# A System Identification Approach to Active Control of Thermoacoustic Instabilities

by

Brent Jerome Brunell

B.S., Mechanical Engineering
Rensselaer Polytechnic Institute, 1994

Submitted to the Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Mechanical Engineering

at the

Massachusetts Institute of Technology

February 2000



Signature of Author .....
Department of Mechanical Engineering
January 14, 2000

Certified by .....
Anuradha M. Annaswamy
Associate Professor of Mechanical Engineering
Thesis Supervisor

Accepted by .....
Ain A. Sonin
Chairman, Department Committee on Graduate Students

# System Identification and Control of Thermoacoustic Instability in Continuous Combustion Process

by

Brent Jerome Brunell

Submitted to the Department of Mechanical Engineering
on January 14, 2000, in Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Mechanical Engineering

## Abstract

Continuous combustion processes are encountered in many applications ranging from heating and power generation to aircraft propulsion. When continuous combustion takes place in an acoustic resonator, the interaction between acoustic waves and the unsteady heat release of combustion may lead to growing pressure oscillations, or thermoacoustic instabilities. These instabilities are undesirable and lead to vibrations, high noise level, poor emissions, and high burn and heat transfer rates. Active combustor control has been increasingly used to suppress these instabilities by introducing continuously modulated external input into the combustion process. Since our understanding of the theoretical mechanisms that govern the instabilities is limited, it is difficult to obtain a control oriented physics-based model of these combustion processes. This thesis uses system identification techniques to identify the important dynamic elements and create a state space model of the combustor instabilities. This dynamic model, based on the input-output data from the combustor, is used to create a model-based active controller to suppress the instability. A methodology for creating the active control based on system identification is presented and evaluated on a nonlinear combustor model and an experimental combustor. The performance of several controller designs including lead-lag, LQG, and 'Bang-Bang' are investigated. Results show that the system identification based controllers are effective in suppressing the instability. Results from the nonlinear model show that LQG control results in the fastest transient time to a stable operating point, and 'Bang-Bang' control has the best steady state performance. The system identification based control on the experimental combustor reduces the minimum peak values by a factor of 4 and reduces the RMS pressure by 10 percent compared to the non-model based active control. The system identification procedure also revealed that energy can be transferred from the longitudinal mode to the bulk mode due to closed loop action and hence a successful active-control strategy must take into account the dynamic behavior at both of these frequencies.

Thesis Supervisor: Anuradha M. Annaswamy
Title: Associate Professor of Mechanical Engineering

# Acknowledgments

I have to thank Dave Michaud, my study partner from R.P.I, for carrying me through my introduction to computers and simulations. Without your help I probably would not have made it into graduate school.

Many people in the MIT community helped tremendously with my graduate school experience. Thank-you Professor Anuradha Annaswamy for your guidance and allowing me to work on an exciting and challenging project. Jean-Pierre Hathout, Mahmoud Fleifil, and Professor Ghoniem were all instrumental in assisting with my understanding of combustion and helping solve many technical problems. Aleksandar Kojic, I appreciated the many candid conversations during the war. J.P., thanks for helping show me everything else at MIT.

Many thanks to GE for their generous support. My gratitude and thanks go to my coworkers on the CF34 team for picking up when I couldn't. Special thanks to Jan Beseler for all the small and very large favors.

My gratitude to Dr. Ken Yu, formerly of China Lake Naval Air Weapons Station, and Ken Wilson for their experimental collaboration and to the Office of Naval Research for funding to run the experiment.

Heather, you mean everything. The world is a much brighter place with you in it.

# Table of Contents

# List of Appendices

# List of Figures

# List of Tables

# 1. Introduction

Continuous combustion processes are encountered in many applications ranging from heating and power generation to aircraft propulsion. When continuous combustion takes place in an acoustic resonator, the interaction between acoustic waves and the unsteady heat release of combustion may lead to growing pressure oscillations, or thermoacoustic instabilities. These instabilities are undesirable and lead to vibrations, high noise level, poor emissions, high burn and heat transfer rates. They can also trigger fan or compressor surge, or combustion blowouts when operating in poor flame stabilization conditions. The problem is becoming more prevalent as manufacturers are driven to low $NO_x$ emissions due to regulations.

Rayleigh [ 1 ] first hypothesized the theory that this growing pressure oscillation was caused by an interaction between the heat release rate and the pressure. Combustion instabilities first appeared as an engineering problem in turbojet afterburners in the late 1940's. At that time the afterburners were small, had high entry pressures, and low volumetric heat release rates. This allowed the oscillations to be taken care of by simple acoustic liners. As the trend towards larger combustors with low entry pressures and high volumetric heat release rate continued, instabilities became more frequent. Most attempts to reduce the pressure oscillations involved passive control of the instability by changing the fuel injection spatial placement, adding addition pilot injectors, modifying the geometry of the combustor, changing the secondary airflow, or adding acoustic baffles. All of these involve costly and time consuming hardware changes and do not guarantee that they will work under changing operating conditions.

With the advent of actuators, sensors, and processors that are fast, accurate, reliable, and cost effective, active combustor control has been increasingly used to suppress these instabilities by introducing continuously modulated external input into the combustion process. Combustor control systems typically contain several major components, as shown in Figure 1-1.

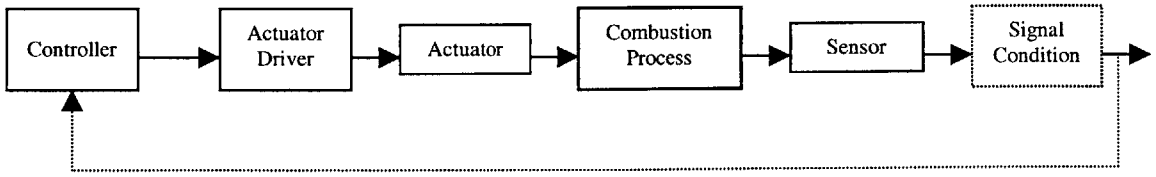



Figure 1-1: Typical Combustion Control System.

The control can be either open loop or closed loop, as indicated by the dashed line. The actuator is often a fuel injector for commercial size experiments [ 31 ], [ 28 ], but speakers [ 13 ] have been used successfully on smaller scale combustors . The sensor is typically a pressure transducer which is used to capture the acoustic behavior, however heat emission has also been used for feedback. Often times if the measured parameter is noisy or the magnitude is unacceptable for measurement it will be conditioned with filters and amplifiers.

There are two families of active controllers: experimentally derived and model-based controllers. Examples of the first category include [ 3 ], [ 4 ], and [ 28 ]. The experimental controllers are typically analog phase-shifter or phase-lock that are tuned based on trial and error experiments until the instability is suppressed. Because the controls are tuned for a single operating point and a single frequency, they do not perform well when secondary frequencies arise or the operating conditions change.

The model-based controllers can again be divided into two general categories by looking at the method for creating the combustor model, there are: physics based, and data or system identification based. Examples of physics based modeling and control can be found in [ 8 ], [ 13 ], and [ 30 ]. For this branch, the fundamental laws that govern the dynamics of the acoustics and heat release in combustion are utilized. While the physical modeling of simple laminar premixed combustion is maturing, our current understanding of the theoretical mechanisms that govern the instabilities in large scale commercial combustion where there is significant turbulence is limited. Therefore, it is difficult to obtain a control oriented physics-based model of these combustion processes. This leads logically to the second family of models based on input-output data using system identification techniques. Reference [ 31 ] used the input-output system identification approach to successfully identify all of the significant natural frequencies and damping accurately on a commercial scale combustor. This thesis uses system identification techniques to identify the dominant dynamics and then create a stabilizing controller. Once such a dynamic model and controller are determined, they could be used to pinpoint physical mechanisms that may be responsible for the unstable behavior. This information provides clues for obtaining physically based models.

A methodology, as outlined in Figure 1-2, for creating the active control based on system identification is evaluated on a nonlinear laminar combustor model and a turbulent experimental combustor. Very few results have appeared on the system identification based approach to control

thermoacoustic instabilities [ 29 ] and [ 31 ] , both of which concern control of laminar combustors. In both reports the system identification and control tools were used on the laminar combustor dynamics, they were not extended to turbulent combustors where no models currently exist. In this thesis, for the first time, an attempt is made to carry out active control using system identification methods on a near full-scale combustion rig under turbulent flow conditions.

To gain experience at identifying and controlling combustors, an evaluation model of a nonlinear, laminar, pre-mixed combustor with a fuel injector actuator is developed in Chapter 2. Models of two pulsed fuel injectors are created from experimental data and added to the evaluation model as actuators. The methodology of Figure 1-2 is followed and an acceptable system identification model is obtained. Model based controls including linear quadratic gaussian (LQG) and 'Bang-Bang' are investigated to determine the best performance. Results from the nonlinear evaluation model show that LQG control results in the fastest transient time to a stable operating point, and 'Bang-Bang' control has the best steady state performance.

Using the insights gained from the system identification method used on the nonlinear model, a similar procedure is used in the context of a 500 kW experimental dump combustor under turbulent flow conditions. Chapter 3 reports on the findings of the liquid fueled experimental dump combustor. Results from [ 28 ] show that the pressure oscillations in this combustor can be attenuated using experimentally derived controls. Using system identification tools, an initial model of the open loop uncontrolled combustor is obtained and validated. This corresponds to identifying the nonlinear region shown in Figure 1-3. A partially stabilizing control is obtained from the open loop model and the performance is compared to the experimentally derived controller. System identification is then accomplished on the closed loop system to obtain a model that captures the unstable dynamics of the combustor in the linear region of Figure 1-3. The control resulting from the closed loop model reduces the minimum peak values by a factor of 4 over the non-model based active control and reduces the RMS pressure by 10 percent.

The results and conclusion from the evaluation model and the experimental combustor are presented in chapter 4.

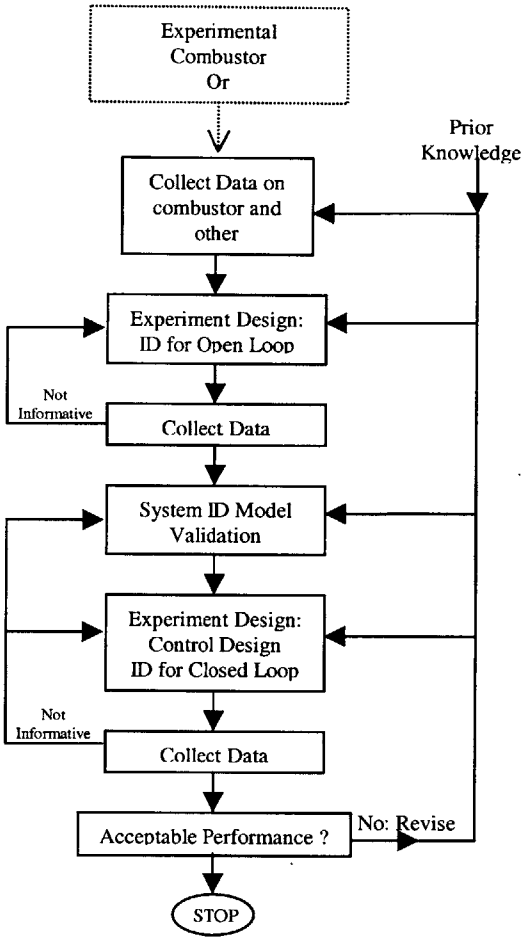



Figure 1-2: Methodology for system identification and active control of combustion

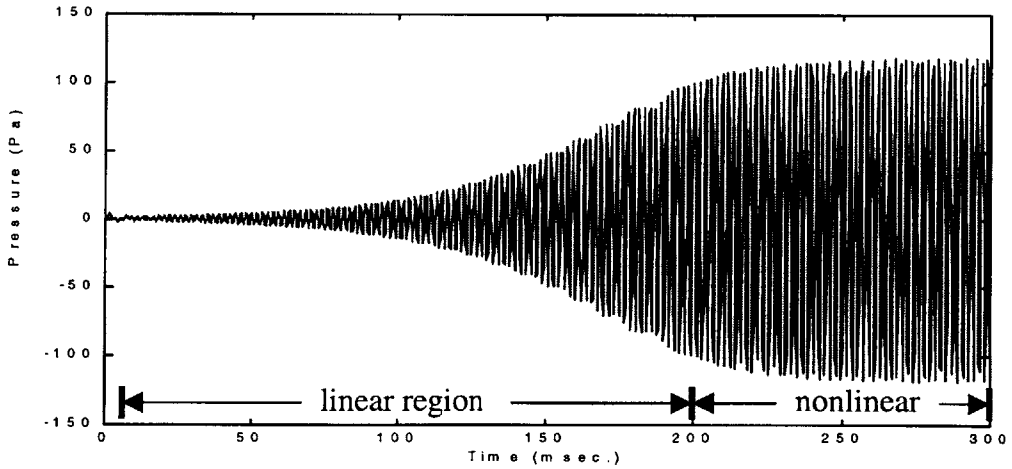



Figure 1-3: Linear and nonlinear regions of a typical combustor pressure response

# 2. System Identification and Control of a Laminar Combustor: A Numerical Study

## 2.1 Evaluation Model Development

System identification and control are investigated on a physically based finite dimensional evaluation model of a continuous combustion process that was developed at MIT; see [ 5 ] through [ 13 ]. The model demonstrates the characteristics of combustion instability. Nonlinearities, a pulsed fuel injector actuator, a microphone sensor, and noise are added to the model to complete the control system and make the model more realistic. The complete evaluation model is developed below.

### 2.1.1 Linear Combustion Model

Combustion instabilities result from the coupling of the acoustics and the flame heat release. The heat release rate at the flame front is a strong driver on the acoustic dynamics. The resulting unsteady acoustic pressure and velocity act as feedback affecting the heat release dynamics. The result is a feedback loop or dynamic coupling.

A physically based finite dimensional model of a continuous combustion process was developed at MIT that demonstrates the characteristics of combustion instability; see [ 5 ] through [ 13 ]. Some development of the dynamic combustor model is given in Appendix B. The resulting dynamic equations are given as

$$\dot{q}_f' = -b_3 q_f' + b_2\left[\bar{\phi}\tilde{u}' + \bar{u}\phi'\right], \tag{1}$$

$$\ddot{\eta}_i = -\omega_i \eta_i - 2\zeta\omega_i \dot{\eta}_i + \tilde{b}_i \dot{q}_f', \tag{2}$$

$$\tilde{u}' = \sum_{i=1}^{n} (\tilde{c}_i \dot{\eta}_i), \tag{3}$$

where $\phi'$ is the equivalence ratio perturbation, $\bar{\phi}$ is the mean equivalence ratio, $\dot{q}_f$ is the rate of heat release, $\tilde{u}'$ is the flow velocity perturbation, $\bar{u}$ is the mean flow velocity, $\eta_i$ is the time

varying component of pressure for the i$^{th}$ mode, $\zeta$ is the damping used for computational stability, $\omega_i$ is the natural frequency of the i$^{th}$ mode, and $b_3, b_2, \tilde{b}_i, \tilde{c}_i$ are constants defined in Appendix B. The system parameters are set to be similar to the MIT combustor rig, and are defined in Appendix C. The output equations to get pressure are

$$P = \overline{P} \sum \tilde{c}_{c_i} \eta_i \tag{4}$$

where $\overline{P}$ is the mean static pressure, and $\tilde{c}_{c_i}$ is a constant based on sensor location. The model performance versus the MIT lab pre-mixed combustor was verified in [ 13 ]. A system with 2 coupled acoustic modes is considered here. The pole zero locations for the linear model is shown in Figure 2-1, and listed in Table 1. The eigenvalues at 168 Hz and 538 Hz represent the acoustic modes of the combustor, and the eigenvalue at 58.7 Hz represents the heat release dynamics. The corresponding bode plot is given in Figure 2-2.

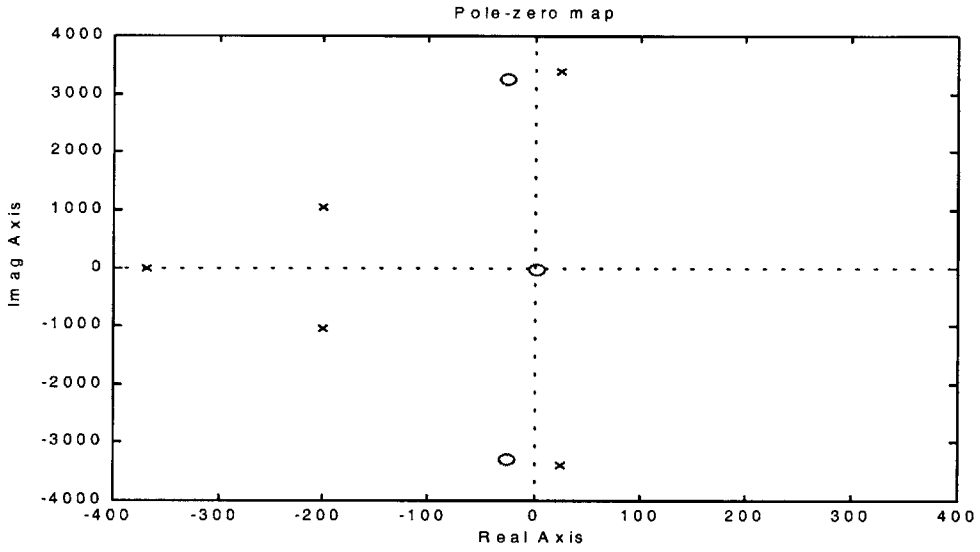



Figure 2-1: Linear combustor model pole-zero locations for input = $\phi'$, output = pressure. Poles are shown as x's and zeros are shown as 0's.

| Eigenvalues | Damping | Freq. (Hz) |
|---|---|---|
| -3.69e+002 | - - - | 58.7 |
| -2.02e+002 +/- 1.04e+003i | 1.91e-001 | 168.7 |
| 1.8729e+001 +/- 3.3815e+003i | -5.54e-003 | 538.2 |

Table 1: Linear combustor model eigenvalues.

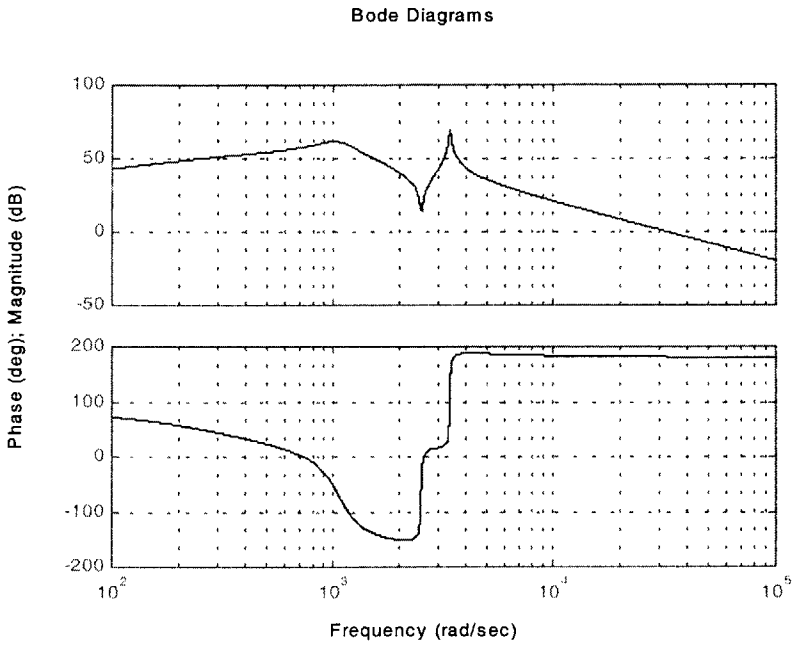



Figure 2-2: Linear combustor model bode plot for input = $\phi'$, output = pressure.

The time response of the linear model is shown in Figure 2-3. The characteristics of the linear model are that the pressure will grow towards infinity. In real combustors the exponentially diverging oscillations transition into limit-cycle behavior. A model for this nonlinear limit cycle behavior is developed below.

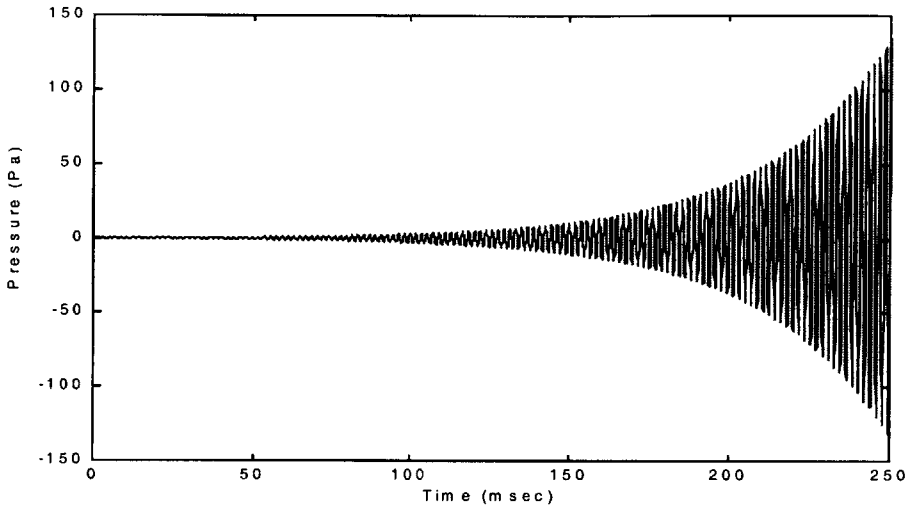



Figure 2-3: Pressure response of the linear combustor model.

### 2.1.2 Nonlinear Model

The characteristic dynamic behavior of thermoacoustic instabilities involve exponentially diverging oscillations which transition into limit-cycles. This limit-cycle behavior indicates the presence of a stabilizing nonlinearity. The linear model of section 2.1.1 captures the exponential growth characteristics of the instability. We quickly describe how the addition of nonlinearities to the linear model leads to limit-cycle behavior (see [ 13 ] for details).

In combustion processes, nonlinearities can occur in both the acoustics and the heat release. However, the nonlinearities in the heat release are the dominant drivers of the limit-cycle behavior. To model this, a nonlinear component is added between the unsteady velocity and the unsteady heat release as shown in Figure 2-4 as f(.).

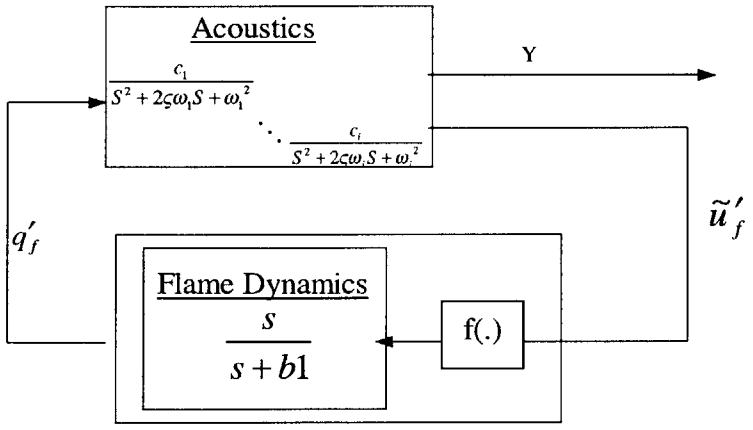



Figure 2-4: Nonlinear model of the thermoacoustic instability.

Phase - plane analysis can predict a limit cycle as being a phase change, gain change or both. One tool for predicting nonlinear oscillations is to use describing functions. Any system that can be transformed into the form shown in Figure 2-5 can be used in describing function analysis. Neglecting the $\phi'$ perturbation, the complete nonlinear model can be described by the equations

$$\dot{q}'_f + b_3 q'_f = b_2 \bar{\phi} f(\tilde{u}'_f) \,, \tag{5}$$

$$\ddot{\eta}_i = -\omega_i \eta_i - 2\zeta\omega_i \dot{\eta}_i + \tilde{b}_i \dot{q}'_f \,, \tag{6}$$

( 7 ) $$\tilde{u}'_f = \sum_{i=1}^{n} (\tilde{c}_i \dot{\eta}_i),$$

and more compactly, in operator form, as

( 8 ) $$\tilde{u}'_f = G(s) u'_n,$$

( 9 ) $$u'_n = -f(\tilde{u}'_f).$$

The negative sign appears in equation ( 9 ) to fit into the describing function form as shown in Figure 2-5.

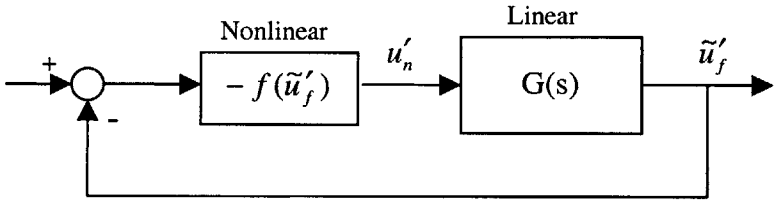



Figure 2-5: Nonlinear model in describing function form.

The goal is to evaluate the conditions on $f$ under which the nonlinear model in ( 8 ) and ( 9 ) generates limit-cycles. One condition that will cause the limit-cycle behavior involves a phase change between the unsteady velocity and the heat release rate. Assuming a nonlinearity $f = f_1$ where

( 10 ) $$f_1(u) = c_1 u - c_2 u^3,$$

where $c_1$ and $c_2$ are positive, the resulting describing function is given by

( 11 ) $$N(A,\omega) = \frac{1}{A}(b_1 + a_1 j) = c_1 - \frac{3}{4} c_2 A^2,$$

where A is the amplitude of the sinusoidal input to the nonlinearity, $\omega$ is the frequency, and $a_1 = 0$, $b_1 = c_1 A - \frac{3}{4} c_2 A^3$ are the Fourier coefficients. Due to the odd nature of the nonlinear function the describing function is a real function only of the amplitude of the input. From [ 25 ] the describing function analysis predicts the limit-cycle behavior when

( 12 ) $$G(j\omega) = -\frac{1}{N(A,\omega)}.$$

Both sides of ( 12 ) are plotted in the complex plane for a range of $A$ and $\omega$, as shown in Figure 2-6. The resulting intersection of the plot yields the solution value for $A$.

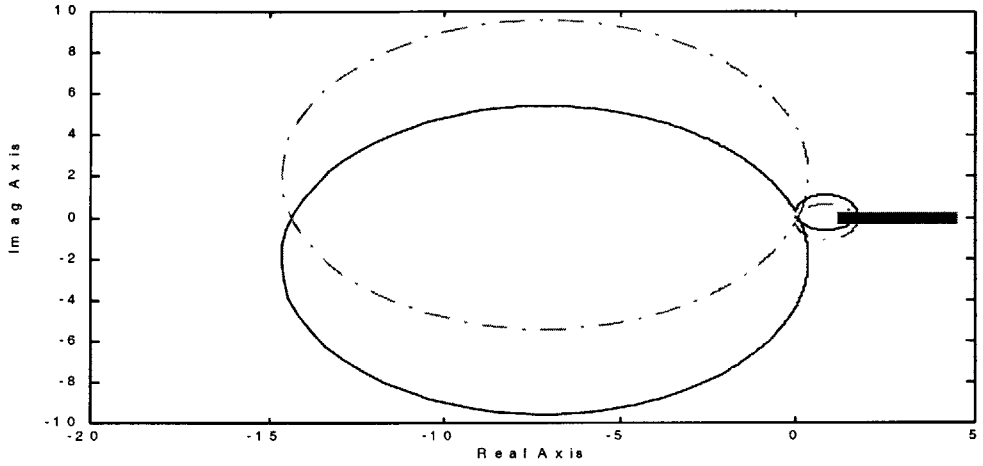



Figure 2-6: Nyquist diagram and describing function of combustor with $f_1$ (thin-line, dash- linear system Nyquist diagram: thick-line- describing function).

By adding the nonlinearity to the evaluation model the exponentially growing oscillations now transition into a limit-cycle behavior as shown in Figure 2-7.

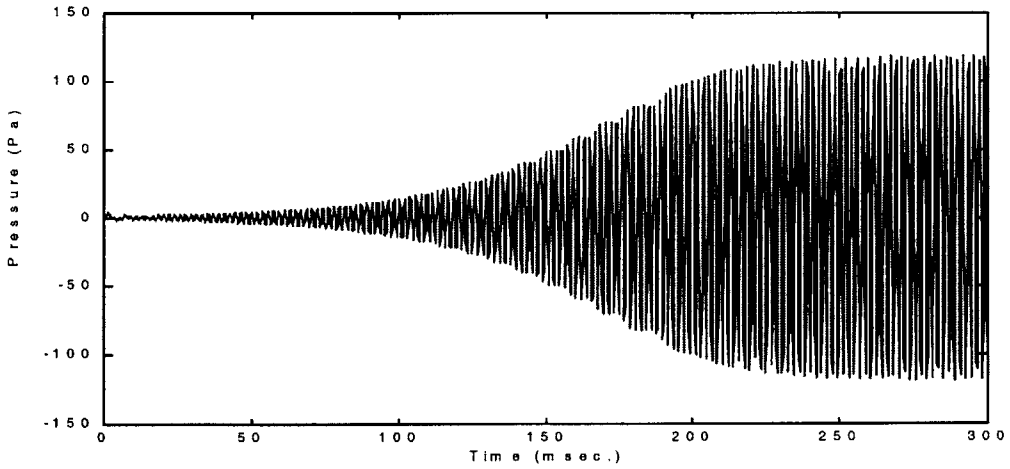



Figure 2-7: Pressure response of the combustor with nonlinear component.

### 2.1.3 Actuator and Sensor Dynamics

There have been several actuators used to successfully control combustor instabilities including pulsed fuel injectors, seen in section 3 of this paper, and proportional fuel injectors that directly affect the heat equation, while speakers have been shown to be effective at influencing the acoustics [ 8 ]. One scheme of actuator-sensor pair is shown in Figure 2-8. For the evaluation model a pressure transducer such as a microphone will be used as the sensor. A pulsed fuel injector will be used as the actuator for control authority of the combustor instability.

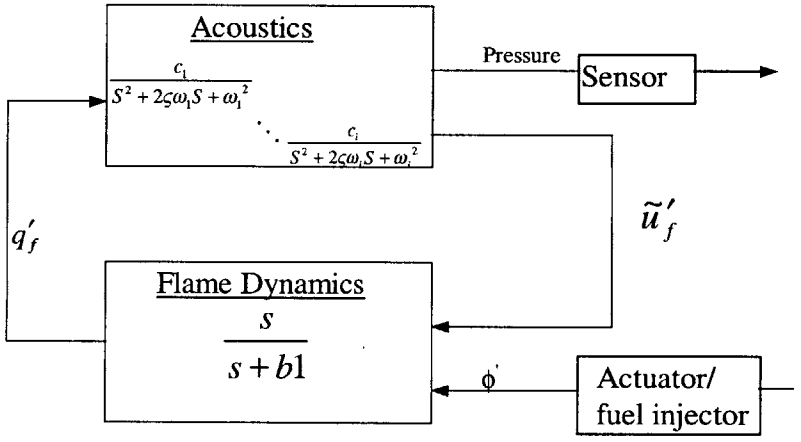



Figure 2-8: Input-output model schematic.

Microphones measure pressure, which is a good indication of the combustor oscillations. Typically, microphones have a flat frequency response over the acoustic frequencies of interest. Therefore, the microphone will be modeled as a pure gain.

The fuel injector is used to influence the heat dynamics through perturbations in phi prime while affecting only slightly phi bar, u bar and u prime. The equation is presented again for clarity.

( 13 ) $$\dot{q}'_f = -b_3 q'_f + b_2\left[\bar{\phi}\tilde{u}' + \bar{u}\phi'\right]$$

For a fuel injector to work properly there are several system issues to consider including; injector dynamics and bandwidth, max and min mass flows, flow velocity, mixing, flushing of mixing zone, secondary jet wake, noise, and delay. A study of two pulsed fuel injectors is accomplished

below. Both injectors are investigated for their dynamic response, which is used to create a transfer function that is implemented in the evaluation model.

### 2.1.3.1 Fuel Injector Experiment with Parker Model #9-130-905

The goal of this experiment is to determine the dynamic transfer function or model between the voltage input to the fuel injector, and the velocity or mass flow out. The data used from the experiment also gives insights into the feasibility of using the fuel injectors on a working combustor rig. The experimental setup to determine the injector dynamics is shown in Figure 2-10. The injector is supplied source air of 5 psi using a regulator. The valve driver, seen in Figure 2-9 and reference [ 24 ], sends a command voltage to open the valve when the reference signal from the computer steps from 0 to 10 volts. The injector will be overdriven at V1 = 38 volts for the first 0.5 msec to and then will hold at V2 = 13.5 volts. The injector operating range is 12 - 24 volts to stay open.

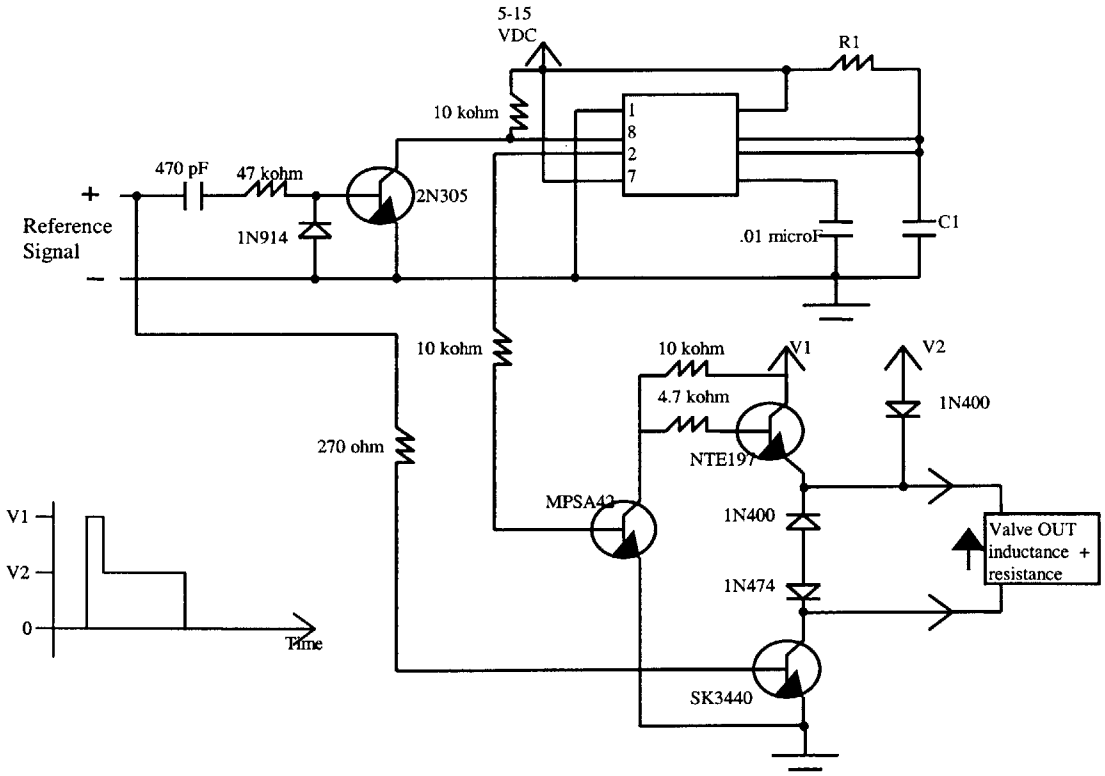



Figure 2-9: Valve driver schematic and voltage characteristic.

When the valve is opened this allows air to flow through the valve and over the hot film anemometer. The anemometer measures fluid velocity by sensing changes in heat transfer from the small electrically-heated sensor exposed to the air flow. The cooling effect resulting from the fluid flowing past the sensor is compensated for by increasing the current flow to the sensor. The magnitude of the current increase needed to keep the temperature constant is directly related to heat transfer and thus, flow velocity. The computer stores both the reference signal and the flow velocity for analysis.

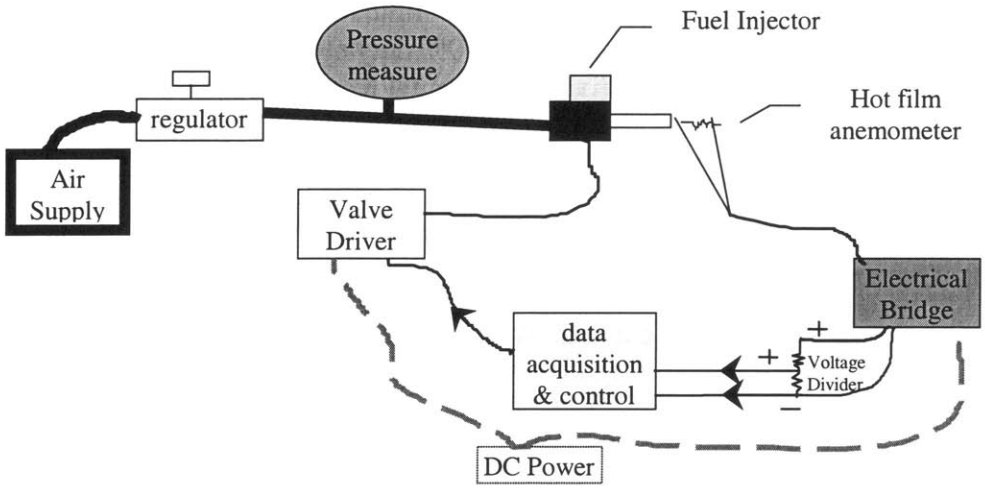



Figure 2-10: Fuel injector experimental setup.

