO system is represented in Figure 5.10. Of the 9 sensor outputs, CH4, CH5, and CH6 are chosen as the most suitable for capturing the compressor system fast dynamics in the system identification due to their location between the leading edge of the rotor blade to approximately the mid-chord length. Figure 5.11 shows the transfer function model of the spline interpolated autocorrelation coefficient data for $\phi = 0.56$ at sensor CH5. In this figure, the spline interpolated autocorrelation coefficient data (black) is compared to the transfer function model data (blue). The system identification plots clearly show that the goodness of fit for the transfer function model data is fairly low, generally around 10 percent or less. However, this is to be expected as the data is based on a point source (spatially) within a very noisy environment. Observing the characteristics, in particular the compressor's stall frequency, and comparing these with the extracted system models, it is believed that the true system dynamics are likely captured in these transfer function models. This is because these models appear to be able to capture the compressor natural frequency and also appear to show certain pole trends as the stall point is approached.

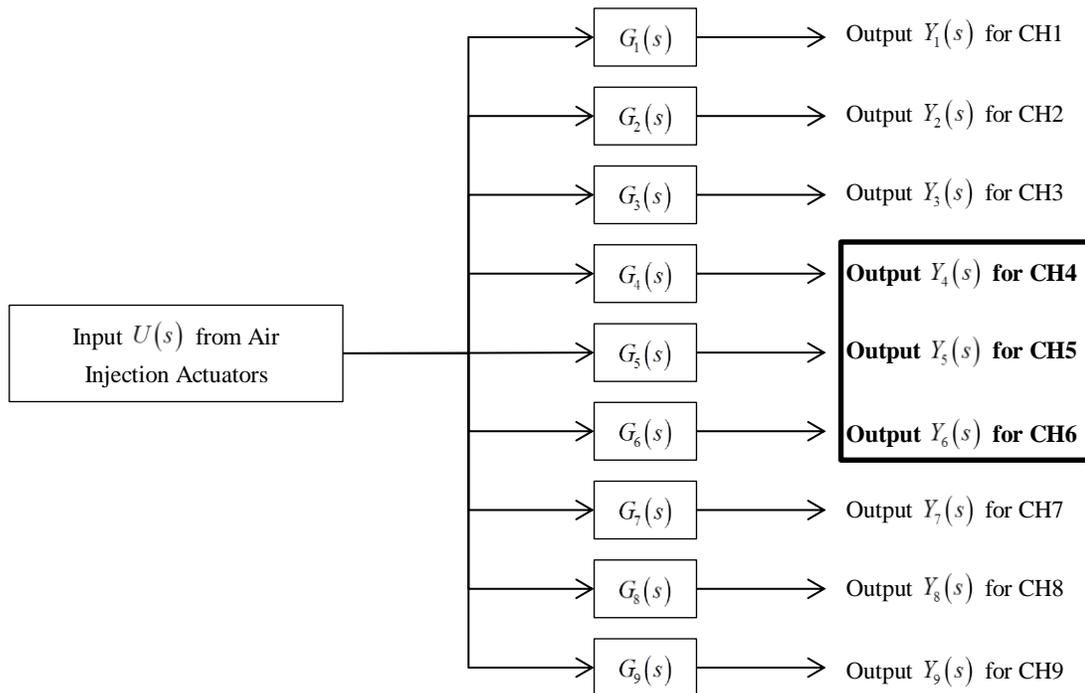


Figure 5.10: Block Diagram Representation of the SIMO System

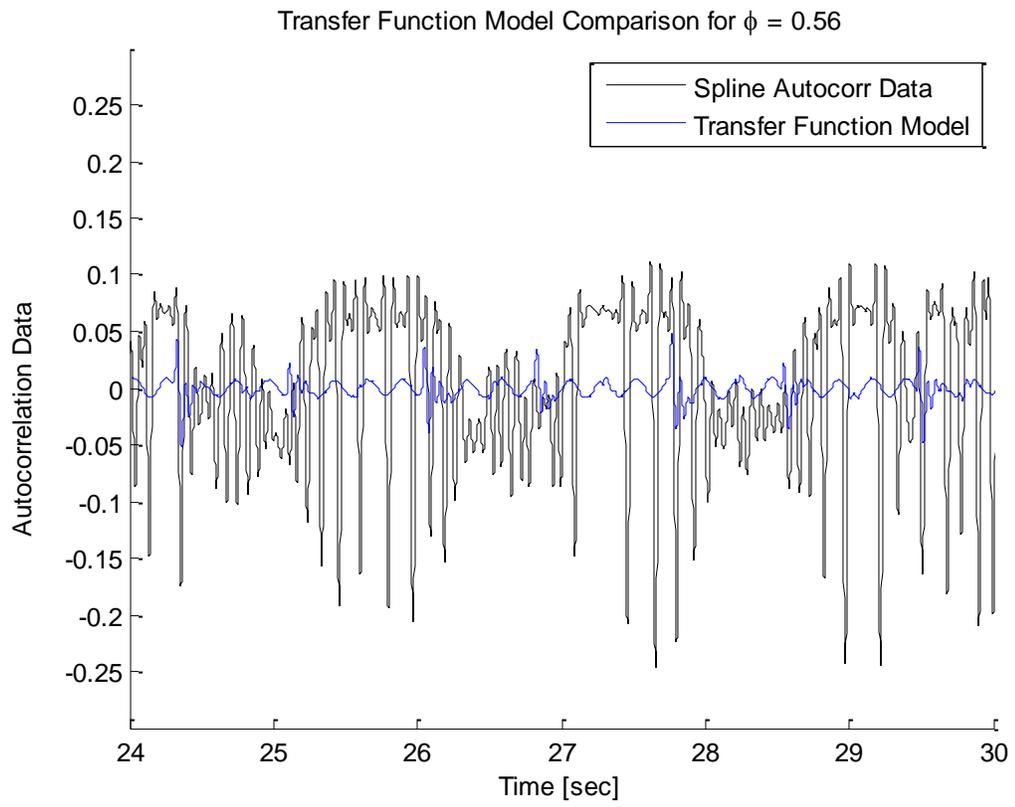


Figure 5.11: Transfer Function Model Comparison for $\phi = 0.56$ at Sensor CH5

CHAPTER 6 RESULTS AND DISCUSSION

6.1 Slow Dynamics Analysis Results

6.1.1 Pole Behavior of Slow Dynamics Transfer Function Models

In addition to Bode plot behavior, the pole locations of the slow dynamics transfer function models are also analyzed. Table 6.1 shows the four poles and three zeros that are obtained for each flow coefficient set point. The natural frequencies f_n of the transfer function model are also calculated from the magnitude of the poles p . The following equation shows the natural frequency calculation in units of Hz for poles with real and imaginary parts.

$$f_n = \frac{\omega_n}{2\pi} = \frac{1}{2\pi} \sqrt{\text{real}(p)^2 + \text{imag}(p)^2} \quad (6.1)$$

In this equation, *real* and *imag* designate the real and imaginary portions of a pole, respectively. ω_n is the natural frequency in units of radians per second.

Further inspection of the pole and zero locations shows some interesting behavior. All poles for the transfer function models are in the left-half of the complex s-plane. This implies that all models are stable. On the other hand, some zeros have a positive real part indicating that these zeros are in the right-half of the complex s-plane. All of the transfer function models have at least one pole in the left-half plane meaning that these models have zeros in both halves of the complex s-plane. Systems with zeros on both halves of the complex s-plane are called mixed-phase systems and can still be stable and causal. If a system has zeros only in the right-half plane, this is indicative of a non-minimum phase system which may lead to complications when developing a control system. However, this does not appear to be the case for the models that are developed.

Table 6.1: Slow Dynamics Transfer Function Model Data

Flow Coefficient	Poles	Zeros	Natural Frequency (Hz)
0.60	-93.14 +103.90i	-59.7136 + 0.0000i	22.208
	-93.14 - 103.90i	-3.9675 + 4.9705i	22.208
	-1.05 + 8.81i	-3.9675 - 4.9705i	1.412
	-1.05 - 8.81i	-	1.412
0.58	-151.31 + 0.0000i	-2.6630 + 2.1010i	24.081*
	-0.90 + 0.0000i	-2.6630 - 2.1010i	0.142*
	-1.27 + 7.61i	-0.9387 + 0.0000i	1.228
	-1.27 - 7.61i	-	1.228
0.56	-98.7756 + 0.0000i	0.2180 + 3.3594i	15.721*
	-0.0000 + 0.0000i	0.2180 - 3.3594i	0.000*
	-0.4920 + 4.6978i	-0.0159 + 0.0000i	0.752
	-0.4920 - 4.6978i	-	0.752
0.54	-70.0642 +55.7367i	-31.9806	14.249
	-70.0642 -55.7367i	-8.5732	14.249
	-1.4062 + 7.1566i	0.1784	1.161
	-1.4062 - 7.1566i	-	1.161
0.52	-99.12 +165.04i	-277.8443	30.640
	-99.12 - 165.04i	-3.7397	30.640
	-1.01 + 6.53i	0.627	1.052
	-1.01 - 6.53i	-	1.052
0.51	-129.62 + 0.0000i	0.5255 + 2.9100i	20.629*
	-0.69 + 0.0000i	0.5255 - 2.9100i	0.110*
	-0.57 + 4.71i	-2.2965 + 0.0000i	0.755
	-0.57 - 4.71i	-	0.755

* The natural frequency of a pole on the real axis is damped out

In the pole location analysis, it is hoped that the transfer function model poles show some sort of trend as the flow coefficient approaches the stall point. Conversely, the actual pole behavior that is observed appeared to be fairly unpredictable and erratic in nature. Henceforth, the four poles are referred to as pole 1, pole 2, pole 3, and pole 4, where pole 3 and pole 4 are the low magnitude poles that are symmetric with each other across the horizontal real axis. Pole 1 and pole 2 are symmetric about the real axis for flow coefficient set points of 0.60, 0.54, and 0.52. For set points of 0.58, 0.56, and 0.51, pole 2 moves along the real axis in the vicinity of the imaginary axis and near instability. See Figure 6.1 for a plot of the pole behavior for pole 1 and pole 2. Pole 3 and pole 4 are symmetric about the real axis, but appear to oscillate erratically as the flow coefficient set point approaches stall. Figure 6.2 demonstrates the behavior for pole 3 and pole 4.

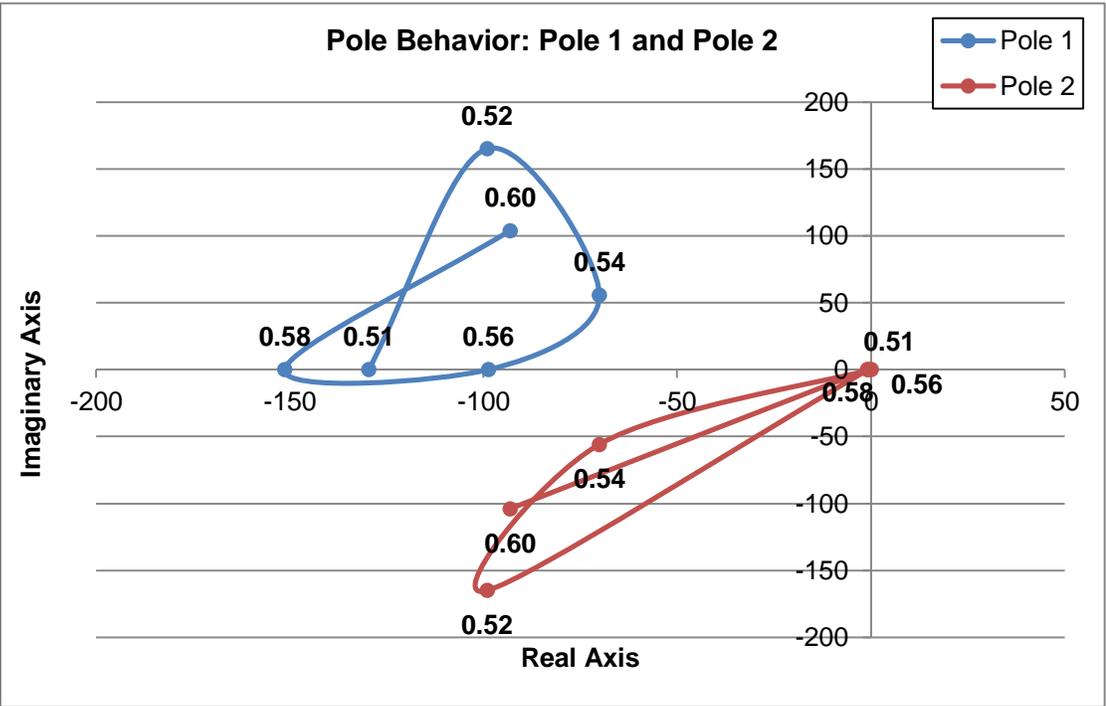


Figure 6.1: Pole Behavior of Poles 1 and 2 for Slow Dynamics Model

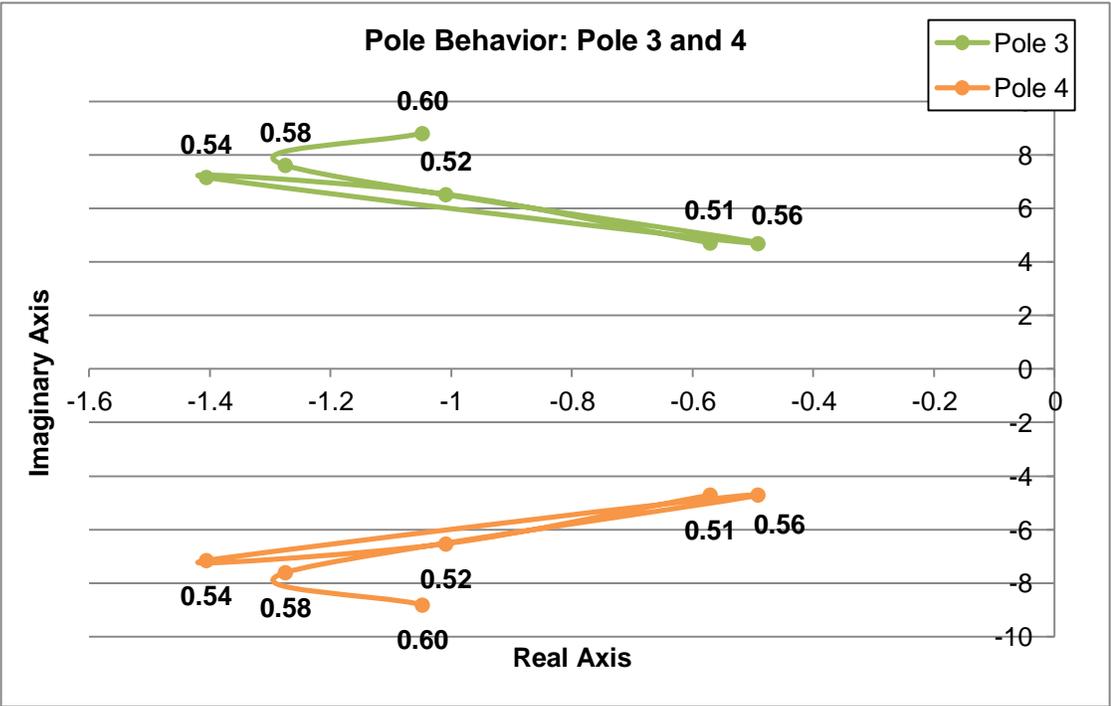


Figure 6.2: Pole Behavior of Poles 3 and 4 for Slow Dynamics Model

The erratic pole location trends in Figure 6.1 and Figure 6.2 are most likely caused by noise in the data as well as uncertainty of the actual measurements. In order to better understand this behavior, a pole variation investigation is conducted. This pole variation investigation attempts to create new models for the data and analyze the covariance estimate of the pole parameters of these models in hopes of quantifying error in the measurement of the data signals. The following section explains the steps taken in order to characterize the uncertainty in the experimental measurements and better understand the pole variation.

6.1.2 Pole Variation Investigation

For the pole variation investigation, a new series of experiments are conducted using the axial compressor setup at the IET. Four flow coefficient set points are selected at 0.58, 0.55, 0.52, and 0.495. Seven throttle oscillation experiments M are then conducted at each of the four flow coefficient set points. These experiments are each truncated to have an equal number of data points set to a value of L . The goal of this pole variation analysis is to develop M autoregressive exogenous (ARX) models for each flow coefficient set point and compare the model parameter sets θ of these ARX models. The details of this investigation are explained in the following paragraphs.

Rather than looking at a realized (state-space or transfer function) system dynamics variation such as $G(j\omega)$, it is easier to investigate the variation of the deterministically obtained parameter coefficient matrix and the resulting pole variations. The proposed approach is outlined in Figure 6.3.