The experimental results of the flow velocity response from the fuel injector for a 50% duty cycle reference voltage is shown in Figure 2-11 (ii) and Figure 2-11 (iv) for 50 Hz and 100 Hz frequencies respectively.

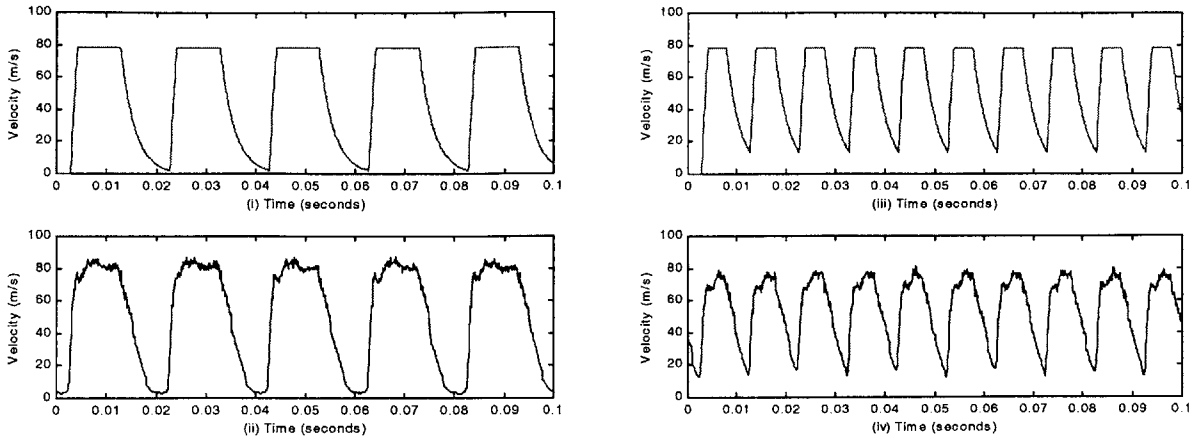



Figure 2-11: Velocity response of fuel injector: (i) simulation response of 50 Hz 50% duty cycle reference, (ii) experimental response of 50 Hz 50% duty cycle reference, (iii) simulation response of 100 Hz 50% duty cycle reference, (iv) experimental response of 100 Hz 50% duty cycle reference.

From these results it is determined that the injector behaves like a first-order system that is saturated. The saturation is due to the solenoid valve reaching its maximum open position and the flow through the valve becoming choked. A state space representation of the fuel injector model is developed;

$$\dot{V} = \frac{-1}{\tau} V + E_{in} \,, \tag{14}$$

$$Y = \frac{g_{ain}}{\tau} V \,, \tag{15}$$

where $\tau \sim 0.0028$ and $g_{ain} \sim 20$. To include the saturation effect in the model, constraints need to be implied on V

$$V = \int \dot{V}(t)dt \rightarrow if \ \min < V < \max \,, \tag{16}$$

$$V = \max \rightarrow V > \max \,, \tag{17}$$

$$V = \min \rightarrow V < \min \,. \tag{18}$$

The complete first-order model with saturation and a delay ~ 2.5 ms is developed using Simulink® and is shown in Figure 2-12. A comparison between the experiment and simulation results is

given in Figure 2-11 and Figure 2-13. The response to the 50% duty cycle input at 50 and 100 Hz is very similar between the model and the experimental tests, except for the saturation being a constant value for the model and being variable from the real injector. This is due to the solenoid not seating properly when the injector is commanded open.

The duty cycle sweeps of Figure 2-13 shows that the model again reproduces most of the experimental data. For the 100 Hz small duty cycles the real injector does not fire correctly. This is due to the injector being unable to open before the voltage is removed to close the valve.

This dynamic study determined that the time constant of the fuel injector is only 2.8 msec (bandwidth of 57 Hz), but for control effectiveness the bandwidth of the fuel injector needs to be equal to or greater than the 538 Hz unstable acoustic frequency.

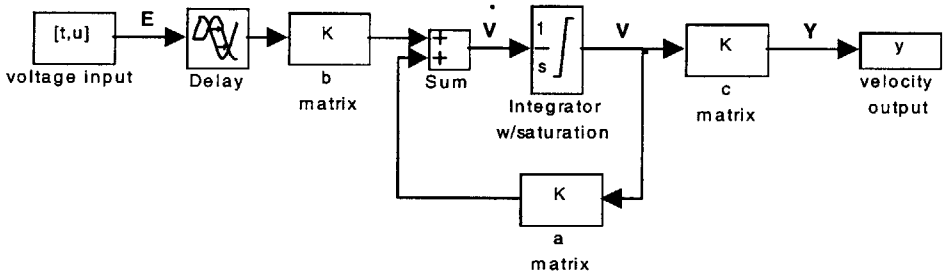



Figure 2-12: Fuel injector simulation model.

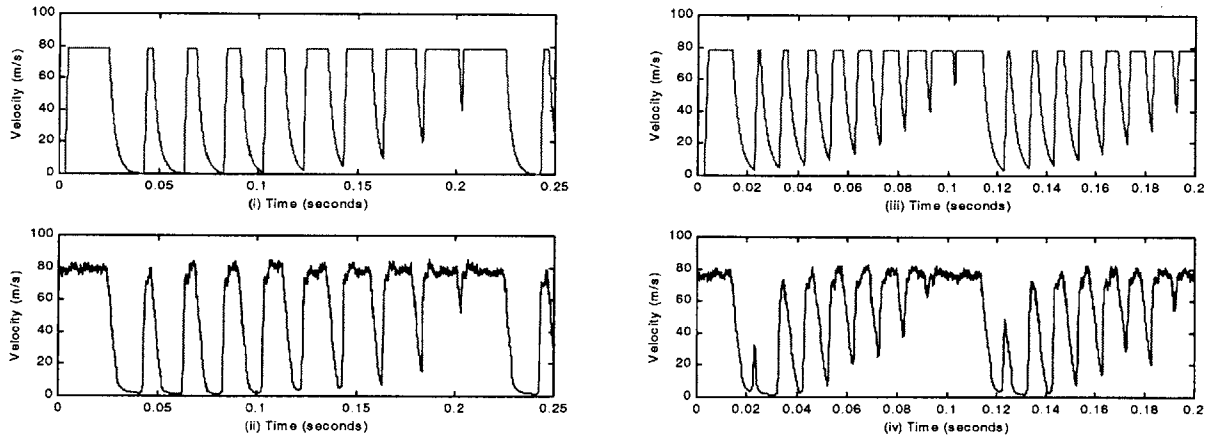



Figure 2-13: Velocity response of fuel injector: (i) simulation result of 50 Hz duty cycle sweep reference, (ii) experimental result of 50 Hz duty cycle sweep reference, (iii) simulation result of 100 Hz duty cycle sweep reference, (iv) experimental result of 100 Hz duty cycle sweep reference.

### 2.1.3.2 High Speed Fuel Injector Experiment with Parker Model # 9-633-900

The goal of this experiment is to determine the dynamic transfer function of a high speed pulsed fuel injector for the evaluation model. The experimental setup to determine the injector dynamics is shown in Figure 2-10. The injector is supplied source air of 10 psi using a regulator and has a 0.030 inch inner diameter 1 inch long Teflon extension tube. The valve driver used is the Parker Iota One. Shown in Figure 2-14 is experimental results for 5 volt, TTL square waves reference signals of 100 Hz and 500 Hz. The outputs are the experimental injector fuel flow output, and the derived model output fuel flow. The model response is from a first order plant with a 0.5 msec time delay.

$$\frac{Vo}{Vi} = \frac{K}{\tau s + 1}, \qquad (19)$$

Where k = 0.72 and $\tau$ = 2.5 msec. Figure 2-14 shows that the time response of the first order plant is a good approximation to the pulsed fuel injector.

If this injector was used on a working combustor rig with a 538 Hz unstable acoustic frequency it would be unable to reduce the pressure oscillations because the time constant for the fuel injector of 2.5 msec (bandwidth of 64 Hz) is too slow. Other characteristics of the injector made it unfeasible to be used on a pre-mixed laminar combustor. As stated earlier, there are several system issues to consider before a fuel injector will work properly on a gaseous, pre-mixed, laminar combustor including: max and min mass flows, flow velocity, mixing, flushing of mixing zone, secondary jet wake, noise, and delay. The min flow should be small relative to the mean flow. Because this injector bandwith is very low, the mean flow to min flow ratio is almost unity at the unstable frequency, see Figure 2-14. The combustor will see this as mainly a mean addition of heat, not a perturbation. The flow from the fuel injector must get mixed before burning or the flame will end up with a locally rich zone. This will change the flame characteristicand not have the intended control action or the fuel will pass through the pre-mixed flame and create a second flame that has no effect on the pre-mixed flame dynamics. Mixing gases at this high of a frequency and retainging a fuel perturbation is very tough. Other concerns are that if the fuel is injected near the flame sight the flow wakes will disrupt the laminar pre-mixed flow and create vortices upstream of the flame that will alter the burning characteristics in an uncontrolled fashion. The final concern is noise. If the solenoid valves are loud and a microphone is used to measure the

pressure, the microphone may capture the noise from the solenoid valves and make it very difficult to measure the combustor pressure correctly. This is usually not a problem in large turbulent combustors where the combustor acoustics are very loud compared to the fuel injector solenoids.

In the final pulsed fuel injector model that is used on the complete evaluation model, it will be assumed that the injector bandwidth is greater than the unstable frequency, there is complete mixing of the secondary fuel with the pre-mixed air fuel, there is no secondary jet wake, and that the min flow is zero.

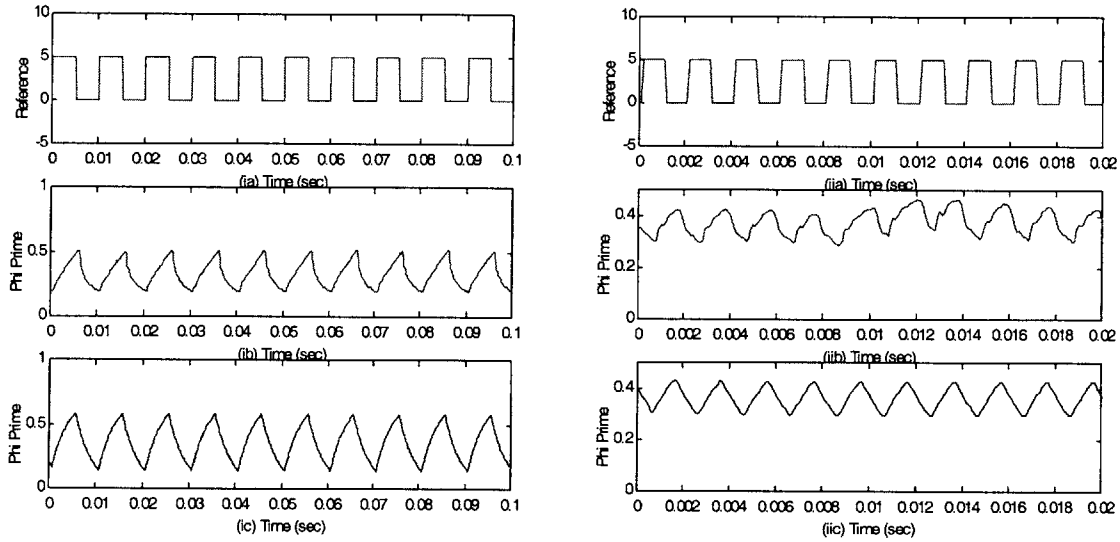



Figure 2-14: Fuel injector reference and response to 100 Hz (case-i.) and 500 Hz (case-ii). Shown for each case are a) reference, b) fuel injector response, c) model response.

### 2.1.3.3 Injector Model

The above dynamic study determined that the time constant of these fuel injectors is only 2.5 – 2.8 msec (bandwidth ~ 60 Hz), but for control effectiveness the bandwidth of the fuel injector needs to be on the order of 538 Hz or greater. The final injector model used has a bandwidth of 500 Hz, use the nonlinear characteristic similar to section 2.1.3.1, and have a 0.5 msec delay time. The time response of the final injector model to the input voltage is shown in Figure 2-15. For this model it is assumed that there is complete mixing of the secondary fuel with the pre-mixed air fuel after the 0.5 msec delay, there is no secondary jet wake, it does not affect the mean heat, and it has no direct affect on the pressure measurement.

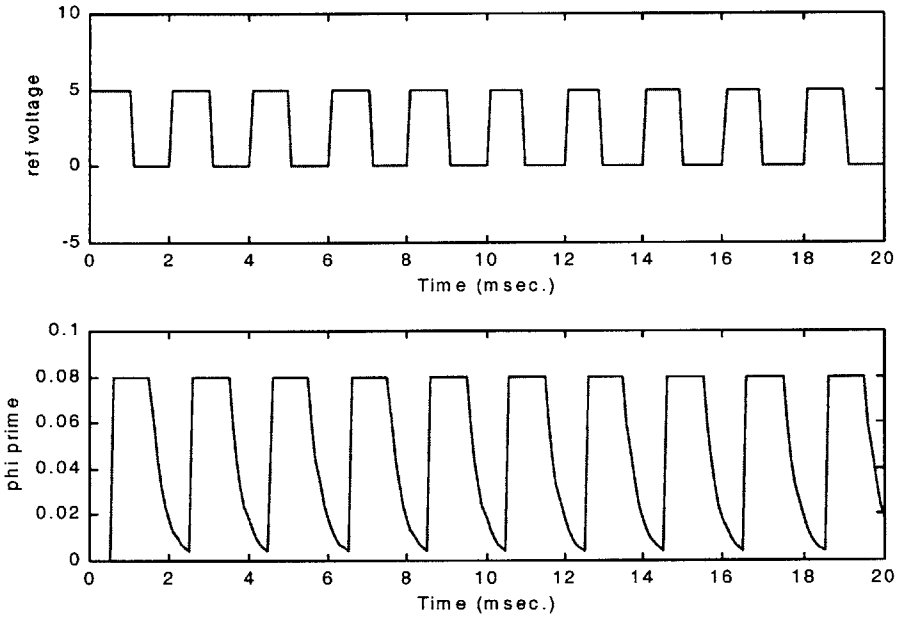



Figure 2-15: Evaluation model fuel injector response.

### 2.1.4 Noise Model

Noise is added to the model to simulate high and low frequency measurement noise. The high frequency noise will have an amplitude on the order of 1% of the maximum pressure measured and the frequency will be greater than the unstable frequency of the evaluation model. The high frequency noise is created by sending a random signal through a high 2nd order Butterworth filter with a corner frequency of 700 Hertz. The low frequency noise which is trying to mimic some process noise will have an amplitude on the order of 3% of the maximum pressure measured and the frequency will be less than the unstable frequency. The low frequency noise is created by sending a random signal through a high 2nd order Butterworth filter with a roll-off frequency of 80 Hertz.

The high and low frequency filters are then combined into one bandstop filter with a transfer function $H$. If $e(t)$ is the white noise input, then

$$v(t) = H(q)e(t),$$

where q is the forward shift operator represented by $qu(t) = u(t+1)$, $H(q)$ is shown in the frequency domain representation of Figure 2-16, and $v(t)$ the linear noise time response is shown in Figure 2-17.

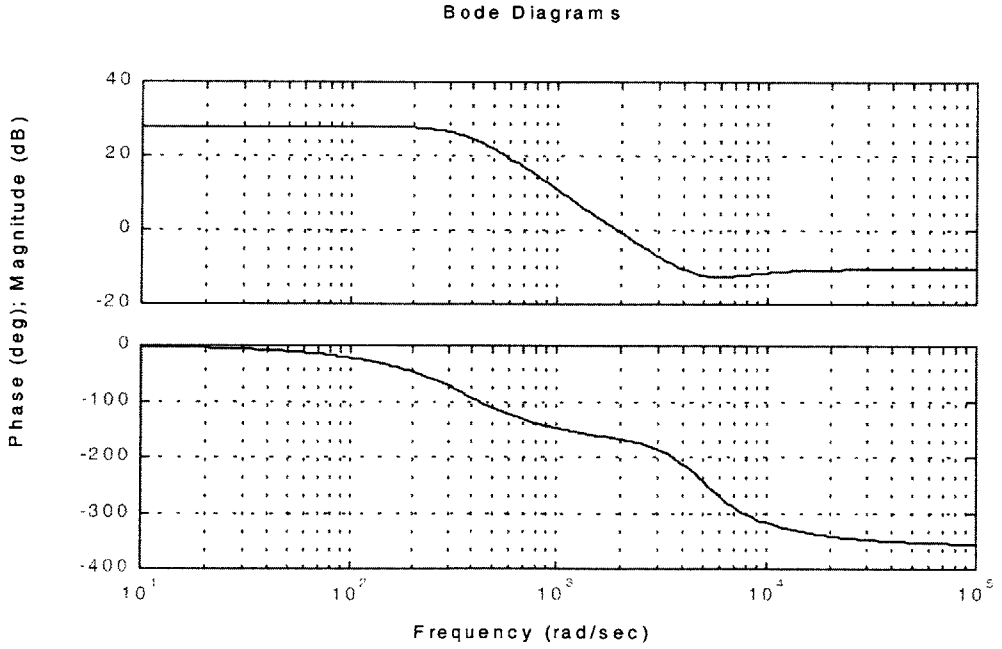



Figure 2-16: Noise model bandstop filter.

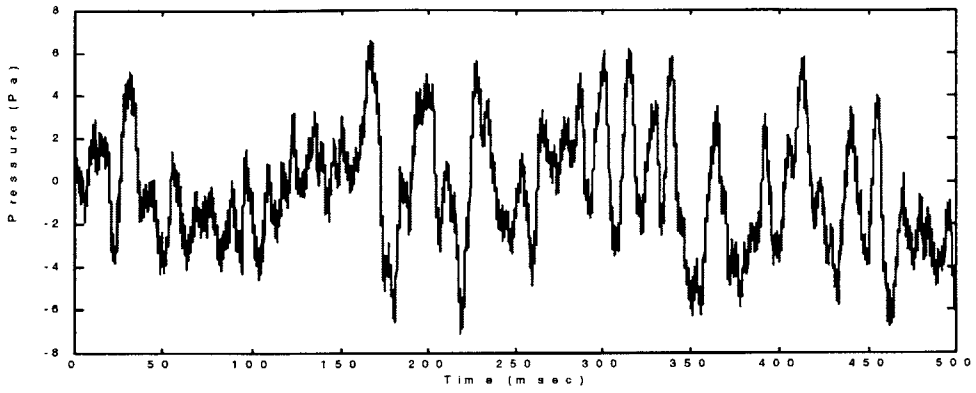



Figure 2-17: Time response of linear disturbance.

A nonlinear absolute value is added to make the noise model more complex. The output of the noise model is

$$v(t) = \left[H(q) * \left|H(q)\right| * c_z\right] e(t), \tag{20}$$

where $\left|\ \right|$ is the absolute value, and $c_z$ is a constant. The time response for equation ( 20 ) is shown in Figure 2-18. Equation ( 20 ) will be used as the representation of noise in the complete evaluation model.

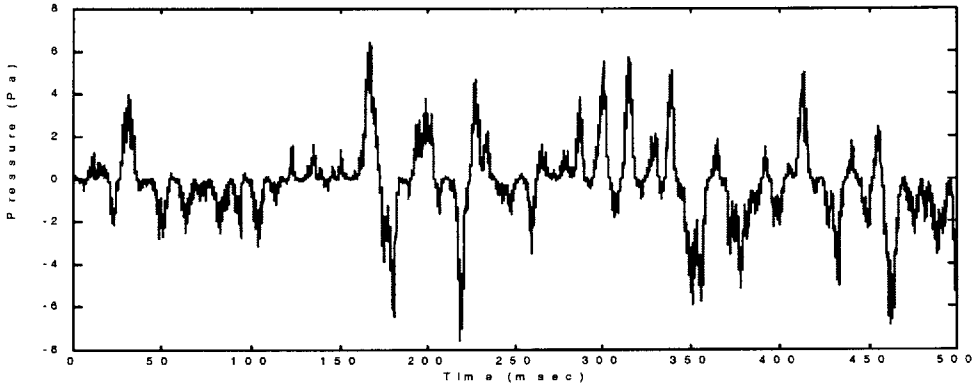



Figure 2-18: Time response of complete nonlinear disturbance.

### 2.1.5 Complete Evaluation Model

The complete combustor evaluation model is created by assembling the nonlinear combustor model of section 2.1.2, the fuel injector model of section 2.1.3.3, and the noise model of 2.1.4. Figure 2-19 displays a block diagram of the complete model, where Cp is a constant gain matrix to compute the pressure, P, in pascals. Appendix C contains the script file with the system parameter values and the simulation block diagram. Reference [ 13 ] verified that the underlying combustor model is a valid representation of an experimental combustor.

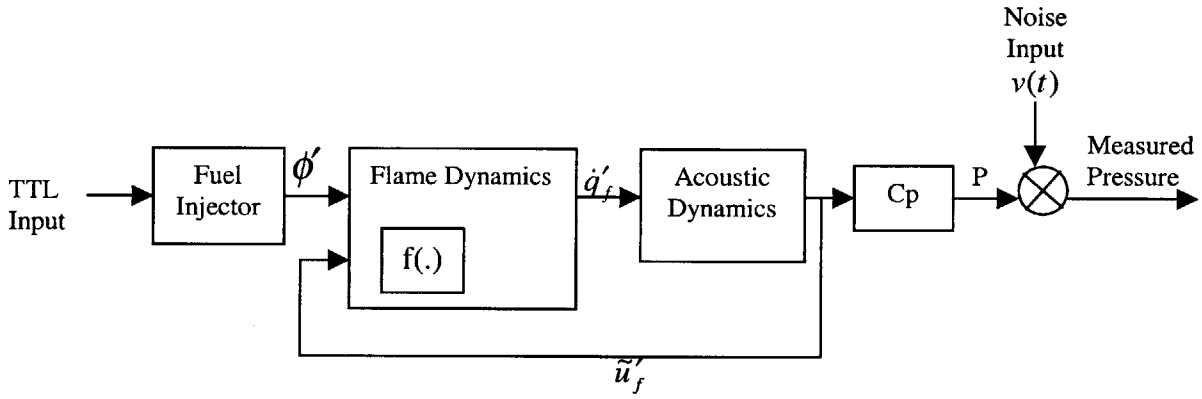



Figure 2-19: Evaluation model block diagram.

By using white Gaussian noise as an input to excite the evaluation model, the power spectrum is revealed and shown in Figure 2-20. The power spectrum shows two pronounced frequencies, one at 1200 rad/s (190 Hz) and the other 3200 rad/s (510 Hz). These peaks correspond closely to the linear model modes at 168 Hz and 538 Hz. The initial condition time response of Figure 2-21 shows the typical linear exponentially growing oscillations followed by the nonlinear limit-cycle behavior. The evaluation model is now ready for the system identification and control studies.

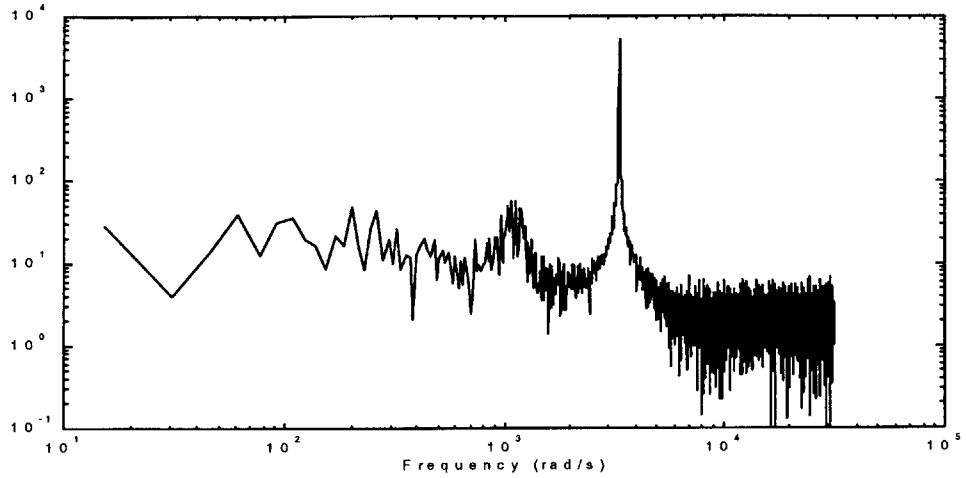



Figure 2-20: Evaluation model power spectrum.

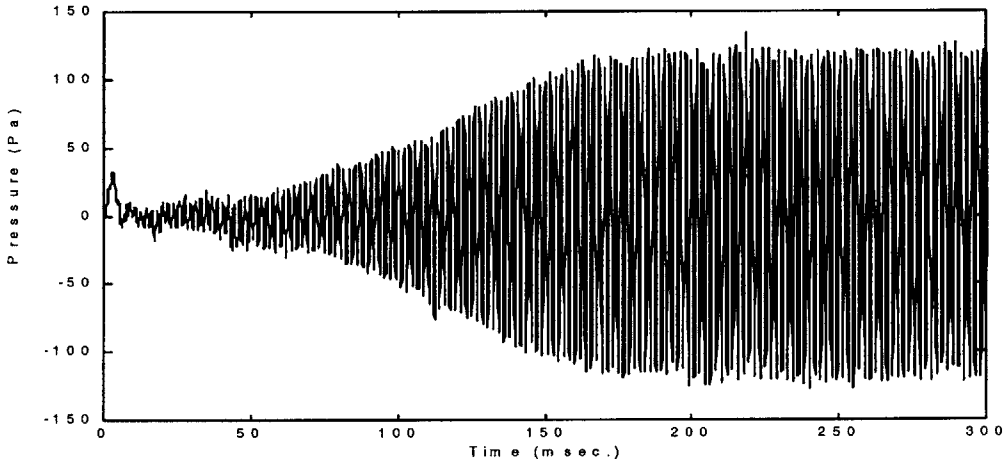



Figure 2-21: Evaluation model time response.

## 2.2 Experiment Design: Identification for open loop

The goal of the open loop experiment is to obtain informative input-output data that displays the relevant system dynamics and to determine whether the system actuator configuration will have the potential for control authority of the instability. In physical systems much effort would be focused here as to the placement and characteristics of sensors and actuators so as to excite and measure the system during the experiment. See Figure 3-7 for the block diagram of the open loop system with the identification input. The complete evaluation model is used to replace the combustor and fuel injector in Figure 3-7 for this experiment.

### 2.2.1 Input Design for Open Loop Experiment

In order to get an informative experiment the input must be rich. The system must be excited and forced to display its dynamic properties. From [ 26 ] an open loop experiment is informative if the input is persistently exciting. If the input signal $u(t)$ has a spectrum $\Phi_u(\omega)$, then for $u$ to be persistently exciting it must have nonzero $\Phi_u(\omega)$ at n points, where n is the number of parameters to be estimated. Therefore, the optimal input design will depend on the system. Since most often the system order is unknown apriori, then it is wise to input many frequencies to be able to validate against higher order system identification models. The interesting frequencies to examine can be found by finding the evaluation model spectra. This was accomplished in 2.1.5 and is shown in Figure 2-20.

The power spectrum shows two pronounced frequencies, one at 1200 rad/s and the other 3200 rad/s. The input should be designed to excite the evaluation model around these frequencies, and attenuate other frequencies. This is accomplished by sending Gaussian noise through a $6^{th}$ order bandpass Butterworth filter with a lower corner point of 500 rad/s and upper cut-off frequency of 9000 rad/s. The break frequencies allow for margin on both ends of the frequency band. The resulting signal versus time is plotted in Figure 2-22 with its power spectrum given in Figure 2-23.

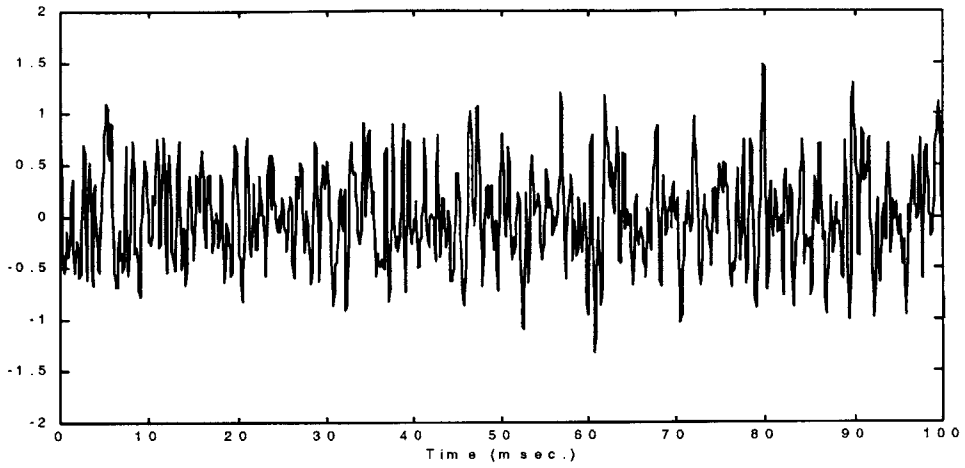



Figure 2-22: Gaussian noise through bandpass filter.

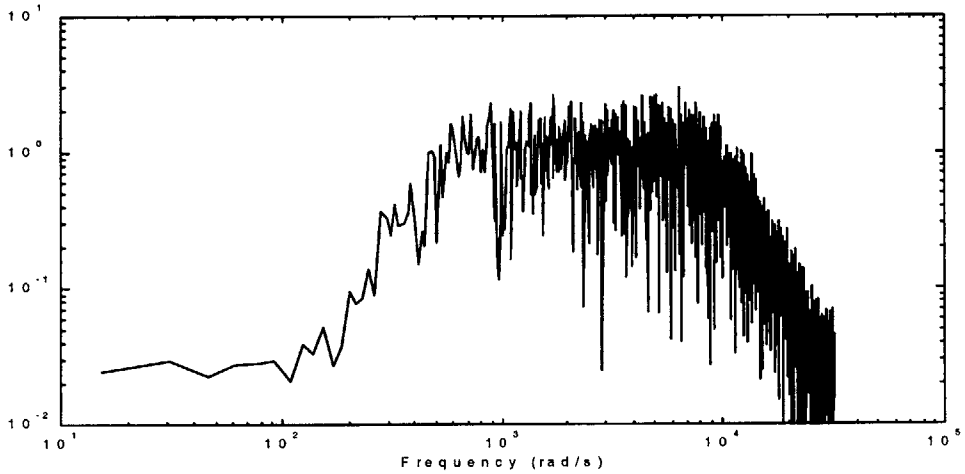



Figure 2-23: Spectra of the Gaussian signal through bandpass filter.

Because the fuel injector used for this model is a pulsed fuel injector the input signal to the fuel injector must be binary in character. This is created by setting a small positive threshold, setting the output to 5 if the input is greater than the threshold, and setting the output to 0 if it is less than the threshold. A time plot of the binary signal is shown in Figure 2-24. The spectra of the binary input signal is given in Figure 2-25. In using the binary signal control over the shape of the input spectrum is lost, but there is still some attenuation for the frequencies we are not interested in identifying.

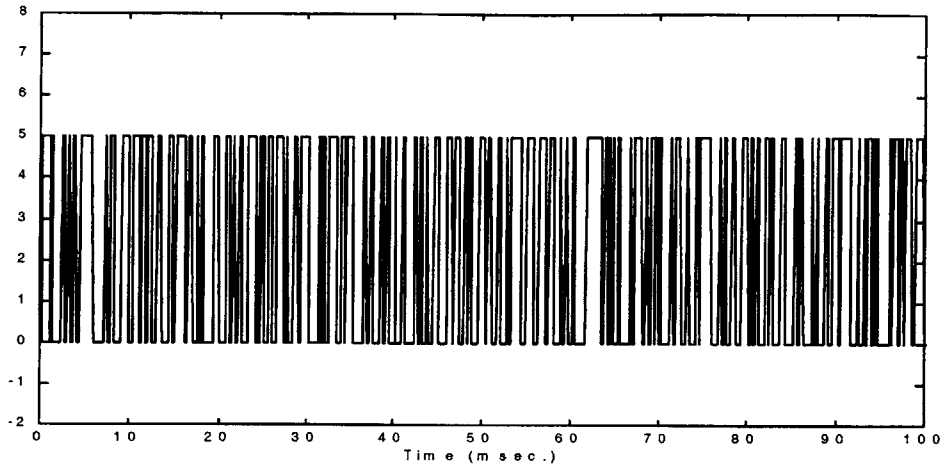



Figure 2-24: Random binary noise input signal.

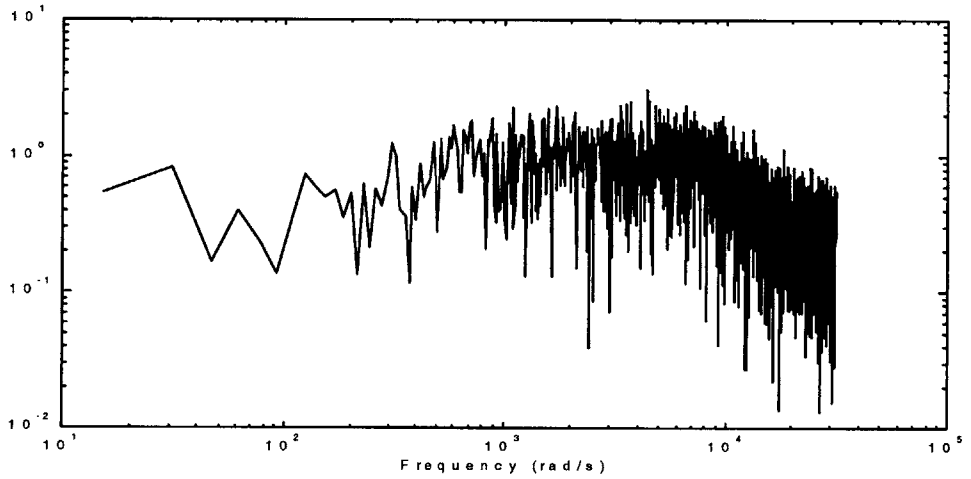



Figure 2-25: Spectra of binary input signal.

### 2.2.2 Open Loop Data Collection

The resulting evaluation model response to the system identification of section 2.2.1 is shown in Figure 2-26. By comparing the response of the model to the shaped system identification input, Figure 2-26, to the response of the model to the Gaussian input, Figure 2-27, we see that the shaped input excites the system more. The excitation is displayed as large deltas from the steady limit-cycle operating point. Figure 2-28 shows the power spectrum of the evaluation model response to the system identification input. This data will be used to create a system identification model for control purposes.

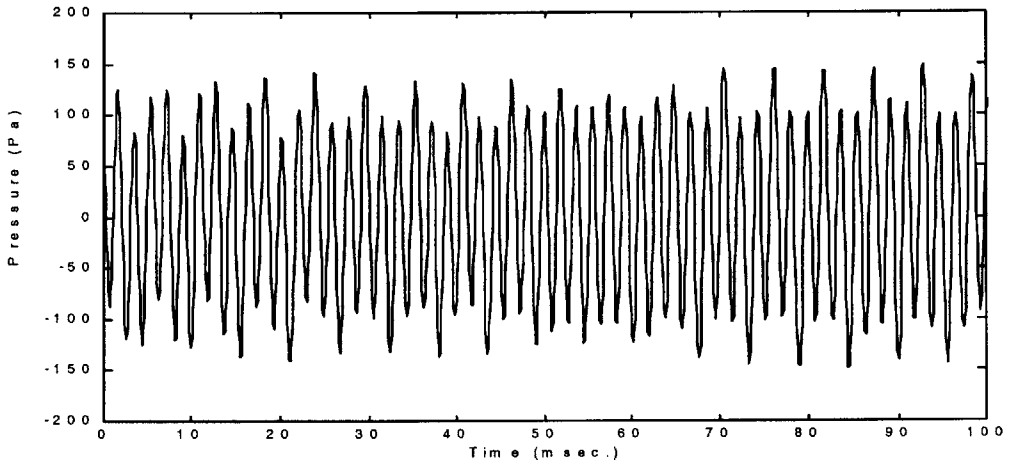



Figure 2-26: Evaluation model open loop time response to system id input.

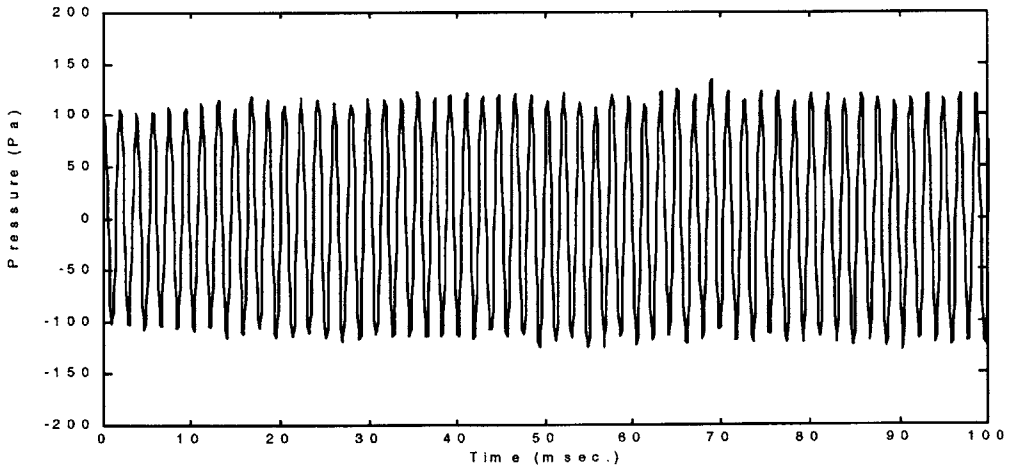



Figure 2-27: Evaluation model open loop time response to Gaussian input.

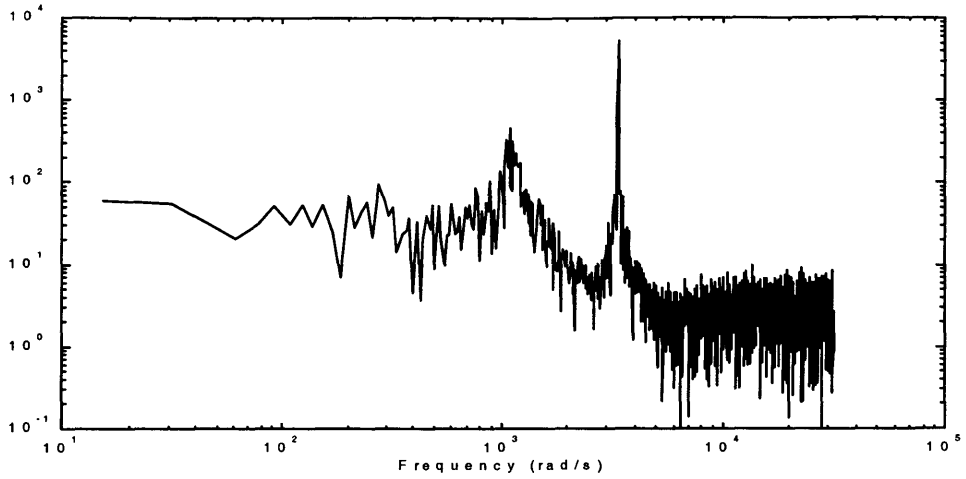



Figure 2-28: Evaluation model open loop power spectrum.

## 2.3 System Identification and Validation: Open loop

The method of finding a suitable system identification model is iterative. A simple approach is to create a number of different model structures using different fit criterion and then compare the resulting models to each other and to the observed data. Multiple model structures with different estimation methods are created and compared to each other by looking at the estimated models frequency response, residuals, time response, and pole-zero locations. Models that do well with this stage of validation analysis will then be used as the model for control design. The final validation is the ability of the system identification model based control to attenuate the combustor instability pressure oscillations.

### 2.3.1 Open loop identification of $4^{th}$ order sub-space model

The data from the open loop system identification is split into estimation and validation data. With the estimation data collected, the open loop system identification model can be created. Similar to section 3.4.1, the system identification model will be a time-invariant linear system approximation. It is also assumed the system is causal and that all disturbances can be lumped into a single term. The model set and the criterion to fit the data are chosen by trial and error to determine a system identification model that allows a successful stabilizing control design. See 3.4.1 for model structures and predictor definition. A $4^{th}$ order sub-space model is selected as the best fit.

### 2.3.2 Open loop model validation

During model validation the quality of the model is evaluated. As stated earlier, the final validation will be a model that allows a successful control design. Using residual analysis, time response plots, power spectrum, bode plot, and pole-zero locations the quality of the model is evaluated.

The model power spectrum , Bode plot, pole-zero plot, and eigenvalues of the $4^{th}$ order sub-space model are shown in Figure 2-29 through Figure 2-31, and Table 2. The power spectrum

of the identification model and evaluation model have similar profiles at both of the acoustic mode frequencies. This shows that the system identification tools are able to capture the frequencies and magnitudes of the acoustic modes correctly. The Bode plot shows the phase of the 4$^{th}$ order model, and the pole-zero plot shows that the model is marginally stable. It is expected that the open loop identification will not identify the poles as unstable because the data came from the nonlinear or limit-cycle region of the response. The eigenvalue frequencies of 176 Hz and 536 Hz correspond very closely to the linear model eigenvalues of 168 Hz and 538 Hz.

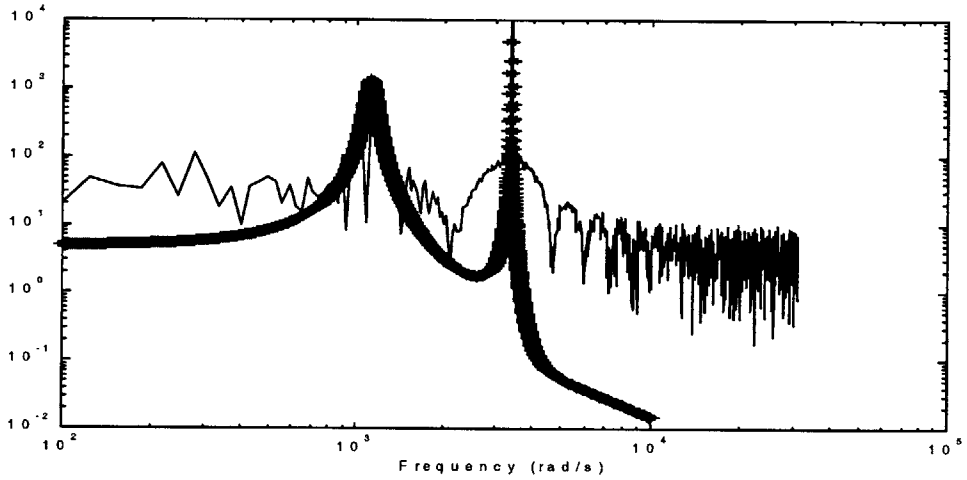



Figure 2-29: Power spectrum plot. Thin-line estimation data; thick-line 4$^{th}$ order sub-space. model.

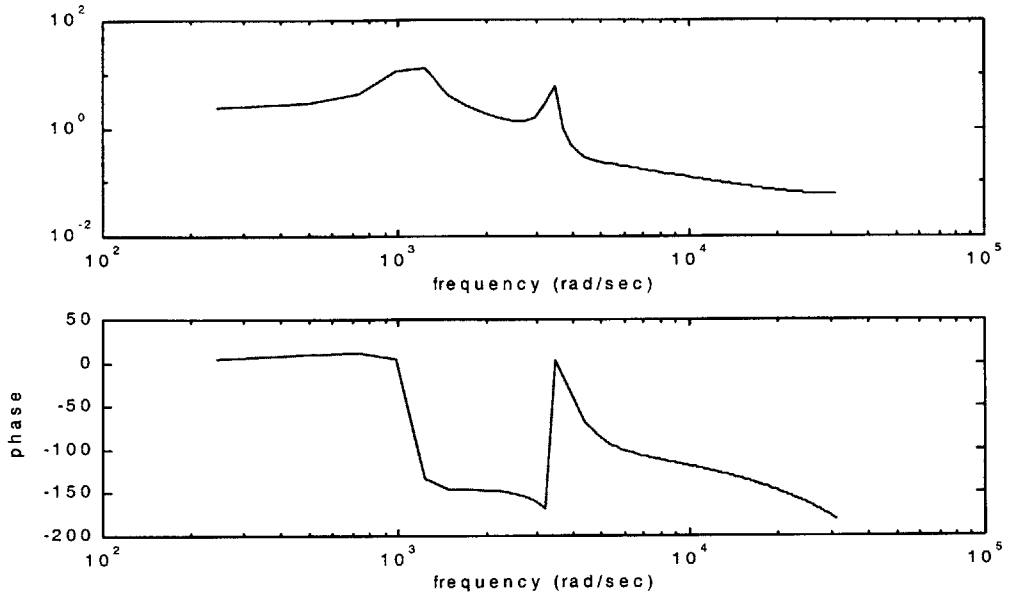



Figure 2-30: Bode plot of 4th order state space system id model.

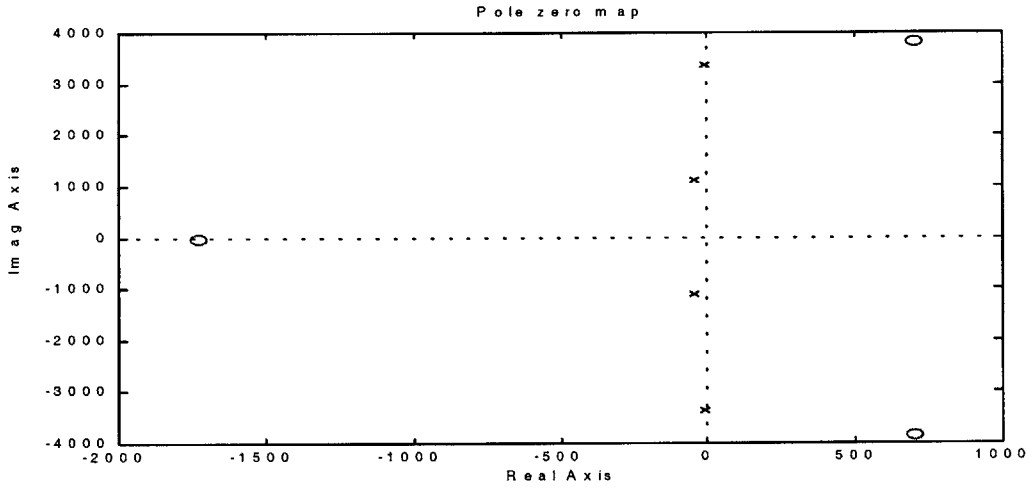



Figure 2-31: Pole-zero plot of 4th order state space system id model. Poles are shown as x's and zeros are shown as 0's.

| Eigenvalues | Damping | Freq. (Hz) |
|---|---|---|
| -4.16e+001 +/- 1.11e+003i | 3.73e-002 | 176.6 |
| -2.16e+000 +/- 3.37e+003i | 6.4e-004 | 536.4 |

Table 2: 4th order sub-space model eigenvalues.

The 4th order sub-space model transfer function is given as

$$(21) \qquad \frac{Y(s)}{U(s)} = \frac{1.1786*10^3 S^3 + 4.1370*10^5 S^2 + 1.5078*10^{10} S + 3.1087*10^{13}}{S^4 + 87.433 S^3 + 1.2626*10^7 S^2 + 9.5137*10^8 S + 1.4150*10^{13}}.$$

In order to ensure a good test of the model, its response will be evaluated on a fresh data set or the validation data. Such a procedure is known as *cross-validation*. The residual time plot for the 4th order state space model is shown in Figure 2-32. It is also informative to look at the basic statistics of the residuals as defined in section 3.4.2 and presented in Table 3. The R value cannot be computed for this model because the residual is larger than the pressure. This is a good indication that the phase of the system identification model does not match the evaluation model. Other system identification models examined were unable to perform better at this test. This indicates that some pre-filtering of the data, a different experimental setup, or a different system identification input is required. Work in later sections changes both the identification inputs and the pre-processing of the data to create a better system identification model.

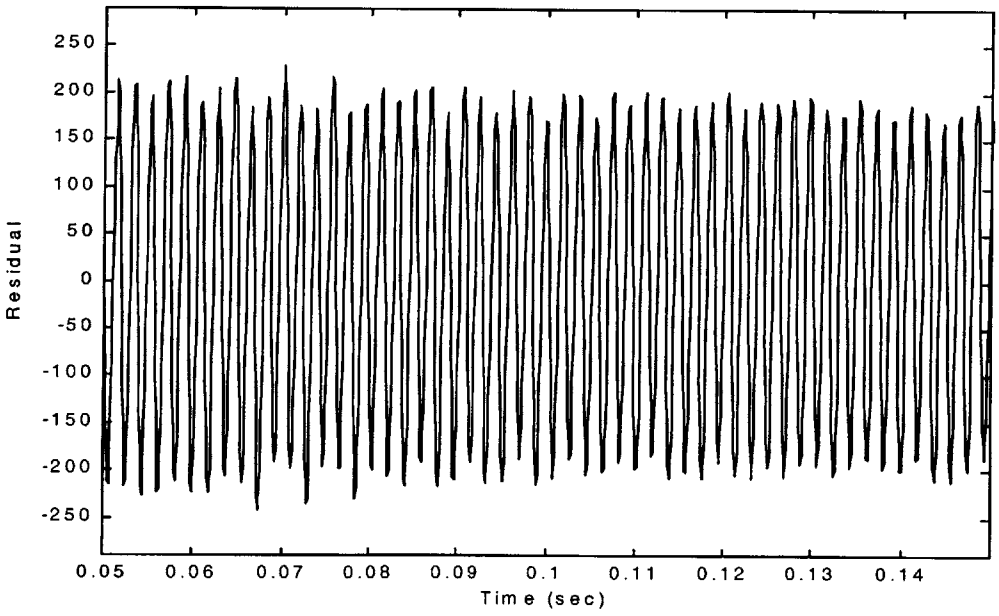



Figure 2-32: Residual plot of 4th order sub-space model versus validation data.

| $S_1$ | $S_2$ | $J_k$ | $J_y$ | $R$ |
|---|---|---|---|---|
| 127.6 | 241.5 | 20210 | 7137.3 | NAN |

Table 3: 4th order sub-space model validation statistics.

All of the statistics shown thus far assume that the residuals do not depend on something that is likely to change. Since the control input will be different in frequency content it would be good to see that the residuals do not depend heavily on the particular input used. The cross-correlation from [ 26 ] using the covariance between residuals and past inputs is:

$$\hat{R}_{eu}(\tau) = \frac{1}{N}\sum_{t=1}^{N}\varepsilon(t)u(t-\tau). \quad (22)$$

Similarly, the auto-correlation among the residuals is:

$$\hat{R}_{\varepsilon}(\tau) = \frac{1}{N}\sum_{t=1}^{N}\varepsilon(t)\varepsilon(t-\tau). \quad (23)$$

The plots of equations ( 22 ) and ( 23 ) for the 4th order sub-space model are shown in Figure 2-33.

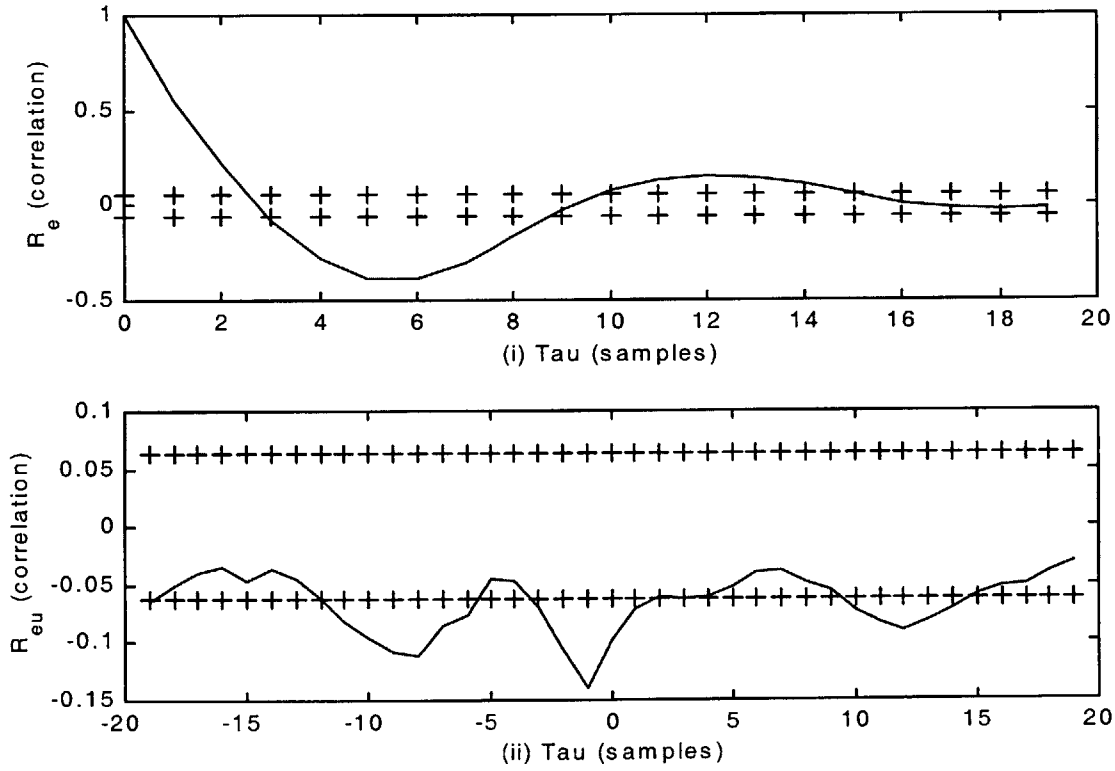


Figure 2-33: Correlation function of residuals: (i) autocorrelation function of residuals using covariance, (ii) cross-correlation function between input and residuals using covariance.

Figure 2-33 (i) shows the whiteness of the residuals. Since much of the correlation function is outside of the confidence region, shown as '+', the model has not captured all of the significant dynamics. The independence between the input and the residual is shown in Figure 2-33 (ii). Those values outside of the 99% confidence intervals have a significant correlation. The correlation for the negative lag of -1 is an indication of feedback in the system. This may be due to the thermoacoustic feedback loop in the evaluation model. Other models were unable to perform better in the residual correlation function analysis. This model will be used to develop an LQG controller in section 2.4.2.

## 2.4 Control Development

Two types of controllers are developed for the evaluation model. The first is an adjustable lead-lag control. This controller allows the amount of phase lead to vary from 0 to 360 degrees. The best phase is determined experimentally. The lead-lag is used in case the LQG based on the open loop nonlinear region does not provide a stabilizing effect. The model-based LQG control is the same as that developed in section 3.5. The goal is to reduce the combustor oscillations in the shortest amount of time and with the smallest steady state oscillations.

### 2.4.1 Adjustable Lead-Lag Control

An easily tunable, discrete, multiple lead-lag control is created to try to stabilize the pressure oscillations. The lead-lag is adjustable from 0 to 360 degrees of phase lead. It uses the Tustin-transformation and 6 lead-lags in series. Six lead-lags are used because each lead-lag is only capable of effectively creating 60 degrees of phase lead on its own. Figure 2-34 shows the discrete Tustin lead-lag block diagram where $T$ is the update time, $\tau_1$ and $\tau_2$ are the lead lag time constants, and $Z^{-1}$ is the past value.

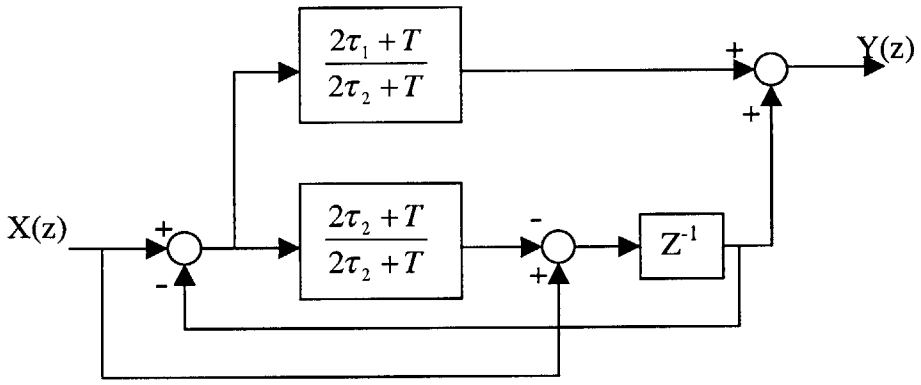



Figure 2-34: Tustin lead-lag block diagram.

Each lead-lag accounts for 1/6 of the total desired phase lead. In order to make the controller unity gain at the unstable frequency for different phase leads, a gain factor that is the inverse of the relationship shown in Figure 2-35 is added.

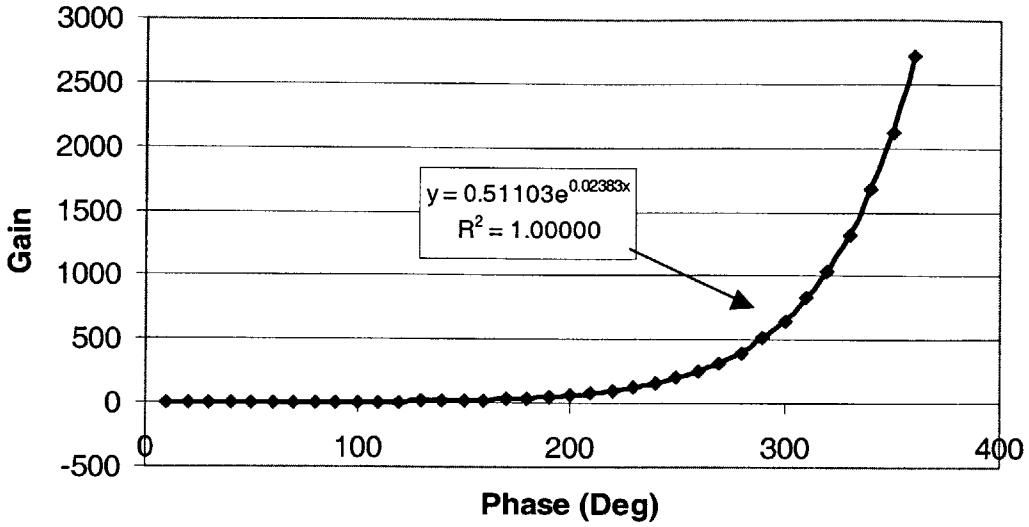



Figure 2-35: 6 Lead-lag gain versus phase relationship.

Shown in Figure 2-36 are the lead-lag controllers for 90, 135, 180, and 270 degree phase lead. Note the controller has a transient during initialization that last less than one period due to the initial conditions response of the downstream lead-lags. This could be eliminated by calculating the correct initial conditions for the downstream lead-lags. The MATLAB® script file to create this controller algorithm is shown in Appendix F: Lead-lag control algorithm.

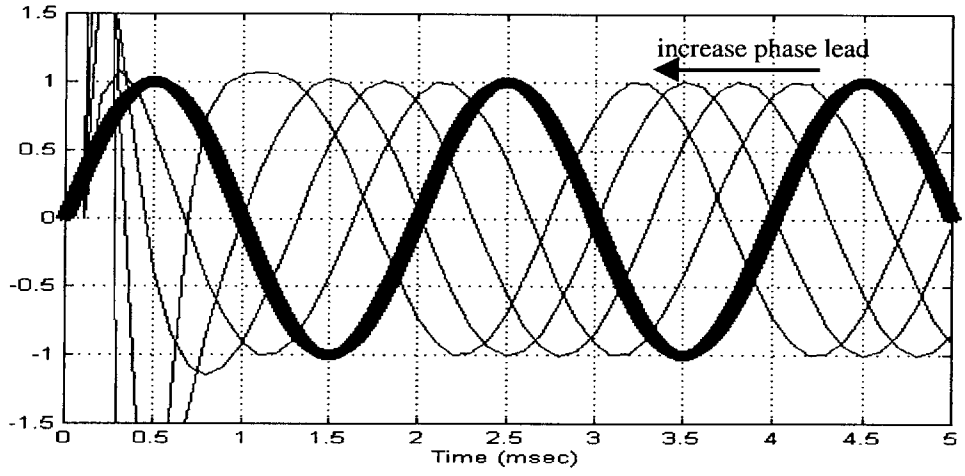



Figure 2-36: 6 Lead-lag control output versus input signal. Thick-line is the input signal to the lead-lag. Thin-lines are lead-lag controller set to 90, 135, 180, and 270 degrees of phase lead.

Using the 6 lead-lag control, a study is run to determine if the controller will have a stabilizing effect on the evaluation model. The pressure response of the evaluation model to the lead-lag

control as the phase is varied from 0 to 360 degrees at 30 degree increments is shown in Figure 2-37. The study indicates that the 60 degree phase lead controller has a stabilizing effect and the quickest transient time. The pressure response of the evaluation model to this control, the control effort, and the binary input to the fuel injector are shown in Figure 2-38. The control is turned on at 0.2 seconds into the time plot. In the steady state the evaluation model exhibits a sustained 110 Hz oscillation. This may be due to exciting the lower frequency acoustic mode. The transient summary of the lead-lag control is shown in Table 4. The transient time is measured from the onset of the control action until the pressure oscillations stay below 30% of the maximum uncontrolled oscillation.

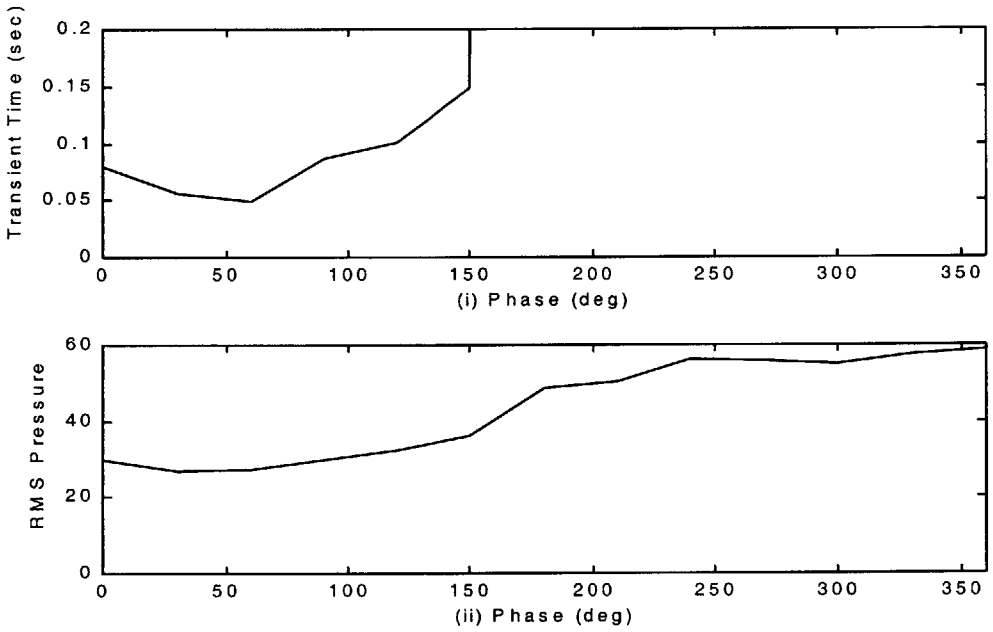



Figure 2-37: Study of lead-lags from 0 – 360 degrees of lead: (i) transient time versus phase (any phase with a transient value greater than 0.2 seconds did not stabilize), (ii) RMS pressure versus phase.

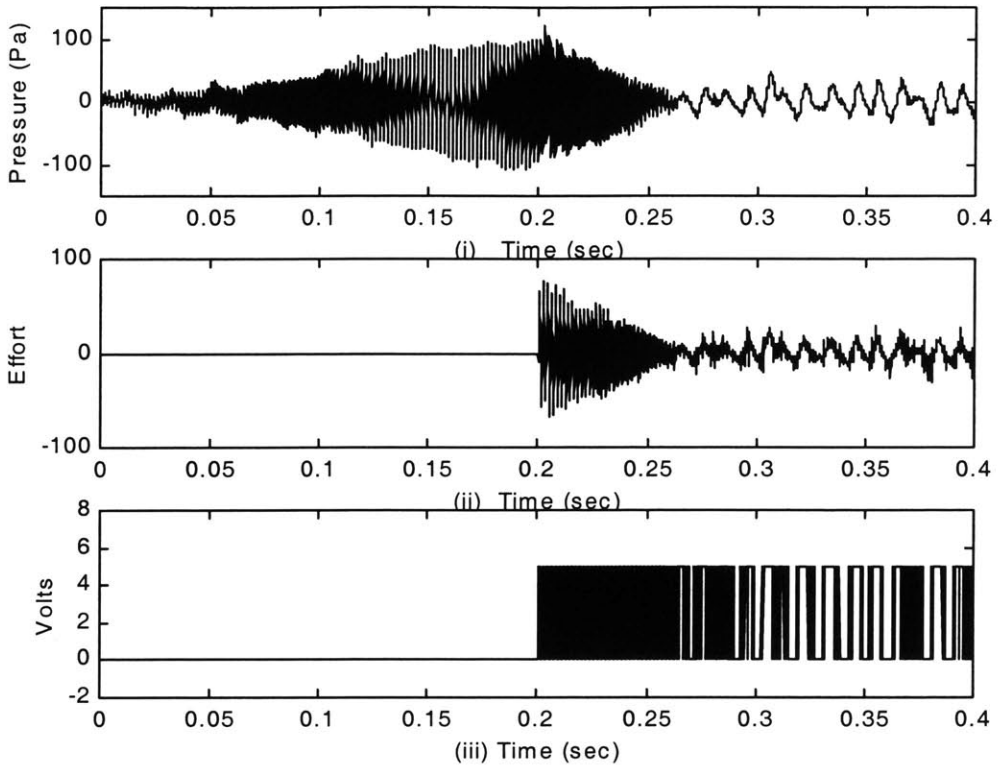



Figure 2-38: 60 degree lead-lag performance: (i) pressure output versus time, (ii) lead-lag output versus time, (iii) binary voltage into the pulse injector versus time.

| Controller | Transient Time | RMS Pressure | Max Pressure |
|---|---|---|---|
| Lead-Lag with 60 degrees of lead | 0.049 | 13.58 | 47.3 |

Table 4: Summary of evaluation model transient performance with 60 degree phase lead-lag control.

### 2.4.2 LQG with Threshold based on open loop 4th order sub-space model

Using the same LQG control technique as shown in section 3.5 and depicted in block diagram form in Figure 3-13, a new LQG control is developed based on the 4th order sub-space model. The script file to develop the control and the simulation block diagram are given in Appendix D. Several different values of the control parameters $\rho$ and $\mu$ were tried to get the best transient and steady state response. The final values of the control parameters are given in Table 5. The time response of the evaluation model, the control effort, and the binary input to the injectors is shown in Figure 2-39. While the control is able to stabilize the combustor oscillations, the transient performance is 3 times longer than the experimentally determined lead-lag control of the previous section. This control performs poorly because the phase of the 4th order sub-space model is incorrect as stated in section 2.3.2. However, since it is stabilizing it will be used for closed-loop system identification below.

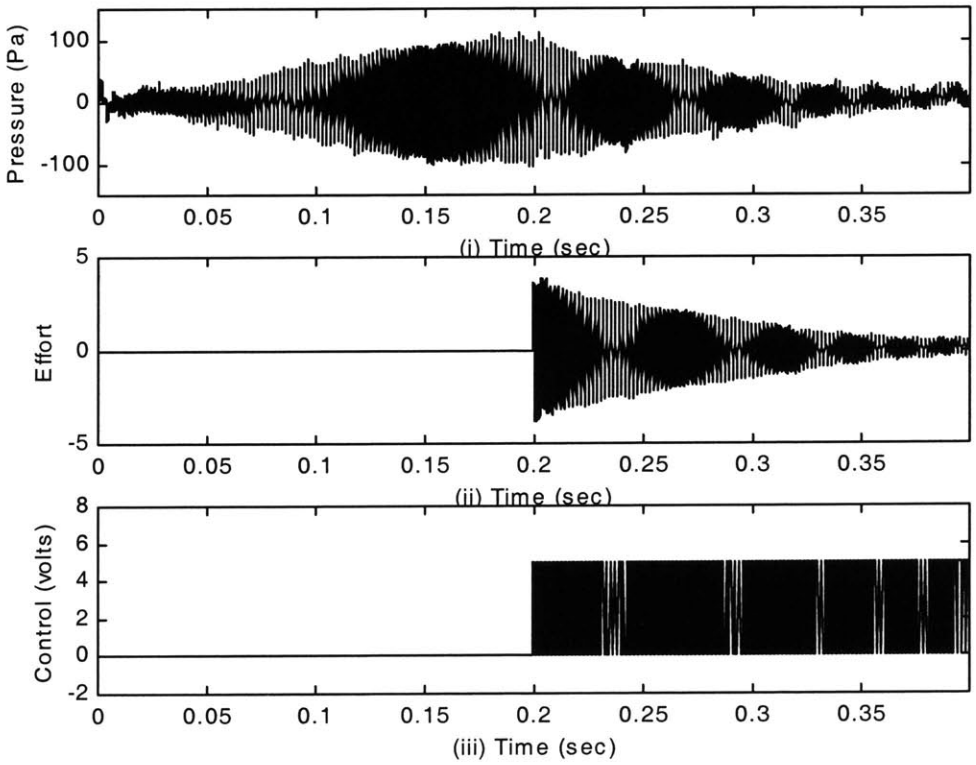



Figure 2-39: Evaluation model response to LQG control based on 4th order sub-space. open loop model: (i) combustor pressure response, (ii) LQG control effort, (iii) effort after threshold.

| Controller | $\mu$ / $\rho$ | Transient Time | RMS Pressure | Max Pressure |
|---|---|---|---|---|
| LQG based on open loop 4th order S.S model. | 5e-1 / 5e2 | 0.148 | 8.57 | 33.12 |

Table 5: Summary of evaluation model transient performance to LQG control based on 4th order sub-space. model.

## 2.5 System Identification, Validation, and LQG Control: Closed loop

In order to get more informative data, a study is accomplished on the system identification input to determine the most effective magnitude. Using U(t) and Y(t) of Figure 3-15 to do direct system identification, several model structures, model orders, and estimation methods are examined to find a good fit to the data. Many of the models place dynamic elements at frequencies higher and lower than the range of interest. In an effort to have the estimation methods focus more on the frequency range of interest, between 100 and 1000 Hz, the pressure data Y(t) is bandpass filtered before being used for identification. Finally, two different closed loop system identification models are created and used to generate LQG controllers.

### 2.5.1 Input Design for Closed Loop Experiment

Since it was difficult from the open loop data to obtain a good system identification model, some effort was focused on the design of the closed loop system identification input. Since the system identification input is added to the control effort before the threshold operation, as shown in Figure 3-15, the magnitude of the identification input relative to the control effort is important. If the identification is large compared to the control then there will be less control authority. If the identification is small compared to the control then the identification has little effect on the combustor performance. Since the idea is to have an informative experiment by exciting the combustor, but still capture the growth associated with the unstable poles of a stabilized system, there must be an optimal ratio of identification input to control. By changing the magnitude of the identification input to be multiples from 1 to 12 of the steady state control effort, storing data, and creating the same system identification model the goodness of fit versus the input magnitude can be plotted. Figure 2-40 shows that by using an identification input that is 8 times larger than the steady state stabilized control effort produces the best system identification model. By using an input of this level the evaluation model is excited enough to be between the controlled operating point and the limit-cycle operating point. This allows the system identification tools to capture the dynamics correctly. The system identification input that is 8 times larger than the control effort will be used to obtain a system identification model from the closed loop evaluation model.

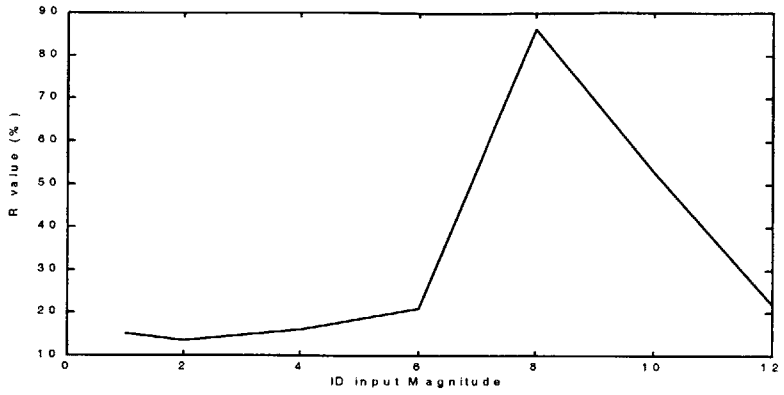



Figure 2-40: System identification input study results. The ordinate displays the $6^{th}$ order prediction-error model pure simulation R value. The abscissa shows the ratio of identification input magnitude to stable control effort magnitude.

### 2.5.2 Closed Loop Data Collection

The closed loop evaluation model response to the system identification input is shown in Figure 2-41. The LQG control based on the open loop 4$^{th}$ order sub-space model is used to initially stabilize the evaluation model. The system identification input is inserted into the loop as shown in Figure 3-15. The identification input is started at 0.1 seconds and the evaluation model responds by increasing the amplitude of the oscillations. The amplitude of the oscillations do not grow to the limit-cycle level. This data is used to create two system identification models: a 6$^{th}$ order prediction-error model, and a 6$^{th}$ order sub-space model.