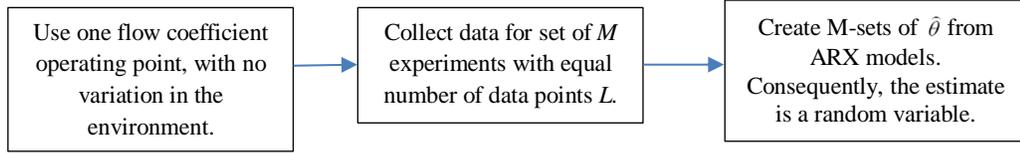


Figure 6.3: Approach for Pole Variation Investigation using Model Parameters of ARX Model

The ARX model that is used for the pole variation investigation is of the following form.

$$y_n(k) = \sum_{i=1}^p a_{i_n} y_n(k-i) + \sum_{i=1}^p b_{i_n} u_n(k-i) + e_n(k) \quad (6.2)$$

The assumption is made that the error follows a Gaussian distribution, $e_n(k) \sim N(0, \sigma^2)$,

$$\text{and } E[e_n(k) * e_m(k-\tau)] = \begin{cases} \sigma_n^2 \delta(\tau) & \text{for } n = m \\ 0 & \text{for } n \neq m \end{cases}, \text{ where } n \text{ is the experiment number.}$$

From this, it follows that the model parameters are a vector containing the ARX model coefficients.

$$\hat{\theta}_n = [a_{1_n} \ b_{1_n} \ a_{2_n} \ b_{2_n} \ \dots \ a_{p_n} \ b_{p_n}]^T \quad (6.3)$$

The model parameters are also assumed to follow a Gaussian distribution, $\hat{\theta}_n \sim N(\theta_0, \Gamma)$, where Γ is the covariance and θ_0 is the mean of the new random variable vector $\hat{\theta}_n$. Also, $\hat{\theta}_n$ and e_n are mutually independent $\forall n$.

The next step of the pole variation investigation is to conduct the M experiments, each collecting the same number of data sets containing L points. M ARX models can then be extracted from the M experiments. In order to keep the variation based only on the actual data scatter from the dynamics and noise, the order must be the same for all of the extracted ARX models. The most effective method of determining the model order is to use some type of statistical criteria to measure the relative quality of a specific model order. For this investigation, the Akaike Information Criterion (AIC) is used to determine the model order p , which is then set as a constant for all M models. The minimum AIC of an ARX model with model order p is used to determine the most statistically suitable model order.

$$p \rightarrow \min(AIC[ARX(k)]) \forall k \quad (6.4)$$

The Matlab™ [54] command “aic” is used to calculate the AIC for each of the ARX models.

Once the model order p is determined, we set this value constant and apply the value to all M models. The average least-squares estimate $\hat{\theta}_0$ of the model parameters can then be calculated as shown below.

$$\hat{\theta}_0 = \frac{1}{M} \sum_{n=1}^M \hat{\theta}_n^{LS} \quad (6.5)$$

In the previous equation, $\hat{\theta}_n^{LS}$ represents the least-squares estimate of the n th model. The average least-squares estimate can then be used in the calculation of the empirical unbiased covariance estimate. The equation for the empirical unbiased covariance estimate is shown in Equation (6.6).

$$\hat{\Gamma}^E = \frac{1}{M-1} \sum_{n=1}^M \left[(\hat{\theta}_n^{LS} - \hat{\theta}_0)(\hat{\theta}_n^{LS} - \hat{\theta}_0)^T \right] \quad (6.6)$$

Since $\hat{\theta}^{LS}$ is now also a random variable vector with its own covariance \hat{P}^{LS} , the extra uncertainty in Equation (6.6) can be removed (or reduced) by subtracting the average of the covariance of each of the $\hat{\theta}^{LS}$ random variable vectors.

$$\hat{\Gamma} = \hat{\Gamma}^E - \frac{1}{M} \sum_{n=1}^M \hat{P}_{\hat{\theta}_n}^{LS} \quad (6.7)$$

The new empirical unbiased covariance estimate $\hat{\Gamma}$ can then be plotted around each of the ARX model poles for each flow coefficient.

Figure 6.4 shows the circles representing the variation at each pole for the four flow coefficient set points ($\phi = 0.58, 0.55, 0.52,$ and 0.495) as stall is approached. More specifically, these circles are plotted in the complex s-plane and centered at each of the model poles. The radii r of these circles is equal to three times the square root of the unbiased covariance estimate or $r = 3\sqrt{\hat{\Gamma}^E}$, where $\hat{\Gamma}^E$ is calculated from Equation (6.6). An additional consideration that may need to be addressed with this estimate is the very finite number of data sets M , due to the limited number of experiments that are run. Ideally, M is large, but this is not practical for running experiments at the IET. If M is small, it may be necessary to modify Γ . For future pole variation investigations, either a larger number of experiments need to be conducted in order to obtain more data sets or some type of shrinkage estimate has to be used to compensate for the small number of data sets.

The results of the pole variation investigation show that the covariance estimate does not appear to significantly increase as stall is approached. The AIC tends to be minimized at larger model orders, so a model order of 10 is chosen as a means of producing a reasonable estimate while also not being too large. Therefore, there are 10 poles at each flow coefficient set point to investigate. The covariance estimates for the poles at a flow coefficient of 0.495 appear to increase due to the fact that only four data sets could be collected before stall occurred. However, even this increase in variation is fairly minimal. A more likely cause of the random erratic behavior of the transfer function model poles for the slow dynamics is that the models are not capturing the correct frequencies that are altered as the compressor stall point is approached. The pole

variation investigation program that is used to perform these calculations is included in Appendix A5.

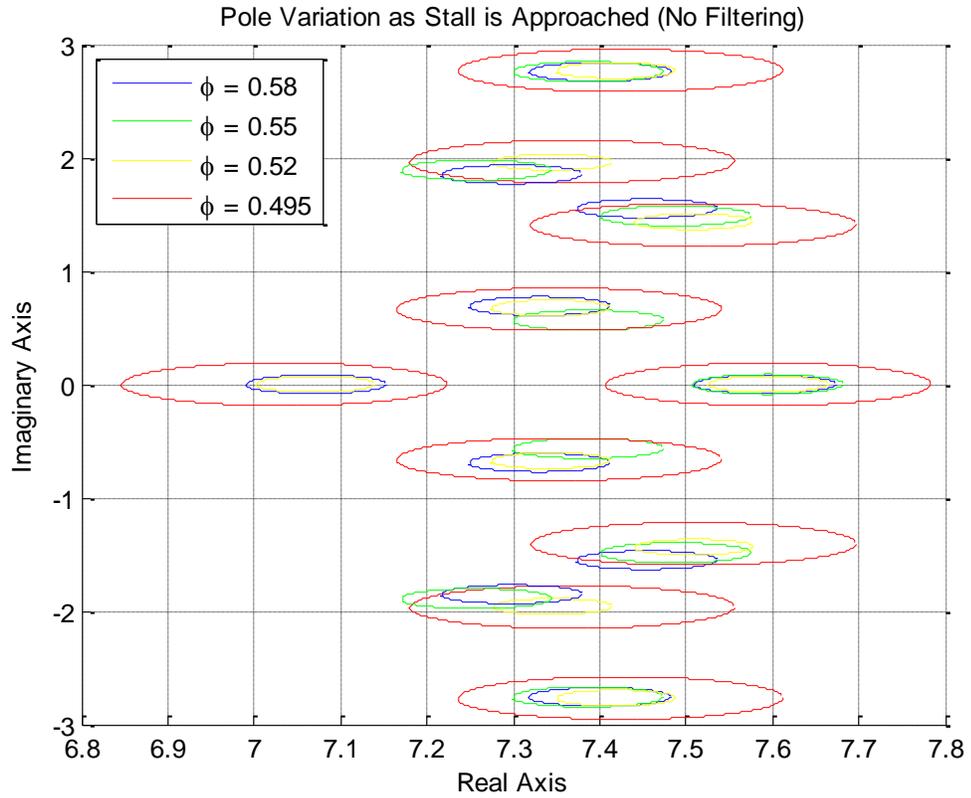


Figure 6.4: Pole Variation as Stall is Approached

6.2 Fast Dynamics Analysis Results

6.2.1 Pole Behavior of Fast Dynamics Transfer Function Models

Two similar experiments are carried out for the fast dynamics analysis. The first experiment (designated Experiment 1) was carried out on July 8, 2014. The second experiment (designated Experiment 2) was performed to compare to the results of Experiment 1 and was carried out on December 25, 2014. Experiment 1 involved collecting dynamic pressure sensor data at the following flow coefficients: 0.58, 0.56, 0.54, 0.52, and 0.51. Experiment 2 involved the collection of dynamic pressure sensor data at the following flow coefficients: 0.58, 0.55, 0.52, and 0.51. Stall occurred for both of these experiments at a flow coefficient approximately equal to 0.50. In these experiments, the air injection actuators are semi-randomly activated during data collection to excite the system. The air injection actuators in Experiment 2 are actuated at a higher average frequency than in Experiment 1. The higher average actuation frequency of Experiment 2 may lead to extra turbulence in the compressor flow and the creation of more significant vortices within the rotor blade passages. This turbulence may add extra error to the data from some sensors.

Sensors located near the leading edge of the rotor blade to approximately mid-chord length are found to be best suited for the fast dynamics system identification analysis. Namely, sensors CH4, CH5, and CH6 are used to collect pressure data for the autocorrelation coefficient calculation and subsequent system identification. The pole behavior analysis of the simple spline curve interpolation of the autocorrelation coefficient points and re-sampling yields some interesting results when system identification is applied. As stated before, the transfer function models empirically chosen to be best suited for modeling the data are three zero and four pole transfer functions. Table 6.2, Table 6.3, and Table 6.4 show the transfer function model data for the system identification of the data from sensors CH4, CH5, and CH6, respectively, for Experiment 1. Similarly, Table 6.5, Table 6.6, and Table 6.7 show the transfer function model data for the system identification of the data from sensors CH4, CH5, and CH6, respectively, for Experiment 2. The natural frequency in this table is calculated from the real and imaginary portions of the transfer function poles using Equation (6.1).

Table 6.2: Fast Dynamics Transfer Function Model Data for Sensor CH4 (Experiment 1)

Flow Coefficient	Poles	Zeros	Natural Frequency (Hz)
0.58	-0.0000 +86.8311i	-133.71 + 0.0000i	13.820
	-0.0000 -86.8311i	-1.80 + 55.55i	13.820
	-7.8998 +20.4642i	-1.80 - 55.55i	3.491
	-7.8998 -20.4642i	-	3.491
0.56	-34.7345 +88.9319i	2.1037 +32.1154i	15.195
	-34.7345 -88.9319i	2.1037 -32.1154i	15.195
	-0.0000 +30.7684i	3.4567 + 0.0000i	4.897
	-0.0000 -30.7684i	-	4.897
0.54	-0.3740 +86.5791i	-5.1659 +28.3489i	13.780
	-0.3740 -86.5791i	-5.1659 -28.3489i	13.780
	-0.0361 +21.9828i	13.1705 + 0.0000i	3.499
	-0.0361 -21.9828i	-	3.499
0.52	-76.0179 +44.0606i	-0.0315 +23.5119i	13.984
	-76.0179 -44.0606i	-0.0315 -23.5119i	13.984
	-0.0347 +23.5191i	6.0286 + 0.0000i	3.743
	-0.0347 -23.5191i	-	3.743
0.51	-79.9413 +30.8245i	1.6096 +26.0384i	13.636
	-79.9413 -30.8245i	1.6096 -26.0384i	13.636
	-0.4161 +25.3275i	1.1757 + 0.0000i	4.032
	-0.4161 -25.3275i	-	4.032

Table 6.3: Fast Dynamics Transfer Function Model Data for Sensor CH5 (Experiment 1)

Flow Coefficient	Poles	Zeros	Natural Frequency (Hz)
0.58	-110.61 + 58.57i	19.8603 + 0.0000i	19.919
	-110.61 - 58.57i	-0.1802 + 7.5664i	19.919
	-0.18 + 7.50i	-0.1802 - 7.5664i	1.194
	-0.18 - 7.50i	-	1.194
0.56	-16.70 + 113.78i	0.0191 +28.1214i	18.303
	-16.70 - 113.78i	0.0191 -28.1214i	18.303
	-0.0000 + 26.95i	9.7887 + 0.0000i	4.289
	-0.0000 - 26.95i	-	4.289
0.54	-0.3206 +86.3066i	48.5023 + 0.0000i	13.736
	-0.3206 -86.3066i	-0.8536 +31.2561i	13.736
	-0.0000 +31.6615i	-0.8536 -31.2561i	5.039
	-0.0000 -31.6615i	-	5.039
0.52	-88.0397 +32.0824i	46.9253 + 0.0000i	14.913
	-88.0397 -32.0824i	0.6126 +22.8844i	14.913
	-0.0226 +23.4365i	0.6126 -22.8844i	3.730
	-0.0226 -23.4365i	-	3.730
0.51	-70.2878 + 0.0000i	0.2443 +25.9897i	11.187*
	-11.3883 + 0.0000i	0.2443 -25.9897i	1.813*
	-0.1389 +25.3343i	0.7730 + 0.0000i	4.032
	-0.1389 -25.3343i	-	4.032

* The natural frequency of a pole on the real axis is damped out

Table 6.4: Fast Dynamics Transfer Function Model Data for Sensor CH6 (Experiment 1)

Flow Coefficient	Poles	Zeros	Natural Frequency (Hz)
0.58	-88.7959 +60.8620i	-0.5506 +18.1437i	17.133
	-88.7959 -60.8620i	-0.5506 -18.1437i	17.133
	-0.0006 +17.4192i	10.7467 + 0.0000i	2.772
	-0.0006 -17.4192i	-	2.772
0.56	-92.4162 +60.8268i	-0.0011 +23.3122i	17.609
	-92.4162 -60.8268i	-0.0011 -23.3122i	17.609
	-0.0019 +23.3145i	10.7301 + 0.0000i	3.711
	-0.0019 -23.3145i	-	3.711
0.54	-65.3445 +52.5376i	-0.2362 +91.8688i	13.344
	-65.3445 -52.5376i	-0.2362 -91.8688i	13.344
	-0.0921 +88.6676i	73.4270 + 0.0000i	14.112
	-0.0921 -88.6676i	-	14.112
0.52	-12.9371 +15.0633i	-7.7264 +95.9229i	3.160
	-12.9371 -15.0633i	-7.7264 -95.9229i	3.160
	-0.0797 +87.4101i	-11.6611 + 0.0000i	13.912
	-0.0797 -87.4101i	-	13.912
0.51	-58.5959 + 0.0000i	4.4825 +76.3377i	9.326*
	-0.8703 + 0.0000i	4.4825 -76.3377i	0.139*
	-0.3802 +81.7580i	-0.0969 + 0.0000i	13.012
	-0.3802 -81.7580i	-	13.012

* The natural frequency of a pole on the real axis is damped out

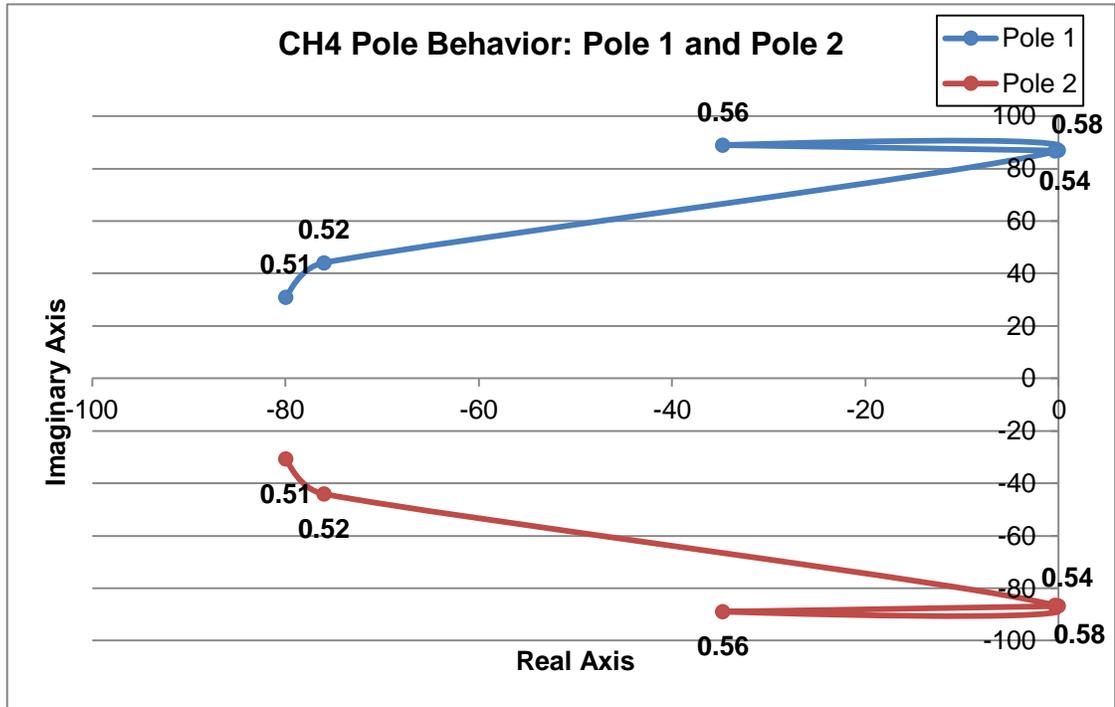


Figure 6.5: Pole Behavior of Poles 1 and 2 for Sensor CH4 (Experiment 1)