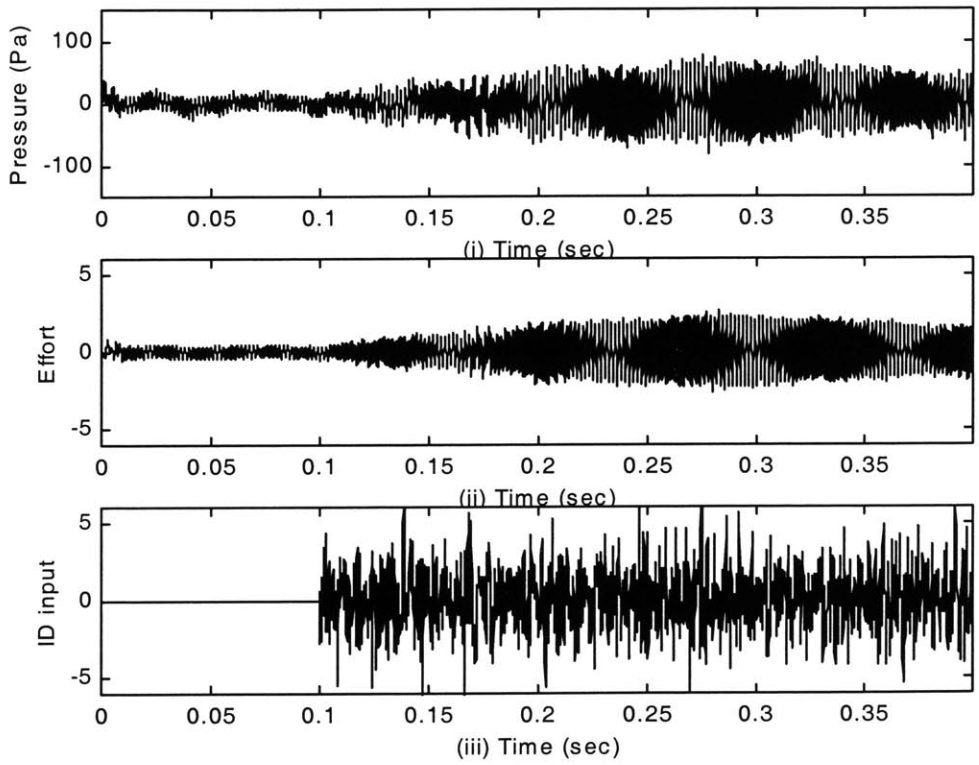



Figure 2-41: Closed loop system identification: (i) evaluation model pressure response, (ii) 4$^{th}$ order open loop LQG control effort, (iii) system identification input.

### 2.5.3 Closed Loop Identification of 6th order prediction-error model

The data from section 2.5.2 is first pre-processed before being used for system identification. It is bandpass filtered between 100 Hz and 1000 Hz to focus the tools on the important frequencies. One of the resulting models is a 6th order prediction-error model. The power spectrum of the system identification model versus the filtered closed loop evaluation model response shows acoustic frequencies around 162 Hz and 536 Hz are correctly identified, as seen in Figure 2-42 and Table 6. Even though all of the poles of the model are stable, see Figure 2-43, it will be used to create a controller because it is able to simulate the response well.

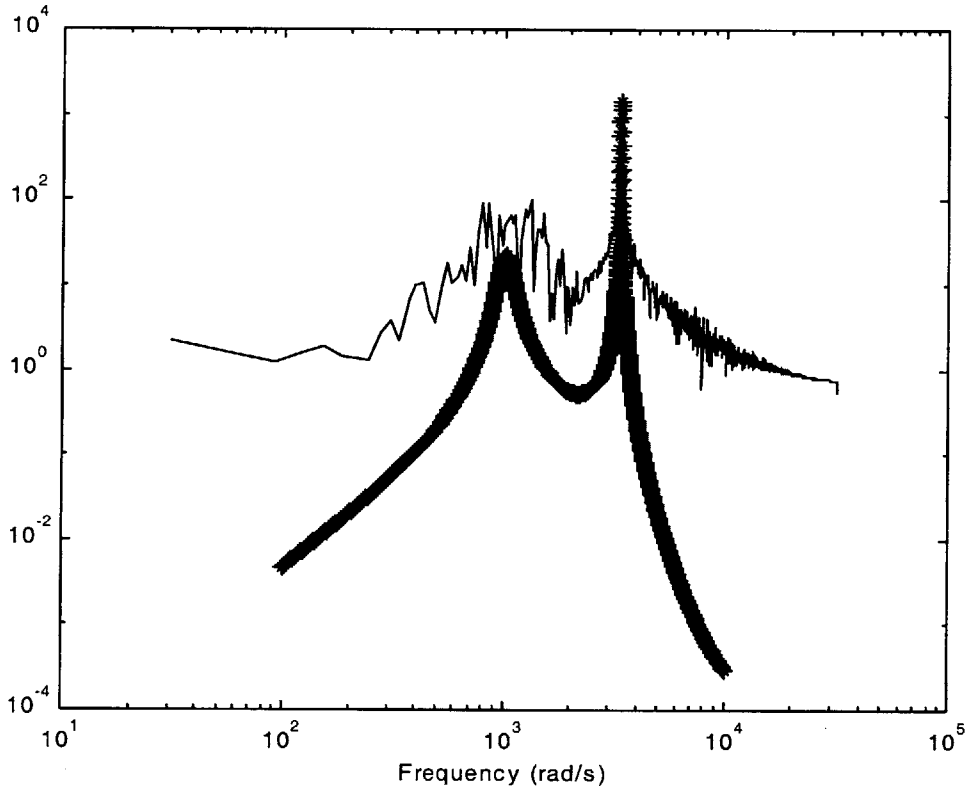



Figure 2-42: Power spectrum plot. Thin-line filtered estimation data; thick-line 6th order prediction-error model.

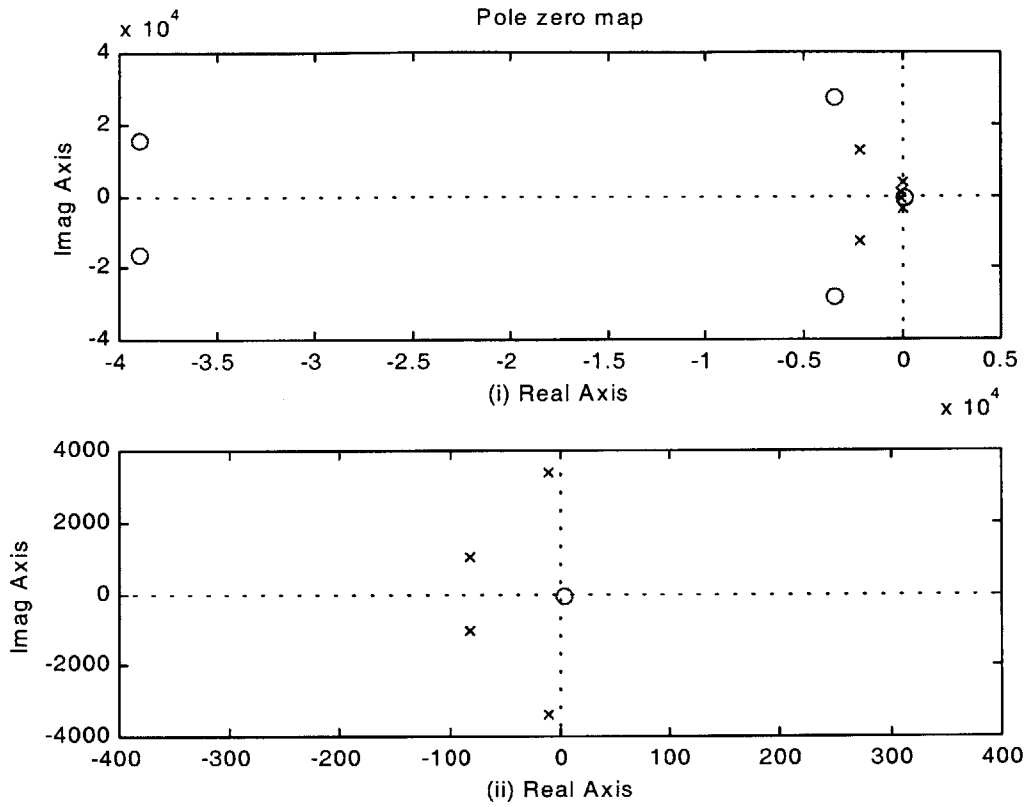



Figure 2-43: 6th order prediction-error model poles and zeros. Poles are shown as x's and zeros are shown as 0's.

| Eigenvalues | Damping | Freq. (Hz) |
|---|---|---|
| -8.1912e+001 +/- 1.0144e+003i | 8.05e-002 | 162.3 |
| -1.0858e+001 +/- 3.3695e+003i | 3.22e-003 | 536.4 |
| -2.2042e+003 +/- 1.2445e+004i | 1.74e-001 | 2005 |

Table 6: 6th order prediction-error model eigenvalues.

The transfer function for the 6th order prediction-error model is

( 24 )

$$\frac{Y(s)}{U(s)} = \frac{0.91494*S^5 + 7.7895*10^4 S^4 + 2.8447*10^9 S^3 + 6.7355*10^{13} S^2 + 1.271*10^{18} S - 4.6147*10^{18}}{S^6 + 4.5939*10^3 S^5 + 1.7294*10^8 S^4 + 8.6152*10^{10} S^3 + 1.9996*10^{15} S^2 + 3.5253*10^{17} S + 1.784*10^{21}}.$$

### 2.5.4 Closed Loop Validation of 6th order prediction-error model

The quality of the 6th order prediction-error model is validated by looking at the residuals and the correlation functions defined in equations ( 22 ) and ( 23 ). Figure 2-44 (i) shows the whiteness of the residuals using the auto-correlation function. By having some values of the autocorrelation (excluding the lag at zero which is by definition equal to 1) outside of the confidence intervals for different lags indicates either some of the dynamics of the thermoacoustics have not been captured or the noise is not modeled correctly. The independence between the input and the residual, or cross-correlation is shown in Figure 2-44 (ii). Those values outside of the 99% confidence intervals have a significant correlation. The correlation for the negative lag of -1 is an indication of feedback in the system. To say it differently, there is a correlation between the past output and the current input. The sharp peak at a lag of 10 indicates either too few zeros or incorrectly modeled delays. While there are some excursions from the confidence region in the residual correlation analysis, much of the model response is within the confidence region. This shows that some of the important dynamic elements of the evaluation model are captured correctly and that the model should respond correctly to inputs that are different from the system identification input.

The time response of the evaluation model validation data and the simulated response of the 6th order prediction-error model is shown in Figure 2-45. The model fits 86.1% of the data, as shown in Table 7. This is a good fit to the data and shows that the some of the modeled phase and magnitude may be correct. These are positive signs that a competent controller will be produced.

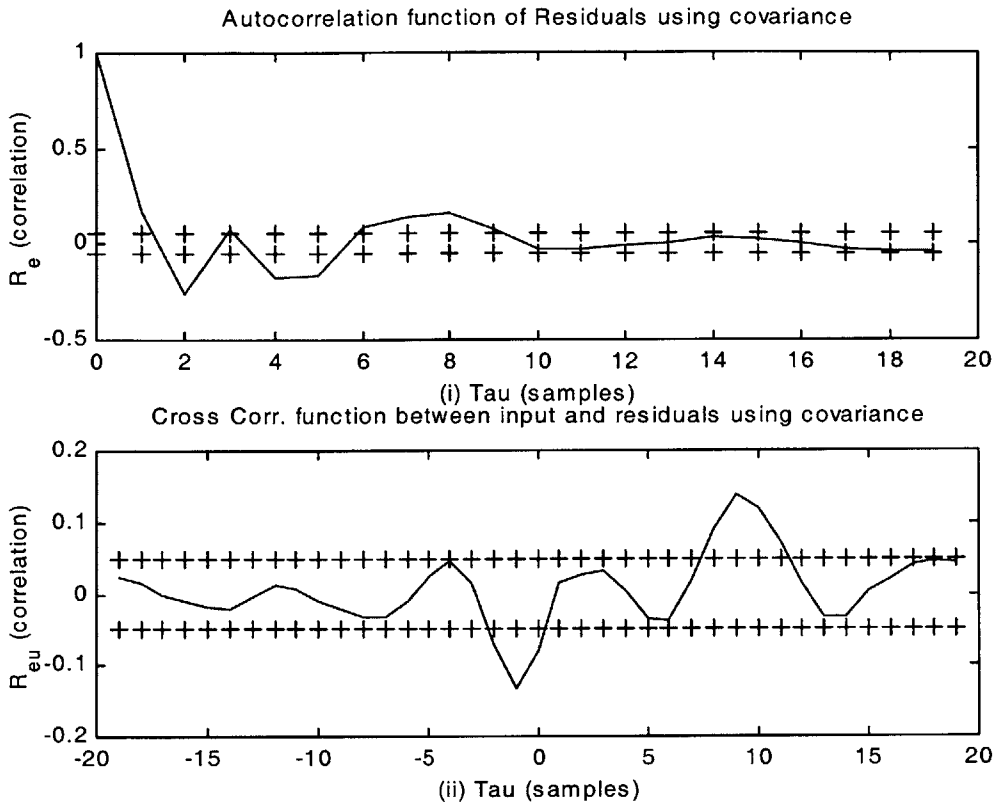



Figure 2-44: Residual analysis of 6th order prediction-error model: (i) autocorrelation of residuals, (ii) cross-correlation of residuals.

| $S_1$ | $S_2$ | $J_k$ | $J_y$ | $R$ |
|---|---|---|---|---|
| 14.8 | 36.3 | 288.9 | 1121.56 | 86.1% |

Table 7: 6th order closed loop prediction-error model validation statistics

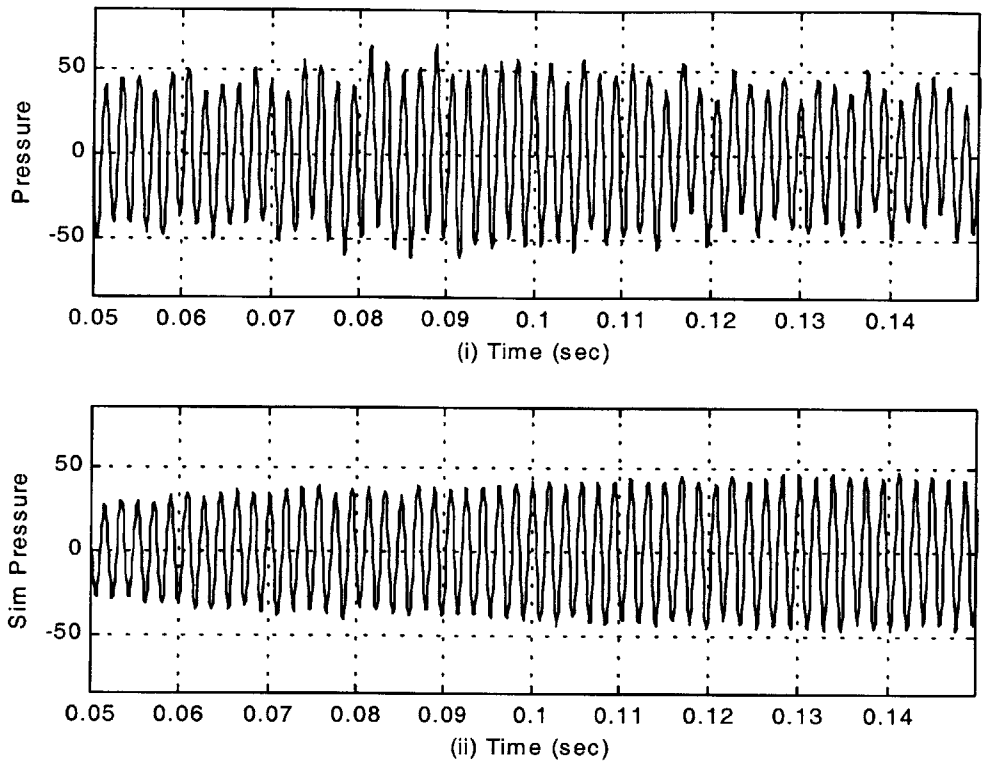



Figure 2-45: Time response data: (i) evaluation model pressure response, (ii) 6$^{th}$ order prediction-error model simulation.

### 2.5.5 LQG control based on closed loop 6$^{th}$ order prediction-error model

Using the model developed in section 2.5.3 and validated in section 2.5.4 an LQG control is created. The script file to develop the control and the simulation block diagram are given in Appendix D. When the controller is applied to the complete evaluation model it is able to stabilize the pressure oscillations in .058 seconds, as shown in Figure 2-46 and Table 8. While the transient time is 18% longer than the performance of the lead-lag controller of section 2.4.1 the RMS pressure post transient is 35% less and the maximum pressure is 32% less. The non-model based lead-lag controller also exhibited a prevalent lower frequency oscillation in the steady state that does not occur under this controller. This is due to the controller design being based on a nearly correct model of the lower frequency acoustic modes. Therefore, it is able to compensate for those dynamics correctly. This feature of being able to accommodate other dynamics other than the prevalent unstable dynamics is one of the driving forces to using model-based control designs.

While this model produced an improvement of the lead-lag controller in some regards, it does not perform better transiently. This may be due to the system identification model not capturing the unstable dynamics correctly. A model that does capture the unstable dynamics is presented in the following section.

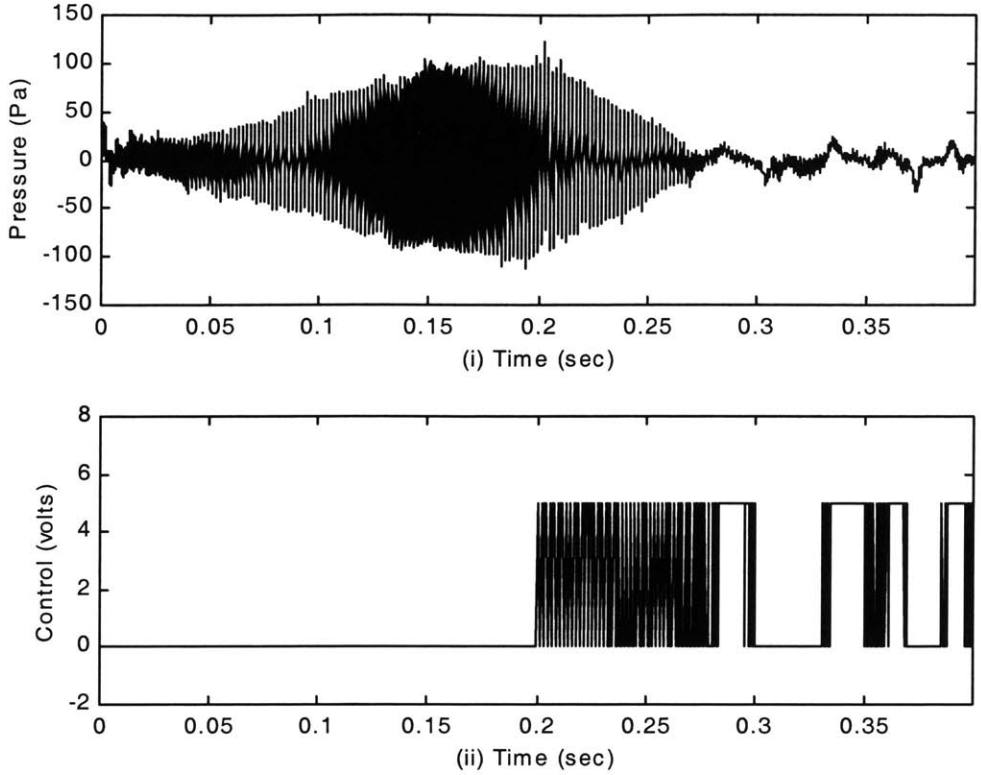



Figure 2-46: Evaluation model response to LQG control based on 6th order prediction-error closed loop model: (i) combustor pressure response, (ii) LQG effort after threshold.

| Controller | $\mu$ / $\rho$ | Transient Time | RMS Pressure | Max Pressure |
|---|---|---|---|---|
| LQG based on closed loop 6th order prediction-error model. | 5e1 / 5e-2 | 0.058 | 8.83 | 32.00 |

Table 8: 6th order prediction-error model transient performance.

### 2.5.6 Closed Loop Identification of 6th order sub-space model

In an effort to find a controller that will perform better during transients, the model and fit criterion space is searched again for a model that captures the unstable dynamics of the evaluation model. The data from section 2.5.2 is pre-processed by bandpass filtering between 100 Hz and 1000 Hz to focus the tools on the important frequencies. The final model from this data is a 6th order sub-space model. The power spectrum of the system identification model versus the filtered closed loop evaluation model response shows acoustic frequencies correctly identified around 176 Hz and 536 Hz, see Figure 2-47 and Table 9. This is similar to the 6th order prediction-error model of section 2.5.3, except this model identifies the poles at 536 Hz to be unstable, as seen in Figure 2-48.

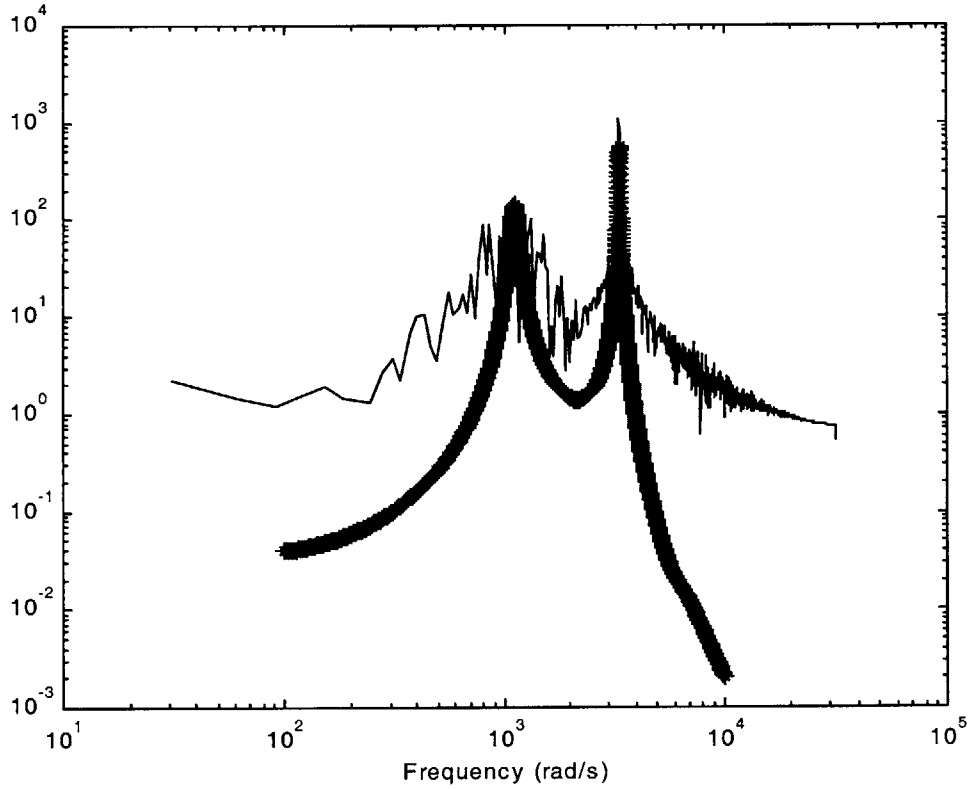



Figure 2-47: Power spectrum of 6th order sub-space model versus evaluation data. Thin-line is evaluation model spectrum; thick-line is 6th order sub-space model spectrum.

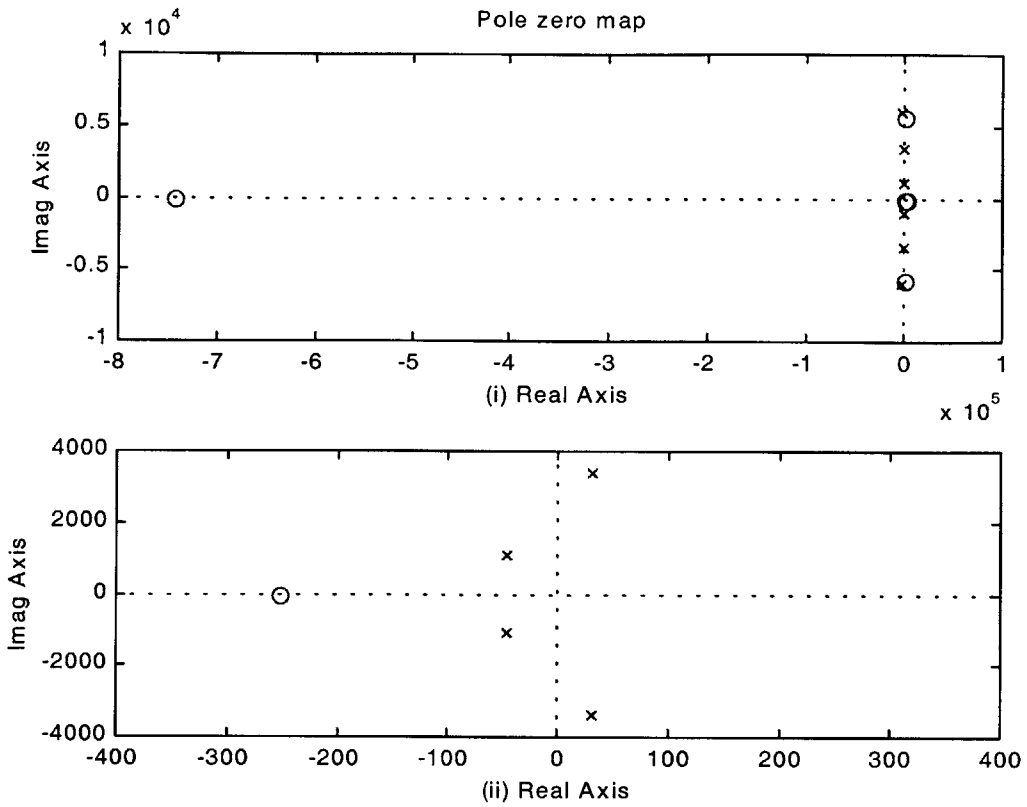



Figure 2-48: Pole-zero plot of 6th order sub-space model. Poles are shown as x's and zeros are shown as 0's

| Eigenvalues | Damping | Freq. (Hz) |
|---|---|---|
| -4.5902e+001 +/- 1.1118e+003i | 4.13e-002 | 176.6 |
| 3.1487e+001 +/- 3.3724e+003i | -9.34e-003 | 536.4 |
| -1.2235e+003 +/- 5.9091e+003i | 2.03e-001 | 959.7 |

Table 9: 6th order sub-space model eigenvalues.

The transfer function for the 6th order sub-space model is

( 25 )

$$\frac{Y(s)}{U(s)} = \frac{-5.2079S^5 - 3.8477*10^6 S^4 + 1.7327*10^{10} S^3 - 1.4719*10^{14} S^2 + 3.2938*10^{17} S + 9.331*10^{19}}{S^6 + 2.4758*10^3 S^5 + 4.9092*10^7 S^4 + 3.2865*10^{10} S^3 + 4.7551*10^{14} S^2 + 6.9647*10^{16} S + 5.1285*10^{20}}.$$

### 2.5.7 Closed Loop Validation of 6th order sub-space model

The quality of the 6th order sub-space model is validated by looking at the residual correlation functions defined in equations ( 22 ) and ( 23 ), and the time response of the model simulation. Figure 2-49 (i) shows the whiteness of the residuals using the auto-correlation function. By having most of the values of the autocorrelation inside the confidence intervals it shows that most of the dynamics and the noise model are captured. The independence between the input and the residual, or cross-correlation is shown in Figure 2-49 (ii). Since all of the values are within the 99% confidence region then there is no significant correlation between any inputs and residuals. This shows that the model should respond similar to the evaluation model for inputs that are different from the system identification input.

The time response of the evaluation model validation data and the simulated response of the 6th order sub-space model is shown in Figure 2-50. A pure simulation could not be used on the unstable 6th order sub-space model because the time response would grow exponentially and be unbounded. Instead it is simulated using a 40 step ahead simulation, see [ 26 ]. Table 10 shows that using this technique the model fits 98.6% of the data. This is an excellent fit and should make an excellent model to use in the creation of an LQG control.

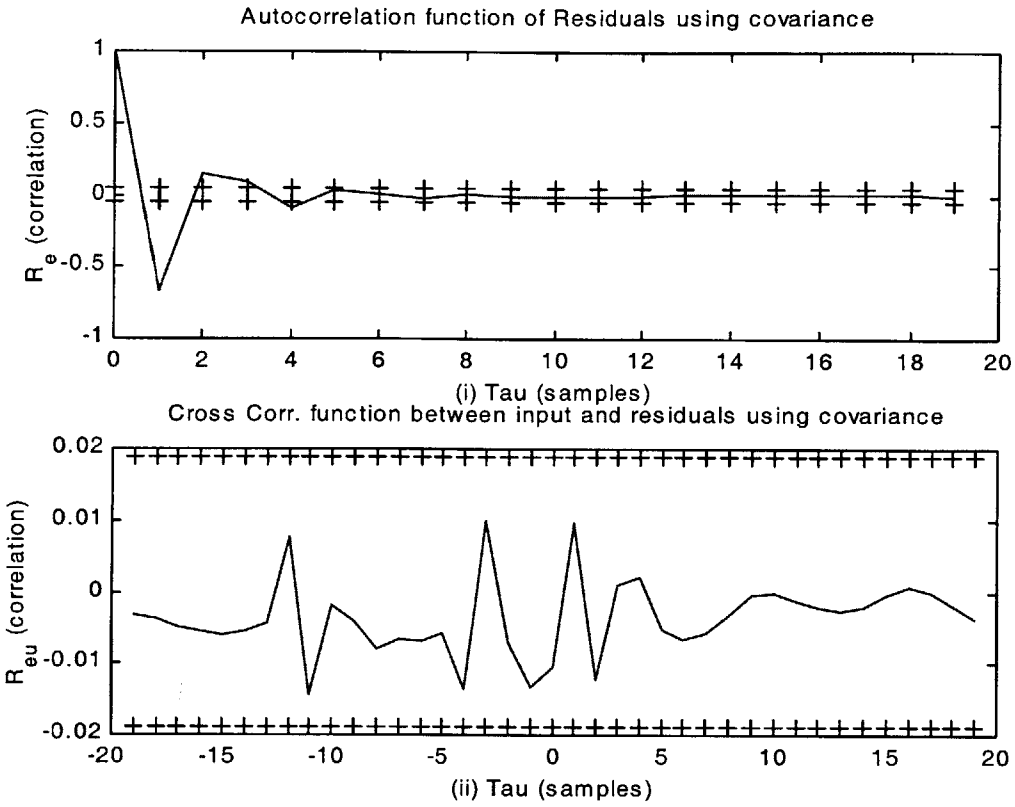



Figure 2-49: Residual analysis of 6th order sub-space model: (i) autocorrelation of residuals, (ii) cross-correlation of inputs and residuals.

| $S_1$ | $S_2$ | $J_k$ | $J_y$ | $R$ |
|---|---|---|---|---|
| 4.3 | 15.2 | 29.9 | 1121.6 | 98.6% |

Table 10: 6th order closed loop sub-space model validation statistics.

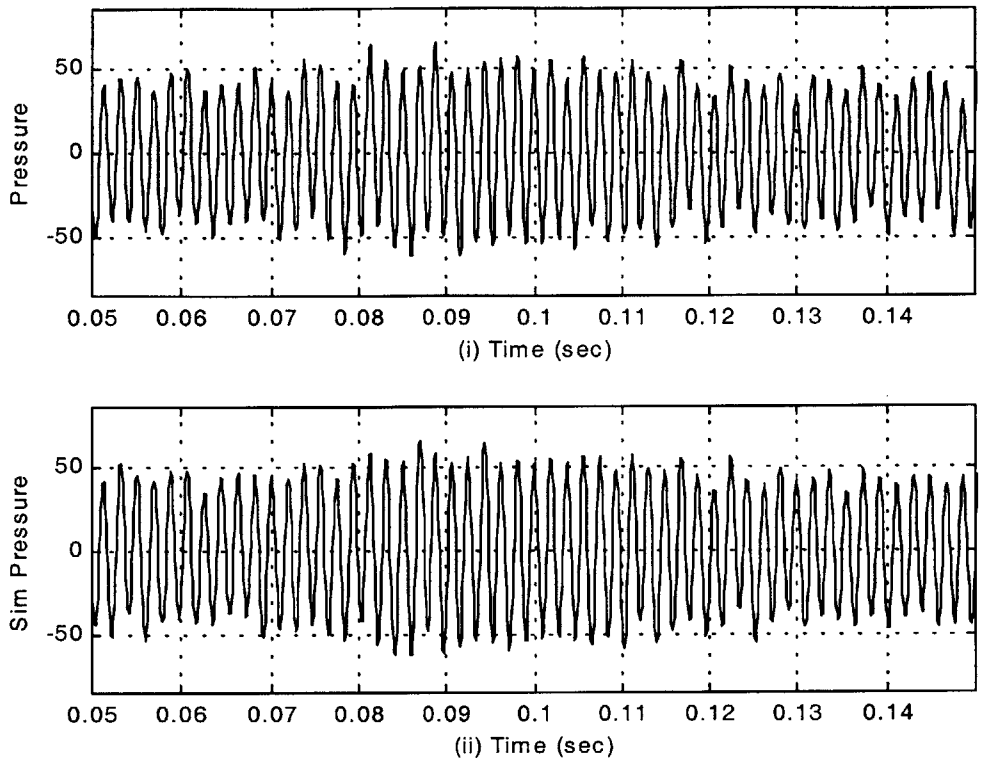



Figure 2-50: Time response data. (i) Evaluation model data, (ii) 40 step ahead simulated $6^{th}$ order sub-space model.

### 2.5.8 LQG control based on closed loop 6th order sub-space model

Using the model developed in section 2.5.6 and validated in section 2.5.7 an LQG control is created. The script file to develop the control and the simulation block diagram are given in Appendix D. The values of the control design parameters $\mu$ and $\rho$, defined in section 3.5., are changed many times until the transient time response of the evaluation model to the LQG control is minimized. With $\mu$ equal to 5 and $\rho$ equal 5e-5 the minimum transient time is found to be 0.041 seconds, see Figure 2-51 and Table 11. This transient time is 16% less than the lead-lag controller. The faster transient time is due to the 6th order sub-space model containing poles that represent the unstable high frequency acoustic mode of the evaluation model.

This LQG has better steady state performance also. The RMS value after the transient is 15% lower than the lead-lag, and the maximum pressure after the transient is 22% smaller than the lead-lag. While this control performs better than the lead-lag in all categories, it has an RMS value after the transient that is 28% higher than the LQG based on the 6th order prediction-error model.

While the performance of this controller is good, there is always room for improvement. One method to make the overall performance better is to use a switch to change between the LQG based on the 6th order sub-space model for the transient and then switch to the LQG based 6th order prediction-error for the steady state. Another possibility is to change the fuel injector characteristics. By making it a proportional instead of on-off will aid in quicker transients and better steady state performance. The last option may be to reformulate the control law to better fit the physical characteristics of the on-off fuel injector actuator. This is accomplished in section 2.6.

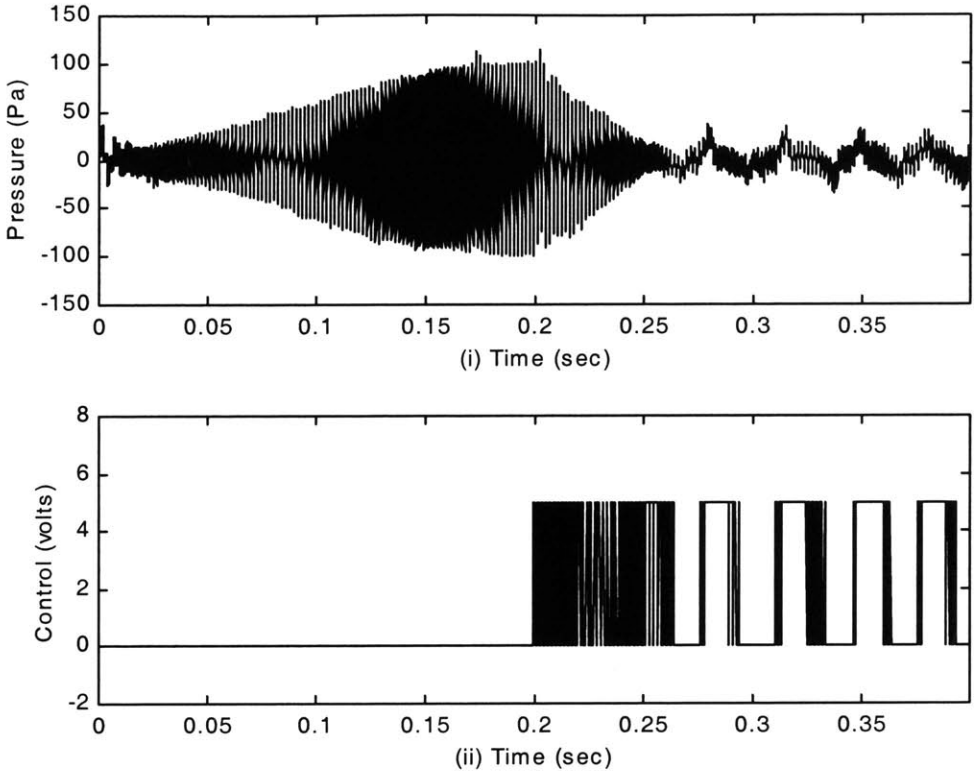



Figure 2-51: Combustor response: (i) evaluation model pressure response, (ii) control effort from LQG control based on 6th order sub-space model.

| Controller | $\mu$ / $\rho$ | Transient Time | RMS Pressure | Max Pressure |
|---|---|---|---|---|
| LQG based on closed loop 6th order S.S. model. | 5e0 / 5e-5 | 0.041 | 11.5 | 36.98 |

Table 11: Summary of evaluation model transient performance

## 2.6 System Identification, Validation, and Bang-Bang Control

The construction of the LQG control assumes that there is an actuator that can reproduce the linear control effort. Because the actuator used in this study is a pulsed fuel injector two things happen: 1. fuel is only added by the injector, 2. there is only on-off control commands. Figure 2-52 (i) shows the LQG linear control and Figure 2-52 (ii) shows the binary signal that results after the threshold function. Since the binary signal has no negative components and does not recreate the shape of the linear effort on the positive side, more than 50% of the desired control effort is not being used as input to control the instabilities. This amounts to a net loss in control action and a decrease in control authority over the linear control design intent.

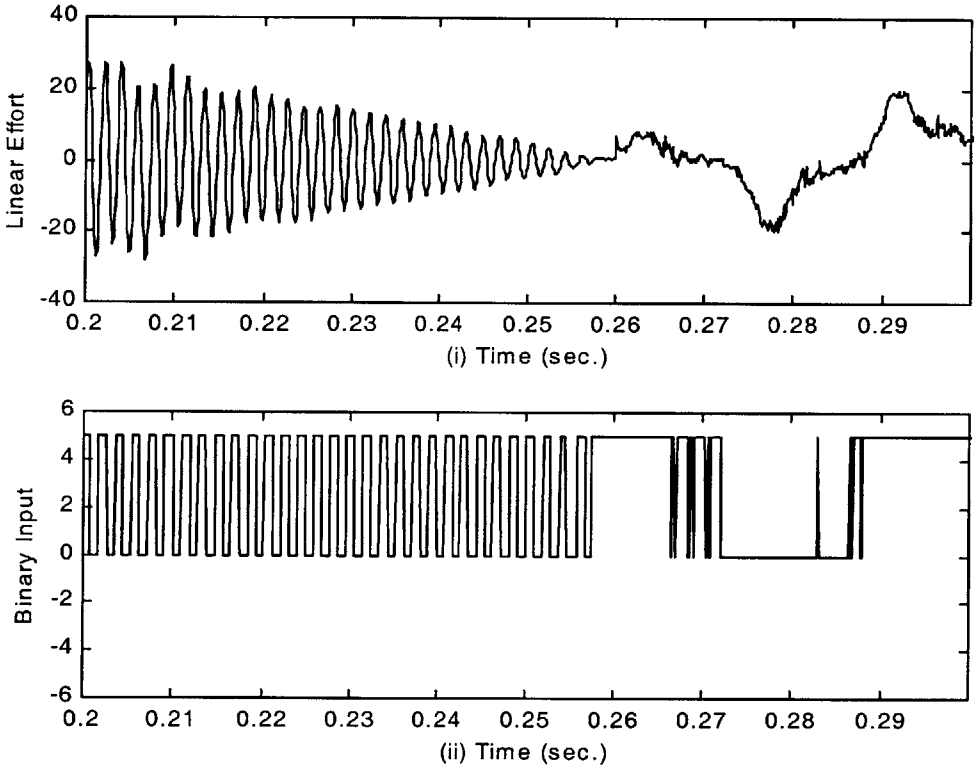



Figure 2-52: 6th order sub-space model LQG control signal comparison: (i) linear LQG control effort, (ii) binary control effort after the threshold and before the fuel injector.

Since much of the control information is not used in the LQG control method utilizing a threshold switch, an optimal method that incorporates the characteristics of on-off action was

sought. The so called 'Bang-Bang' optimal control will determine the optimal routine for such an on-off characteristic. The development of this control as given below was done by [ 17] through [ 22 ]. The central element in this control scheme is the switching function. The switching function can be determined optimally by solving the equations, or sub-optimally by fitting curves to the backwards time trajectories. In this thesis the sub-optimal approach is taken and an algorithm following work in [ 18 ] is utilized.

### 2.6.1 Closed Loop Identification of 2nd order output-error model

The 'Bang-Bang' control is developed based on a model. For the sub-optimal solution the model supplies the backwards time trajectories to get the switching functions. This all requires the creation of a suitable model of the thermoacoustic instability. The development in this thesis of the 'Bang-Bang' control is based only on a 2nd order system model. This drives the system identification model to be 2nd order also.

The data from section 2.5.2 is pre-processed by bandpass filtering between 100 Hz and 1000 Hz to focus the identification tools on the important frequencies. The best fit resulted from a 2nd order sub-space model. The power spectrum of the system identification model versus the filtered closed loop evaluation model response shows the acoustic frequency around at 538 Hz is correctly identified, see Figure 2-53 and Table 12. The SISO transfer function resulting from this model is

$$\frac{Y(s)}{U(s)} = \frac{-1.7676*10^3 S + 2.8857*10^5}{S^2 + 30.686S + 1.1448*10^7}. \quad (26)$$

The Bode plot of this transfer function is shown in Figure 2-54. This model has the poles as being stable, or placed in the left half plane, see Figure 2-55. None of the system identification model structures or criterion of fit resulted in unstable poles for a 2nd order model.

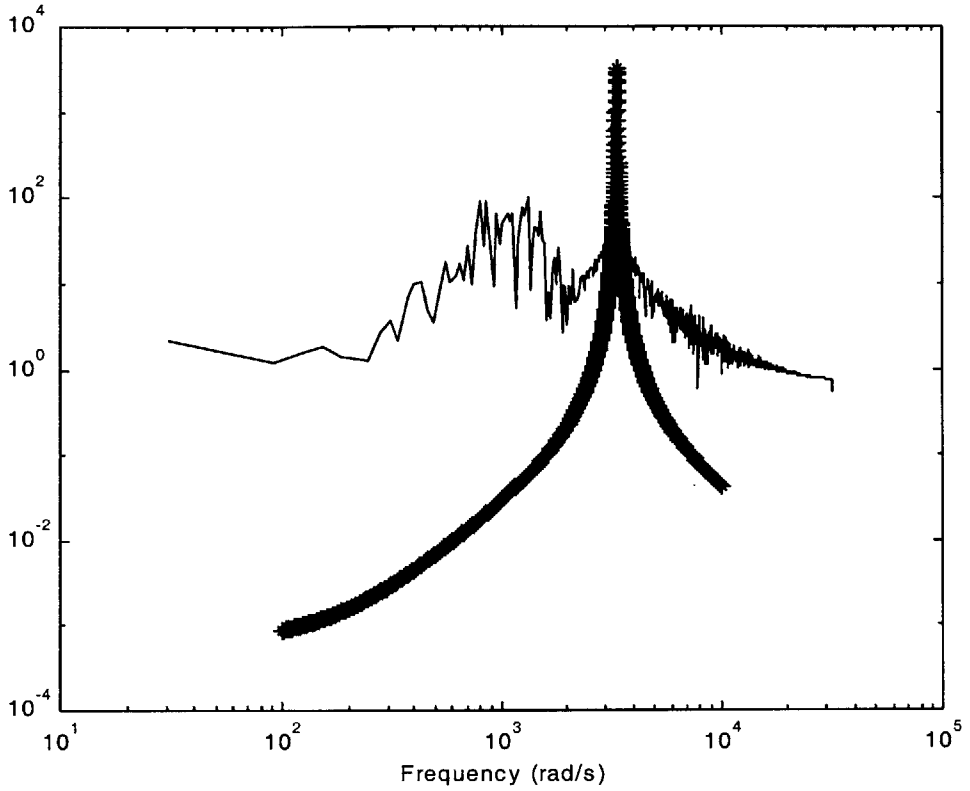



Figure 2-53: Power spectrum of 2nd order output-error model versus evaluation data. Thin-line is evaluation model spectrum; thick-line is 6th order sub-space model spectrum.

| Eigenvalues | Damping | Freq. (Hz) |
|---|---|---|
| -1.5343e+001 +/- 3.3835e+003i | 4.53e-003 | 537.9 |

Table 12: 2nd order output-error model eigenvalues.

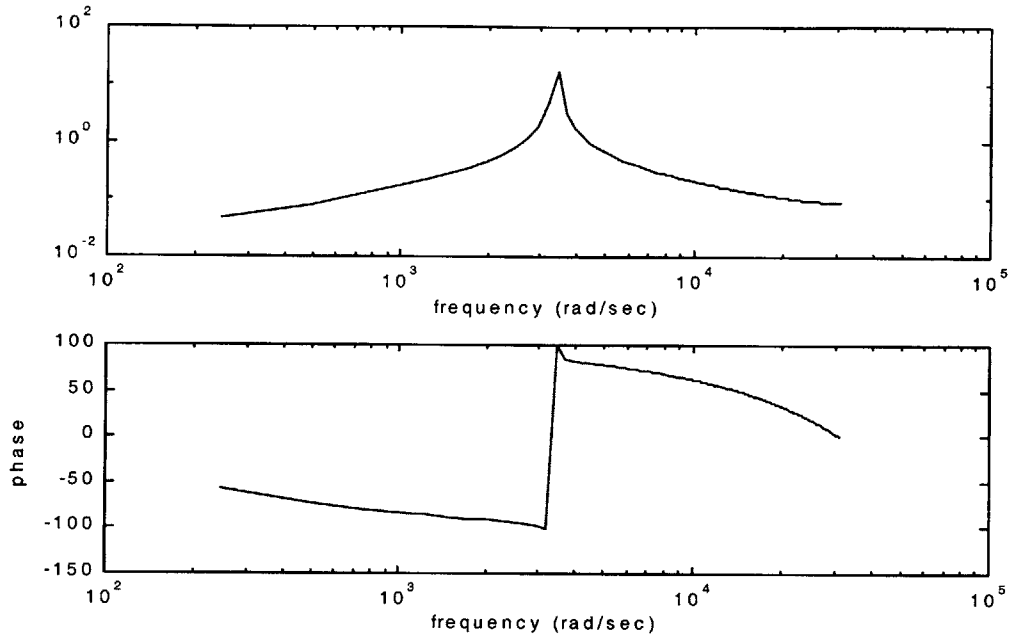



Figure 2-54: Bode plot of 2nd order output-error model.

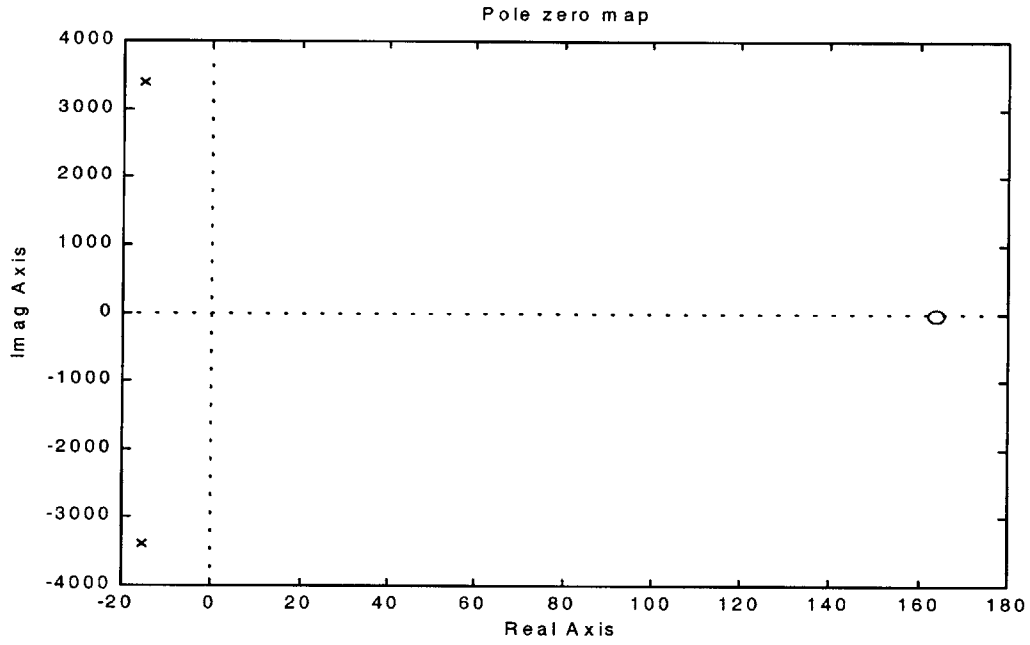



Figure 2-55: Pole-zero plot of 2nd order output-error model. Poles are shown as x's and zeros are shown as 0's.

### 2.6.2 Closed Loop Validation of 2nd order output-error model

The quality of the 2nd order output-error model is validated by looking at the residual correlation functions defined in equations ( 22 ) and ( 23 ), and the time response of the model simulation. Figure 2-56 (i) shows the whiteness of the residuals using the auto-correlation function. By having most of the values of the autocorrelation outside the confidence intervals shows that most of the dynamics and the noise model are not captured. This means that a higher order model is required to capture more of the dynamics behavior. The independence between the input and the residual, or cross-correlation is shown in Figure 2-56 (ii). Since much of the values are outside of the 99% confidence region then there is significant correlation between the inputs and residuals. This shows that the model may not respond similar to the evaluation model for inputs that are different from the system identification input. All of these poor results are expected since the order of the model is required to stay small for the creation of the 'Bang-Bang' control used in this thesis.

The simulated time response of the 2nd order output-error model shows very good correspondence with the evaluation model validation data, see Figure 2-57. Even though the system identification model only contains 2 dynamic elements it reproduces 94.6% of the data, see Table 13. This is a good sign that the 'Bang-Bang' control based on the phase plane response may work well.

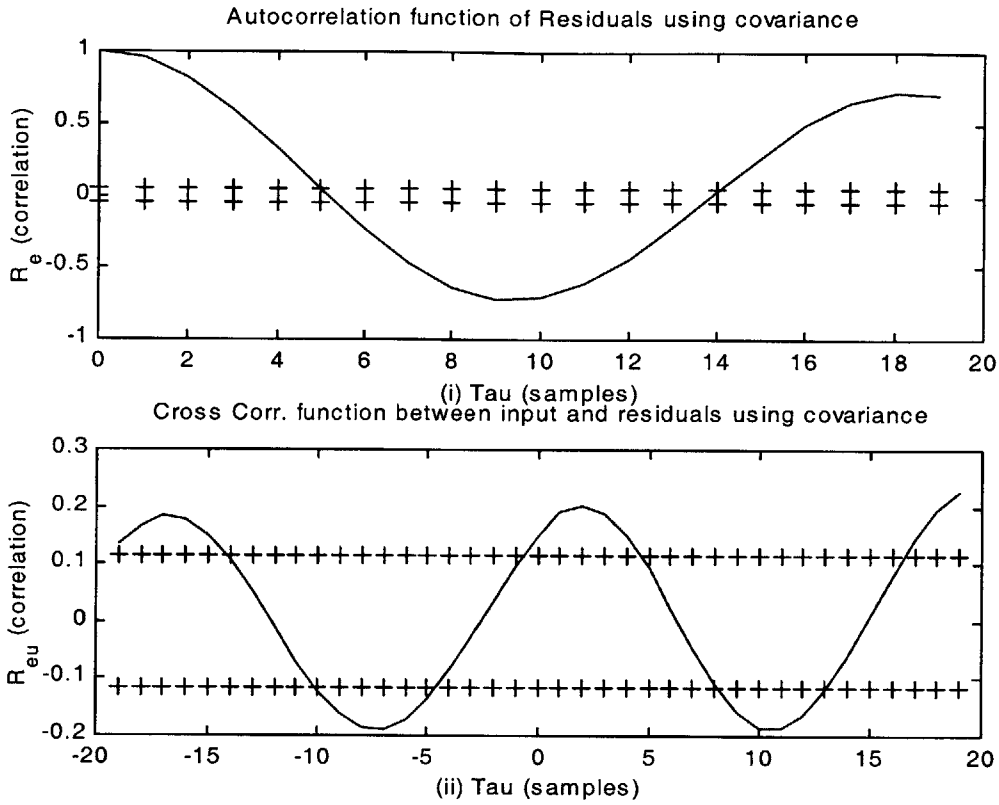



Figure 2-56: Combustor response: (i) evaluation model pressure response, (ii) control effort from LQG control based on 2nd order output-error model.

| $S_1$ | $S_2$ | $J_k$ | $J_y$ | $R$ |
|---|---|---|---|---|
| 8.8 | 25.3 | 116.7 | 1121.6 | 94.6% |

Table 13: 2nd order closed loop output-error model validation statistics

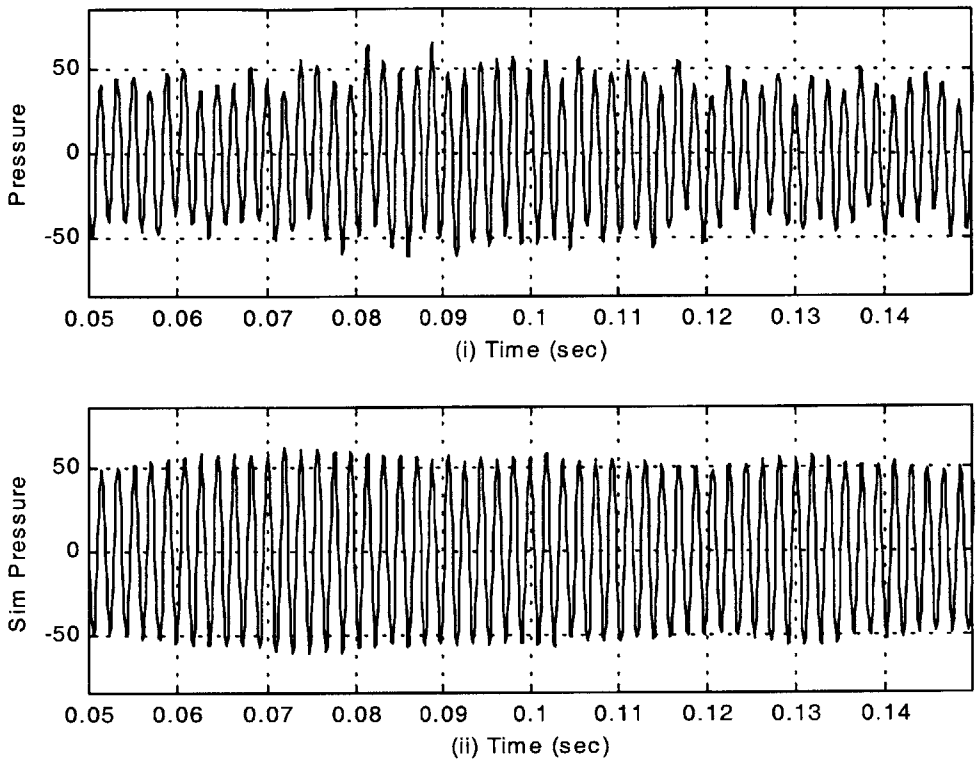



Figure 2-57: Time response data. (i) Evaluation model data, (ii) simulated 2$^{nd}$ order output-error model.

### 2.6.3 Bang-Bang Control Theory

The development of the 'Bang-Bang' controller is shown in [ 17 ] through [ 21 ]. A short presentation of the theory is given in Appendix E. The goal is to take an n dimensional linear system with a constant input,

( 27 ) $$\dot{x} = Ax(t) + Bu(t)$$

where $-U_{max} \leq u(t) \leq U_{max}$, and transfer it to the origin in the minimum amount of time using the performance index J:

( 28 ) $$J = \int_{t_0}^{t_f} L dt = \int_{t_0}^{t_f} 1 dt = (t_f - t_0).$$

The solution involves minimizing the switching function $\lambda^T B$. This results in a control that is compactly stated as

( 29 ) $$u(t) = -\text{sgn}(\lambda^T B)$$

By allowing the sgn function to be either $U_{min}$ or $U_{max}$ in ( 29 ) the time to go response and the phase plane trajectories for both values of sgn can be determined either optimally by solving the equations given in Appendix E, or sub-optimally by fitting curves to the switching functions. The sub-optimal approach is taken in this thesis with the curve fitting and control scheme presented below.

### 2.6.4 Bang-Bang Control Sub-Optimal Solution based on 2nd order output-error model

For the Bang-Bang control at any given time there are only two control alternatives: $u = U_{min}$ or $u = U_{max}$. Since the input can only take on 2 values it is easy to determine the time domain and the phase plane response of the 2nd order output-error model. The states versus time are realized by running the state equations backwards in time in response to $u = 1$ with the final condition (initial condition of inverse time) at the origin. The state trajectory is then mapped to the phase plane in Figure 2-59. This trajectory is the basis for the state feedback control law. A least squares curve fit is used to create the switching function from the phase plane trajectory. The equation for the switching function is

$$(30)\quad X_2 = -9.4574e\text{-}001 * X_1 + 2.4155e+002 * X_1^3 - 1.7798e+005 * X_1^5 + 6.9560e+007 * X_1^7.$$

The control law using the switching function is displayed graphically in Figure 2-60. The switching function is made up of the $U_{min}$ and the $U_{max}$ trajectories with straight lines added at the min and max of the trajectories.

Suppose the initial state is as shown in Figure 2-60. Then to bring the plant to the origin an input $u = U_{min}$ would transfer the model states following the $U_{min}$ trajectory until the switching function is reached. The input will change to $u = U_{max}$ and follow the $U_{max}$ trajectory closer to the origin. Continuing on with this logic will bring the system to the origin in the minimum time. The script file used to apply the 'Bang-Bang' control and the simulation block diagram are given in Appendix E.

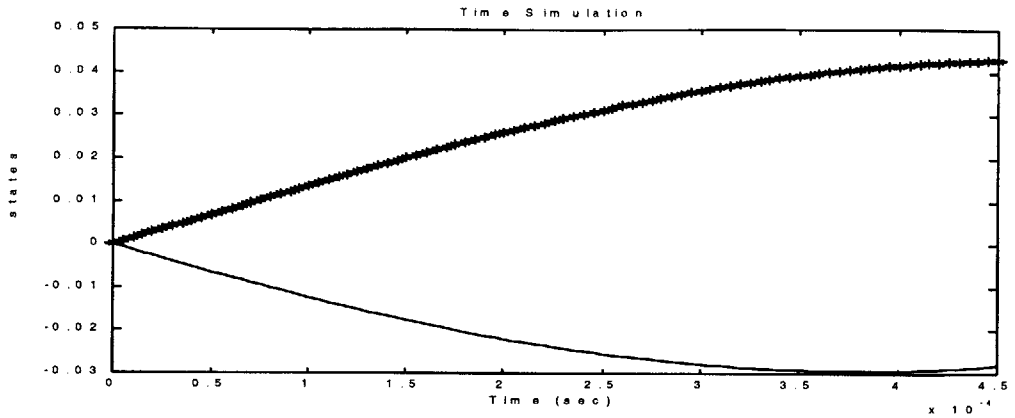



Figure 2-58: State inverse time response to an input u=1: Thick-line is X1, thick line is X2.

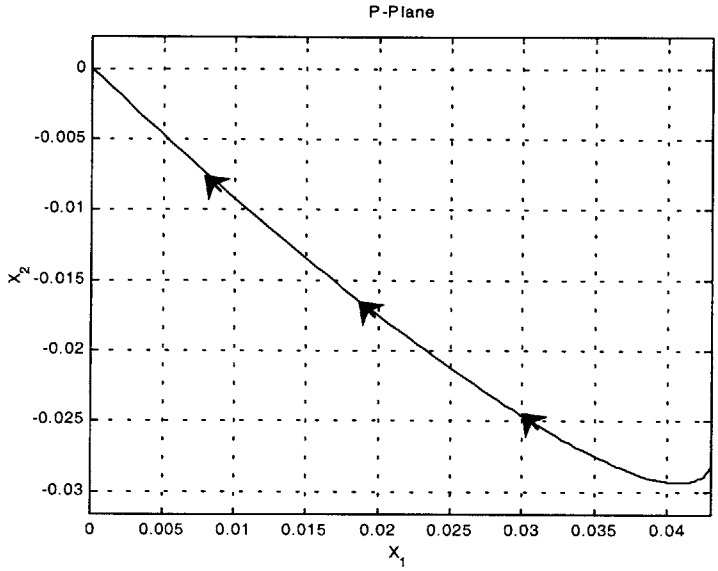



Figure 2-59: Phase plane trajectory of 2nd order output-error model to an input u=1.

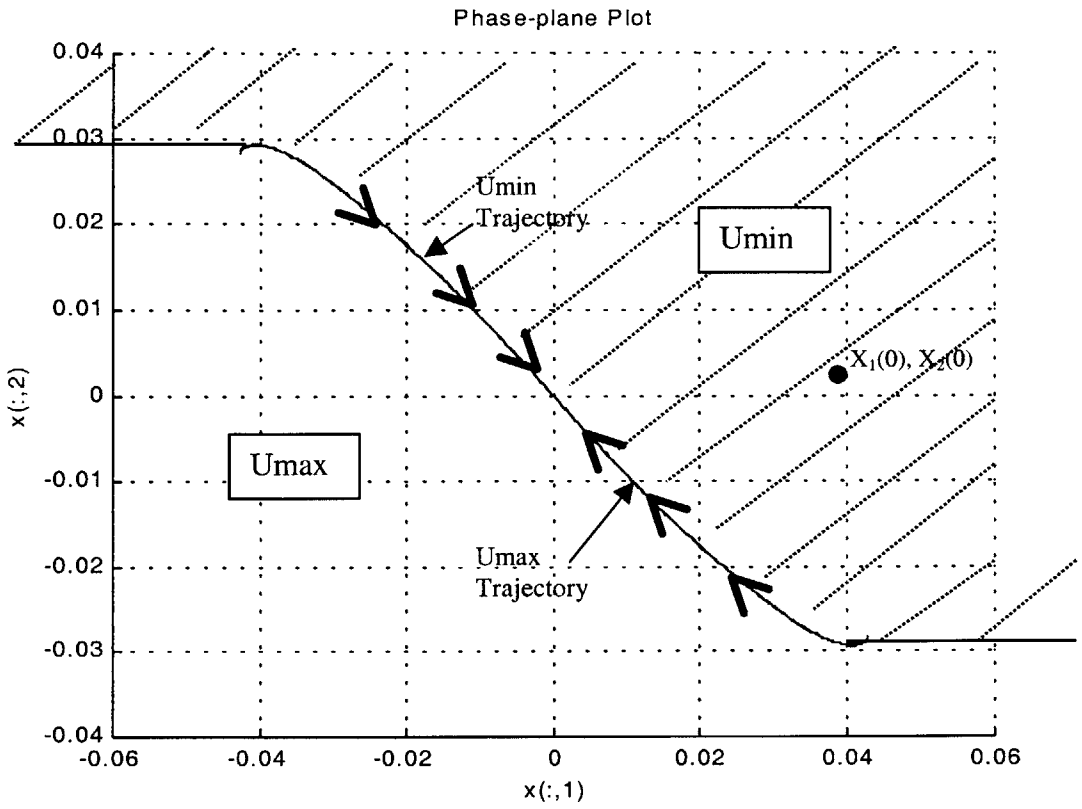



Figure 2-60: 'Bang-Bang' state feedback control law.

The performance of the 'Bang-Bang' control is investigated on the evaluation model. When the 'Bang-Bang' control is applied at 0.2 seconds the evaluation model is stabilized, see Figure 2-61. The transient time of 0.065 is 50% slower than the LQG based on the 6$^{th}$ order sub-space model, but the steady state performance is better than any of the other controls, see Table 14. The transient performance would probably be improved if the model identified the 500 Hz acoustic mode as being unstable. The steady state performance is the best for the 'Bang-Bang' because it only applies control action when it is necessary.

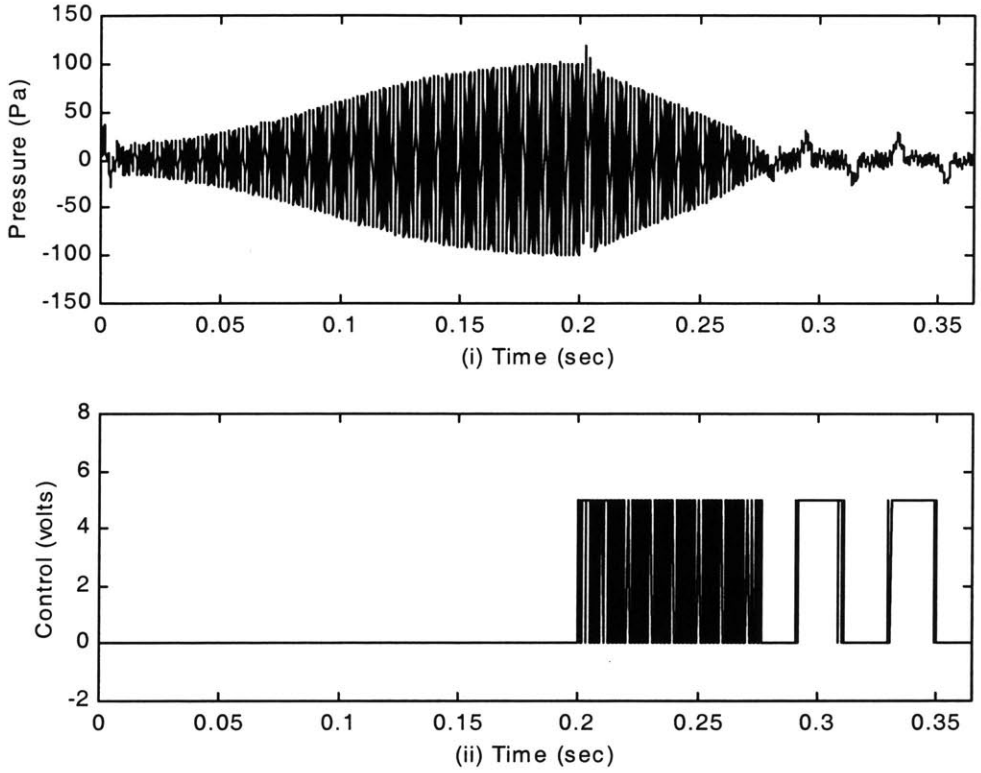



Figure 2-61: Combustor response to 'Bang-Bang' controller: (i) pressure response, (ii) 'Bang-Bang' controller input.

| Controller | Transient Time | RMS Pressure | Max Pressure |
|---|---|---|---|
| 'Bang-Bang' Control based on 2$^{nd}$ order model | 0.065 | 8.12 | 31.20 |

Table 14: Summary of evaluation model transient performance in response to 'Bang-Bang' control.

## 2.7 Evaluation Model System Identification and Control Summary

Through this study several active controls were investigated on a nonlinear evaluation model of thermoacoustic instability. An experimentally determined lead-lag control was used as a tool to stabilize the thermoacoustics. The performance showed fast transient times, but poor steady state performance. The best transient performance was displayed by the LQG based on the $6^{th}$ order sub-space model, see Table 15. This control performed the best transiently because the unstable dynamics of the evaluation model were correctly identified, which allows the LQG to optimize the control correctly.

The steady state performance is best for the on-off 'Bang-Bang' control, see Table 15. This is due to the 'Bang-Bang' delivering the desired magnitude of control at all times. In contrast, the LQG control effort magnitude is very rarely equal to the magnitude of the binary input to the fuel injectors. This causes the input to the fuel injectors to either be too strong or too weak as compared to what is desired from the LQG. This method works during the transient because the maximum control effort out of the fuel injectors is obtained. But in the steady state adding too much fuel will reexcite the instability and make it difficult to control.

| Controller | Transient Time | RMS Pressure | Max Pressure |
|---|---|---|---|
| Lead-Lag | 0.049 | 13.58 | 47.3 |
| LQG based on open loop $4^{th}$ order S.S model., rho=5e2, mu=5e-1 | 0.148 | 8.57 | 33.12 |
| LQG based on closed loop $6^{th}$ order prediction-error, rho=5e-2, mu =5e1 | 0.058 | 8.83 | 32.00 |
| LQG based on closed loop $6^{th}$ order S.S. model rho=5e-5,mu=5e0 | 0.041 | 11.5 | 36.98 |
| 'Bang-Bang' Control based on $2^{nd}$ order model | 0.065 | 8.12 | 31.20 |

Table 15: Summary of evaluation model transient performance

# 3. System Identification and Control of a Turbulent Combustor: An Experimental Study

System identification and control are investigated experimentally on a model ramjet dump combustor. Liquid fuel is used as both the main fuel and the secondary control fuel. The liquid fuel is supplied by four pulsed fuel injectors that spray directly into the combustor. Two of the fuel injectors are run open loop and supply the main fuel. The other two injectors can be used in either open loop or closed loop to suppress the combustor pressure oscillations.

This is the same combustor that was used by Dr. Ken Yu at the China Lake Naval Air Warfare Center as reported in [ 28 ] as case number 5. Dr. Yu's experiments demonstrated that combustion instabilities can be successfully suppressed using a phase-lock closed loop control on a liquid fuel actuated combustor. The results from the phase-lock control are used as a baseline example of good performance in this experimental investigation.

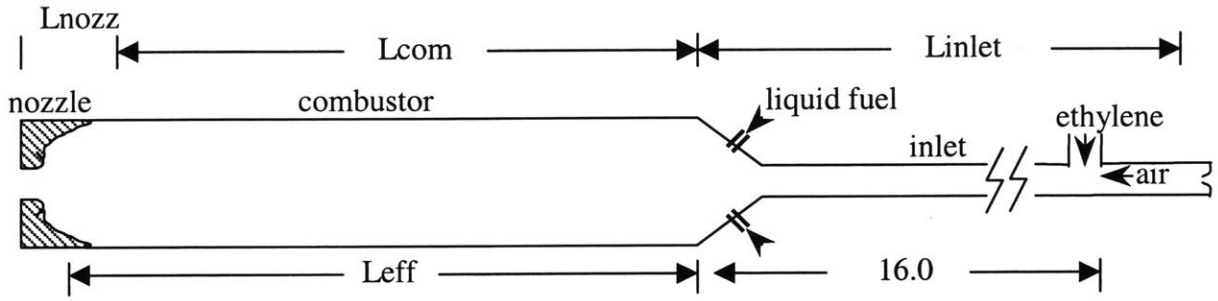
Both open loop and closed loop identification models are created from the combustor experimental data. Such a model-based controller developed in this thesis is shown to perform better than the experimentally derived phase-lock control.

## 3.1 Experimental Setup

### 3.1.1 Combustor

Experiments were performed on a dump combustor as shown in Figure 3-1. Air is supplied from a high pressure storage tank. A choked orifice is used to measure the air flow through the combustor, and provide an upstream acoustic boundary. Ethylene is introduced upstream of the dump zone only as a pilot. The pulsed fuel injectors then introduce ethanol ($C_2H_6O$) and the ethylene is turned off for the remainder of the experiment. An exhaust nozzle is used at the downstream end of the combustor. The values for the parameters in Figure 3-1 (i) are given in Table 16.

(i)




(ii)

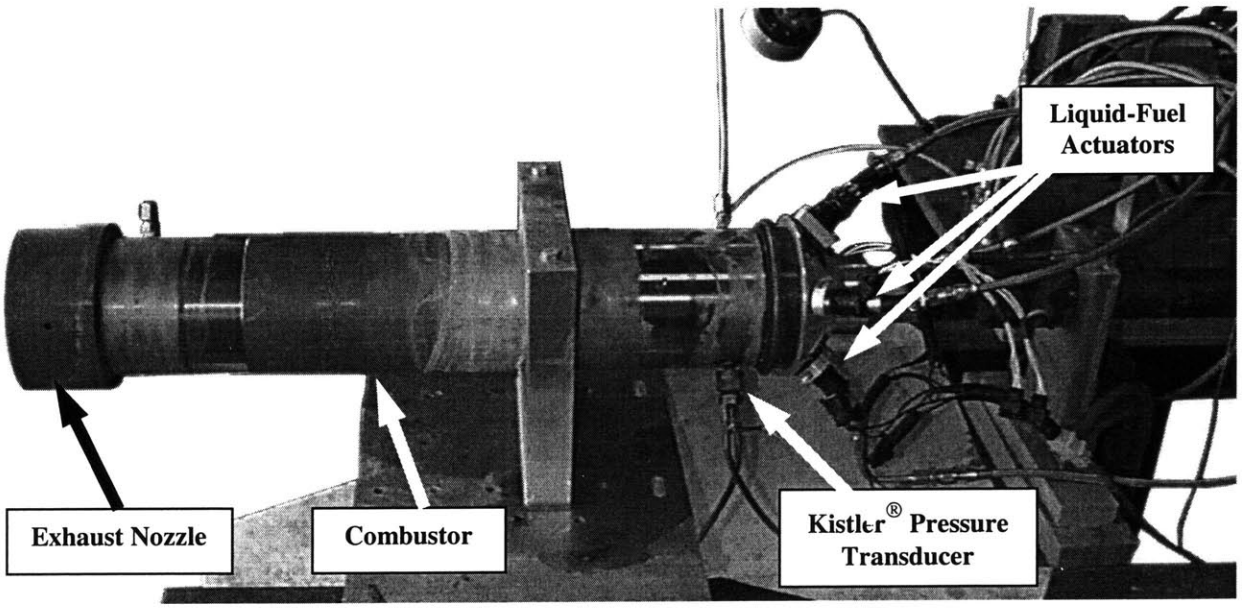



Figure 3-1: Axisymmetric dump combustor set-up: (i) schematic, (ii) picture of combustor.

| $L_{inlet}$ | $L_{comb}$ | $L_{nozz}$ | $D_{inlet}$ | $D_{comb}$ | $D_{nozz}$ |
|---|---|---|---|---|---|
| 254 cm | 64.7 cm | 5.6 cm | 4.19 cm | 10.16 cm | 5.4 cm |

Table 16: Dump combustor configuration

During unstable operations with all four of the injectors running open loop, the combustor pressure is 0.8 psi above ambient. The other flow conditions for the combustor are given in Table 17.

| $V_{inlet}$ | $\dot{m}_{air}$ | $\dot{m}_{fuel}$ | $\phi$ | $R_e$ | $D_{32}$ |
|---|---|---|---|---|---|
| 17.7 m/s | 28.5 gm/sec | 1.63 gm/sec | 0.51 | 47300 | 40 μm |

Table 17: Flow conditions during unstable operation with 4 injectors at 50% duty cycle

where $\phi$ is the equivalence ratio, $R_e$ is the Reynold's number, and $D_{32}$ is the Sauter-mean diameter.