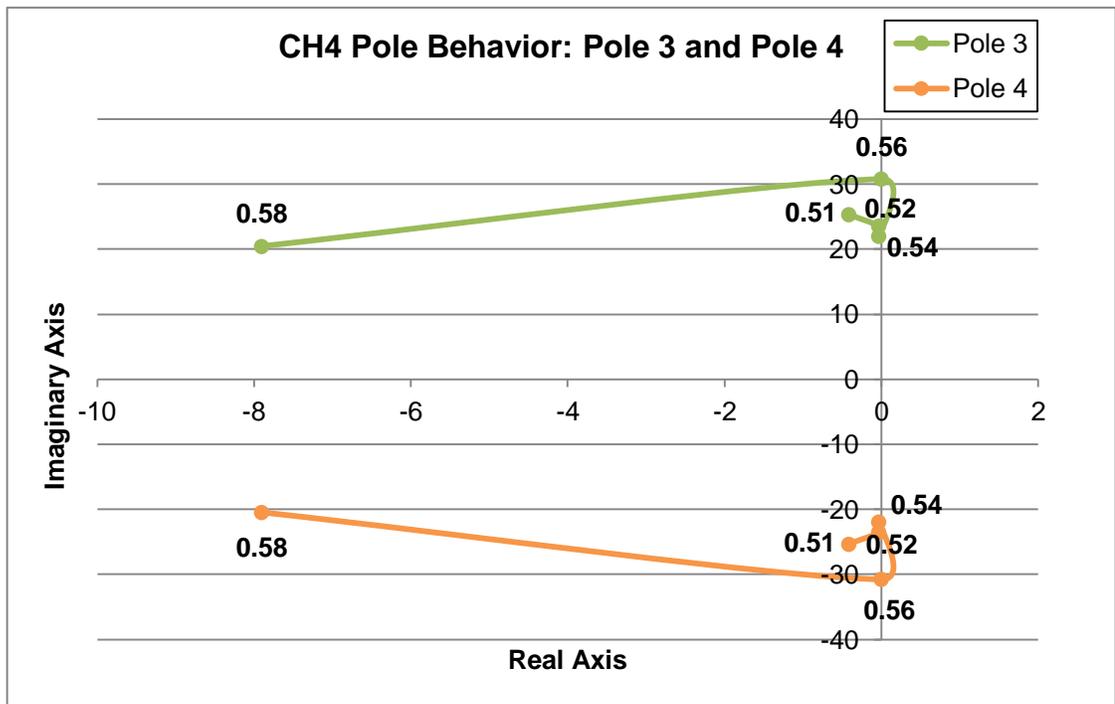


Figure 6.6: Pole Behavior of Poles 3 and 4 for Sensor CH4 (Experiment 1)

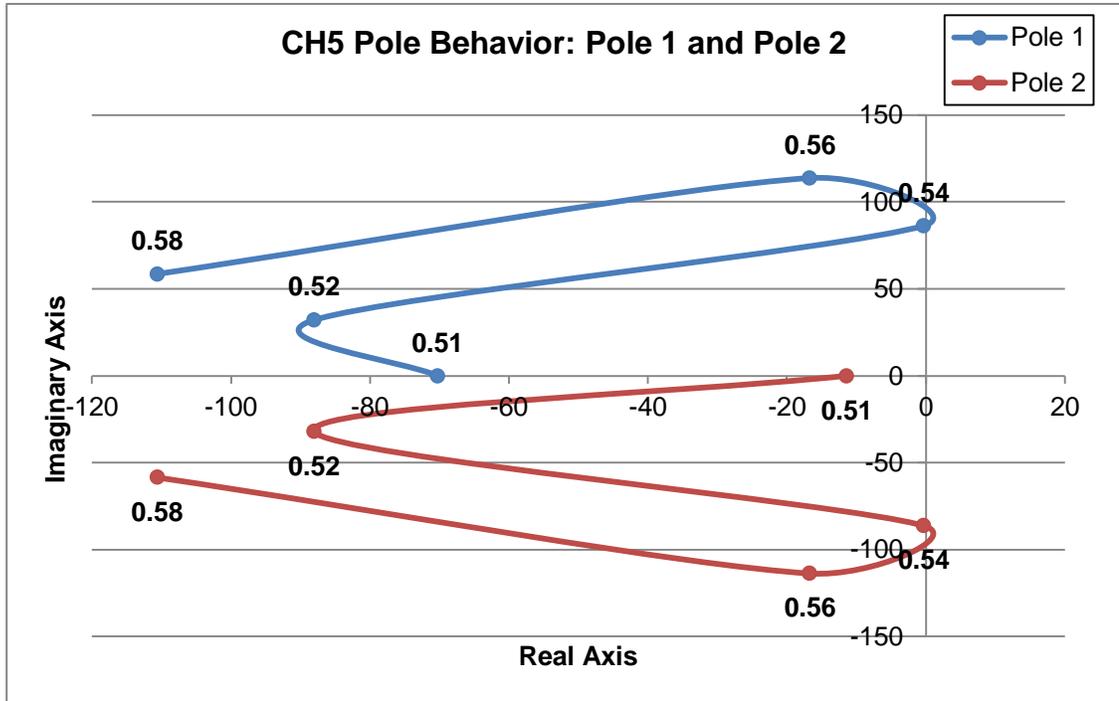


Figure 6.8: Pole Behavior of Poles 1 and 2 for Sensor CH5 (Experiment 1)

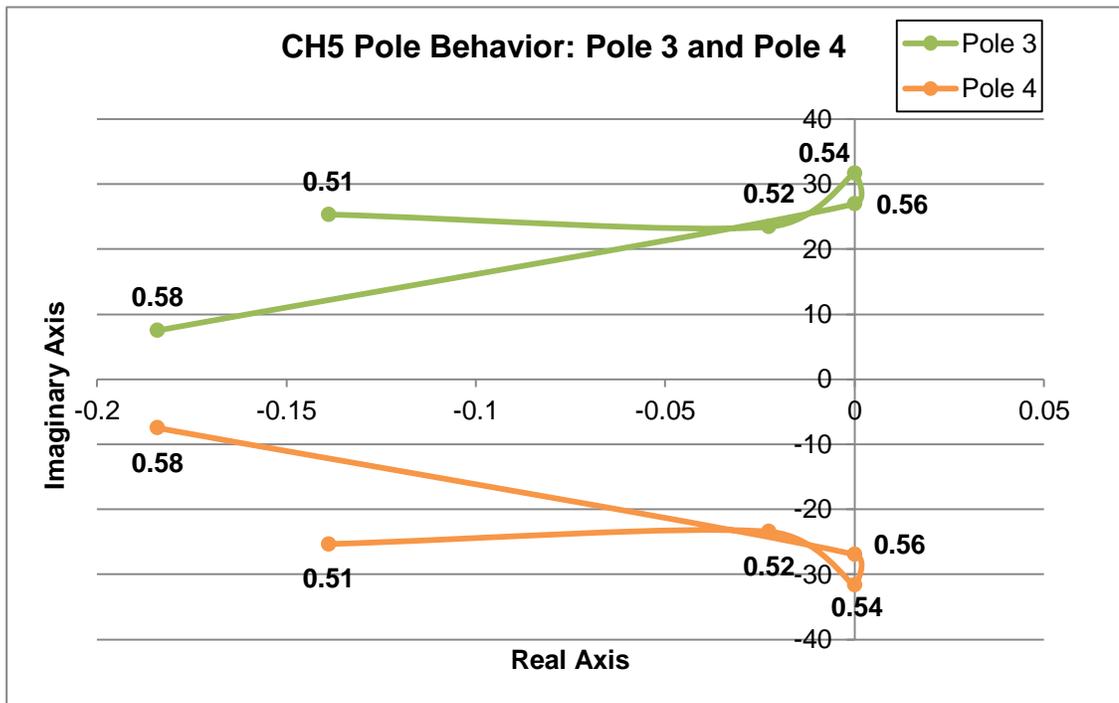


Figure 6.7: Pole Behavior of Poles 3 and 4 for Sensor CH5 (Experiment 1)

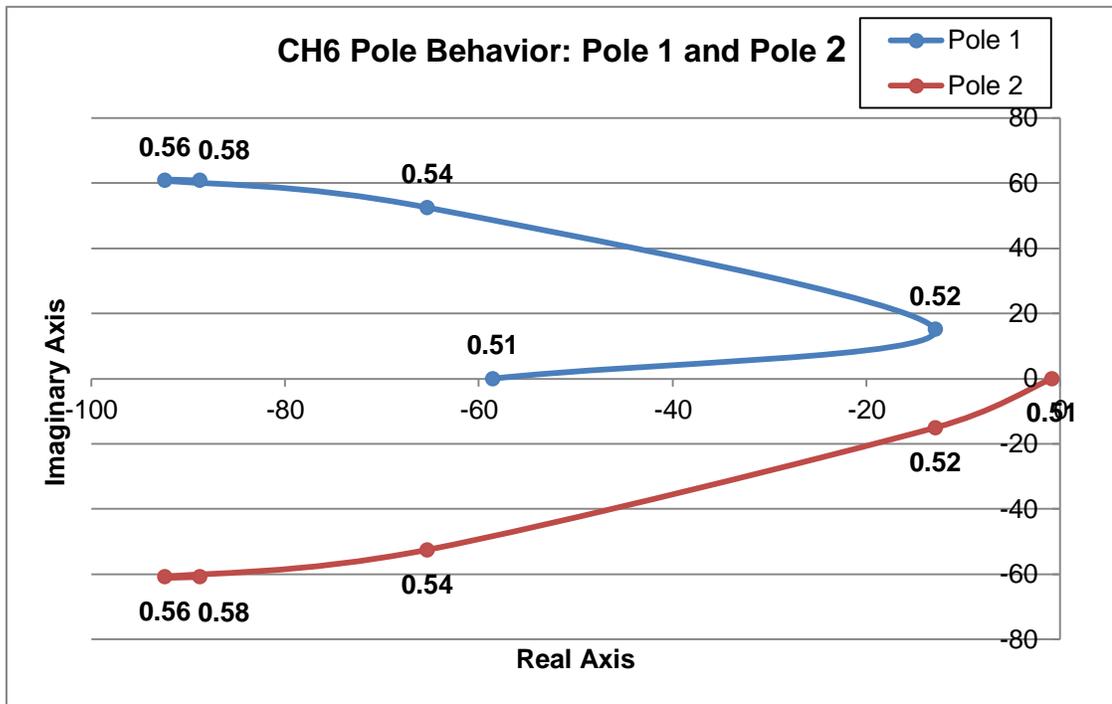


Figure 6.10: Pole Behavior of Poles 1 and 2 for Sensor CH6 (Experiment 1)

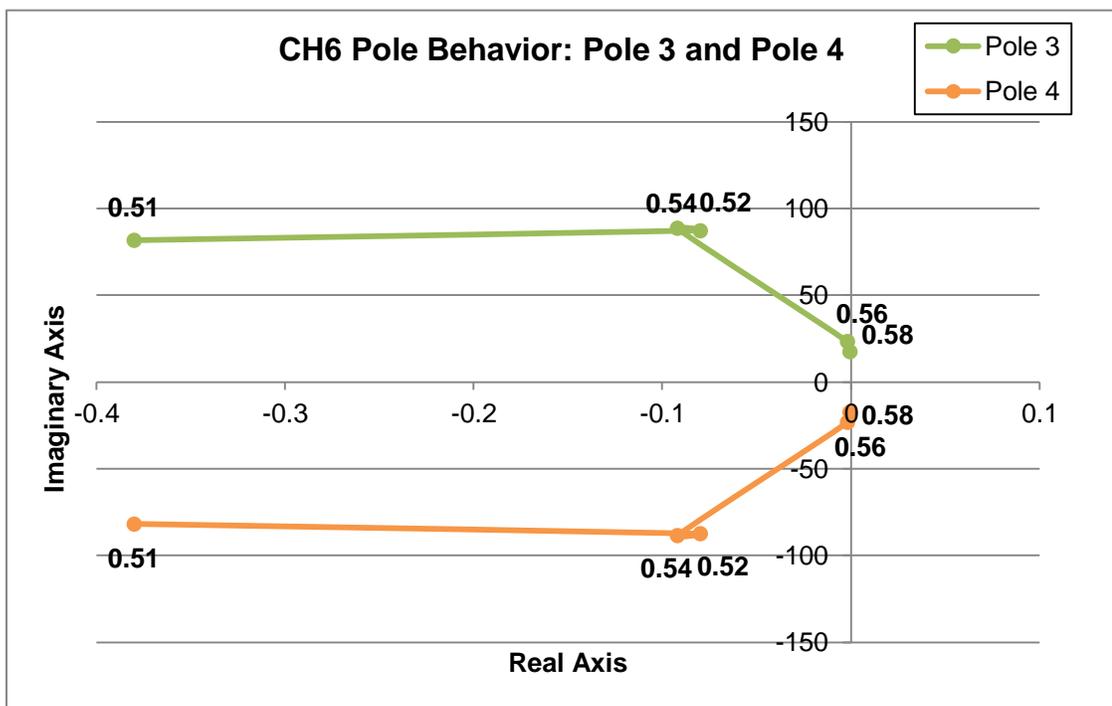


Figure 6.9: Pole Behavior of Poles 3 and 4 for Sensor CH6 (Experiment 1)

Table 6.5: Fast Dynamics Transfer Function Model Data for Sensor CH4 (Experiment 2)

Flow Coefficient	Poles	Zeros	Natural Frequency (Hz)
0.58	-0.0000 +91.8247i	-5.7593 +35.8042i	14.614
	-0.0000 -91.8247i	-5.7593 -35.8042i	14.614
	-0.0866 +22.8682i	4.3735 + 0.0000i	3.640
	-0.0866 -22.8682i	-	3.640
0.55	-50.8872 +56.1665i	102.37 + 0.0000i	12.062
	-50.8872 -56.1665i	-3.62 + 32.80i	12.062
	-1.0834 +32.3166i	-3.62 - 32.80i	5.146
	-1.0834 -32.3166i	-	5.146
0.52	-23.57 + 103.21i	17.5915 -91.3253i	16.849
	-23.57 - 103.21i	0.3515 + 0.0000i	16.849
	-111.94 + 0.0000i	17.5915 +91.3253i	17.815*
	-2.97 + 0.0000i	-	0.473*
0.51	-15.6778 +86.7238i	65.81 + 231.51i	14.026
	-15.6778 -86.7238i	65.81 - 231.51i	14.026
	-59.6922 + 0.0000i	-0.10 + 0.0000i	9.500*
	-0.2148 + 0.0000i	-	0.034*

* The natural frequency of a pole on the real axis is damped out

Table 6.6: Fast Dynamics Transfer Function Model Data for Sensor CH5 (Experiment 2)

Flow Coefficient	Poles	Zeros	Natural Frequency (Hz)
0.58	-0.0494 +86.2249i	0.6613 +85.5851i	13.723
	-0.0494 -86.2249i	0.6613 -85.5851i	13.723
	-24.8661 +20.0039i	-45.6292 + 0.0000i	5.079
	-24.8661 -20.0039i	-	5.079
0.55	-0.0761 +82.6960i	3.2461 +82.6426i	13.161
	-0.0761 -82.6960i	3.2461 -82.6426i	13.161
	-77.8056 + 0.0000i	0.4314 + 0.0000i	12.383*
	-3.0339 + 0.0000i	-	0.483*
0.52	-39.9046 +77.9154i	79.0283	13.932
	-39.9046 -77.9154i	-13.5749	13.932
	-8.0218 +11.8928i	-6.3414	2.283
	-8.0218 -11.8928i	-	2.283
0.51	-36.2845 + 0.0000i	-0.0679 + 0.0000i	5.775*
	-0.1209 + 0.0000i	-	0.019*
	-0.2171 +38.8524i	-1.1366 +40.1447i	6.184
	-0.2171 -38.8524i	-1.1366 -40.1447i	6.184

* The natural frequency of a pole on the real axis is damped out

Table 6.7: Fast Dynamics Transfer Function Model Data for Sensor CH6 (Experiment 2)

Flow Coefficient	Poles	Zeros	Natural Frequency (Hz)
0.58	-62.5110 + 0.0000i	-63.0494 + 0.0000i	9.949*
	-35.3722 + 0.0000i	0.2770 + 12.2539i	5.630*
	-0.0000 + 13.4747i	0.2770 - 12.2539i	2.145
	-0.0000 - 13.4747i	-	2.145
0.55	-0.4476 + 68.6131i	6.7801 + 71.1216i	10.920
	-0.4476 - 68.6131i	6.7801 - 71.1216i	10.920
	-34.3090 + 0.0000i	-0.0225 + 0.0000i	5.460*
	-1.5807 + 0.0000i	-	0.252*
0.52	-0.1027 + 77.4728i	145.8282	12.330
	-0.1027 - 77.4728i	-50.392	12.330
	-2.3111 + 12.0673i	-0.1081	1.955
	-2.3111 - 12.0673i	-	1.955
0.51	-0.5820 + 66.0659i	216.21 + 0.0000i	10.515
	-0.5820 - 66.0659i	-0.18 + 23.21i	10.515
	-0.1746 + 23.2212i	-0.18 - 23.21i	3.696
	-0.1746 - 23.2212i	-	3.696

* The natural frequency of a pole on the real axis is damped out

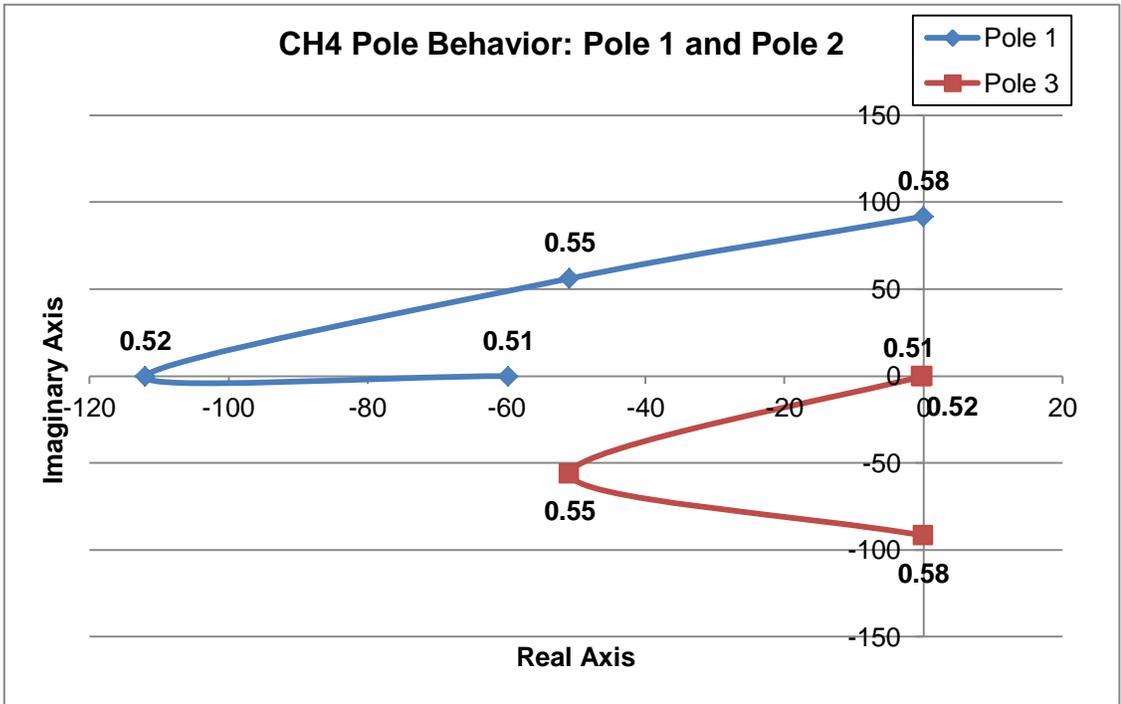


Figure 6.12: Pole Behavior of Poles 1 and 2 for Sensor CH4 (Experiment 2)

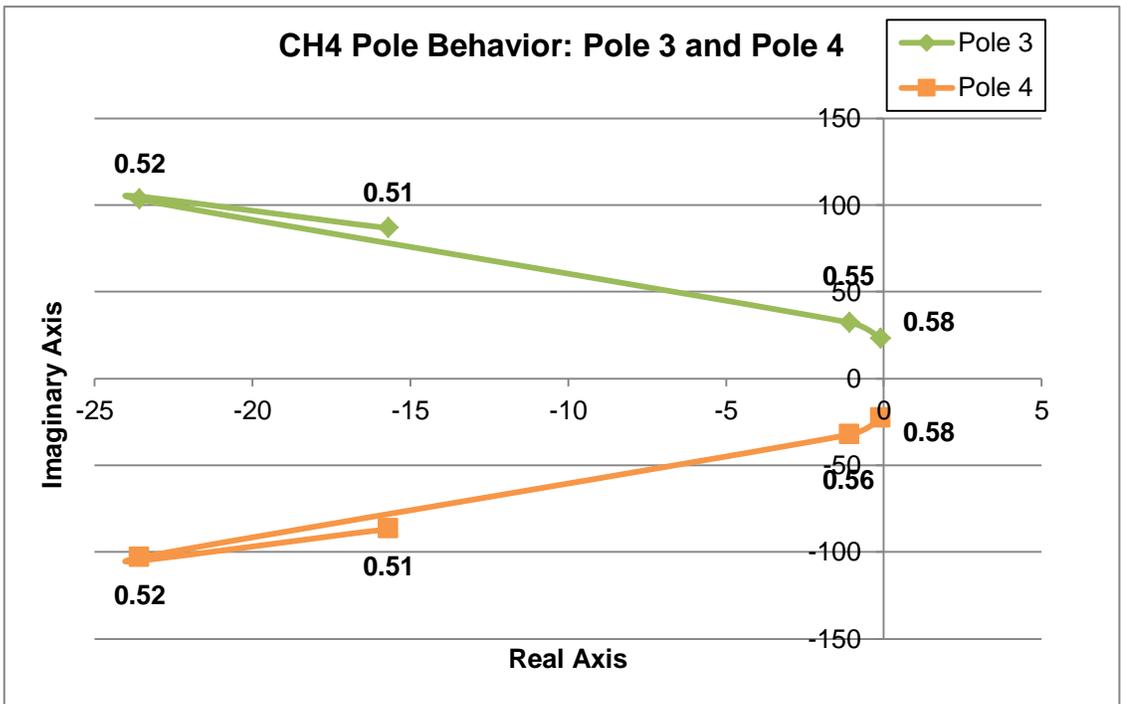


Figure 6.11: Pole Behavior of Poles 3 and 4 for Sensor CH4 (Experiment 2)

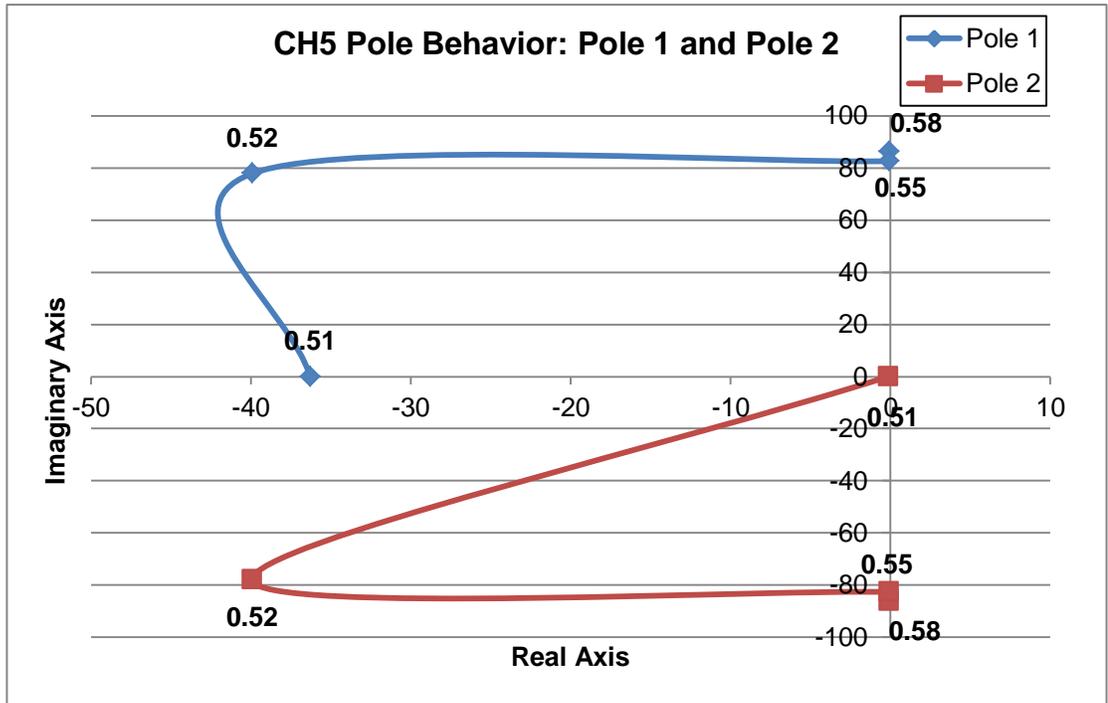


Figure 6.13: Pole Behavior of Poles 1 and 2 for Sensor CH5 (Experiment 2)

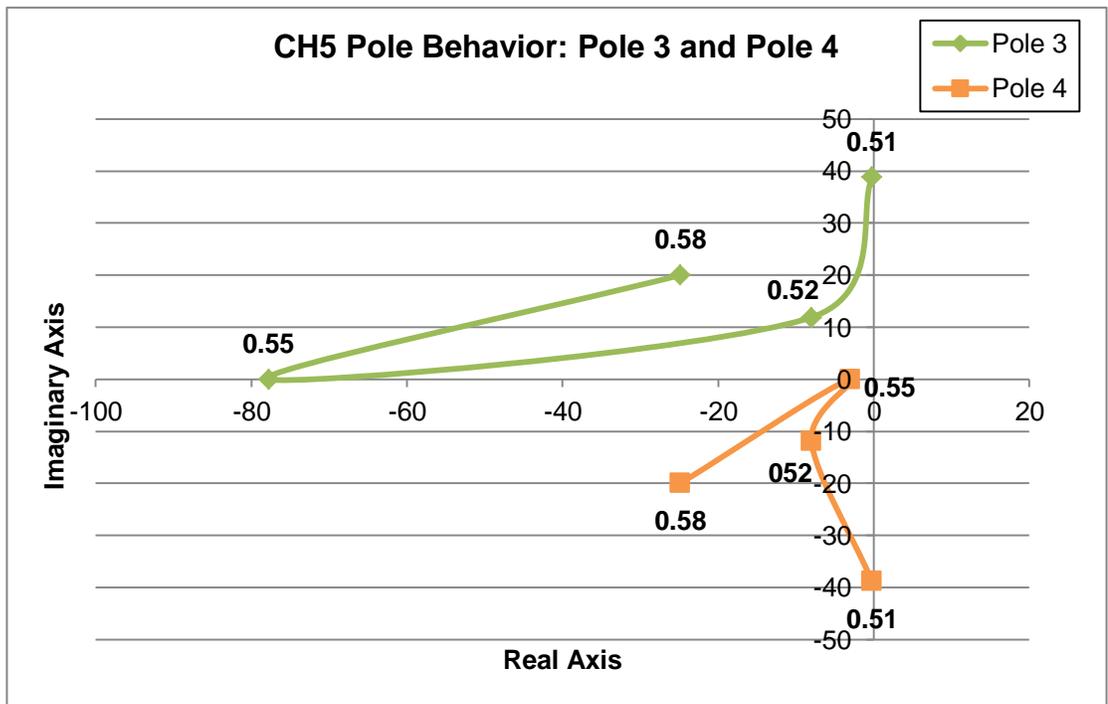


Figure 6.14: Pole Behavior of Poles 3 and 4 for Sensor CH5 (Experiment 2)

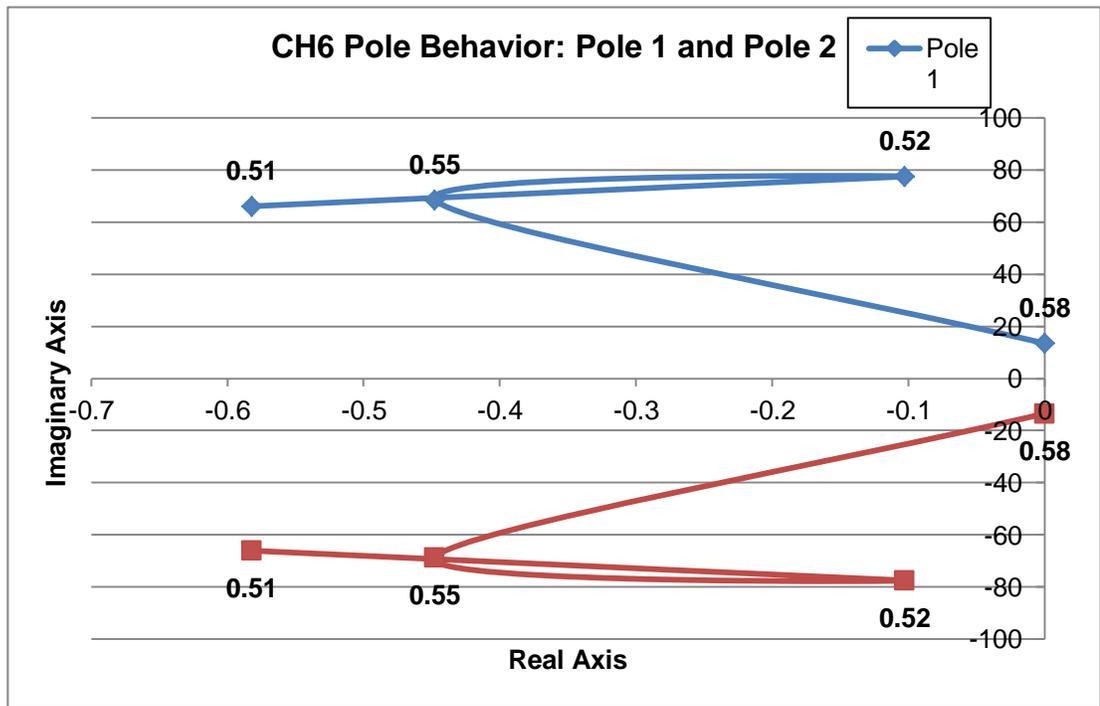


Figure 6.15: Pole Behavior of Poles 1 and 2 for Sensor CH6 (Experiment 2)

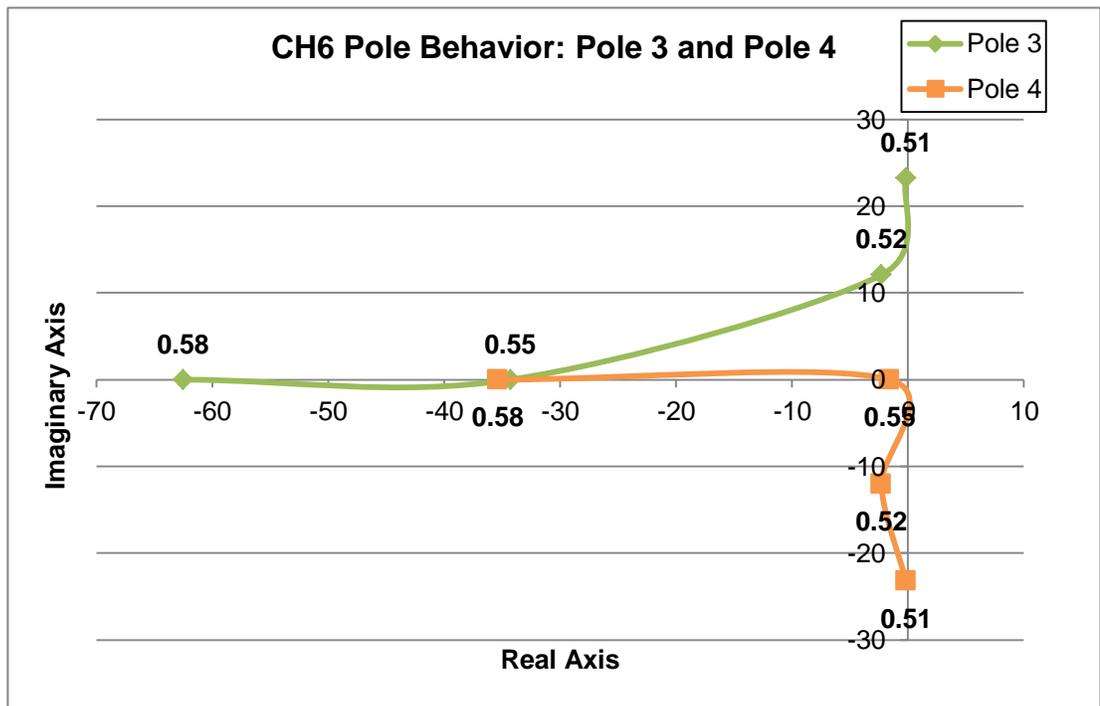


Figure 6.16: Pole Behavior of Poles 3 and 4 for Sensor CH6 (Experiment 2)

The natural frequency band of the compressor, approximately in the range of 13 to 19 Hz for this specific compressor, appeared to be roughly captured in the system identified models (transfer functions with three zeros and four poles). Additionally, the poles of these transfer functions appear to produce a trend as the stall point is approached. These trends are hypothesized to take the form of a traditional root locus with the poles approaching the real axis and then “splitting” along the real axis as the flow coefficient nears the stall point. The poles also appear to edge toward instability as one of the poles on the real axis moves toward the origin of the complex plane. Figure 6.17 and Figure 6.18 demonstrate the extrapolated root locus pole behavior for sensors CH5 and CH6, respectively, for Experiment 1. If root locus behavior can be confirmed for some of the model poles, this may aid in the development of a reliable and efficient control system for axial compressor stall margin improvement.