### 3.1.2 Actuators

The instantaneous heat release is influenced in this experiment by high-frequency pulsed fuel injectors. The injectors are "off-the-shelf" automotive fuel injectors that provide high-frequency response as well as high volume flow rate. A 0 to 5 volt square wave is used to command the integrated circuit relays to drive the fuel injector solenoid. The fuel injector exit has a customized swirl-based atomizer to reduce the mean diameter of the droplets and impart angular velocity for better mixing. Four fuel injectors are used throughout this experiment. The injectors are angled at 45 degrees with respect to the air flow direction. The command from the PC to the drive circuit relays is transformed from a continuous to a binary input by using a threshold logic block.

### 3.1.3 Sensor

A Kistler® piezo pressure transducer mounted 1 diameter downstream of the dump region is utilized as feedback for the control. The pressure sensor completes the control system as shown in Figure 3-2.

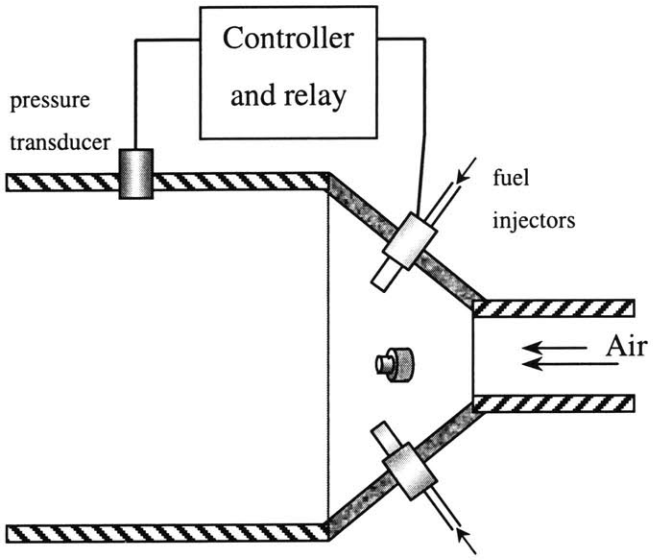



Figure 3-2: Combustor control system schematic.

### 3.1.4 Control

Experiments accomplished in [ 28 ] concluded that open loop control is not effective in suppressing the oscillation amplitude, and closed loop controlled injection is needed. Four fuel injectors each with 90 degrees of separation are used throughout this experiment. To incorporate closed loop control, two injectors are always commanded open loop and two injectors can be operated in either open or closed loop.

The two open loop injectors, 180 degrees apart, are commanded by a function generator. The function generator is set to command the injectors with 50% duty cycle at the unstable

frequency thereby creating the largest oscillation amplitude. The other two injectors can be driven in either open or closed loop.

The closed loop command can be generated by a Wavetek™ Variable Phase synthesizer or by digital signal processing based on a personal computer. The Wavetek™ has the ability to use simple phase-lock and delay control onto the pressure signal and is implemented in-line with a 40 to 140 Hz Butterworth bandpass filter and amplifier as shown in Figure 3-3 (i). The digital signal processing is implemented using a Keithly MetraByte DAS1801-AO data acquisition board sampling data at 10 kHz and a 166 MHz Pentium® PC as shown in Figure 3-3 (ii). Using the PC allows great flexibility in the design of the control. Two control methods, lead-lag and LQG, are evaluated experimentally. C-code is used to implement the control and communicate with the data acquisition board.

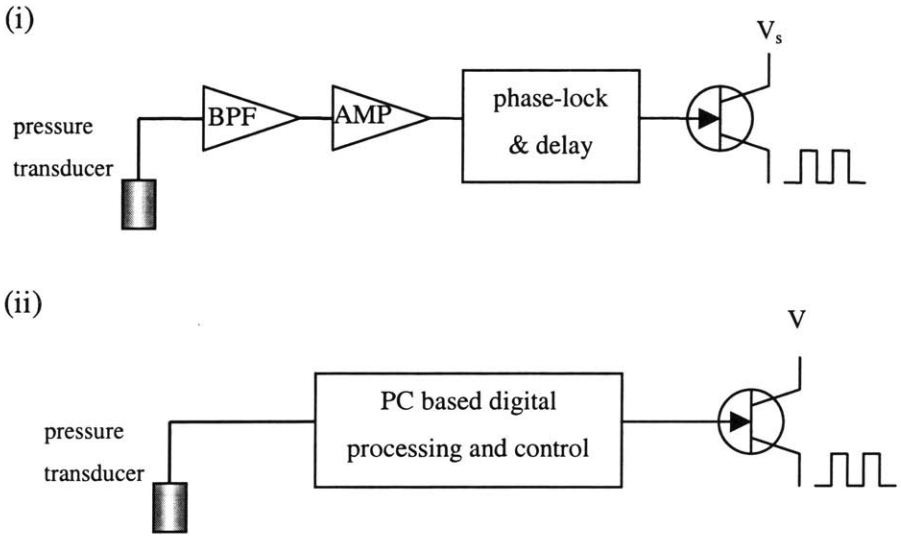



Figure 3-3: Closed loop control of dump combustor: (i) Wavetek™ phase-lock control, (ii) PC based digital processing and control.

## 3.2 Baseline Combustor Performance

### 3.2.1 Open loop combustor response

The combustor was run with all 4 injectors running open loop off of the function generator at the unstable frequency to determine the baseline characteristics of the combustor instability. Figure 3-4 shows the raw pressure response to the 4 injectors. The plotted pressure values are in volts, and the relationship between voltage and pressure is ~1 volt per 1 psi. The plot of the combustor pressure power spectrum of Figure 3-4 (iii) shows the combustor instability is at 90 Hz.

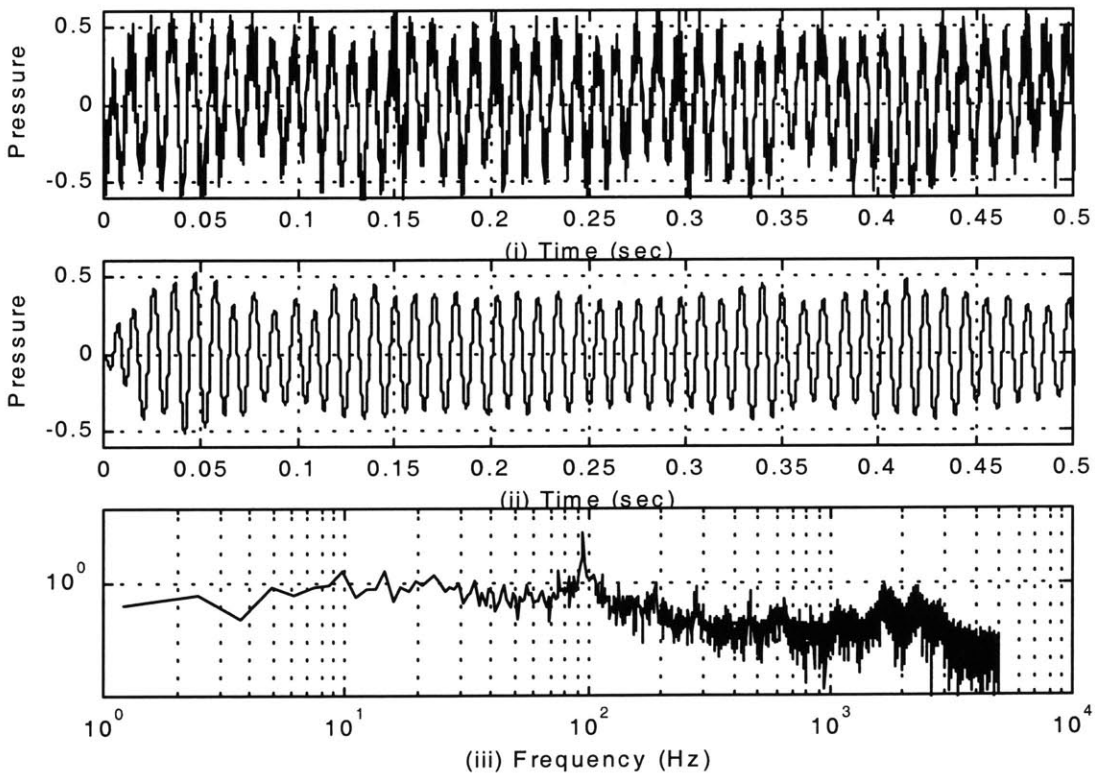



Figure 3-4: Combustor response to all 4 injectors running open loop at unstable frequency. (i) Unfiltered time response, (ii) bandpass filtered time response, (iii) power spectrum of unfiltered time response.

### 3.2.2 Active control with Wavetek™ phase-delay

Following the experiments accomplished in [ 28 ], the Wavetek™ is used to close the loop on pressure as shown in Figure 3-2 and Figure 3-3 (i). The approach taken is to pulse the liquid fuel at the instability frequency and change the timing or phase using the Wavetek™. A range of phases from 0 to 360 degrees is tried experimentally to determine the most effective control. The goal is to reduce the amplitude of the pressure oscillations by changing the phase of the heat release so that the heat release oscillations interfere destructively with the pressure oscillations. The Wavetek™ employs a phase-lock algorithm that forces the fuel injectors to be commanded every cycle. This ensures that the equivalence ratio, $\phi$, stays near the 4 injector open loop value of 0.51. Figure 3-5 shows that using a phase of 300 degrees has the greatest stabilizing effect on the pressure oscillations with an RMS pressure value of 0.22 psi, average peak pressure of 0.31 psi, max peak pressure of 0.58 psi, and a min pressure over 2 cycles of 0.12 psi. Therefore the pressure reduction from the average peak to the min peak is 61.2%, and the maximum reduction in pressure from the max to min pressure is 79%. The results are summarized in Table 18 and Figure 3-6 shows the time response of the combustor to the 300 degree phase control.

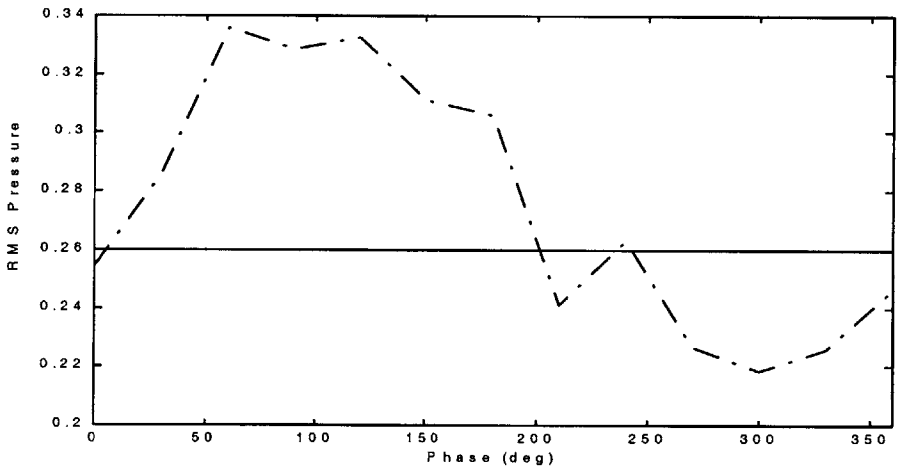



Figure 3-5: Pressure oscillation RMS value as a function of injection timing. The solid straight line shows the RMS level for the open loop 4 injector case.

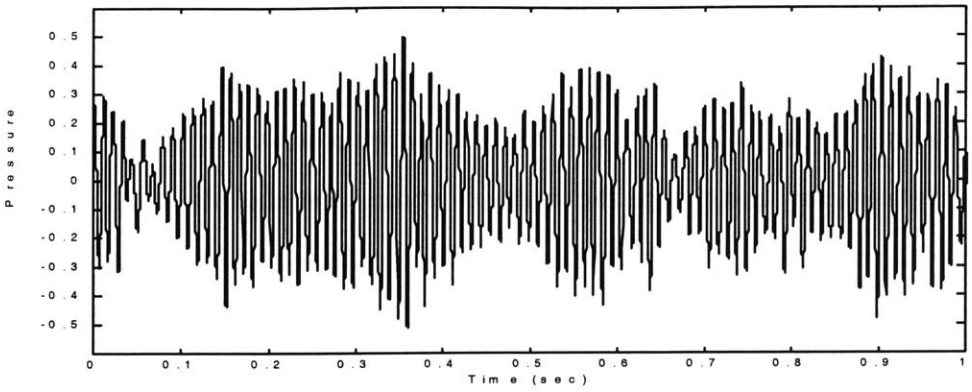



Figure 3-6: Time response of combustor to Wavetek™ control at 300 degrees phase.

| Avg. Peak (psi) | Max Peak (psi) | Min Peak (psi) | Reduction from Avg | Max Reduction | RMS (psi) |
|---|---|---|---|---|---|
| 0.31 | 0.58 | 0.12 | 61.2% | 79% | 0.22 |

Table 18: Wavetek™ control with 300 degree phase transient summary

## 3.3 Experiment Design: Open loop Identification

The goal of the open loop experiments is to get the dynamic input-output relationships between fuel injector command and combustor pressure response so a dynamic model can be built for control purposes. This is accomplished by driving the fuel injectors through a threshold with a system identification input R(t), recording the fuel injector command U(t) and the pressure response output Y(t). Then, using system identification tools a dynamic model is fit to the data. See Figure 3-7 for the block diagram of the open loop identification system.

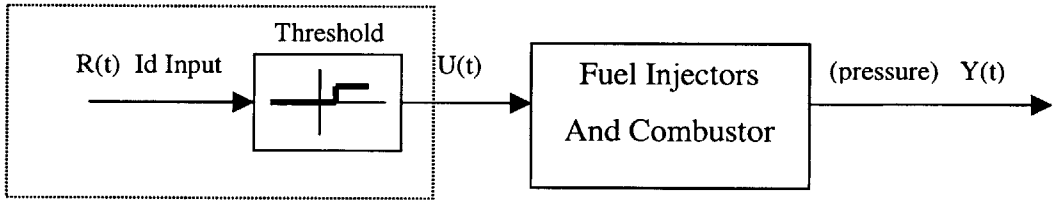



Figure 3-7: Block diagram of open loop system with identification input. Dotted line encloses the processing in the computer.

Much work was accomplished in [ 27 ] and [ 28 ] to determine a fuel injector configuration that has control authority over the combustor instability.

### 3.3.1 Input Design for Open Loop Experiment

Since it is already established that the dump combustor with the pressure sensor and fuel injectors has some amount of control authority over the instabilities, it is wise to use the same actuators and sensors for the system identification experiment. The goal of the open loop experiment is to force the combustor to display its dynamic characteristics by exciting it with 2 of the 4 fuel injectors. In order to get an informative experiment the input must be rich. From [ 26 ] an open loop experiment is informative if the input is persistently exciting. If the input signal $u(t)$ has a spectrum $\Phi_u(\omega)$, then for $u$ to be persistently exciting it must have nonzero $\Phi_u(\omega)$ at n points, where n is the number of parameters to be estimated. Therefore, the optimal input design will depend on the system. Since the combustor system order is unknown a priori, then it is wise to

input many frequencies around the interesting regions to be able to validate against higher order system identification models. The interesting frequencies to examine can be found by looking at the combustor output spectra as shown in Figure 3-4.

The power spectrum shows a pronounced frequency at ~90 Hz. The input should be designed to excite this frequency and attenuate others. This is accomplished by sending random noise through a $6^{th}$ order bandpass Butterworth filter with a lower corner point of 25 Hz and upper cut-off frequency of 270 Hz. The break frequencies allow for margin on both ends of the frequency band. The time response for the filtered random noise is shown in Figure 3-8 (i). Because the fuel injector is pulsed or on-off, the input is converted to a binary signal as shown in Figure 3-8 (ii). In converting from a continuous signal to a binary signal the frequency spectra of the input signal loses the sharp attenuations at the corner points as shown in Figure 3-8 (iii). The binary input signal will be the signal used to excite the combustor for identification purposes.

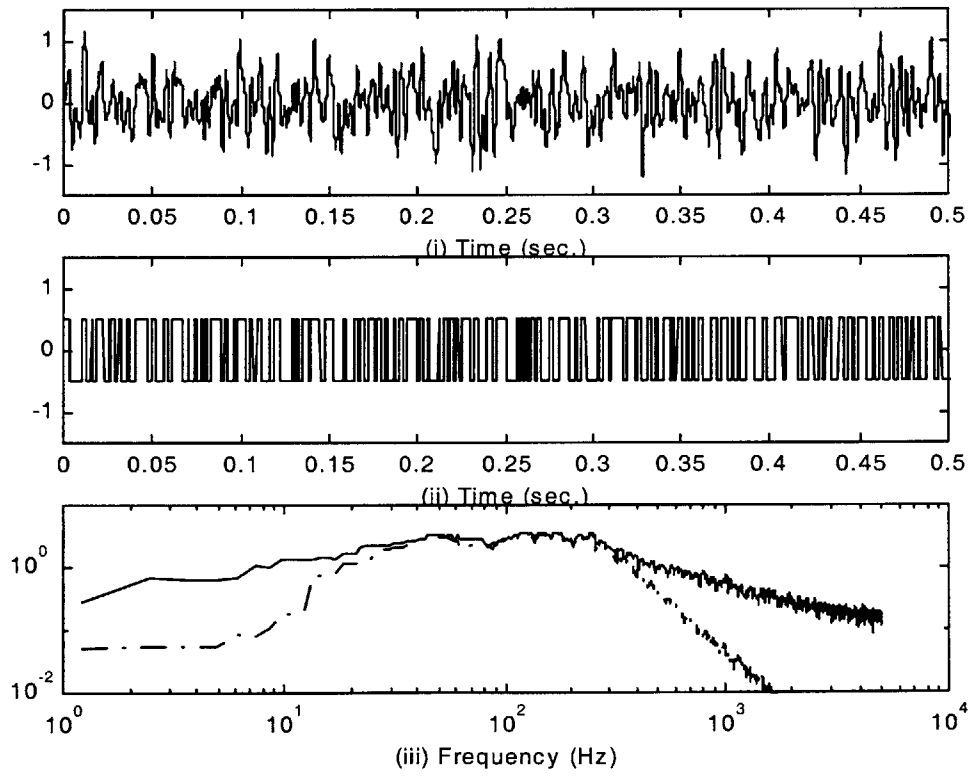



Figure 3-8: System Identification Input signal. (i) Random noise signal through bandpass filter, (ii) random binary noise signal, (iii) spectra of the signals. Dashed line: random noise through filter. Solid line: binary noise signal.

### 3.3.2 Open Loop Data Collection

The combustor response to the binary system identification input is shown in Figure 3-9 (ii). The pressure spectrum of Figure 3-9 (iii) shows again that there are dominant dynamics around 95 to 100 Hz. Some activity is also seen at low frequencies around 20 Hz and at high frequencies around 2500 Hz. Note that the frequencies are shifted due to recording time delays on the PC, not due to a shift in dynamics.

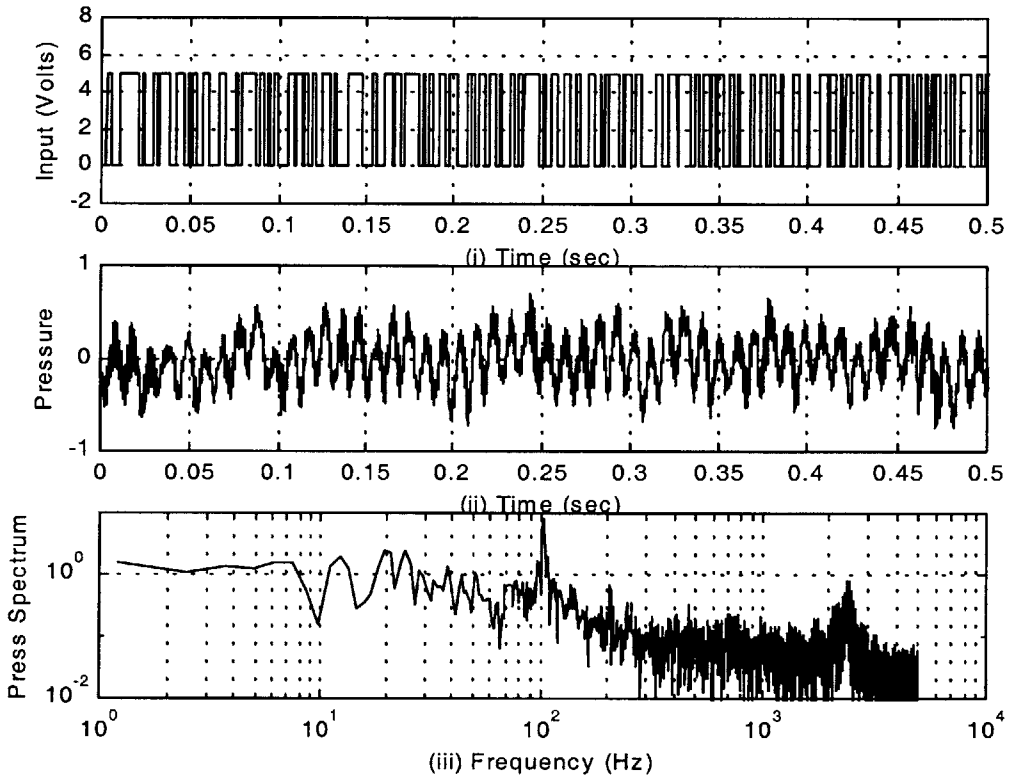



Figure 3-9: Open loop identification. (i) System identification input, (ii) unfiltered combustor time response, (iii) combustor pressure spectrum.

## 3.4 System Identification and Validation: Open loop

Uncontrolled combustors that display instabilities have pressure responses that are saturated and operating in the nonlinear region as shown in Figure 1-3. Since the combustor is operating at saturation and not displaying the transient growth in pressure that is typical of unstable dynamics, it is difficult for the system identification methods to capture the unstable dynamics. The goal of the open loop or nonlinear region system identification is to obtain an initial model that allows creation of a partially stabilizing control. The method of finding a suitable system identification model is iterative. A simple approach is to create a number of different model structures using different fit criterion and then compare the resulting models to each other and to the observed data. Models that do well with this stage of validation analysis will then be used as the model for control design. The final criterion for a good system identification model is one that allows a successful stabilizing control.

### 3.4.1 Open loop identification

With the estimation data collected from the initial open loop design, the open loop system identification model can be created. While the combustor contains numerous nonlinearities, the system identification model will be a time-invariant linear system approximation. This type of model simplifies the design of a controller. It is also assumed the system is causal and that all disturbances can be lumped into a single term. The model set and the criterion to fit the data are chosen by trial and error to determine a system identification model that allows a successful stabilizing control design. The model structure can take on many forms. One model description, see [ 26 ] is:

$$y(t) = \sum_{k=1}^{\infty} g(k)u(t-k) + \sum_{k=0}^{\infty} h(k)e(t-k), \tag{31}$$

where $u(t)$ is the system input, $e(t)$ is a normally distributed random input with zero mean and variance $\lambda$. By introducing the backward shift operator q

$$q^{-1}u(t) = u(t-1), \tag{32}$$

the system can be described in transfer function form

( 33 ) $$y(t) = G(q)u(t) + H(q)e(t),$$

where $G(q) = \sum_{k=1}^{\infty} g(k)q^{-k}$, and similarly $H(q) = \sum_{k=1}^{\infty} h(k)q^{-k}$. This can be expanded to a more general family of model structure

( 34 ) $$A(q)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t),$$

where A, B, C, D, and F are polynomials. Many combinations of the polynomials have been used for system identification in the literature. The number of parameters to fit equals the combined order of all of the polynomials used in the model structure description.

Another popular model description used for system identification is the state-space model

( 35 ) $$\dot{x}(t) = F(\theta)x(t) + G(\theta)u(t)$$
$$y(t) = C(\theta)x(t) + v(t).$$

For this model structure, the number of parameters to fit in $\theta$ is on the order of the number of energy storage elements in the system.

The estimation data from the open loop experiment data is then fit to one of the model structures in equation ( 34 ) or ( 35 ) using the least-squares, prediction-error, or sub-space method. See [ 26 ] for further detail on these methods. The computation of the models is facilitated by using the MATLAB® identification toolbox. One predictor of the model response is

( 36 ) $$\hat{y}(t \mid \theta) = \varphi^T(t)\theta,$$

where $\hat{y}(t \mid \theta)$ is prediction of y given $\theta$,

( 37 ) $$\varphi(t) = [-y(t-1)\ldots - y(t-n) \quad u(t-1)\ldots u(t-m)]^T$$

is the regressor or past value vector, and

( 38 ) $$\theta = [a_1 \ldots a_n \quad b_1 \ldots b_m]^T$$

is the estimated parameter vector. The predictor of equation ( 36 ) can be transformed to fit either of the model structures shown in equations ( 34 ) or ( 35 ).

After trying many of the model structures and estimation methods, a 6th order model of the form shown in ( 34 ), where $F$ and $D$ equal 1, and $A$, $B$, $C$ have 3, 6, and 5 estimated parameters respectively. The prediction-error method is used to fit the data to this 6th order model structure. The analysis and validation of this model is given below in section 3.4.2.

### 3.4.2 Open loop model validation

During model validation the quality of the model is evaluated. As stated earlier, the final validation will be a model that allows a successful control design. Since it is not practical to create and evaluate a control for every model, other validation tools are used to filter out poor models. Some common tools used for this purpose are residual analysis, time response plots, power spectrum, bode plot, and pole-zero locations. Each of these tools will be shown in turn below.

The model power spectrum , pole-zero plot, and eigenvalues of the 6$^{th}$ order prediction-error model are shown in Figure 3-10, Figure 3-11, and Table 19. While the power spectrum of the model and combustor do have similar profiles, the model magnitude is much lower than the combustor. This could imply that the correlation between the output magnitude and the input is not strong enough. This could be caused by the input magnitude not being large enough to influence the output pressure, or the input is small or similar in magnitude compared to other system disturbances. Another cause could be the presence of significant nonlinear dynamics.

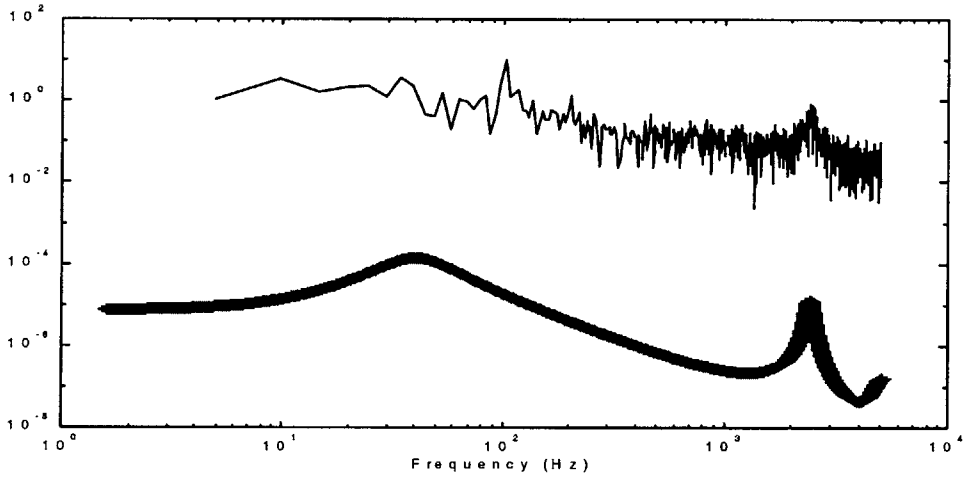



Figure 3-10: Open loop identification. Thin-line is combustor frequency spectrum; thick-line is the 6$^{th}$ order prediction error system identification model frequency spectrum.

The system identification tools placed poles at 41, 2400 , and 5000 Hz. The dominant unstable 95 Hz dynamics are not identified uniquely by this model. The tools instead identified significant dynamics between the 20 and 95 Hz of the combustor.

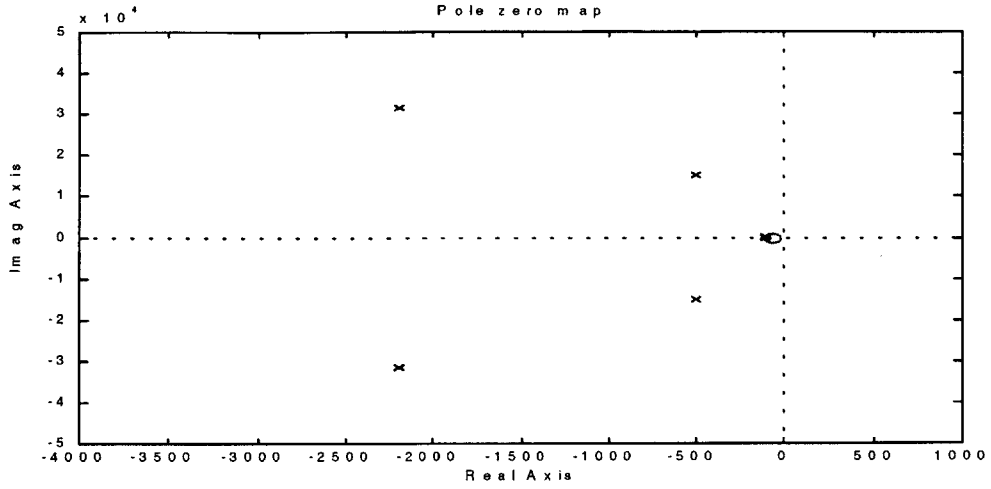



Figure 3-11: 6th order prediction error model poles and zeros. Poles are shown as x's and zeros are shown as 0's.

| Eigenvalues | Damping | Freq. (Hz) |
|---|---|---|
| -1.10e+002 +/- 2.35e+002i | 4.23e-001 | 41.2 |
| -5.02e+002 +/- 1.51e+004i | 3.31e-002 | 2403 |
| -2.20e+003 +/- 3.14e+004i | 6.99e-002 | 5013 |

Table 19: 6th order prediction-error model eigenvalues.

The corresponding single-input single-output transfer function is

( 39 )

$$\frac{Y(s)}{U(s)} = \frac{4.24*10^{-1}S^5 + 1.06*10^4 S^4 + 1.24*10^9 S^3 + 4.41*10^{13} S^2 + 5.65*10^{17} S + 4.15*10^{19}}{S^6 + 5.62*10^3 S^5 + 1.23*10^9 S^4 + 2.27*10^{12} S^3 + 2.28*10^{17} S^2 + 5.0*10^{19} S + 1.53*10^{22}}.$$

In order to ensure a good test of the model, its response will be evaluated on a fresh data set, known as the validation data. Such a procedure is known as *cross-validation.* One tool for looking at the goodness of fit of the model is to determine if the model agrees with the observed data by looking at the differences between the new data set and the model output. Figure 3-12 compares the pressure response from the combustor versus the model output. While the response of the model is quite similar and contains a similar high frequency component, it is delayed by approximately .01 seconds. The system identification modeled delay was reduced to a single update of .0001 seconds in an effort to reduce this model output delay.

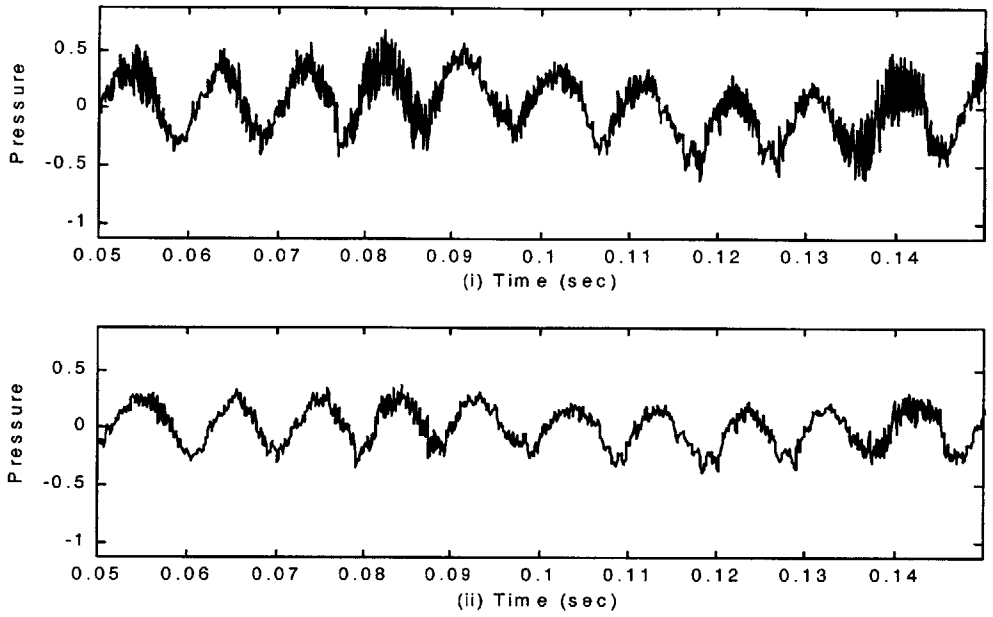



Figure 3-12: Time response: (i) combustor pressure response, (ii) 20-step ahead simulation of the system identification model.

The difference between the combustor response and the model response can be quantified by looking at the residual,

$$\varepsilon(t) = y(t) - \hat{y}_k(t \mid m), \tag{40}$$

where $\hat{y}_k(t \mid m)$ is the k-step ahead predicted value given the model $m$, and $y(t)$ is the combustor pressure response. It is also informative to look at the basic statistics of the residuals as shown in Table 20.

| $S_1$ | $S_2$ | $J_k$ | $J_y$ | $R$ |
|---|---|---|---|---|
| 0.20 | 0.616 | 0.058 | 0.064 | 32% |

Table 20: $6^{th}$ order prediction-error model validation statistics.

Where $S_1$, $S_2$, $J_k$, $J_y$, and $R$ are defined by equations ( 41 ) through ( 45 ).

$$S_1 = \frac{1}{N}\sum_{t=1}^{N}\varepsilon(t), \tag{41}$$

$$S_2 = \max_t\left|\varepsilon(t)\right|, \tag{42}$$

( 43 ) $$J_k(m) = \frac{1}{N}\sum_{t=1}^{N}\left|\varepsilon(t)\right|^2 ,$$

( 44 ) $$J_y(m) = \frac{1}{N}\sum_{t=1}^{N} y^2(t) .$$

Equations ( 43 ) and ( 44 ) are combined to get a normalized measure of the model fit

( 45 ) $$R = \sqrt{\left(1 - \frac{J_k}{J_y}\right)} * 100 ,$$

where $R$ is the part of the output explained by the model. The R value of 32% shows that the model is only explaining 1/3 of the dynamic behavior. While this R value is low, this model is the best fit of all of the models investigated. This model will be used to create a model based controller.

## 3.5 LQG Control Development

For closed loop control, we use the dynamic model created by the system identification methods and create model-based controls. The linear quadratic Gaussian (LQG) optimal control will be used to control the combustor. LQG poses to create a control input that brings the initial state of the system to a designated final state by minimizing a performance index. LQG implements a state observer to obtain full system state information. The performance index is setup to weight the control or observer as the designers see appropriate. A brief summary is given below. For more details refer to [ 17 ] through [ 19 ].

Given the linear system model:

$$\begin{aligned} \dot{x} &= Ax(t) + Bu(t) \\ y &= Cx(t) \end{aligned} \tag{46}$$

where x is the n-component state vector, u is the control vector, y is the system output. The control input and the desired final state is obtained by minimizing the performance index made up of quadratic terms in the terminal state plus an integral of quadratic terms in the state and control:

$$J(t_0) = \frac{1}{2}x^T(t_f)S(t_f)x(t_f) + \frac{1}{2}\int_{t_0}^{t_f}(x^TQx + u^TRu)dt\,, \tag{47}$$

where Q and R represent matrices that weight the various output and inputs appropriately, and $x(t_f) = 0$ is the final state of the system. Since the states of the model derived from the system identification methods may or may not have physical significance, it is difficult to weigh the influence of one state differently than another. One choice is to set $Q = I$ and $R = \rho I$ so that $\rho$ is a scaling factor that dictates the relative effort between faster transients and the magnitude of the control signal input. The solution to get the controller $K$ is of the form [ 17 ], [ 18 ]

$$K = R^{-1}B^TP\,, \tag{48}$$

where $P$ is the solution of the Riccati equation

$$A^TP + PA + C^TC - PBR^{-1}B^TP = 0\,. \tag{49}$$

The design of the observer of the following form is required for LQG

$$\dot{\hat{x}} = A\hat{x} + Bu + H(y - C\hat{x})$$
$$u = -K\hat{x} \tag{50}$$

where $\hat{x}$ is the estimated state of the system, and $H$ is the feedback gain matrix. $H$ can be found in a similar fashion as $K$ by posing the problem as the design of a Kalman filter. The solution is of the form [ 16 ], [ 17 ]

$$H = R_f^{-1} C^T P_f \tag{51}$$

where $P_f$ is the solution of

$$P_f A^T + A P_f + B B^T - P_f C^T R_f^{-1} C = 0. \tag{52}$$

Similar to the control weighting, choose $R_f = \mu I$ .