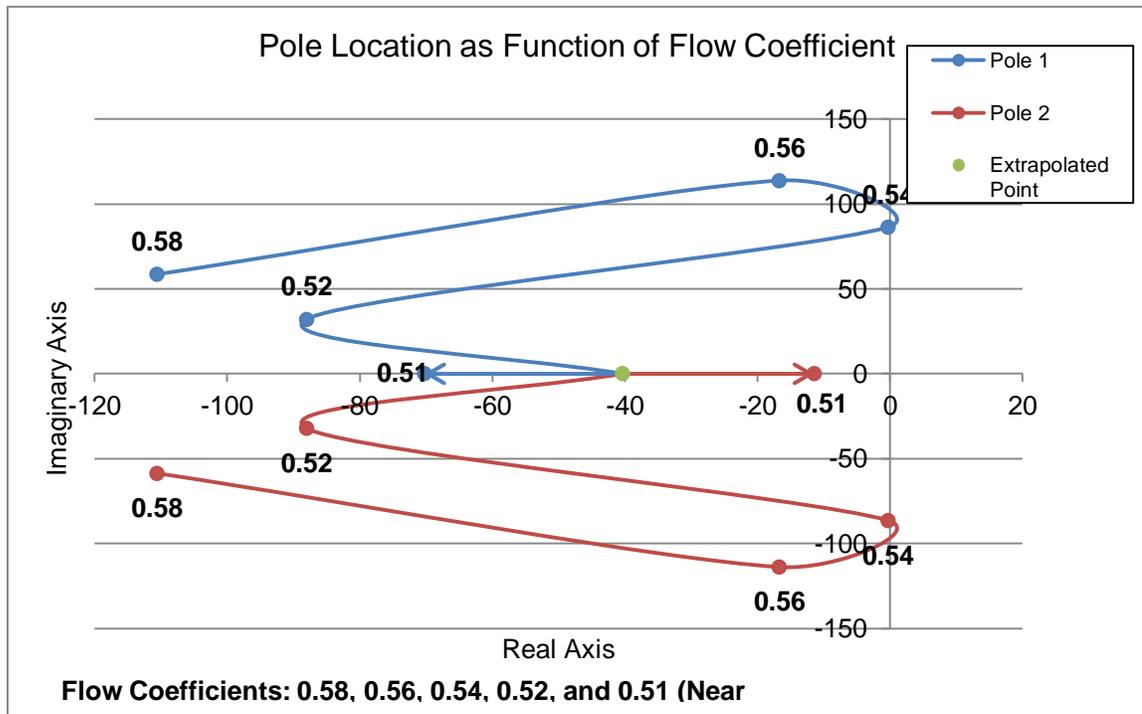


Figure 6.17: Extrapolated Root Locus Behavior of Poles at Sensor CH5

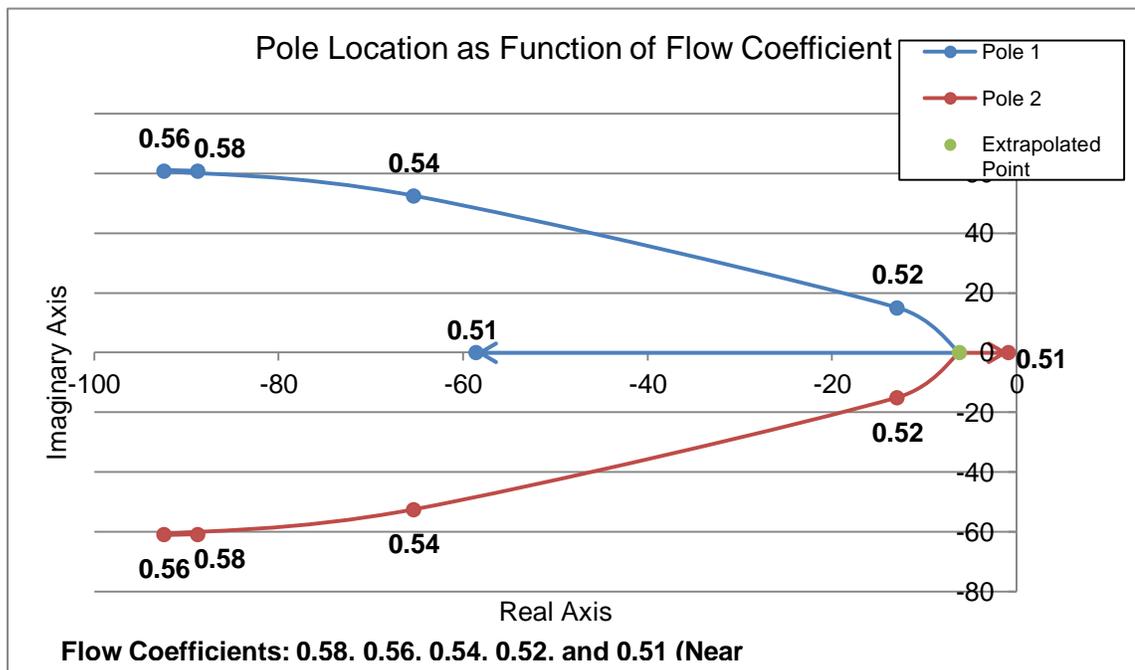


Figure 6.18: Extrapolated Root Locus Behavior of Poles at Sensor CH6

An interesting aspect of the fast dynamics behavior that appears to be captured in the model is a region of instability at and around sensors CH4 and CH5 when the compressor is throttled to a flow coefficient near 0.54. This instability causes a spike in the compressor natural frequency at and around 0.54 and then settles back down as throttling continues. Currently, the impetus behind this instability is unknown. Further research conducted into determining the flow dynamics that cause this behavior would be a worthwhile area of study.

Magnitude Bode plots are also created for each of the models obtained from sensors CH4, CH5, and CH6 from Experiment 1. These plots clearly demonstrate some interesting trend behavior of the system identified models that could not as easily be observed in the pole trend behavior plots. For the models obtained from sensor CH5, shown in Figure 6.19, the spike in the natural frequency at and around a flow coefficient of 0.54 is obvious. This same behavior is observed for the models obtained from sensor CH4. Additionally, for the models obtained from sensor CH6, shown in Figure 6.20, a small distinct spike in the natural frequency is observed at around 0.54. As the flow coefficient is throttled to the stall point there is a sharp increase in the magnitude of the compressor natural frequency. This increase in magnitude of the natural frequency, captured in the models of sensor CH6, could possibly be used as a stall inception warning indicator.

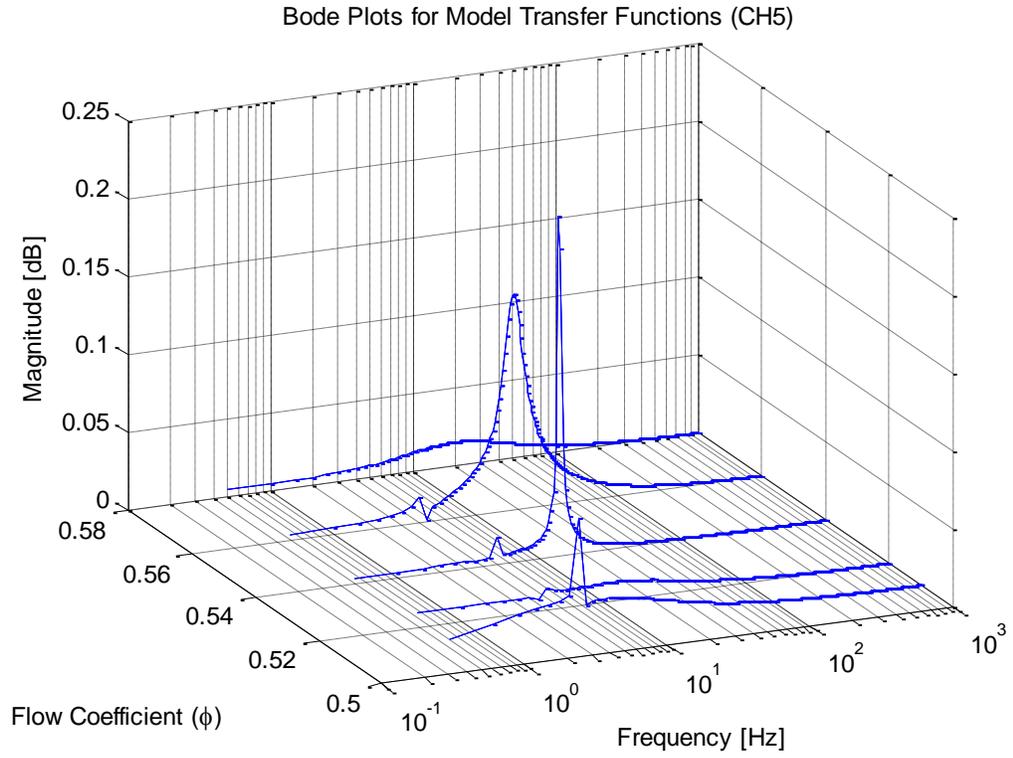


Figure 6.19: Magnitude Bode Plots for the Fast Dynamics Model Transfer Functions (CH5)

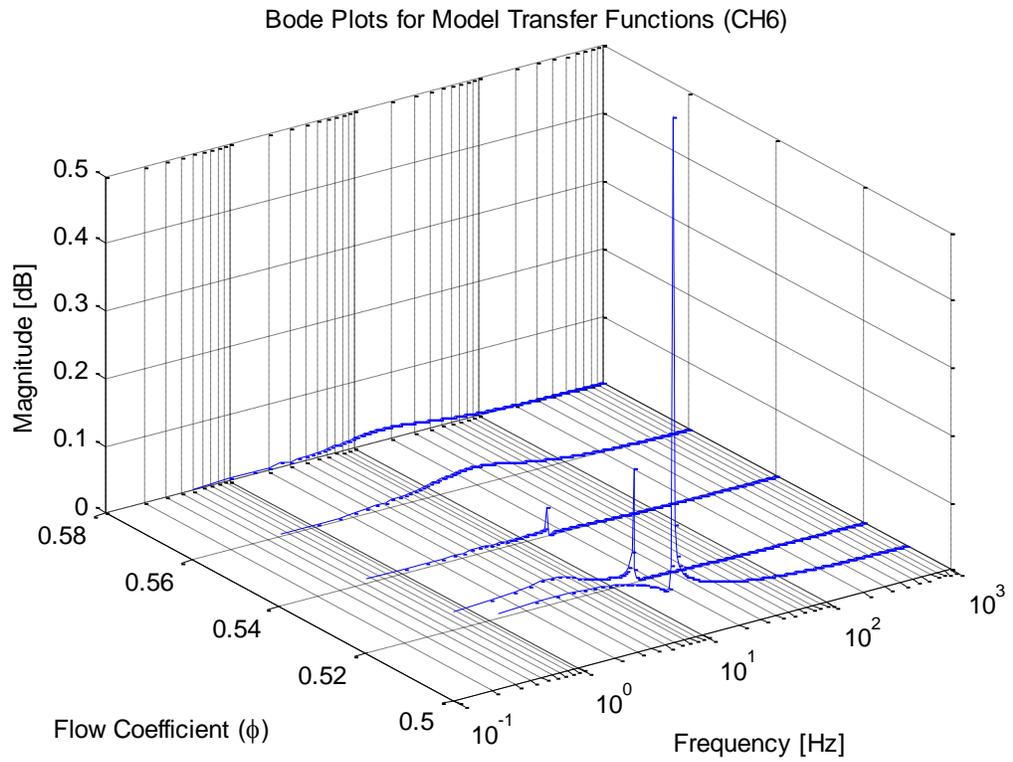


Figure 6.20: Magnitude Bode Plots for the Fast Dynamics Model Transfer Functions (CH6)

CHAPTER 7 CONCLUSIONS AND FUTURE WORK

7.1 Summary and Conclusions

The research presented in this thesis provides a baseline for studying the fast and slow dynamics of an axial flow compressor through the use of system identification. The development of a model to characterize the flow dynamics is necessary to advance the efficiency of control systems for improving the stall margin of a jet engine axial compressor. System identification, signal filtering, filter parameter optimization, and pole behavior of transfer function models are applied or analyzed in this research.

More specifically, in the presented research, a method is demonstrated for optimizing the noise filter parameters for system identification of the slow dynamics. This filter optimization method uses a particle swarm algorithm to produce the optimal Bayesian filter parameters based on the goodness of fit for the system identified model. This optimization method is not directly used in the research presented in this document, but can be employed in future slow dynamics system identification.

For the compressor slow dynamics, the transfer function models with three zeros and four poles are found to produce very accurate fits for the pressure rise coefficient data. These transfer function models are found to have a high goodness of fit in the range of approximately 70 to 80 percent with obvious visual correlation. However, the poles of these transfer functions do not appear to have an obvious trend as the compressor stall point is approached. This is most likely due to noise and error in the signal or the inability of the models to capture the frequencies that capture the dynamics of the compressor system.

The fast dynamics research produces more promising results in terms of pole behavior of the system identified models. Using the autocorrelation coefficient data with cubic splines to interpolate the missing data problem, system identification is used to create models. The system identification uses the air injection pressure signal as the input and the autocorrelation coefficient of the dynamic pressure data across the rotor blades as the output. From this system identification, transfer function models that also use three zeros and four poles are found to capture the dynamic frequencies of the system, in spite of fairly low goodness of fit values, generally less than 10 percent.

The pole behavior of the fast dynamics models appeared to roughly capture the natural compressor frequency, approximately around 17 Hz, and produced an apparent trend as the compressor stall point is approached. This trend appeared to take on the behavior of a traditional root locus, with a conjugate pole pair approaching the real axis and “splitting” along the real axis as the compressor flow coefficient approaches the stall point. In other words, as the compressor moves closer to stall (i.e. the flow coefficient is reduced from 0.58 to 0.51), the poles of the identified models move towards the imaginary axis of the complex plane, indicating an impending instability. Lastly, the magnitude Bode plots obtained from the transfer function models demonstrated behavior indicative of stall inception as well as highlighting a not well understood spike in the compressor natural frequency at and around a flow coefficient of 0.54.

7.2 Future Work

Future work on this project includes the use of the solenoid actuated throttle modules, which are currently being manufactured at the IET, to excite the slow dynamics of the system. These throttle modules will be able to excite the compressor system at a higher frequency than was previously possible with the current equipment at the IET. The ability to excite the system at a frequency higher than compressor natural frequency of approximately 17 Hz is a necessary next step in modeling the slow dynamics of the compressor. With data from experiments using these new throttle modules, the noise filter optimization that is discussed previously can be used to produce more useful system identified models than are developed in this research.

Additionally, the pole behavior of the fast dynamics transfer function models is another area that requires further study. More experiments to confirm the pole behavior as a function of the flow coefficient would be useful for validation purposes. Models obtained from an alternate system identification program or an analytical flow model to compare with the models developed from this research would also be useful for further validation. Applying some type of filter to the autocorrelation coefficient data in order to reduce random noise might be beneficial for performing future system identification research. Overall, the application of system identification to characterize the dynamics of an axial flow compressor is an emerging field with many new opportunities for dynamic model development.

REFERENCES

- [1] Przybylko, S. J., 1997, "Active-Control Technologies for Aircraft Engines," AIAA 97-2769, IAA/ASME/SAE/ASEE 33rd Joint Propulsion Conference & Exhibit.
- [2] Boyce, M. P., 2012, Gas Turbine Engineering Handbook, 4th ed., Elsevier Inc., Waltham, MA, Chap. 7.
- [3] Moore, F. K., and Greitzer E. M., 1986, "A Theory of Post-Stall Transients in Axial Compression Systems: Part I-Development of Equations," Journal of Engineering for Gas Turbines and Power, 108, pp. 68-76.
- [4] Paduano, J. D., Greitzer, E. M., and Epstein, A. H., 2001, "Compression System Stability and Active Control," Annual Review of Fluid Mechanics, 33(1), pp. 491-517.
- [5] Chi, J. N., and Paduano, J. D., 2008, "New Concepts for Active Control of Rotating Stall and Surge," American Control Conference, Seattle, WA, pp. 2435-2442.
- [6] Uddin, N., and Gravdahl, J. T., 2011, "Active Compressor Surge Control Using Piston Actuation," ASME 2011 Dynamic Systems and Control Conference and Bath/ASME Symposium on Fluid Power and Motion Control Arlington, VA, pp. 69-76.
- [7] Ohta, Y., Fujita, Y., and Morita, D., 2012, "Unsteady Behavior of Surge and Rotating Stall in an Axial Flow Compressor," Journal of Thermal Science, 21(4), pp. 302-310.
- [8] Saxer-Felici, H. M., Saxer, A. P., Inderbitzin, A., and Gyarmathy, G., 2000, "Numerical and Experimental Study of Rotating Stall in an Axial Compressor Stage," AIAA Journal, 38(7), pp. 1132-1141.
- [9] Du, W., and Léonard, O., 2012, "Numerical Simulation of Surge in Axial Compressor," International Journal of Rotating Machinery, pp. 1-10.
- [10] Greitzer, E. M., 1976, "Surge and Rotating Stall in Axial Flow Compressors, Part I: Theoretical Compression System Model," ASME J. Eng. for Power, 98(2), pp. 190-198.
- [11] Chen, J. P., Hathaway, M. D., and Herrick, G. P., 2008, "Prestall Behavior of a Transonic Axial Compressor Stage via Time-Accurate Numerical Simulation," ASME Journal of Turbomachinery, 130(4).
- [12] Biollo, R., and Benini, E., 2013, "Recent Advances in Transonic Axial Compressor Aerodynamics," Progress in Aerospace Sciences, 56, pp. 1-18.