The state space block diagram of the system with the control and observer is shown in Figure 3-13.

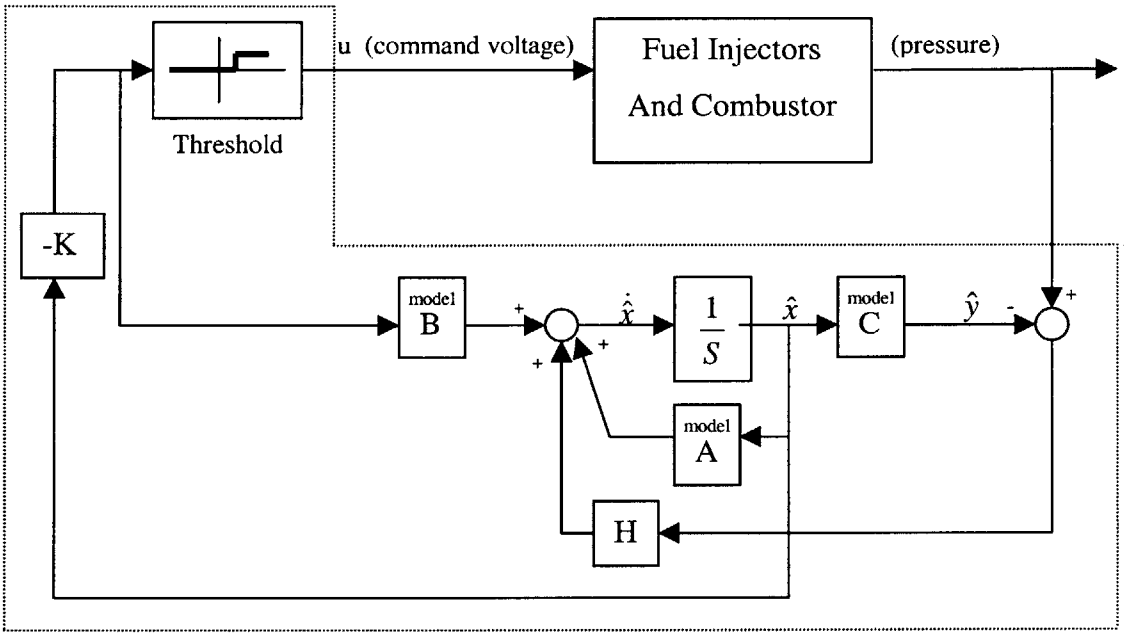



Figure 3-13: LQG state space block diagram. Dotted line surrounds LQG controller in the PC.

Where model A, B, and C came from the system identification model developed in section 3.4. The control design variables are K and H. The use of a mathematical computer program such as MATLAB® can aid in the computation of K and H efficiently by choosing $\mu$ and $\rho$ to allow fast transients without unrealistic control effort (smaller $\mu$ equals a faster observer, and smaller $\rho$

equals more control effort). With the control and feedback gain matrix determined, the LQG can be defined in a single state space representation

$$\dot{\hat{x}} = A\hat{x} + Bu + H[y - C\hat{x}] \qquad (53)$$
$$u = -K\hat{x}$$

or

$$\dot{\hat{x}} = (A - BK - HC)\hat{x} + Hy \qquad (54)$$
$$u = -K\hat{x}$$

The output of the control defined in equation ( 54 ) is u. This output is a continuous signal based off of the linear control. The fuel injectors are pulsed, therefore this control needs to be converted into a binary signal. This is accomplished by comparing the output to a threshold. If the value of the output is greater than the threshold then the command voltage takes on a high value. If the value of the output is less than the threshold then the command voltage takes on a low value.

## 3.6 LQG control based on open loop 6th order prediction-error model

The LQG control developed in section 3.5 based on the system identification model from section 3.4 is now implemented, using the PC, on the combustor rig. The control input, filtered and unfiltered pressure responses are shown in Figure 3-14.

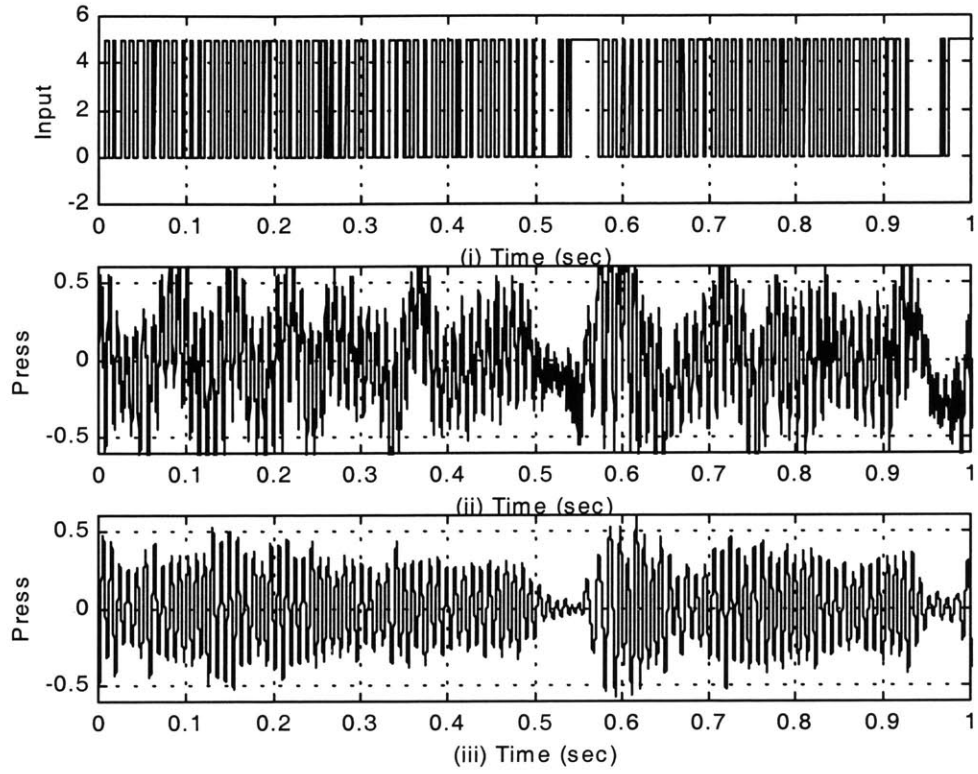



Figure 3-14: (i) LQG control input based on 6th order prediction-error model, (ii) unfiltered combustor response, (iii) combustor response bandpass filtered between 40 – 140 Hz.

This controller is able to stabilize the pressure for a short period of time. This stabilization is immediately followed by a very rapid growth region. This ability to control the pressure response at one period of time but not at another shows either that the underlying combustor dynamics are changing or that the oscillations are being driven by an external force. This same phenomenon is displayed on the closed loop controller and is analyzed further in section 3.8. Table 21 shows the resulting pressure oscillations with the LQG control exhibiting an RMS pressure value of 0.23 and a max absolute value of 0.65. While this controller displays minimum oscillations that are smaller than the lead lag controller, the lead lag still performs better on the average.

| $\mu$ / $\rho$ | Avg Peak (psi) | Max Peak (psi) | Min Peak (psi) | Reduction from Avg | Max Reduction | RMS (psi) |
|---|---|---|---|---|---|---|
| 6e-4 / 5e-7 | 0.325 | 0.65 | 0.07 | 78% | 89% | 0.23 |

Table 21: LQG control based on open loop 6$^{th}$ order prediction-error model transient summary.

## 3.7 System Identification and Validation: Closed loop

The goal of the system identification is to obtain a 'good' model of the unstable combustor dynamics. The combustor must be moved from the marginally stable or nonlinear region of Figure 1-3 to identify the unstable poles of the combustor. This is accomplished by inputting a system identification signal into a controlled closed loop system as shown in Figure 3-15. The direct system identification method is used by taking U(t) as the input and Y(t) as the output data. The LQG algorithm from section 3.6 is used as a partially stabilizing control. The identification input developed in section 3.3.1 is used as the identification input to the system as R(t) in Figure 3-15.

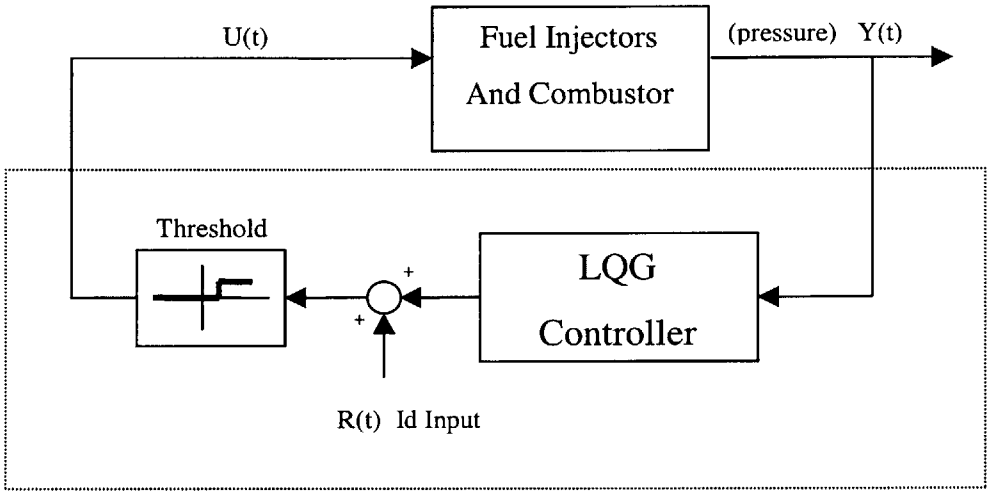



Figure 3-15: Block diagram of closed loop system with identification input. Dotted line encloses the processing in the computer.

### 3.7.1 Closed Loop Identification of 7th order sub-space model

Using U(t) and Y(t) several model structures, model orders, and estimation methods are examined to find a good fit to the data. Many of the models place dynamic elements at frequencies higher and lower than the range of interest. In an effort to have the estimation methods focus more on the frequency range of interest, between 20 and 300 Hz, the pressure data Y(t) is bandpass filtered before being used for identification. Finally, a 7th order state space model of the form shown in equation ( 35 ) is developed using sub-space methods (see [ 26 ] for details on the sub-space identification algorithm).

### 3.7.2 Closed Loop model validation

The closed loop system identification did identify unstable dynamics. As shown in Figure 3-16 and Table 22 the system identification model determined that the unstable dynamic is represented by a pole on the positive real axis at 341 radians or 54.3 Hz.

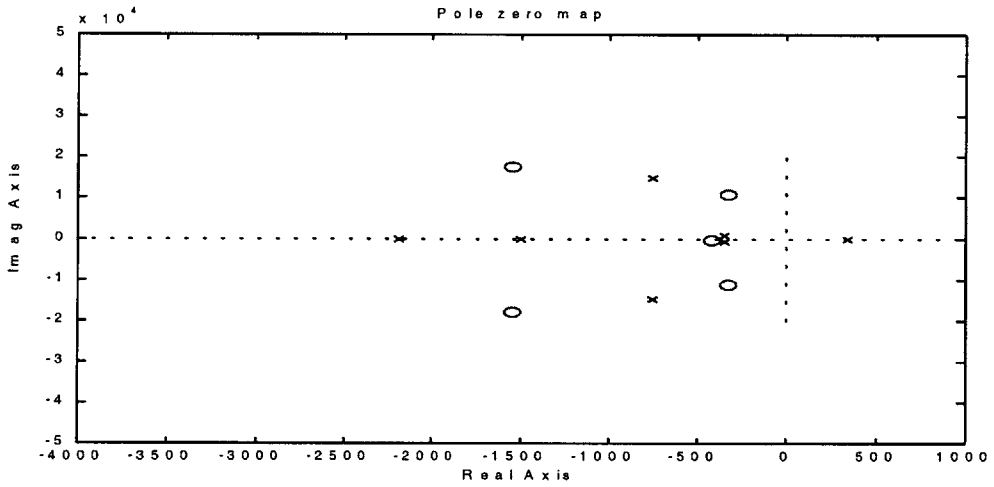



Figure 3-16: 7th order sub-space model poles and zeros. Poles are shown as x's and zeros are shown as 0's.

| Eigenvalues | Damping | Freq. (Hz) |
|---|---|---|
| 3.41e+002 | --- | 54.3 |
| -3.46e+002 +/- 7.57e+002i | 4.16e-001 | 132.6 |
| -1.49e+003 | --- | 237 |

| | | |
|---|---|---|
| -2.19e+003 | --- | 348 |
| -7.58e+002 +/- 1.49e+004i | 5.07e-002 | 2371 |

Table 22: 7th order sub-space model eigenvalues.

The corresponding single-input single-output transfer function is

( 55 )

$$\frac{Y(s)}{U(s)} = \frac{-4.23*10^{-6}S^7 - 0.194S^6 - 1.576*10^3 S^5 - 4.53*10^7 S^4 + 7.64*10^{10} S^3 - 3.475*10^{15} S^2 + 2.66*10^{19} S + 1.2*10^{22}}{S^7 + 5.55*10^3 S^6 + 2.34*10^8 S^5 + 9.1*10^{11} S^4 + 1.1*10^{15} S^3 + 5.8*10^{17} S^2 + 1.37*10^{20} S - 1.73*10^{23}}.$$

The power spectrum of the 7th order model versus the combustor pressure data is given in Figure 3-17. The figure shows that even after bandpass filtering the pressure, the model still does not fit to the data power spectrum well between 20 and 300 Hz.

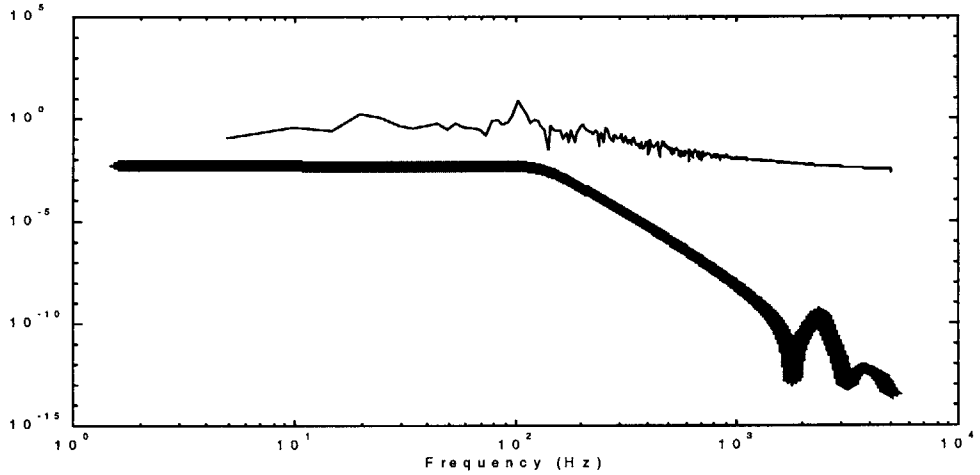



Figure 3-17: Closed loop identification. Thin-line is combustor frequency spectrum; thick-line is the 7th order sub-space system identification model frequency spectrum.

Cross-validation is used to determine if the model agrees with the observed data by looking at the differences between a new data set and the model output. While the power spectrum does not fit the experimental data well, the time simulation of the model recreated 88% of the dynamic characteristics of the validation data. The simulation output versus the combustor pressure response is shown in Figure 3-18

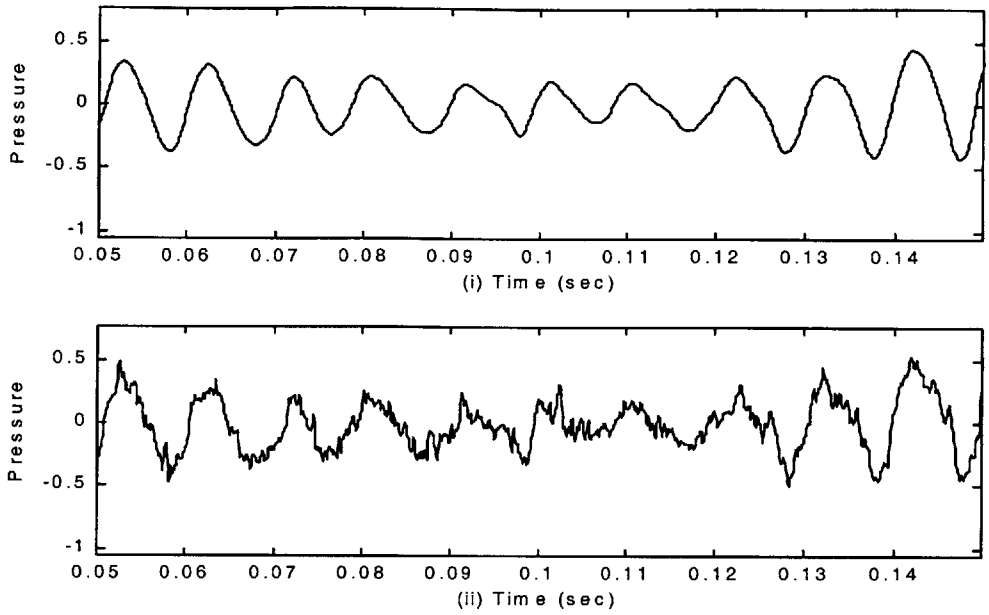



Figure 3-18: Time response: (i) combustor pressure response, (ii) 20-step ahead simulation of the system identification model.

It is also informative to look at the basics statistics of the residuals as shown in Table 23. By all invariant residual measures the closed loop model is much improved over the open loop model given above. The average residual, $S_1$, is reduced by a factor of 3, the maximum residual, $S_2$, is down by 30%, and the overall fit, R, is greater than 2 times better. Using this model, another LQG controller is developed below.

| $S_1$ | $S_2$ | $J_k$ | $J_y$ | $R$ |
|---|---|---|---|---|
| 0.07 | 0.42 | 0.009 | 0.039 | 88% |

Table 23: $7^{th}$ order sub-space model validation statistics.

## 3.8 LQG control based on 7th order sub-space model

The LQG control developed in section 3.5 based on the system identification model from section 3.7 is now implemented on the combustor rig. The c-code used to implement the LQG control on the combustor rig is shown in Appendix A. For this LQG controller, defined by $\mu$ equal to 1e-4 and $\rho$ equal to 5e-7 (see section 3.5), the minimum peak values are reduced by a factor of 4 over the Wavetek™ control. The RMS pressure is reduced by 10% compared to the Wavetek™ and 15% compared to the open loop controller. In Table 24 more extensive comparisons between the model-based controllers and the Wavetek™ control are made. The improvements are several fold: the average peak value is reduced by 11%, the maximum peak value is reduced by 4%, and the maximum reduction is improved by 15.5%.

| Case | Avg Peak (psi) | Max Peak (psi) | Min Peak (psi) | Reduction from Avg | Max Reduction | RMS (psi) |
|---|---|---|---|---|---|---|
| 1 | 0.31 | 0.58 | 0.12 | 61% | 79% | 0.22 |
| 2 | 0.325 | 0.65 | 0.07 | 78% | 89% | 0.23 |
| 3 | 0.28 | 0.56 | 0.031 | 89% | 94.5% | 0.20 |

Table 24: Experimental combustor transient summary. Case 1: Wavetek™ control with 300 degrees of phase. Case 2: LQG based on open loop 6th order prediction-error model. Case 3: LQG based on final 7th order sub-space model.

## 3.9 Turbulent Combustor Experimental Study Discussion and Conclusions

This experimental study illustrated the viability of using system identification to create a model for active control of thermoacoustic instabilities. While the LQG control performed better than the Wavetek™, the performance numbers reported in Table 24 could be improved upon. Speculation on the possible ways in which this can be achieved are discussed below..

The first comment is regarding the changing dynamics of the combustor when in closed loop. As can be seen over the first 0 to 1 second the model-based controller is quite successful in bringing the pressure response down, see Figure 3-19. However, the pressure then rises, only to be reduced again ( see 0.4 – 0.7 seconds). This pattern was repeated over the entire experimental run. The question that arises is what the likely cause is for such a dynamic behavior. This ability to control the pressure response at certain periods and not at others suggests the presence of combustion dynamics at frequencies other than at the unstable one (~95 Hz).

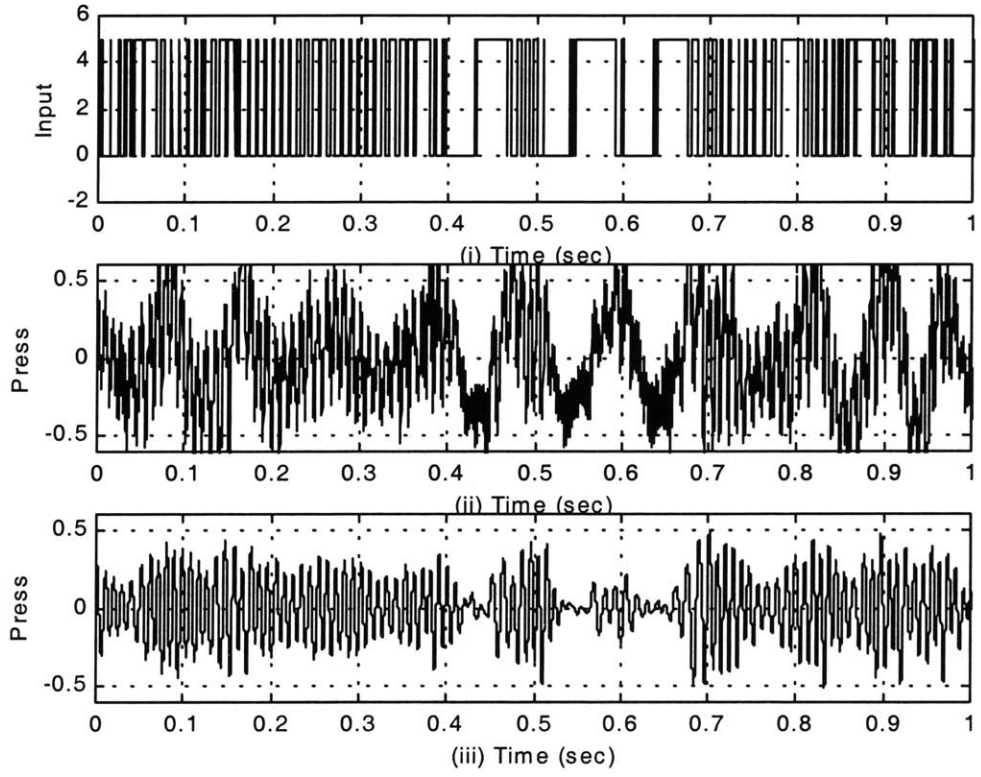



Figure 3-19: (i) LQG control input based on 7$^{th}$ order sub-space model, (ii) unfiltered combustor response, (iii) combustor response bandpass filtered between 40 – 140 Hz.

Combustion dynamics at frequencies other than at the unstable one are present. There are two characteristics in Figure 3-21 and Figure 3-22 to be noted. The first is the unmistakable presence of a mode at 10 Hz. This is likely be due to a bulk-mode in the combustor. The bulk mode may be dormant during open loop operations and gets excited due to a closed loop action. The second is the presence of multiple peaks 10 Hz apart between 10 and 100 Hz. The periodicity in these peaks suggests the presence of a time delay, possibly due to the burning of the control fuel addition. Figure 3-20 shows the baseline 4 injector open loop and the Wavetek™ control frequency spectrum for comparative purposes.

Both of the identification models presented only captured dynamics near the unstable frequency and did not identify the bulk mode and 95 Hz frequencies independently. In addition, neither of the models incorporated any delays larger than .0002 seconds. Using appropriate system identification procedures that accommodates these two characteristics it is quite likely to lead to a significant improvement in the pressure response. The final experiment incorporating these changes was not accomplished due to time limitations.

During the experiment with the final LQG control, the bulk mode and the unstable frequency change characteristics. Just before stabilization the 95 Hz frequency is dominant, then after stabilization the bulk mode becomes dominant. Figure 3-21(i) shows that before stabilization the 100 Hz instability is very prevalent with only a small amount of energy at 10 Hz. During rapid pressure growth Figure 3-21 (ii) shows the lower frequency mode increases by greater than a factor of 3, and the single frequency at 100 Hz is reduced and split into a pair of frequencies at 100 and 110 Hz. This behavior repeats itself as shown in Figure 3-22 and in other experiments with this controller. This shows that the combustor dynamics are shifting or changing after becoming stabilized. One cause of the stabilization and then regrowth may simply be longer on-off times during stabilization that excite a lower frequency dynamic mode of the combustor. This would take energy away from the 100 Hz instability and change the combustor characteristics.

Another factor that may be changing the combustor operation is the equivalence ratio. Figure 3-23 (i) shows that the control pulses on and off for different durations of time, not at a steady 50% duty cycle like the 4 injector open loop command of section 3.2.1 and the Wavetek™ control of section 3.2.2. Figure 3-23 (ii) shows some equivalence ratio calculations. The thick line shows the equivalence ratio for 4 injectors pulsing at 50% duty cycle. The thin line shows the

predicted equivalence ratio of the combustor assuming a constant burn rate, 2 injectors are pulsing at 50% duty cycle, and 2 injectors supply fuel proportionally to the on-off control command. Under LQG control the combustor is operating at a lower equivalence ratio than the Wavetek™ or the 4 injector open loop case. Also, large positive and negative slopes in Figure 3-23 (ii), which correspond to long on and off times in Figure 3-23 (i), show that during stabilized periods the controller is quiet and either keeps fuel flow off or on for long time periods. These fuel flow changes affect the burning and create locally lean and rich spots. These will cause a change in the burning characteristics and thus change the combustor instability dynamics.

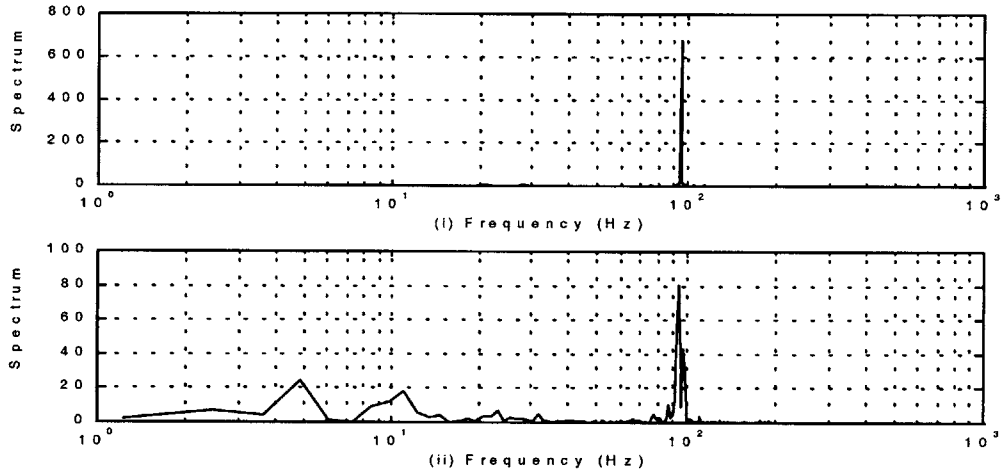



Figure 3-20: (i) Spectrum of the baseline 4 injector open loop case, (ii) spectrum of the Wavetek™ control at with 300 degrees phase lead.

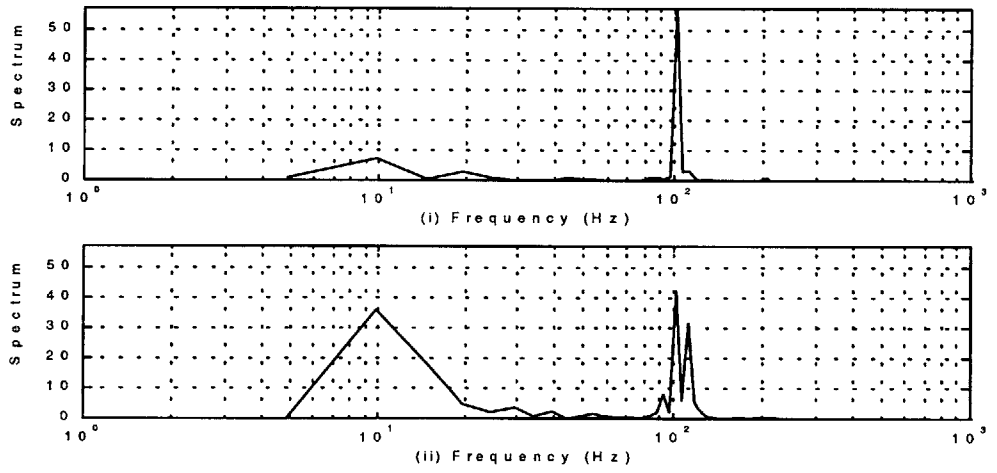



Figure 3-21: (i) Spectrum from 0 to 0.4 seconds of unfiltered combustor response to 7$^{th}$ order sub-space model, (ii) spectrum from 0.7 to 1.1 seconds of unfiltered combustor response to 7$^{th}$ order sub-space model.

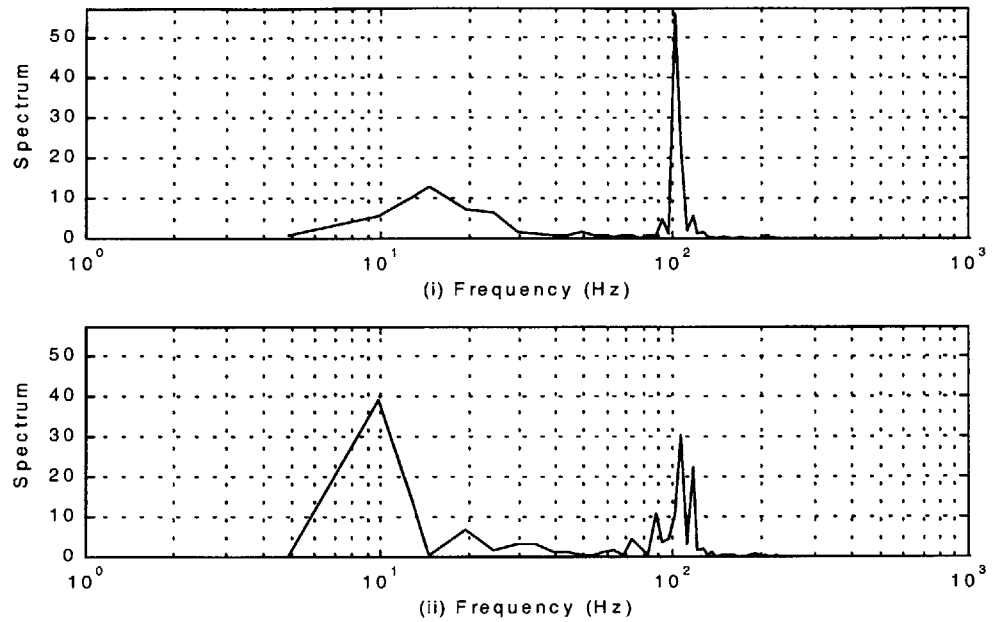



Figure 3-22: (i) Spectrum from 2.5 to 2.7 seconds of unfiltered combustor response to 7$^{th}$ order sub-space model, (ii) spectrum from 2.8 to 3.0 seconds of unfiltered combustor response to 7$^{th}$ order sub-space model.

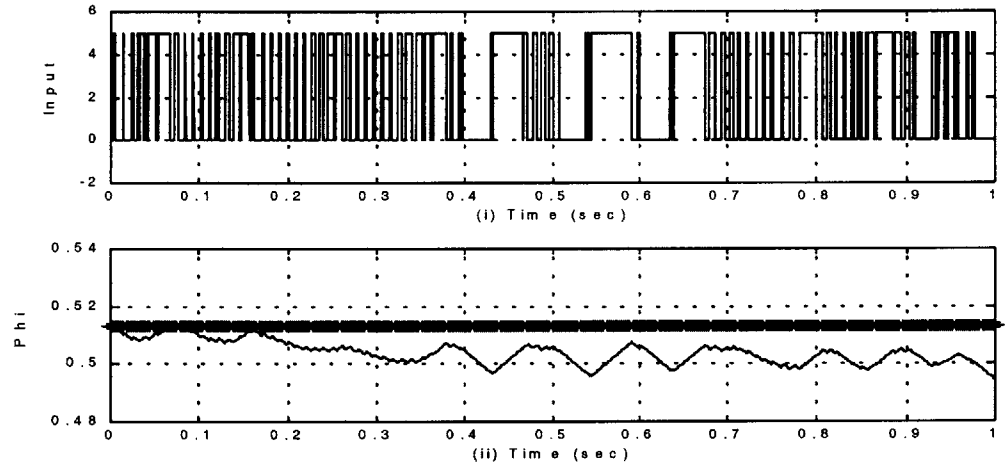



Figure 3-23: (i) LQG control input, (ii) equivalence ratio. Thick-line: 4 injector 50% duty cycle steady state equivalence ratio. Thin-line: calculated equivalence ratio assuming constant fuel burn rate.

Additional complexities are also present due to experimental constraints. The first is the presence of an external forcing function. Because this experiment is operated with two injectors always pulsing at the combustor instability frequency, they act like continuous forcing functions trying to destabilize any external control. In most industrial combustors the instability is

homogeneous or just a function of the dynamics between the heat release and acoustics without any forcing functions.

Another complicating factor during the experiment is that the sample period changes by up to 15% depending on which program is running. This results in an inability to be able to see if the combustor was shifting frequency, and its effects on the controller stability. These problems can be reconciled by sampling more slowly, using a faster processor, or a better I/O board.

While the linear LQG controller displayed the ability to have a stabilizing effect on the combustor it is designed to work on a proportional actuator not an on-off actuator. Because the linear output of the LQG is put through a threshold much of the control information is lost. It would be better to pose the controller in a form that is compatible with this binary configuration. One recommendation is to use the 'Bang-Bang' controller, see section 2.6.3 and [ 18 ] through [ 21 ].

# 4. Discussion and Conclusions

Active control of combustors will become a critical part of combustor technology as industry tends to large, low emission, high volumetric heat release combustors. A robust control design method is required to ensure that the instabilities will be suppressed over a range of operating conditions.

The numerical studies illustrated many system identification techniques. An easily tunable control for stabilizing the pressure to get closed loop data was presented in detail. In closed loop system identification studies, the relative magnitude between the system identification input and the control effort was shown to be important for obtaining informative data. Pre-processing of the identification data by bandpass filtering before using system identification tools proved to increase the quality of the model. Linear LQG controllers were successful in quickly suppressing the instability, despite the nonlinearities in the system. A 'Bang-Bang' controller proved to provide the best steady state performance.

The experimental studies outlined in this thesis illustrate the viability of using system identification for model-based active control of thermoacoustic instabilities on large scale turbulent combustors. Investigations on the experimental combustor showed that using the LQG control based on $7^{th}$ order sub-space model proved to reduce the minimum peak values of the pressure by a factor of 4 over the experimentally derived controller. Because the system identification model only created dynamics near the unstable frequency and did not create separate dynamics for the bulk mode and 95 Hz frequencies separately, lower frequencies besides the 95 Hz instability became prevalent in the pressure response. Data also indicated the presence of delays in the combustor dynamics. If a system identification model was obtained that separated out the bulk mode and the unstable frequency, and included provisions for the delay, then the model based control would reduce the pressure more effectively.

One other significant advantage to using the system identification method is to aide in pinpointing the physical mechanism that may be responsible for the combustion dynamic behavior thereby giving clues for obtaining physically based models. System identification methods were used in [ 31 ] to verify the system dynamics and validate system models. This method of gaining insight will become more prevalent as model-based control of thermoacoustic instabilities gains popularity.

All of the results presented in this thesis indicate the advantage of using system identification for active control of thermoacoustic instabilities. The next steps for the experimental combustor used in this thesis is to develop a model of the 'off the shelf' pulsed liquid fuel injectors, incorporate the delay in the system identification model, and implement the control on a more robust I/O board. If a pulsed fuel injector continues to be used, then formulation of a 'Bang-Bang' controller for higher order systems will optimize the control for steady state operation.

Careful expansion of the system identification models and methods illustrated in this thesis will lead to the success of model-based control design on combustion systems for power generation and propulsion applications.