- [13] Iura, T., and Rannie, W. D., 1953, "Observations of Propagating Stall in Axial-Flow Compressor," California Institute of Technology, Pasadena, CA.
- [14] Marble, F. E., 1953, "Propagation of Stall in a Compressor Blade Row," *Journal of the Aeronautical Sciences*, 22(8), pp. 541-554.
- [15] Greitzer, E. M., 1976, "Surge and Rotating Stall in Axial Flow Compressors, Part II: Experimental Results and Comparison with Theory," *ASME J. Eng. for Power*, 98(2), pp. 199-217.
- [16] Moore, F. K., 1984, "A Theory of Rotating Stall of Multistage Axial Compressor: Part I, II, and III," *Journal of Engineering for Gas Turbines and Power*, 106(2), pp. 313-336.
- [17] Brons, M., 1988, "Bifurcations and Instabilities in the Greitzer Model for Compressor System Surge," *Mathematical Engineering in Industry*, 2(1), pp. 51-63.
- [18] Haynes, J. M., Hendricks, G. J., and Epstein, A. H., 1994, "Active Stabilization of Rotating Stall in a Three-Stage Axial Compressor," *ASME Journal of Turbomachinery*, 116, pp. 226-239.
- [19] Paduano, J. D. e. a., 1994, "Modeling for Control of Rotating Stall," *Automatica*, 30(9), pp. 1357-1373.
- [20] Simon, J. S. e. a., 1993, "Evaluation of Approaches to Active Compressor Surge Stabilization," *ASME Journal of Turbomachinery*, 115, pp. 57-67.
- [21] Hendricks, G. J., Sabnis, J. S., and Feulner, M. R., 1997, "Analysis of Instability Inception in High-Speed Multistage Axial-Flow Compressors," *ASME Journal of Turbomachinery*, 119, pp. 714-722.
- [22] Weigl, H. J., Paduano, J. D., and Frechette, L. G., 1998, "Active Stabilization of Rotating Stall and Surge in a Transonic Single-Stage Axial Compressor," *ASME Journal of Turbomachinery*, 120(4), pp. 625-636.
- [23] D'Andrea, R., Behnken, R. L., and Murray, R. M., 1997, "Active Control of an Axial Flow Compressor via Pulsed Air Injection," *ASME Journal of Turbomachinery*, 119, pp. 742-752.
- [24] Humbert, J. S., and Krener, A. J., 1998, "Dynamics and Control of Entrained Solutions in Multi-Mode Moore-Greitzer Compressor Models," *Int. J. Control*, 71(5), pp. 807-821.

- [25] Haddad, W. M. e. a., 1999, "Nonlinear Robust Disturbance Rejection Controllers for Rotating Stall and Surge in Axial Flow Compressors," *IEEE Transactions on Control Systems Technology*, 7(3), pp. 391-398.
- [26] Kang, W. e. a., 1999, "Bifurcation Test Functions and Surge Control for Axial Flow Compressors," *Automatica*, 35, pp. 229-239.
- [27] Bohagen, B., and Gravdahl, J. T., 2008, "Active Surge Control of Compression System using Drive Torque," *Automatica*, 44, pp. 1135-1140.
- [28] Nayfeh, M. A., and Eyad E. H., 2002, "High-Gain Feedback Control of Rotating Stall in Axial Flow Compressors," *Automatica*, 38, pp. 995-1001.
- [29] Chaturvedi, N. A., and Bhat, S. P., 2006, "Output-Feedback Semiglobal Stabilization of Stall Dynamics for Preventing Hysteresis and Surge in Axial-Flow Compressors," *IEEE Transactions on Control Systems Technology*, 14(2), pp. 301-307.
- [30] Sun, X., and Liu, X., 2013, "A General Theory of Flow-Instability Inception in Turbomachinery," *American Institute of Aeronautics and Astronautics*.
- [31] Tahara, N., Kurosaki, M., Ohta, Y., Outa, E., Nakakita, T., and Tsurumi, Y., 2004, "Early Stall Warning Technique for Axial Flow Compressors," *Proc. of ASME Turbo Expo 2004, Power for Land, Sea, and Air, Vienna, Austria*.
- [32] Chen, J. P. e. a., 2009, "High Performance Computing of Compressor Rotating Stall and Stall Control," *Integrated Computer-Aided Engineering*, 16, pp. 75-89.
- [33] Chima, R. V., 2006, "A Three-Dimensional Unsteady CFD Model of Compressor Stability," *Proc. of ASME Turbo Expo 2006, Power for Land, Sea, and Air, Barcelona, Spain*.
- [34] Cornelius, C. e. a., 2014, "Experimental and Computational Analysis of a Multistage Axial Compressor Including Stall Prediction by Steady and Transient CFD Methods," *ASME Journal of Turbomachinery*, 136.
- [35] Emmons, H. W., Pearson, C. E., and Grant, H. P., 1955, "Compressor Surge and Stall Propagation," *Trans. ASME*, 79, pp. 455-469.
- [36] McDougall, N. M., Cumpsty, N. A., and Hynes, T. P., 1990, "Stall Inception in Axial Compressors," *ASME Journal of Turbomachinery*, 112(1), pp. 116-123.

- [37] Garnier, V. H., Epstein, A. H. and Greitzer, E. M., 1991, "Rotating Waves as a Stall Inception Indication in Axial Compressors," ASME Journal of Turbomachinery, 113(2), pp. 290-302.
- [38] Tryfonidis, M., Etchevers, O., Paduano, J. D., Epstein, A. H. and Hendricks, G. J., 1995, "Prestall Behavior of Several High-Speed Compressors," ASME Journal of Turbomachinery, 117(1), pp. 62-80.
- [39] Spakovszky, Z. S., 2004, "Backward Traveling Rotating Stall Waves in Centrifugal Compressors," ASME Journal of Turbomachinery, 126, pp. 1-12.
- [40] Day, I. J., 1993, "Stall Inception in Axial Flow Compressors," ASME Journal of Turbomachinery, 115(1), pp. 1-9.
- [41] Camp, T. R., and Day, I. J., 1998, "A Study of Spike and Modal Stall Phenomena in a Low-Speed Axial Compressor," ASME Journal of Turbomachinery, 120(3), pp. 393-401.
- [42] Greitzer, E. M., 1981, "The Stability of Pumping Systems," ASME Journal of Fluids Engineering, 103, pp. 193-242.
- [43] Feulner, M. R., Hendricks, G. J., and Paduano, J. D., 1996, "Modeling for Control of Rotating Stall in High-Speed Multistage Axial Compressors," ASME Journal of Turbomachinery, 118, pp. 1-10.
- [44] Frechette, L. G., 1997, "Implications of Stability Modeling for High-Speed Axial Compressor Design," M.S. Thesis, Massachusetts Institute of Technology.
- [45] Khalid, S. A., Khalsa, A. S., Waitz, I. A., Tan, C. S., Greitzer, E. M., Cumpsty, N. A., Adamczyk, J. J., and Marble, F. E., 1999, "Endwall Blockage in Axial Compressors," ASME Journal of Turbomachinery, 121, pp. 499-509.
- [46] Vo, H. D., 2001, "Role of Tip Clearance Flow on Axial Compressor Stability," Doctor of Philosophy, Massachusetts Institute of Technology.
- [47] Lin, F., Chen, J., Nie, C., and Zhang, H., 2009, "Unsteady Tip Leakage Flow and Perspectives to Stall Mechanism in Axial Compressors " 2nd Asian Congress on Gas Turbines Tokyo, Japan.
- [48] Tahara, N., Nakajima, T., Kurosaki, M., Ohta, Y., Outa, E., and Nishikawa, T., 2001, "Active Stall Control with Practicable Stall Prediction System Using Auto-Correlation," AIAA Paper No. 2001-3623.

- [49] Dhingra, M., Neumeier, Y., Prasad, J. V. R., Breeze-Stringfellow, A., Shin, H. W., and Szucs, P. N., 2007, "A Stochastic Model for a Compressor Stability Measure," *Journal of Engineering for Gas Turbines and Power*, 129, pp. 730-737.
- [50] Christensen, D., Cantin, P., Gutz, D., Szucs, P. N., Wadia, A. R., Armor, J., Dhingra, M., Neumeier, Y., and Prasad, J. V. R., 2008, "Development and Demonstration of a Stability Management System for Gas Turbine Engines," *ASME Journal of Turbomachinery*, 130.
- [51] Li, J., Lin, F., Tong, Z., Nie, C., and Chen, J., 2015, "The Dual Mechanisms and Implementations of Stability Enhancement with Discrete Tip Injection in Axial Flow Compressors," *ASME Journal of Turbomachinery*, 137.
- [52] Li, C., Ke, K., Zhang, J., Zhang, H., and Huang, W., 2013, "Experimental and Numerical Investigation of the Unsteady Tip Leakage Flow in Axial Compressor Cascade," *Journal of Thermal Science*, 22(2), pp. 103-110.
- [53] Keesman, K. J., 2011, *System Identification: An Introduction*, Springer, New York.
- [54] MATLAB Release 2013a, The MathWorks, Inc., Natick, Massachusetts, United States.
- [55] Sanger, D. T., 2007, "Bayesian Filtering of Myoelectric Signals," *Journal of Neurophysiology*, 97, pp. 1839-1845.
- [56] Chen, C. H., Bosworth, K. W., and Schoen, M. P., 2007, "Investigation of Particle Swarm Optimization Dynamics," *Proc. of ASME International Mechanical Engineering Congress and Exposition*, Seattle, Washington.

APPENDIX A – Matlab™ Files for Slow Dynamics Analysis

A1 - Bayesian Filter

This code is a modified version of a program developed by Sanger in [55].

```
function [yf1,uf1] = bayes_filt(y1,u1,Fs);

% bayes_filt.m
%
% Dane Sterbentz
% October 11, 2014, Version 1.0
% This code was adapted from a program created in conjunction with the
% following research paper:
% Sanger, D. T., 2007, "Bayesian Filtering of Myoelectric Signals,"
% Journal of Neurophysiology, 97, pp. 1839-1845.
%


---


%
% Computation of Bayesian filtered data of the input (u1) and output
% (y1)
% with a specified sampling frequency (Fs)
%


---


Fs; % Sampling Frequency
Ts = 1/Fs; % Sampling interval

% Preliminary section: load data, set constants, initialize variables
%set parameters
samplerate1 = Fs; %samples per second
noutputs1 = 50; %output quantization levels
ratemax1 = 1; %rectified EMG is normalized to max value of 1
inscale1 = 1; %arbitrary input scaling
alpha1 = 0.1 / samplerate1; %sets diffusion rate
beta1 = 300 / (noutputs1 * samplerate1); %sets probability of
sudden jumps

%calculate rectified EMG after removing the mean, and normalize
y1 = inscale1 * ratemax1 * y1/max(y1); %input prescaling to use
full output range
u1 = inscale1 * ratemax1 * u1/max(u1); %input prescaling to use
full output range

%initialize variables
% x is the latent variable (the driving rate)
% MAP is the output estimate
x1 = linspace(ratemax1/noutputs1, ratemax1, noutputs1)'; %don't start
with zero because requires n=0 exactly to match
y_Filt1 = zeros(length(y1),1); %store the bayes
estimates
```

```

u_Filt1 = zeros(length(u1),1); %store the bayes
estimates
g1 = [(alpha1/2) (1 - alpha1) (alpha1/2)]; %approximate
spatial second derivative operator

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Following is the main section of the algorithm; steps are numbered as
in the text
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Bayesian filtering for output (y1):
%1. Initialize  $p(x,0) = 1$ ;
prior1 = ones(noutputs1,1) / noutputs1; %start with uniform
prior

for t1=1:length(y1) %iterate for each sample

    %2. Forward propagate  $p(x,t) \hat{=} \int p(x-,t-1) + (1-2?)p(x,t-1) + ?p(x+?,t-1) + ?(1-?)p(x,t-1)$ ;
    %
    prior1 = filtfilt(g1, 1, prior1); %drift term by
    convolving with second derivative operator
    prior1 = betal + (1-betal) * prior1; %sets probability
    of a sudden jump

    %3. Measure the rectified emg;
    yvall1 = y1(t1); %if this were online,
    would read a new sample here

    %4. Calculate the posterior likelihood function
    %  $P(x,t) \hat{=} P(emg|x)p(x,t-)$ ;
    measurement_model1 = 2*exp(-
    (yvall1).^2./(2.*(x1).^2))./(2.*pi.*x1.^2).^(1/2); %half Gaussian
    model for  $P(Pres1|x)$ 
    posterior1 = measurement_model1 .* prior1; %calculate posterior
    density using Bayes rule

    %5. Output the signal estimate  $MAP(x(t)) = \text{argmax } P(x,t)$ ;
    pp1 = min(find(posterior1 == max(posterior1))); %find the
    maximum of the posterior density
    if (pp1 > 1 && pp1 < length(posterior1)), %interpolate to
    find the zero
        dL1 = posterior1(pp1-1) - posterior1(pp1);
        dR1 = posterior1(pp1) - posterior1(pp1+1);
        PeakIndex1 = (pp1 - .5 - (dL1/ (dR1 - dL1))); %index runs
    from 1 to noutputs
    else
        PeakIndex1 = pp1; %if maximum occurs at an endpoint do not
    interpolate
    end
    y_Filt1(t1) = (ratemax1 / (noutputs1-1)) * PeakIndex1; %convert
    index of peak value to scaled value

    %6. Divide  $p(x,t)$  by a constant C so that  $\int p(x,t) dx = 1$ ;

```

```

    posterior1 = posterior1 / sum(posterior1);           %normalize the
density

    %7. Repeat from step 2;
    prior1 = posterior1;                               %prior for next
iteration is posterior from this iteration
end

% Bayesian filtering for input (u1):
%1. Initialize  $p(x,0) = 1$ ;
prior1 = ones(noutputs1,1) / noutputs1;              %start with uniform
prior

for t1=1:length(u1) %iterate for each sample

    %2. Forward propagate  $p(x,t) \hat{=}$ 
%            $\alpha p(x^-,t-1) + (1-\alpha)p(x,t-1) + \beta p(x^+,t-1) + (1-\beta)p(x,t-1)$ ;
    prior1 = filtfilt(g1, 1, prior1);                 %drift term by
convolving with second derivative operator
    prior1 = betal + (1-betal) * prior1;             %sets probability
of a sudden jump

    %3. Measure the rectified emg;
    uval1 = u1(t1);                                  %if this were online,
would read a new sample here

    %4. Calculate the posterior likelihood function
%            $P(x,t) \hat{=} P(\text{emg}|x)p(x,t)$ ;
    measurement_model1 = 2*exp(-
(uval1).^2./(2.*(x1).^2))./(2.*pi.*x1.^2).^(1/2);   %half Gaussian
model for  $P(\text{Flow1}|x)$ 
    posterior1 = measurement_model1 .* prior1;        %calculate posterior
density using Bayes rule

    %5. Output the signal estimate  $\text{MAP}(x(t)) = \text{argmax } P(x,t)$ ;
    pp1 = min(find(posterior1 == max(posterior1)));    %find the
maximum of the posterior density
    if (pp1 > 1 && pp1 < length(posterior1)),         %interpolate to
find the zero
        dL1 = posterior1(pp1-1) - posterior1(pp1);
        dR1 = posterior1(pp1) - posterior1(pp1+1);
        PeakIndex1 = (pp1 - .5 - (dL1/ (dR1 - dL1))); %index runs
from 1 to noutputs
    else
        PeakIndex1 = pp1;                             %if maximum occurs at an endpoint do not
interpolate
    end
    u_Filt1(t1) = (ratemax1 / (noutputs1-1)) * PeakIndex1; %convert
index of peak value to scaled value

    %6. Divide  $p(x,t)$  by a constant C so that  $\int p(x,t) dx = 1$ ;
    posterior1 = posterior1 / sum(posterior1);        %normalize the
density

    %7. Repeat from step 2;

```

```

    prior1 = posterior1;                                %prior for next
iteration is posterior from this iteration
end

yf1 = (y_Filt1-mean(y_Filt1))/max(abs(y_Filt1(17:32000)-
mean(y_Filt1(17:32000))));
uf1 = (u_Filt1-mean(u_Filt1))/max(abs(u_Filt1(17:32000)-
mean(u_Filt1(17:32000))));

% Plots of non-filtered data (blue) and filtered data (red) for y1 and
u1:
time = (1:1:length(y1))*Ts;

% Bayesian filtered output (y1) plot:
subplot(2,1,1)
hold on
plot(time,y1-mean(y1), 'b')
plot(time,y_Filt1-mean(y_Filt1), 'r')
xlabel('Time [sec]')
ylabel('Amplitude')
title('Pressure Rise Coefficient (\psi) versus Time')
legend('Measured Data', 'Bayesian Filtered Data')
xlim([1 length(time)*Ts])
hold off

% Bayesian filtered input (u1) plot:
subplot(2,1,2)
hold on
plot(time,u1-mean(u1), 'b')
plot(time,u_Filt1-mean(u_Filt1), 'r')
xlabel('Time [sec]')
ylabel('Amplitude')
title('Flow Coefficient (\phi) versus Time')
legend('Measured Data', 'Bayesian Filtered Data')
xlim([1 length(time)*Ts])
hold off

end

```

A2 – Particle Swarm Optimization for Optimizing Bayesian Filter

This code is a modified version of a program developed in [56].

```
% simplePSO_ab.m
%
% Dane Sterbentz
% September 21, 2014
% Version 1.0
% Basic PSO algorithm using global best methodology
% This code is modified to calculate the optimal parameters for a
Bayesian
% filter (alpha and beta) that produce the maximum goodness of fit for
a
% transfer function system identification model. The cost function for
the
% transfer function system identification model is given in the file:
% "cost_fit_tf.m"
%
%


---


% Initialization of a ns x nx - dimensional swarm S
c1=.1;c2=.1;
input('Nominal 0, or specific 1: ');spec=ans;
input('Input data (Flow Coefficient): ');Flow1=ans;
input('Output data (Pressure Rise Coefficient): ');Pres1=ans;
if spec==0

nx=2;maxxalpha=10;minxalpha=0.01;maxxbeta=120;minxbeta=20;ns=3;nd=2;
else
    input('Dimension of particle nx: ');nx=ans;
    input('Search space, maximum: ');maxx=ans;
    input('Search space, minimum: ');minx=ans;
    input('Number of particles ns: ');ns=ans;
    input('How many iterations to be carried out nd: ');nd=ans;
end

tic
S.x=[(maxxalpha-minxalpha)*rand(ns,1)+minxalpha,(maxxbeta-
minxbeta)*rand(ns,1)+minxbeta]; % uniform distribution of initial
particles

%S.x=rand(ns,nx); %uniform distribution within 0 and 1
V=ones(ns,nx,nd);

% Initial personal and global best computations
for i=1:ns
    y(i,:)=S.x(i,:); % Personal best
    S.cost(i)=cost_fit_tf(S.x(i,:),Pres1,Flow1); % Fitness evaluation
    costp(i,1)=S.cost(i);
end;
[gbestk,element]=min(S.cost); % Minimization problem
yhat=S.x(element,:); % Global best
costg(1)=S.cost(element);
```

```

% Main loop
for k=1:nd
    for i=1:ns
        S.cost(i)=cost_fit(S.x(i,:),k),Pres1,Flow1);
        if S.cost(i)<costp(i,k)%y(i,:)
            y(i,:)=S.x(i,:),k);
            costp(i,k+1)=S.cost(i);
        else
            costp(i,k+1)=costp(i,k);
        end;
        if S.cost(i)<costg(k)%yhat
            yhat=S.x(i,:),k);
            costg(k+1)=S.cost(i);
        else
            costg(k+1)=costg(k);
        end;
    end;
    for i=1:ns
        % Update velocity vector
        V(i,:,k+1)=V(i,:,k)+c1*rand(1,nx).*(y(i,:)-
S.x(i,:),k))+c2*rand(1,nx).*(yhat-S.x(i,:),k));
        % Update position vector
        S.x(i,:,k+1)=S.x(i,:,k)+V(i,:,k+1);
        % Check border of search area
        if S.x(i,1,k+1)>maxxalpha
            S.x(i,1,k+1)=maxxalpha;
        elseif S.x(i,1,k+1)<minxalpha
            S.x(i,1,k+1)=minxalpha;
        else
            end
        if S.x(i,2,k+1)>maxxbeta
            S.x(i,2,k+1)=maxxbeta;
        elseif S.x(i,2,k+1)<minxbeta
            S.x(i,2,k+1)=minxbeta;
        else
            end
    end;
end;
for i=1:ns
    S.cost(i)=cost_fit(S.x(i,:),k+1),Pres1,Flow1);
    if S.cost(i)<costp(i,k+1)% y(i,:)
        y(i,:)=S.x(i,:),k);
    else
        end;
    if S.cost(i)<costg(k)% yhat
        yhat=S.x(i,:),k);
    else
        end;
end;
yhat
toc

```

A3 – Transfer Function Goodness of Fit Cost Function (Includes Bayesian Filter) for PSO

```
function [y] = cost_fit_tf(x,Pres1,Flow1)

% cost_fit_tf.m
%
% Dane Sterbentz
% September 23, 2014, Version 1.0
%

-----

%
% Computation of the cost function (y) for a transfer function system
% identification model.
%

-----

y=0;

Fs = 1000; % Samling Frequency
Ts = 1/Fs; % Sampling interval

% Preliminary section: load data, set constants, initialize variables
%set parameters
samplerate1 = Fs; %samples per second
noutputs1 = 50; %output quantization levels
ratemax1 = 1; %rectified EMG is normalized to max value of 1
inscale1 = 1; %arbitrary input scaling
alpha1 = x(:,1) / samplerate1; %sets diffusion rate
beta1 = x(:,2) / (noutputs1 * samplerate1); %sets probability of
sudden jumps

%calculate rectified EMG after removing the mean, and normalize
Pres1 = inscale1 * ratemax1 * Pres1 / max(Pres1); %input
prescaling to use full output range
Flow1 = inscale1 * ratemax1 * Flow1 / max(Flow1); %input
prescaling to use full output range

%initialize variables
% x is the latent variable (the driving rate)
% MAP is the output estimate
x1 = linspace(ratemax1/noutputs1, ratemax1, noutputs1)'; %don't start
with zero because requires n=0 exactly to match
Pres_Filt1 = zeros(length(Pres1),1); %store the
bayes estimates
Flow_Filt1 = zeros(length(Flow1),1); %store the
bayes estimates
g1 = [(alpha1/2) (1 - alpha1) (alpha1/2)]; %approximate
spatial second derivative operator

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
```

```

% Following is the main section of the algorithm; steps are numbered as
in the text
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%1. Initialize  $p(x,0) = 1$ ;
prior1 = ones(noutputs1,1) / noutputs1;           %start with uniform
prior

for t1=1:length(Pres1) %iterate for each sample

    %2. Forward propagate  $p(x,t-)$ 
    %            $?p(x-?,t-1)+(1-2?)p(x,t-1)+?p(x+?,t-1)+?(1-?)p(x,t-1)$ ;
    prior1 = filtfilt(g1, 1, prior1);           %drift term by
convolving with second derivative operator
    prior1 = beta1 + (1-beta1) * prior1;       %sets probability
of a sudden jump

    %3. Measure the rectified emg;
    Presvall1 = Pres1(t1);                     %if this were
online, would read a new sample here

    %4. Calculate the posterior likelihood function
    %            $P(x,t) \hat{=} P(\text{emg}|x)p(x,t-)$ ;
    measurement_model1 = 2*exp(-
(Presvall1).^2./ (2.*(x1).^2))./(2.*pi.*x1.^2).^ (1/2); %half Gaussian
model for  $P(\text{Pres1}|x)$ 
    posterior1 = measurement_model1 .* prior1;   %calculate posterior
density using Bayes rule

    %5. Output the signal estimate  $\text{MAP}(x(t)) = \text{argmax } P(x,t)$ ;
    pp1 = min(find(posterior1 == max(posterior1))); %find the
maximum of the posterior density
    if (pp1 > 1 && pp1 < length(posterior1)),    %interpolate to
find the zero
        dL1 = posterior1(pp1-1) - posterior1(pp1);
        dR1 = posterior1(pp1) - posterior1(pp1+1);
        PeakIndex1 = (pp1 - .5 - (dL1/ (dR1 - dL1))); %index runs
from 1 to noutputs
    else
        PeakIndex1 = pp1; %if maximum occurs at an endpoint do not
interpolate
    end
    Pres_Filt1(t1) = (ratemax1 / (noutputs1-1)) * PeakIndex1; %convert
index of peak value to scaled value

    %6. Divide  $p(x,t)$  by a constant C so that  $\int p(x,t) dx = 1$ ;
    posterior1 = posterior1 / sum(posterior1); %normalize the
density

    %7. Repeat from step 2;
    prior1 = posterior1; %prior for next
iteration is posterior from this iteration
end

```

```

%1. Initialize p(x,0) = 1;
prior1 = ones(noutputs1,1) / noutputs1;           %start with uniform
prior

for t1=1:length(Flow1) %iterate for each sample

    %2. Forward propagate p(x,t-) Å
    %           ?p(x-?,t-1)+(1-2?)p(x,t-1)+?p(x+?,t-1)+?(1-?)p(x,t-1);
    prior1 = filtfilt(g1, 1, prior1);           %drift term by
convolving with second derivative operator
    prior1 = beta1 + (1-beta1) * prior1;       %sets probability
of a sudden jump

    %3. Measure the rectified emg;
    Flowvall = Flow1(t1);                       %if this were
online, would read a new sample here

    %4. Calculate the posterior likelihood function
    %           P(x,t) Å P(emg|x)p(x,t-);
    measurement_model1 = 2*exp(-
(Flowvall).^2./(2.*(x1).^2))./(2.*pi.*x1.^2).^(1/2); %half Gaussian
model for P(Flow1|x)
    posterior1 = measurement_model1 .* prior1;   %calculate posterior
density using Bayes rule

    %5. Output the signal estimate MAP(x(t)) = argmax P(x,t);
    pp1 = min(find(posterior1 == max(posterior1))); %find the
maximum of the posterior density
    if (pp1 > 1 && pp1 < length(posterior1)),    %interpolate to
find the zero
        dL1 = posterior1(pp1-1) - posterior1(pp1);
        dR1 = posterior1(pp1) - posterior1(pp1+1);
        PeakIndex1 = (pp1 - .5 - (dL1/ (dR1 - dL1))); %index runs
from 1 to noutputs
    else
        PeakIndex1 = pp1; %if maximum occurs at an endpoint do not
interpolate
    end
    Flow_Filt1(t1) = (ratemax1 / (noutputs1-1)) * PeakIndex1; %convert
index of peak value to scaled value

    %6. Divide p(x,t) by a constant C so that ° p(x,t) dx = 1;
    posterior1 = posterior1 / sum(posterior1); %normalize the
density

    %7. Repeat from step 2;
    prior1 = posterior1; %prior for next
iteration is posterior from this iteration
end

output1 = Pres_Filt1/max(Pres_Filt1)-mean(Pres_Filt1/max(Pres_Filt1));
input1 = Flow_Filt1/max(Flow_Filt1)-mean(Flow_Filt1/max(Flow_Filt1));
Pres_data1 = Pres1-mean(Pres1);
Flow_data1 = Flow1-mean(Flow1);

```

```

% System identification transfer function, fit, and cost function
% calculation
%


---


Data = iddata(output1,input1,Ts,'InputName','Flow
Coefficient','OutputName','Pressure Rise Coefficient');

% Transfer function estimation:
Options = tfestOptions;
Options.Display = 'on';

tf1 = tfest(Data, 4, 3,Options);

% Fit calculation for measured and simulated transfer function model
data:
tf1_idss = idss(tf1);
statel = findstates(tf1_idss,Data);
simOpt = simOptions('InitialCondition',statel);
output1_sim = sim(tf1_idss,Data,simOpt);
output1_model = output1_sim.y;
fit = [1 - norm(output1 - output1_model)/norm(output1-
mean(output1))]*100;

% Cost function calculation [%]:
y = abs(100-fit); % fitness evaluation (objective function)
compare(Data,tf1);
end

```

A4 – ARX Goodness of Fit Cost Function (Includes Bayesian Filter) for PSO

```
function [y] = cost_fit_arx(x,Pres1,Flow1)

% cost_fit_arx.m
%
% Dane Sterbentz
% September 23, 2014, Version 1.0
%

-----
%
% Computation of the cost function (y) for an ARX system identification
% model.
%
-----

y=0;

Fs = 2000; % Samling Frequency
Ts = 1/Fs; % Sampling interval

% Preliminary section: load data, set constants, initialize variables
%set parameters
samplerate1 = Fs; %samples per second
noutputs1 = 50; %output quantization levels
ratemax1 = 1; %rectified EMG is normalized to max value of 1
inscale1 = 1; %arbitrary input scaling
alpha1 = x(:,1) / samplerate1; %sets diffusion rate
beta1 = x(:,2) / (noutputs1 * samplerate1); %sets probability of
sudden jumps

%calculate rectified EMG after removing the mean, and normalize
Pres1 = inscale1 * ratemax1 * Pres1 / max(Pres1); %input
prescaling to use full output range
Flow1 = inscale1 * ratemax1 * Flow1 / max(Flow1); %input
prescaling to use full output range

%initialize variables
% x is the latent variable (the driving rate)
% MAP is the output estimate
x1 = linspace(ratemax1/noutputs1, ratemax1, noutputs1)'; %don't start
with zero because requires n=0 exactly to match
Pres_Filt1 = zeros(length(Pres1),1); %store the
bayes estimates
Flow_Filt1 = zeros(length(Flow1),1); %store the
bayes estimates
g1 = [(alpha1/2) (1 - alpha1) (alpha1/2)]; %approximate
spatial second derivative operator

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
```

```

% Following is the main section of the algorithm; steps are numbered as
in the text
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%1. Initialize  $p(x,0) = 1$ ;
prior1 = ones(noutputs1,1) / noutputs1;           %start with uniform
prior

for t1=1:length(Pres1) %iterate for each sample

    %2. Forward propagate  $p(x,t-)$ 
    %            $?p(x-?,t-1)+(1-2?)p(x,t-1)+?p(x+?,t-1)+?(1-?)p(x,t-1)$ ;
    prior1 = filtfilt(g1, 1, prior1);           %drift term by
convolving with second derivative operator
    prior1 = beta1 + (1-beta1) * prior1;       %sets probability
of a sudden jump

    %3. Measure the rectified emg;
    Presvall1 = Pres1(t1);                     %if this were
online, would read a new sample here

    %4. Calculate the posterior likelihood function
    %            $P(x,t) \hat{=} P(\text{emg}|x)p(x,t-)$ ;
    measurement_model1 = 2*exp(-
(Presvall1).^2./ (2.*(x1).^2))./(2.*pi.*x1.^2).^ (1/2); %half Gaussian
model for  $P(\text{Pres1}|x)$ 
    posterior1 = measurement_model1 .* prior1; %calculate posterior
density using Bayes rule

    %5. Output the signal estimate  $\text{MAP}(x(t)) = \text{argmax } P(x,t)$ ;
    pp1 = min(find(posterior1 == max(posterior1))); %find the
maximum of the posterior density
    if (pp1 > 1 && pp1 < length(posterior1)), %interpolate to
find the zero
        dL1 = posterior1(pp1-1) - posterior1(pp1);
        dR1 = posterior1(pp1) - posterior1(pp1+1);
        PeakIndex1 = (pp1 - .5 - (dL1/ (dR1 - dL1))); %index runs
from 1 to noutputs
    else
        PeakIndex1 = pp1; %if maximum occurs at an endpoint do not
interpolate
    end
    Pres_Filt1(t1) = (ratemax1 / (noutputs1-1)) * PeakIndex1; %convert
index of peak value to scaled value

    %6. Divide  $p(x,t)$  by a constant C so that  $\int p(x,t) dx = 1$ ;
    posterior1 = posterior1 / sum(posterior1); %normalize the
density

    %7. Repeat from step 2;
    prior1 = posterior1; %prior for next
iteration is posterior from this iteration
end

```

```

%1. Initialize p(x,0) = 1;
prior1 = ones(noutputs1,1) / noutputs1;           %start with uniform
prior

for t1=1:length(Flow1) %iterate for each sample

    %2. Forward propagate p(x,t-) Å
    %           ?p(x-?,t-1)+(1-2?)p(x,t-1)+?p(x+?,t-1)+?(1-?)p(x,t-1);
    prior1 = filtfilt(g1, 1, prior1);             %drift term by
convolving with second derivative operator
    prior1 = beta1 + (1-beta1) * prior1;         %sets probability
of a sudden jump

    %3. Measure the rectified emg;
    Flowvall = Flow1(t1);                         %if this were
online, would read a new sample here

    %4. Calculate the posterior likelihood function
    %           P(x,t) Å P(emg|x)p(x,t-);
    measurement_model1 = 2*exp(-
(Flowvall).^2./(2.*(x1).^2))./(2.*pi.*x1.^2).^(1/2); %half Gaussian
model for P(Flow1|x)
    posterior1 = measurement_model1 .* prior1;     %calculate posterior
density using Bayes rule

    %5. Output the signal estimate MAP(x(t)) = argmax P(x,t);
    pp1 = min(find(posterior1 == max(posterior1))); %find the
maximum of the posterior density
    if (pp1 > 1 && pp1 < length(posterior1)),      %interpolate to
find the zero
        dL1 = posterior1(pp1-1) - posterior1(pp1);
        dR1 = posterior1(pp1) - posterior1(pp1+1);
        PeakIndex1 = (pp1 - .5 - (dL1/ (dR1 - dL1))); %index runs
from 1 to noutputs
    else
        PeakIndex1 = pp1; %if maximum occurs at an endpoint do not
interpolate
    end
    Flow_Filt1(t1) = (ratemax1 / (noutputs1-1)) * PeakIndex1; %convert
index of peak value to scaled value

    %6. Divide p(x,t) by a constant C so that ° p(x,t) dx = 1;
    posterior1 = posterior1 / sum(posterior1);     %normalize the
density

    %7. Repeat from step 2;
    prior1 = posterior1;                           %prior for next
iteration is posterior from this iteration
end

output1 = Pres_Filt1/max(Pres_Filt1)-mean(Pres_Filt1/max(Pres_Filt1));
input1 = Flow_Filt1/max(Flow_Filt1)-mean(Flow_Filt1/max(Flow_Filt1));
Pres_data1 = Pres1-mean(Pres1);
Flow_data1 = Flow1-mean(Flow1);

```

```

% System identification ARX model, fit, and cost function
% calculation
%


---


Data = iddata(output1,input1,Ts,'InputName','Flow
Coefficient','OutputName','Pressure Rise Coefficient');

% ARX function estimation:
Opt = arxOptions;

arx20201 = arx(Data,[20 20 1], Opt);

% Fit calculation for measured and simulated transfer function model
data:
arx20201_idss = idss(arx20201);
% statel = findstates(arx20201_idss,Data);
% simOpt = simOptions('InitialCondition',statel);
% output1_sim = sim(arx20201_idss,Data,simOpt);
% output1_model = output1_sim.y;
% fit = [1 - norm(output1 - output1_model)/norm(output1-
mean(output1))]*100;
[y,fit,x0]=compare(Data,arx20201);

% Cost function calculation [%]:
y = abs(100-fit); % fitness evaluation (objective function)

% hold on
% time = (1:1:length(Pres1))*Ts;
% plot(time,output1_model,'b')
% plot(time,output1,'k')
% hold off

end

```

A5 – Pole Variation Investigation

```
% polevariation.m
% February 8, 2015
% Dane M. Sterbentz, Marco P. Schoen
%



---


% Using a set of experiments on one operating-point, the variation of
the
% poles of the identified system is analysed. In particular, the
empirical
% unbiased covariance estimate of the parameter estimate is computed.
% Circles are then plotted for each pole with a radius equal to 3 times
the
% square root of the unbiased covariance estimate.
%



---


% 1. Load data
clear
load p58case1.txt;
u1=p58case1(60000:120000,1);
y1=p58case1(60000:120000,2);
% load p55case1.txt;
% u1=p55case1(60000:120000,1);
% y1=p55case1(60000:120000,2);
% load p52case1.txt;
% u1=p52case1(75000:135000,1);
% y1=p52case1(75000:135000,2);
% load p495case1.txt;
% u1=p495case1(60000:120000,1);
% y1=p495case1(60000:120000,2);

% 2. Filter data
Fs = 2000; % Sampling frequency is 2000 Hz
d = fdesign.lowpass('Fp,Fst,Ap,Ast',100,120,0.5,80,Fs);
Hd = design(d); % filter with low pass 100Hz/120Hz
yf1=filter(Hd,y1);

% 3. Compose iddata vector
data1=iddata(y1,u1,(1/Fs));
dataf1=iddata(yf1,u1,(1/Fs));

% 4. Determine optimum order using AIC criteria
min=100;ord=[0,0,0];ordf=ord;minf=min;
for na=1:10
    for nb=1:10
        for nk=1:2
            tmodel=arx(data1,[na,nb,nk]);
            tfmodel=arx(dataf1,[na,nb,nk]);
            infcr(na,nb,nk)=aic(tmodel);
            infcrf(na,nb,nk)=aic(tfmodel);
            if infcr(na,nb,nk)<min
                min=infcr(na,nb,nk);
            end
        end
    end
end
```

```

        ord=[na,nb,nk];
    else
    end
    if infcrf(na,nb,nk)<minf
        minf=infcrf(na,nb,nk);
        ordf=[na,nb,nk];
    else
    end
end
end
end

% 5. Define models with optimum parameter order
modell1=arx(data1,ord);
modelf1=arx(dataf1,ordf);

load p58case2.txt;
u2=p58case2(60000:120000,1);
y2=p58case2(60000:120000,2);
% load p55case2.txt;
% u2=p55case2(60000:120000,1);
% y2=p55case2(60000:120000,2);
% load p52case2.txt;
% u2=p52case2(60000:120000,1);
% y2=p52case2(60000:120000,2);
% load p495case2.txt;
% u2=p495case2(60000:120000,1);
% y2=p495case2(60000:120000,2);
yf2=filter(Hd,y2);
data2=iddata(y2,u2,(1/Fs));
dataf2=iddata(yf2,u2,(1/Fs));
model2=arx(data2,ord);
modelf2=arx(dataf2,ordf);

load p58case3.txt;
u3=p58case3(60000:120000,1);
y3=p58case3(60000:120000,2);
% load p55case3.txt;
% u3=p55case3(60000:120000,1);
% y3=p55case3(60000:120000,2);
% load p52case3.txt;
% u3=p52case3(60000:120000,1);
% y3=p52case3(60000:120000,2);
% load p495case3.txt;
% u3=p495case3(60000:120000,1);
% y3=p495case3(60000:120000,2);
yf3=filter(Hd,y3);
data3=iddata(y3,u3,(1/Fs));
dataf3=iddata(yf3,u3,(1/Fs));
model3=arx(data3,ord);
modelf3=arx(dataf3,ordf);

load p58case4.txt;
u4=p58case4(60000:120000,1);
y4=p58case4(60000:120000,2);
% load p55case4.txt;

```

```

% u4=p55case4(60000:120000,1);
% y4=p55case4(60000:120000,2);
% load p52case4.txt;
% u4=p52case4(60000:120000,1);
% y4=p52case4(60000:120000,2);
% load p495case4.txt;
% u4=p495case4(60000:120000,1);
% y4=p495case4(60000:120000,2);
yf4=filter(Hd,y4);
data4=iddata(y4,u4,(1/Fs));
dataf4=iddata(yf4,u4,(1/Fs));
model4=arx(data4,ord);
modelf4=arx(dataf4,ordf);

load p58case5.txt;
u5=p58case5(60000:120000,1);
y5=p58case5(60000:120000,2);
% load p55case5.txt;
% u5=p55case5(60000:120000,1);
% y5=p55case5(60000:120000,2);
% load p52case5.txt;
% u5=p52case5(60000:120000,1);
% y5=p52case5(60000:120000,2);
yf5=filter(Hd,y5);
data5=iddata(y5,u5,(1/Fs));
dataf5=iddata(yf5,u5,(1/Fs));
model5=arx(data5,ord);
modelf5=arx(dataf5,ordf);

load p58case6.txt;
u6=p58case6(60000:120000,1);
y6=p58case6(60000:120000,2);
% load p55case6.txt;
% u6=p55case6(64000:124000,1);
% y6=p55case6(64000:124000,2);
% load p52case6.txt;
% u6=p52case6(60000:120000,1);
% y6=p52case6(60000:120000,2);
yf6=filter(Hd,y6);
data6=iddata(y6,u6,(1/Fs));
dataf6=iddata(yf6,u6,(1/Fs));
model6=arx(data6,ord);
modelf6=arx(dataf6,ordf);

load p58case7.txt;
u7=p58case7(60000:120000,1);
y7=p58case7(60000:120000,2);
% load p55case7.txt;
% u7=p55case7(60000:120000,1);
% y7=p55case7(60000:120000,2);
% load p52case7.txt;
% u7=p52case7(60000:120000,1);
% y7=p52case7(60000:120000,2);
yf7=filter(Hd,y7);
data7=iddata(y7,u7,(1/Fs));
dataf7=iddata(yf7,u7,(1/Fs));

```

```

model7=arx(data7,ord);
modelf7=arx(dataf7,ordf);

check_495 = exist('y5')

if check_495 == 0
    % 6. Find nominal Parameter Estimate, Theta0
    [ma,nu]=size(model1.A);
    for k=1:nu

Theta0p(k)=model1.A(:,k)+model2.A(:,k)+model3.A(:,k)+model4.A(:,k);

Theta0fp(k)=modelf1.A(:,k)+modelf2.A(:,k)+modelf3.A(:,k)+modelf4.A(:,k)
;
    end;
    Theta0=Theta0p/4;
    Theta0f=Theta0fp/4;

% 7. Compute empirical unbiased Covariance Estimate, GammaE
GammaE=(1/3)*(model1.A-Theta0)*(model1.A-Theta0)'+((model2.A-
Theta0)*(model2.A-Theta0)'+((model3.A-Theta0)*(model3.A-
Theta0)'+((model4.A-Theta0)*(model4.A-Theta0)')));
GammaEf=(1/3)*(modelf1.A-Theta0f)*(modelf1.A-Theta0f)'+((modelf2.A-
Theta0f)*(modelf2.A-Theta0f)'+((modelf3.A-Theta0f)*(modelf3.A-
Theta0f)'+((modelf4.A-Theta0f)*(modelf4.A-Theta0f)')));

else
% 6. Find nominal Parameter Estimate, Theta0
    [ma,nu]=size(model1.A);
    for k=1:nu

Theta0p(k)=model1.A(:,k)+model2.A(:,k)+model3.A(:,k)+model4.A(:,k)+mode
l5.A(:,k)+model6.A(:,k)+model7.A(:,k);

Theta0fp(k)=modelf1.A(:,k)+modelf2.A(:,k)+modelf3.A(:,k)+modelf4.A(:,k)
+modelf5.A(:,k)+modelf6.A(:,k)+modelf7.A(:,k);
    end;
    Theta0=Theta0p/7;
    Theta0f=Theta0fp/7;

% 7. Compute empirical unbiased Covariance Estimate, GammaE
GammaE=(1/6)*(model1.A-Theta0)*(model1.A-Theta0)'+((model2.A-
Theta0)*(model2.A-Theta0)'+((model3.A-Theta0)*(model3.A-
Theta0)'+((model4.A-Theta0)*(model4.A-Theta0)'+((model5.A-
Theta0)*(model5.A-Theta0)'+((model6.A-Theta0)*(model6.A-
Theta0)'+((model7.A-Theta0)*(model7.A-Theta0)')));
GammaEf=(1/6)*(modelf1.A-Theta0f)*(modelf1.A-Theta0f)'+((modelf2.A-
Theta0f)*(modelf2.A-Theta0f)'+((modelf3.A-Theta0f)*(modelf3.A-
Theta0f)'+((modelf4.A-Theta0f)*(modelf4.A-Theta0f)'+((modelf5.A-
Theta0f)*(modelf5.A-Theta0f)'+((modelf6.A-Theta0f)*(modelf6.A-
Theta0f)'+((modelf7.A-Theta0f)*(modelf7.A-Theta0f)')));
end

[nr,nt]=size(roots(model1.A)); % determine number of roots nr
nomroots=roots(Theta0);nomrootsf=roots(Theta0f);

```

```

% Plot circles for each pole with a radius equal to 3 times the square
root
% of the unbiased covariance estimate (No Filtering)
figure
grid;
xlim([6.8, 7.8]);
ylim([-3, 3]);
title('Pole Variation for \phi = 0.58 (No Filtering)')
xlabel('Real Axis');ylabel('Imaginary Axis');hold
radius=3*(GammaE^0.5);
for k=1:nr
    lambdanom(k)=log(nomroots(k)/(1/Fs));
    circle(real(lambdanom(k)),imag(lambdanom(k)),radius);
end;

% % Plot circles for each pole with a radius equal to 3 times the
square root
% % of the unbiased covariance estimate (With Filtering)
% figure
% plot(lambdaf,'*');
% grid;
% title('Pole Variation for \phi = 0.58 (With Filtering)')
% xlabel('Real Axis');ylabel('Imaginary Axis');hold
% radiusf=3*(GammaEf^0.5);
% for k=1:nr
%     lambdanomf(k)=log(nomrootsf(k)/(1/Fs));
%     circle(real(lambdanomf(k)),imag(lambdanomf(k)),radiusf);
% end;

```

APPENDIX B – Matlab™ Files for Fast Dynamics Analysis

B1 – Spline Interpolated Autocorrelation Coefficient Data Set

Calculation

```
% splinecorr.m
%
% Dane Sterbentz, Marco P. Schoen
% January 7, 2015, Version 1.0
%
% The steps for the "Fast Dynamics" data analysis that are executed in
this program are as follows:
%     1. Loads the dynamic pressure data collected at the IET
%     2. Subtracts the mean of the sensor drift from each sensor
signal
%     3. Calibration factors are used to convert the pressure signal
from volts
%         (V) to Pascals (Pa)
%     4. The Hall effect sensor data or "datav(:,11)" is
%         used to collect 30 point pressure data groups for the same
rotor
%         position at each revolution
%     5. Correlation coefficient is calculated between 30 point data
groups using the Matlab function "corrcoef"
%     6. Substitute the correlation coefficients back into the data
array 15 points
%         behind the end of the 30 point data group
%     7. The missing data in between the correlation coefficient
points is set to NaN
%     8. Matlab "spline" function is used to fit a cubic spline curve
to interpolate the missing data
%     9. The spline curve is then resampled at 1000 Hz to produce
%     10. Subtract the mean to obtain the output data set for system
identification, "y_output"
%
%


---


clear;
clc;
% Calibration factors [Pa/V]:
con(1)=968.3;
con(2)=879.9;
con(3)=891.9;
con(4)=603.7;
con(5)=599.2;
con(6)=599.5;
con(7)=596.8;
con(8)=600;
con(9)=917.7;

% 1. Loads the dynamic pressure data collected at the IET
load dynamic_data_056.dat;
% nstart_058 = 2.269e5;
nstart_056 = 3.751e5;
% nstart_054 = 2.63e5;
```

```

% nstart_054_s4_34 = 2.61e5;
% nstart_052 = 2.402e5;
% nstart_051_s6_24 = 2.78e5;
% nstart_051 = 2.887e5;
datav=dynamic_data_056(nstart_056:nstart_056+7.5e5,:); % data in [V]
% datav=dynamic_data_054(:,:); % data in [V]
[L,ca]=size(datav);
Fs = 20000;
Ts=1/Fs;

% 2. Subtracts the mean of the sensor drift from each sensor signal
load dynamic_data_zero.dat;
drift=dynamic_data_zero;
[Ld,col]=size(drift);
for i=1:col
    driftmean(i)=mean(drift(:,i));
    datavm(:,i)=datav(:,i)-driftmean(i);% data with drift adjusted
end

% 3. Calibration factors are used to convert the pressure signal from
volts (V) to Pascals (Pa)
for i=1:9
    datapm(:,i)=datavm(:,i)*con(i); % data in [Pa]
end;
datapm(:,10:12)=datav(:,10:12);

% 4. The Hall effect sensor data or "datav(:,11)" is used to collect 30
point pressure data groups for the same rotor position at each
revolution
count=1;
xd=zeros(30,3000,12);
for i=1:L
    x=datav(i,11);
    if x<1
        if datav(i+1,11)>1 % last point of injection
            %take a window of 30 data points
            xd(1:30,count,:)=datapm(i-30+1:i,:);
            count=count+1;
        else
            end
        else
            end
    end
end

% 5. Correlation coefficient is calculated between 30 point data groups
using the Matlab function "corrcoef"
for sensor=1:10
    for j=2:count-1 % count is the number of blade passages
        xcf=corrcoef(xd(:,j-1,sensor),xd(:,j,sensor));
        corcf(j,sensor)=xcf(2,1);
    end
end

% 6. Substitute the correlation coefficient points back into the data
array 15 points behind the end of the 30 point data group

```

```

% 7. The missing data in between the correlation coefficient points is
set to NaN
count=1;
y_corcf=zeros(L,10);
for i=1:L
    x=datav(i,11);
    if x<1
        if datav(i+1,11)>1
            y_corcf(i-15,:) = corcf(count,:);
            count=count+1;
            y_corcf(i,:)=NaN;
        else
            y_corcf(i,:)=NaN;
        end
    else
        y_corcf(i,:)=NaN;
    end
end

% 8. Matlab "spline" function is used to fit a cubic spline curve to
interpolate the missing data
% Determine index locations of correlation coefficients in data set.
for sensor = 1:10
    count = 1;
    for i = 1:length(y_corcf)
        if isnan(y_corcf(i,sensor))
        else
            yspline(count,sensor) = y_corcf(i,sensor);
            xspline(count,sensor) = i;
            count = count+1;
        end
    end
end

% Use spline function to create new data set and resample every 20
points.
u_input = datav(1:20:end,10);
for sensor = 1:10
% 9. The spline curve is then resampled at 1000 Hz to produce the
output data set for system identification
xxSI = 1:20:L;
Y_spline(:,sensor) = spline(xspline(:,sensor),yspline(:,sensor),xxSI);
% 10. Subtract the mean to obtain the output data set for system
identification, "y_output"
y_output(:,sensor) = Y_spline(:,sensor)-mean(Y_spline(:,sensor));
end

clearvars -except u_input y_output Y_spline yspline y_corcf datav

```