# Appendix A: C-code for LQG control of experimental combustor

```
/*************************************************************************
 Filename: c:\thesis\chinalake\c_code\rbcl_s7.cpp

 Purpose:
 This program is used to acquire data using das1800 board,
 and is used to create the LQG control reference for the China Lake combustor.

 General Operation:
 First run with a constant frequency for as long as operator desires.  This allows the
 operator to run 2 injectors open loop at the unstable frequency.
 Then, by hitting any key, start data recording for 0.5 seconds at steady frequency to capture
 Combustor behavior before control starts.
 Then implement tunable LQG control for 5.5 seconds.
 The total recording time will be 6 seconds.

 The user can choose to input system id input at 1.5 seconds.
 The id input is a random number between -1 and 1 that is bandpass filtered.
 The filter is a butterworth band pass from 25 to 270 Hz.

 Revisions:
 13-Oct-99 B Brunell Copied from inject4.cpp
 13-Oct-99 B Brunell Copied from id_freq.cpp
 13-0ct-99 B Brunell Copied from disc_frq.cpp
 16-Oct-99 B Brunell copied from rand_frq2.cpp
                    Changed the random signal to be filtered by bandpass  6th order Butterworth filter first
 14-Nov-99 B Brunell Copied from c:\thesis\ken_yu\programs_10_99\rand_fq4.cpp
                     added tunable (0 to 360 deg) phase lead control
 17-Nov-99 B Brunell Copied from c_code\lead.cpp
                     Changed from lead control to LQG based on 4th order ss
 17-Nov-99 B Brunell Copied from c_code\disfr_s5.cpp
                      Changed to LQG based on 5th order armax model based
                      off of rand_bp1 data
 17-Nov-99 B Brunell Copied from rbp1_mx5.cpp
                     Changed output voltage to be ~ 8volts vs real signal
 17-Nov-99 B Brunell copied from rb1_m08.cpp
                     Changed control to be based on 5th order SS from  id_ol_rndbp1_filt_s5.m
 17-Nov-99 B Brunell Copied from rb1_s5.m
                      Added 20Hz to 270 Hz filter before the LQG control
 18-Nov-99 B Brunell Copied from rb1_s5f.cpp
                     Changed control to be based on 5th Order SS model from id_cl_rb1s98_f_s5.m
 18-Nov-99 B Brunell Changed control to be based on 7th Order SS model from  id_cl_rb1s98_f_s7.m (unstable)

  Note: see subroutine "GO" for system id routines
*************************************************************************/

/* C INCLUDE FILES */
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>
#include <math.h>
#include <alloc.h>
#include <process.h>
#include <time.h>     /* required by the randomize function */
```

```
#include "dasio.h"

/* DEFINE GLOBAL VARIABLES TO BE USED IN ISR */

typedef unsigned long int uint;

uint points,spoint;  /* NUMBER OF DATA POINTS TO ACQUIRE */
float AD_BitValue;                              /* A/D BIT VALUE */
float DA_BitValue;                              /* D/A BIT VALUE */
unsigned int Samp_Rate;                         /* Sampling rate in KHz*/
unsigned cont1, cont2;        /* Variable used for setting clock */
unsigned  long int noisedelay, noisestart; /* noise generation params (jayesh)*/
float noiseamp;
uint itmpa; /* temporary variable to save time */

/* LQG Control Algorithm Variables */
float Kl;
float x1, x2, x3, x4, x5, x6, x7;
float x1_n, x2_n, x3_n, x4_n, x5_n, x6_n, x7_n;
float a11, a12, a13, a14, a15, a16, a17;
float a21, a22, a23, a24, a25, a26, a27;
float a31, a32, a33, a34, a35, a36, a37;
float a41, a42, a43, a44, a45, a46, a47;
float a51, a52, a53, a54, a55, a56, a57;
float a61, a62, a63, a64, a65, a66, a67;
float a71, a72, a73, a74, a75, a76, a77;
float b1, b2, b3, b4, b5, b6, b7;
float c1, c2, c3, c4, c5, c6, c7;
float d;

float tinvolt, trefvolt;  /* Input/output before we start storing */
float *involt, *refvolt;
uint nu,ny,nx;
FILE *fp;

void main (void)
{
        COUNTS Counts;

        clrscr();
        randomize();
        /* Calculate A/D and D/A Bit Values */
        AD_BitValue = (float)(10.0/4096);
        DA_BitValue = (float)(20.0/4096);
        /* Inform user of code and test configuration */
        gotoxy (14,2);
        printf("China Lake Phase LEAD Control -- RBCL_S7 --\n");
        gotoxy (2,3);
        printf("R (id)   U   +--------+     +-----------+   Y (pressure)\n");
        printf(" ------>O------>| Thresh |----->| Combustor |------->      \n");
        printf("       ^     +--------+     +-----------+  |          \n");
        printf("       |                                   |          \n");
        printf("       |        +-----------+              |          \n");
        printf("        -----------| Control |<--------------          \n");
        printf("                   +-----------+                       \n");
```

```
        Samp_Rate = 10; /* Fixed sample rate kHz */
        Counts.byte = long (500/Samp_Rate);
        cont1 = Counts.bits.count1;
        cont2 = Counts.bits.count2;
        gotoxy (2,8);

        reset();
        set_time();
        set_up_AD();
        set_up_DA();
        GO();
        reset();

}

void set_up_AD(void)
{
CONT_REG_C_REG cont_reg_c_reg;

        outp(AD_STATUS_REG, 0x00);

        /* Initialization of QRAM for A/D reading(Ch0 and 1)  */
        outp(AD_SELECT_REG,0X01);       /* Set Data Select to QRAM  */
        outp(QRAM_ADDR, 0X00);    /* Initialize QRAM for CH0  */
        outpw(QRAM_DATA,0X0000);     /* Set the CH0 gain to 1  */
        outp(QRAM_ADDR, 0X00);     /* Reinitialize QRAM  */

        /* SET UP A/D CONVERSIONS IN INTERUPT MODE */
        /* SET UP AS BIPOLAR, DIFFERENTIAL, DISABLE BURST MODE, INTERNAL PACER CLOCK */
        cont_reg_c_reg.bits.UB = 0;
        cont_reg_c_reg.bits.SD = 0;
        cont_reg_c_reg.bits.UQEN = 1;
        cont_reg_c_reg.bits.CMEN = 0;
        cont_reg_c_reg.bits.BMDE = 0;
        cont_reg_c_reg.bits.S0 = 0;
        cont_reg_c_reg.bits.S1 = 0;
        outp(CONT_REG_C, cont_reg_c_reg.byte);

        /* Set Control Register B for interupt when FIFO Not Empty */
        outp(CONT_REG_B, 0x40);
        outp(CONT_REG_A,0X03);    /* Enable A/D FIFO  */
        outp(AD_SELECT_REG,0x00);
        outp(AD_STATUS_REG, 0x80);       /* Enable A/D Conversions */
        outp(CONT_REG_B, 0xc0);
        outp(CONT_REG_A,0X05);
}

void set_up_DA(void)
{
        DA_CONT_C_REG da_cont_c_reg;
        DA_SELECT_REGS da_select_regs;

        /* Reset D/A */
        outp(DA_CONT_A, 0x00);
        /* Set up operation: Gain of 1, internal Software Clock */
```

```
        da_cont_c_reg.bits.GN0 = 1;
        da_cont_c_reg.bits.S1 = 0;
        da_cont_c_reg.bits.S0 = 0;
        outp(DA_CONT_C, da_cont_c_reg.byte);
        /* Enable D/A FIFO */
        outp(DA_CONT_A, 0x01);
        /* Select DAC 0 for output */
        da_select_regs.bits.DSL1 = 0;
        da_select_regs.bits.DSL0 = 0;
        outp(DA_SELECT_REG, da_select_regs.byte);
         /* Enable D/A Conversions */
        outp(DA_STATUS_REG, 0x80);

        /* PRELOAD ZEROS TO CHANNEL 0 */
        outp(DA_SELECT_REG, 0x00);      /* CHOOSE DA CHANNEL 0 */
        outpw(DA_OUT, 0x0000);                          /* LOAD ZEROS */
}


void set_time (void)
{

        outp(CONT_REG_A, 0x00);
        outp(COUNTER_CLR, 0xb4);
        outp(COUNTER_2, cont1);
        outp(COUNTER_2, cont2);
        outp(COUNTER_CLR, 0x74);
        outp(COUNTER_1, 0x0a);
        outp(COUNTER_1, 0x00);
        outp(CONT_REG_A, 0x04);
        outp(CONT_REG_B, 0x80);
}


void reset(void)
{

        /* RESET OUTPUT TO 0 */

        outpw(DA_OUT, 0x0000);
        outp(DA_STATUS_REG, 0x60);
        outpw(DA_OUT, 0x0000);
        outp(DA_STATUS_REG, 0x60);
        outp(AD_STATUS_REG, 0x00);      /* DISABLE A/D CONVERSION */
        outp(CONT_REG_A, 0x00);         /* DISABLE A/D FIFO */
        outp(DA_STATUS_REG, 0x00);      /* DISABLE D/A CONVERSION */
        outp(DA_CONT_A, 0x00);                          /* RESET D/A FIFO */
}


 void GO (void)
{
        int inp1, outda,wflag=0, idflag, const_fr_flag;
        int period_end_switch=0;
        int out_amp_counts;
```

```
unsigned long int i, j=0, const_freq_cnt, id_start_count;
unsigned long int period_cnt, period_end;
unsigned long int total_samples;
unsigned int seed;
float base_samples_c, period_samples_c;
float past_samples, duty_factor, Gn;
float out_amp_volts, tin_filt;
float duty, thresh, ref_freq;
float tnoise, tnoise_filt;
/* Butterworth filter variable */
float xf1, xf2, xf3, xf4, xf5, xf6, xf1_n, xf2_n, xf3_n, xf4_n, xf5_n, xf6_n;
float af11, af12, af13, af14, af15, af16;
float af21, af22, af23, af24, af25, af26;
float af31, af32, af33, af34, af35, af36;
float af41, af42, af43, af44, af45, af46;
float af51, af52, af53, af54, af55, af56;
float af61, af62, af63, af64, af65, af66;
float bf1, bf2, bf3, bf4, bf5, bf6;
float cf1, cf2, cf3, cf4, cf5, cf6;
float df;
/* Butterworth filter variable for input signal */
float xp1, xp2, xp3, xp4, xp5, xp6, xp1_n, xp2_n, xp3_n, xp4_n, xp5_n, xp6_n;
float apf11, apf12, apf13, apf14, apf15, apf16;
float apf21, apf22, apf23, apf24, apf25, apf26;
float apf31, apf32, apf33, apf34, apf35, apf36;
float apf41, apf42, apf43, apf44, apf45, apf46;
float apf51, apf52, apf53, apf54, apf55, apf56;
float apf61, apf62, apf63, apf64, apf65, apf66;
float bpf1, bpf2, bpf3, bpf4, bpf5, bpf6;
float cpf1, cpf2, cpf3, cpf4, cpf5, cpf6;
float dpf;

char string[20];
FILE *out_file;
uint itmpr;

gotoxy (2,10);
printf("Do you want to save data (y/n) ?");
if (getch () == 'y') {
                        wflag = 1;
                        gotoxy (2,11);
                        printf("I will store 6 sec. of data.  Enter the file name : ");
                        scanf("%s", string);
                        out_file = fopen(string,"w");
}

printf("Please enter the combustor unstable frequency in (Hz): \n");
scanf("%f", &ref_freq);

out_amp_volts = 5.0;

printf("Do you want to run with id reference input at 1.5 sec (y/n) ?\n");
if (getch () == 'y') { idflag = 1; /* id input fla */ }

printf("\nTo start -  press any key     ");
```

```
        getch();

        /**** Set the storage times  ***/
        itmpr = 0; /* to save on data storage */
        total_samples = 60000; /* total # of samples to run */
        const_freq_cnt = 5000; /* # of samples to store const freq */
        if (idflag == 1) { id_start_count = 15000;  }
        else {          id_start_count = total_samples+1; }

        /**** set the threshold value to go from continuous signal to binary ***/
        thresh = 2.0;
        Gn    = 1.0;
        printf("\n\nThe binary threshold is %f ", thresh);
        printf(" the gain is %f \n", Gn);

        /**** Initialize Constant frequency Reference *****/
        const_fr_flag = 1; /* allow constant freq before storage */
        base_samples_c = (Samp_Rate*1000)/ref_freq; /* init # of samples in 1 cycle*/
        period_samples_c = base_samples_c; /* initialize period_samples_c */

        /****  Initialize Random Frequency reference *****/
        /* initialize random number generator */
        seed = 1;
        srand(seed);

        /*** common adjustments ***/
        out_amp_counts = (int)(out_amp_volts/DA_BitValue);/* conv output to counts */
        past_samples = 0; /* initialize past # of samples to 0 */
        period_cnt = 0; /* Define period counter for reference signal */
        duty_factor = 0.5; /* initialize 50% Duty Factor */

        /*** Initialize the Butterworth Filter for input signal ***/
        /** 20 to 270 Hz set */
        xp1 = 0; xp2 = 0; xp3 = 0; xp4 = 0; xp5 = 0; xp6 = 0;
        xp1_n = 0; xp2_n = 0; xp3_n = 0; xp4_n = 0; xp5_n = 0; xp6_n = 0;
        apf11 = 8.5309e-001;
apf12 = 0.0000e+000;
apf13 = 0.0000e+000;
apf14 = 4.2832e-002;
apf15 = 0.0000e+000;
apf16 = 0.0000e+000;
apf21 = 1.3443e-001;
apf22 = 8.4251e-001;
apf23 = -1.4501e-001;
apf24 = 3.1072e-003;
apf25 = 4.2587e-002;
apf26 = -3.3517e-003;
apf31 = 1.0580e-002;
apf32 = 1.4501e-001;
apf33 = 9.8752e-001;
apf34 = 2.4454e-004;
apf35 = 3.3517e-003;
apf36 = 4.5939e-002;
apf41 = -4.2832e-002;
apf42 = 0.0000e+000;
```

```
apf43 = 0.0000e+000;
apf44 = 9.9901e-001;
apf45 = 0.0000e+000;
apf46 = 0.0000e+000;
apf51 = -3.1072e-003;
apf52 = -4.2587e-002;
apf53 = 3.3517e-003;
apf54 = -7.1818e-005;
apf55 = 9.9902e-001;
apf56 = 7.7471e-005;
apf61 = -2.4454e-004;
apf62 = -3.3517e-003;
apf63 = -4.5939e-002;
apf64 = -5.6522e-006;
apf65 = -7.7471e-005;
apf66 = 9.9894e-001;
bpf1 = 2.0636e-001;
bpf2 = 1.4970e-002;
bpf3 = 1.1782e-003;
bpf4 = -4.7698e-003;
bpf5 = -3.4602e-004;
bpf6 = -2.7232e-005;
cpf1 = 3.7405e-003;
cpf2 = 5.1268e-002;
cpf3 = 7.0269e-001;
cpf4 = 8.6458e-005;
cpf5 = 1.1850e-003;
cpf6 = 1.6242e-002;
dpf = 4.1655e-004;

/*** Initialize the Butterworth Filter for Sys ID ***/
xf1 = 0; xf2 = 0; xf3 = 0; xf4 = 0; xf5 = 0; xf6 = 0;
xf1_n = 0; xf2_n = 0; xf3_n = 0; xf4_n = 0; xf5_n = 0; xf6_n = 0;
af11 = 8.5556e-001;
af12 = 0.0000e+000;
af13 = 0.0000e+000;
af14 = 4.7952e-002;
af15 = 0.0000e+000;
af16 = 0.0000e+000;
af21 = 1.3213e-001;
af22 = 8.4537e-001;
af23 = -1.4232e-001;
af24 = 3.4145e-003;
af25 = 4.7688e-002;
af26 = -3.6778e-003;
af31 = 1.0190e-002;
af32 = 1.4232e-001;
af33 = 9.8769e-001;
af34 = 2.6333e-004;
af35 = 3.6778e-003;
af36 = 5.1366e-002;
af41 = -4.7952e-002;
af42 = 0.0000e+000;
af43 = 0.0000e+000;
af44 = 9.9876e-001;
```

```
af45 = 0.0000e+000;
af46 = 0.0000e+000;
af51 = -3.4145e-003;
af52 = -4.7688e-002;
af53 = 3.6778e-003;
af54 = -8.8237e-005;
af55 = 9.9877e-001;
af56 = 9.5042e-005;
af61 = -2.6333e-004;
af62 = -3.6778e-003;
af63 = -5.1366e-002;
af64 = -6.8049e-006;
af65 = -9.5042e-005;
af66 = 9.9867e-001;
bf1 = 2.0251e-001;
bf2 = 1.4420e-002;
bf3 = 1.1121e-003;
bf4 = -5.2334e-003;
bf5 = -3.7265e-004;
bf6 = -2.8739e-005;
cf1 = 3.6027e-003;
cf2 = 5.0317e-002;
cf3 = 7.0275e-001;
cf4 = 9.3100e-005;
cf5 = 1.3003e-003;
cf6 = 1.8161e-002;
df = 3.9319e-004;

/*** Initialize the LQG Control ***/
Kl=1.0;
x1 = 0; x2 = 0; x3 = 0; x4 = 0; x5 = 0; x6 = 0; x7 = 0;
x1_n = 0; x2_n = 0; x3_n = 0; x4_n = 0; x5_n = 0; x6_n = 0; x7_n = 0;
a11 = 5.1351e-001;
a12 = 1.0973e-001;
a13 = 3.1718e-002;
a14 = 1.0168e-002;
a15 = -2.0672e-003;
a16 = 5.4757e-004;
a17 = 2.4088e-003;
a21 = -1.0663e+000;
a22 = 7.0059e-001;
a23 = -9.6075e-002;
a24 = 2.5906e-002;
a25 = -1.2895e-002;
a26 = 2.2206e-003;
a27 = 1.1384e-002;
a31 = 1.3568e+000;
a32 = 6.9150e-001;
a33 = 8.1348e-001;
a34 = -2.7000e-001;
a35 = 2.0421e-002;
a36 = -1.5875e-003;
a37 = -2.2671e-002;
a41 = -9.1350e-001;
a42 = -4.9088e-001;
```

```
  a43 = 1.0915e-001;
  a44 = 1.0116e+000;
  a45 = -1.5111e-001;
  a46 = 6.0037e-002;
  a47 = 9.5726e-002;
  a51 = -7.5637e-001;
  a52 = -4.8675e-001;
  a53 = 1.4179e-001;
  a54 = 8.9429e-004;
  a55 = 4.8856e-001;
  a56 = 7.0932e-001;
  a57 = -4.1462e-001;
  a61 = -3.5724e-001;
  a62 = -2.3901e-001;
  a63 = 5.3001e-002;
  a64 = -3.7974e-002;
  a65 = -4.7145e-001;
  a66 = -3.8810e-003;
  a67 = -8.4729e-001;
  a71 = 2.7783e-001;
  a72 = 2.3182e-001;
  a73 = -3.6829e-002;
  a74 = -9.2326e-003;
  a75 = -1.7969e-001;
  a76 = 6.1741e-001;
  a77 = 3.6801e-001;
  b1 = 2.9896e-002;
  b2 = 1.0320e-002;
  b3 = 2.8156e-002;
  b4 = -2.3921e-002;
  b5 = -3.0957e-002;
  b6 = -1.5353e-002;
  b7 = 1.7794e-002;
  c1 = -1.3375e+004;
  c2 = -4.1458e+003;
  c3 = 6.2366e+002;
  c4 = 2.2220e+002;
  c5 = -1.1320e+002;
  c6 = -7.8509e+000;
  c7 = 1.0999e+002;
  d = 0.0000e+000;

/*************************************************************************
                *** START OF DATA LOOP ****
*************************************************************************/
for (i = 0; i < total_samples; i++) {
        /* READ IN SIGNAL from A/D Channel 0 */
        outpw(AD_IN,0X0000);  /* Initiate A/D for CH0 */
        while(!(inp(AD_STATUS_REG)&0x40) );             /* Wait Till FIFO Not Empty */
                inp1 = inpw(AD_IN);
                /* CONVERT A/D SIGNAL TO VOLTS  */
                tinvolt = float(inp1) * AD_BitValue;

                /*****************************************/
                /*  START - Constant Freq Reference LOOP   */
```

```
        /*****************************************/
        if (const_fr_flag==1){
                        printf("\nTo stop Open Loop and ID Input, hit any key ...");
                        while(!kbhit()) {
                        j=j+1;
                        /* for (j=1; j<18 ; j++) { */
                        /* CREATE REFERENCE SIGNAL */
                        /* Compute past # of samples after every period */
                        if (period_end_switch)
                        { past_samples = past_samples + period_samples_c;
                        }
                        /* CHECK LIMITS OF FREQUENCY ASSUME SAMPLE 10 kHz */
                        /* don't let reference get > 200 Hz (50 samples) or
                        /* less than 20 Hz (500 samples) - TURN OFF DURING TESTING */
                        if (period_samples_c < 50) {
                                        period_samples_c = 50;
                        }
                        else if (period_samples_c > 500) {
                                        period_samples_c = 500;
                        }
                        /* Compute the sample # to switch from hi value to low value  */
                        duty = past_samples + duty_factor*period_samples_c;
                        /* make reference = 8 volts for 1st 1/2 of period */
                        if ( j <= duty ) { outda = out_amp_counts; }
                        /* make reference = 0 volts for 2nd 1/2 of period */
                        else { outda = 0; }
                        /* if at the end of a period then increment the period counter */
                        period_end = past_samples + period_samples_c;
                        period_end_switch = 0;
                        if ( j >= period_end ) { period_cnt = period_cnt + 1;
                                        period_end_switch = 1;
                        }

                        /* re-write reference voltage after limit check */
                        trefvolt = outda*DA_BitValue;

                        /* OUTPUT CURRENT */
                        outpw(DA_OUT, outda);
                        /* Initiate D/A Conversion */
                        outp(DA_STATUS_REG, 0x60);
                        /* check A/D status */
                        while (!(inp(AD_STATUS_REG)&0x04) );
                        outp(AD_STATUS_REG, 0xFB);  /* Magic number replaced */
                        } /* end of kbhit */

        const_fr_flag = 0; /* reset constant freq flag */
        /* Reset Open Loop Parameters for open loop with storage */
        past_samples = 0; /* initialize past # of samples to 0 */
        period_cnt = 0; /* Define period counter for reference signal */
 }
/*****************************************/
/*  END -  Constant Freq Reference  LOOP   */
/*****************************************/
```

```
/*****************************************/
/* Start - Constant Freq Storage      */
if (i < const_freq_cnt){
        /* CREATE REFERENCE SIGNAL */
        /* Compute past # of samples after every period */
        if (period_end_switch)
         { past_samples = past_samples + period_samples_c;
         }
        /* CHECK LIMITS OF FREQUENCY ASSUME SAMPLE 10 kHz */
        /* don't let reference get > 200 Hz (50 samples) or
        /* less than 20 Hz (500 samples) - TURN OFF DURING TESTING */
        if (period_samples_c < 50) {
                        period_samples_c = 50;
        }
        else if (period_samples_c > 500) {
                        period_samples_c = 500;
        }
        /* Compute the sample # to switch from hi value to low value  */
        duty = past_samples + duty_factor*period_samples_c;
        /* make reference = 8 volts for 1st 1/2 of period */
        if ( i <= duty )
        {
                        outda = out_amp_counts;
         /*             trefvolt_prt = out_amp_volts; *//* store for printing */
        }
        /* make reference = 0 volts for 2nd 1/2 of period */
        else
        {               outda = 0;
         /*             trefvolt_prt = 0.0; *//* store for printing */
        }
        /* if at the end of a period then increment the period counter */
        period_end = past_samples + period_samples_c;
        period_end_switch = 0;
        if ( i >= period_end ) { period_cnt = period_cnt + 1;
                                        period_end_switch = 1;     }
 }

if(i==const_freq_cnt) {
        /* Reset Open Loop Parameters for open loop with storage */
        past_samples = 0; /* initialize past # of samples to 0 */
        period_cnt = 0; /* Define period counter for reference signal */
}
/* End - Constant Freq Storage        */
/*****************************************/


/*************************************************/
/*****  Begin Control and Id Creation  ********/
if (i>=const_freq_cnt)
{
        /*****************************************/
        /* Start - Closed Loop Reference       */

        /* band-pass filter the input 20Hz - 270 Hz */
        xp1_n = apf11*xp1+apf12*xp2+apf13*xp3+apf14*xp4+apf15*xp5+apf16*xp6+bpf1*tinvolt;
```

```
xp2_n = apf21*xp1+apf22*xp2+apf23*xp3+apf24*xp4+apf25*xp5+apf26*xp6+bpf2*tinvolt;
xp3_n = apf31*xp1+apf32*xp2+apf33*xp3+apf34*xp4+apf35*xp5+apf36*xp6+bpf3*tinvolt;
xp4_n = apf41*xp1+apf42*xp2+apf43*xp3+apf44*xp4+apf45*xp5+apf46*xp6+bpf4*tinvolt;
xp5_n = apf51*xp1+apf52*xp2+apf53*xp3+apf54*xp4+apf55*xp5+apf56*xp6+bpf5*tinvolt;
xp6_n = apf61*xp1+apf62*xp2+apf63*xp3+apf64*xp4+apf65*xp5+apf66*xp6+bpf6*tinvolt;
/* calculate continuous output */
tin_filt=(cpf1*xp1+cpf2*xp2+cpf3*xp3+cpf4*xp4+cpf5*xp5+cpf6*xp6+dpf*tinvolt);

/* update the filter states */
xp1 = xp1_n;
xp2 = xp2_n;
xp3 = xp3_n;
xp4 = xp4_n;
xp5 = xp5_n;
xp6 = xp6_n;


/* INPUT Voltage Signal to LQG S.S. Control to compute States */
x1_n = a11*x1+a12*x2+a13*x3+a14*x4+a15*x5+a16*x6+a17*x7+b1*tin_filt;
x2_n = a21*x1+a22*x2+a23*x3+a24*x4+a25*x5+a26*x6+a27*x7+b2*tin_filt;
x3_n = a31*x1+a32*x2+a33*x3+a34*x4+a35*x5+a36*x6+a37*x7+b3*tin_filt;
x4_n = a41*x1+a42*x2+a43*x3+a44*x4+a45*x5+a46*x6+a47*x7+b4*tin_filt;
x5_n = a51*x1+a52*x2+a53*x3+a54*x4+a55*x5+a56*x6+a57*x7+b5*tin_filt;
x6_n = a61*x1+a62*x2+a63*x3+a64*x4+a65*x5+a66*x6+a67*x7+b6*tin_filt;
x7_n = a71*x1+a72*x2+a73*x3+a74*x4+a75*x5+a76*x6+a77*x7+b7*tin_filt;

/* CALCULATE OUTPUT TO FUEL INJECTOR */
trefvolt = Kl*(c1*x1+c2*x2+c3*x3+c4*x4+c5*x5+c6*x6+c7*x7+d*tin_filt);
x1 = x1_n;
x2 = x2_n;
x3 = x3_n;
x4 = x4_n;
x5 = x5_n;
x6 = x6_n;
x7 = x7_n;

/*  END - Closed Loop Reference          */
/*****************************************/

/*********************************************/
/****   Begin Random Frequency Creation ******/
if ( i >= id_start_count)
{
                /* Create random gaussian noise between -1 and 1 with 0 mean */
                tnoise = 2.0*(((float)rand() - RAND_MAX/2.0)/RAND_MAX);

                /* band-pass filter the random noise */
                xf1_n = af11*xf1+af12*xf2+af13*xf3+af14*xf4+af15*xf5+af16*xf6+bf1*tnoise;
                xf2_n = af21*xf1+af22*xf2+af23*xf3+af24*xf4+af25*xf5+af26*xf6+bf2*tnoise;
                xf3_n = af31*xf1+af32*xf2+af33*xf3+af34*xf4+af35*xf5+af36*xf6+bf3*tnoise;
                xf4_n = af41*xf1+af42*xf2+af43*xf3+af44*xf4+af45*xf5+af46*xf6+bf4*tnoise;
                xf5_n = af51*xf1+af52*xf2+af53*xf3+af54*xf4+af55*xf5+af56*xf6+bf5*tnoise;
                xf6_n = af61*xf1+af62*xf2+af63*xf3+af64*xf4+af65*xf5+af66*xf6+bf6*tnoise;
                /* calculate continuous output */
                /* noise_filt gain of 3.8 make output go from -1 to 1
```

```
                                        other gains are to make filter output the same order as control */

                        tnoise_filt=6.5*(cf1*xf1+cf2*xf2+cf3*xf3+cf4*xf4+cf5*xf5+cf6*xf6+df*tnoise)*3.8;

                        /* update the states */
                        xf1 = xf1_n;
                        xf2 = xf2_n;
                        xf3 = xf3_n;
                        xf4 = xf4_n;
                        xf5 = xf5_n;
                        xf6 = xf6_n;
                        /*** COMBINE ID INPUT AND CONTROL INPUT ***/
                        trefvolt = trefvolt + tnoise_filt;
        }
        /********  End Random Frequency *********/
        /*****************************************/
        /* Threshold check and conversion from continuous to binary */
        /* trefvolt_prt = trefvolt; */ /* store for printing */
        if (trefvolt < thresh)
        {
                        trefvolt = 0.0;
        }
        else if (trefvolt >= thresh)
        {
                        trefvolt = out_amp_volts;
        }
        /* Convert from Volts to counts */
        outda = (int)(trefvolt/DA_BitValue);
}
/*********************************************/
/*****  Begin Control and Id Creation  ********/

/* CHECK LIMITS OF REFERENCE OUTPUT */
if (outda < 0)
{
        outda = 0;
}
else if (outda >= out_amp_counts)
{
        outda = out_amp_counts;
}
/* re-write reference voltage after limit check */
trefvolt = outda*DA_BitValue;


/* OUTPUT CURRENT */
outpw(DA_OUT, outda);
/* Initiate D/A Conversion */
outp(DA_STATUS_REG, 0x60);

/* STORE DATA */
if( (i>=itmpr) & (wflag) )
{  fprintf(out_file, "%f %f\n", trefvolt, tinvolt);
/* fprintf(out_file, "%f %f %f %f\n", trefvolt, tinvolt, tnoise, tnoise_filt);*/
 }
```

```
        /* check A/D status */
        /* if(inp(AD_STATUS_REG)&0x04)  printf("sample missed\n"); */
        while (!(inp(AD_STATUS_REG)&0x04) );
        outp(AD_STATUS_REG, 0xFB);  /* Magic number replaced */

}
printf("\n\ni = %u \n", i );
printf("total_samples = %u \n", total_samples );
printf("To continue -  press any key     ");
getch();
/*************************************************************************
                                *** END OF DATA LOOP ***
*************************************************************************/

/* WRITE DATA TO A FILE */
if(wflag)
{ fclose(out_file);
          }

farfree(involt);
farfree(refvolt);

}
```

# Appendix B: Thermoacoustic Instability Physics based Model

**Dynamic model of the combustion system**

For full details on the development of the finite dimensional, laminar, pre-mixed combustor model see works from MIT. in [ 5 ] through [ 13 ]. A brief presentation is given below. The finite dimensional model contains physics based on the thermoacoustics of combustion. The presentation is broken into the physics of the acoustics and the flame dynamics. Figure B-1 shows the assumed system configuration for the analysis.

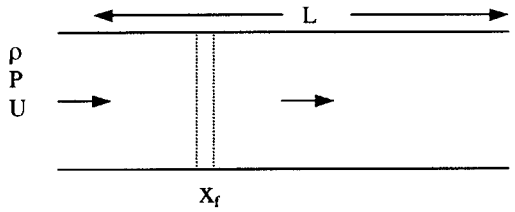



Figure B-1: One dimensional reacting fluid flow with flame at $x = x_f$

**Acoustics**

Assuming one dimensional flow, the conservation of mass, momentum, and the energy for a compressible fluid can be written as in [ 14 ]

( 56 ) $$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} = 0$$

( 57 ) $$\rho \frac{\partial u}{\partial t} + \rho u \frac{\partial u}{\partial x} + \frac{\partial P}{\partial x} = 0$$

( 58 ) $$\frac{\partial P}{\partial t} + u \frac{\partial P}{\partial x} + \gamma P \frac{\partial u}{\partial x} = (\gamma - 1) q$$

Where t and x are time and space, respectively, ρ is the density, u is the velocity, P is the pressure, γ is the specific heat ratio, and q is the heat release rate per unit volume.

Following the development in [ 6 ] and [ 12 ], the conservation equations are simplified by decomposing the system variables into their mean and perturbation components. Governing equations for the perturbationsare revealed by assuming the mean flow is steady and substituting the variable decompositions into the governing equations. Assuming that the flame zone is localized spatially, as shown in

Figure B-1, the heat release can be expressed as $q(x,t) = \delta(x-x_0)q_0(t)$, where $q_0$ is the heat release per unit area, and $\delta(.)$ is the Dirac delta function. This implies that the mean variables are essentially constant over the length of the combustor except for a step change @ $x_0$. Reference [ 6 ] shows the step change in pressure is negligible compared to the change in the mean velocity or mean density at $x_0$. Assuming that the spatial derivative effects are negligible and do some further algebraic manipulation yields

$$\left[\frac{\partial^2 P'}{\partial t^2} - \left(\bar{c}(x) - \bar{u}(x)\right)\frac{\partial^2 P'}{\partial x^2}\right] + 2\bar{u}(x)\frac{\partial^2 P'}{\partial x \partial t} = (\gamma - 1)\left(\frac{\partial q'}{\partial t} + \bar{u}(x)\frac{\partial q'}{\partial x}\right) \quad (59)$$

$$\frac{\partial P'}{\partial t} + \bar{u}(x)\frac{\partial P'}{\partial x} + \gamma\bar{P}\frac{\partial u'}{\partial x} = (\gamma - 1)q' \quad (60)$$

where P', u', $q_f$' denote the unsteady components of the pressure, velocity, and heat release rate, respectively, $\bar{P}$ is the mean pressure. All of the mean variables can be considered as constants if we assume the mean heat is negligible. Then, if the mean flow is neglected

$$\left(\frac{\partial^2 P'}{\partial t^2} - \bar{c}^2\frac{\partial^2 P'}{\partial x^2}\right) = (\gamma - 1)\frac{\partial q'}{\partial t} \quad (61)$$

$$\frac{\partial P'}{\partial t} + \gamma\bar{P}\frac{\partial u'}{\partial x} = (\gamma - 1)q' \quad . \quad (62)$$

Equations ( 61 ) and ( 62 ) display the dominant acoustic characteristics of the combustor. Then separate the variables into spatial and temporal changes using the Galerkin method [ 15 ]. Expressing unsteady pressure as spatial $\psi_i(x)$ and temporal $\eta_i(t)$, where i is the mode number, and assuming spatial mode shapes allows the differential equations to be decoupled in x and t. Restricting our attention to the single mode case and substituting into ( 61 ) and ( 62 ) gives

$$\bar{P}\psi(x)\ddot{\eta} - c^2\bar{P}\frac{d^2\psi}{dx^2}\eta(t) = (\gamma - 1)\frac{\partial q'}{\partial t} \quad (63)$$

$$\ddot{\eta} - \frac{c^2}{\psi(x)}\frac{d^2\psi}{dx^2}\eta(t) = \frac{(\gamma - 1)}{\bar{P}\psi(x)}\frac{\partial q'}{\partial t} \quad . \quad (64)$$

This is simplified further by multiplying by $\psi$ and integrating [0,L],

$$\ddot{\eta} + \omega^2\eta = \frac{\gamma a_0}{E}\psi(x_0)\dot{q}_0'(t), \quad (65)$$

where $a_0 = \frac{\gamma - 1}{\gamma\bar{P}}$, $\omega = \bar{c}k$, $E = \int_0^L \psi^2(\delta)d\delta$ .

Similarly ( 60 ) can be integrated to yield

$$u'(x,t) = \frac{1}{\gamma k^2}\dot{\eta}(t)\frac{d\psi(x)}{dx} + a_0 q_0'(t)\mathrm{H}(x - x_0) \qquad (66)$$

where $\mathrm{H}(z) = 1$ for $z \geq 0$, $\mathrm{H}(z) = 0$ for $z \leq 0$.

Equations ( 65 ) and ( 66 ) together describe the time domain interactions between pressure amplitude $\eta(t)$, heat release rate $q_0'(t)$, and velocity $u'(x,t)$.

**Flame Dynamics**

The unsteady heat release from a flame is dynamically related to the unsteady component of the flow velocity. The reader is referred to [ 5 ] for further details on this relationship and the ensuing model. The governing equations from [ 5 ] for the flame surface,

$$\frac{\partial \xi}{\partial t} = S_u \frac{\partial \xi}{\partial r} + u_0(t) \qquad (67)$$

where $u_0(t)$ denotes flow velocity @ $x = x_0$, $S_u$ is the speed of propagation of the lfame normal to itself, and r is the radial displacement.

When $S_u$ is constant, the instantaneous unsteady component of the heat release $Q_0'$ is:

$$Q_0'(t) = \rho S_u \Delta h_r A_f'(t), \qquad (68)$$

where

$$A_f' = 2\pi \int_0^{r_d} \xi'(r,t)dr. \qquad (69)$$

Therefore we obtain from ( 68 ) and ( 69 )

$$Q_0'(t) = 2\pi\rho S_u \Delta h_r \int_0^{r_d} \xi'(r,t)dr = k\int_0^{r_d} \xi'(r,t)dr \qquad (70)$$

where $k = 2\pi\rho S_u \Delta h_r$.

By solving the flame model of ( 67 ) with boundary conditions and assuming the effect of high frequencies on the flame dynamics is negligible as in [ 5 ] results in

$$Q_0'(t) = k\int_0^{r_d} \xi'(r,t)dr \sim \frac{kr_d}{2}\xi'(0,t) \qquad (71)$$

Combining the time derivative of ( 70 ) using ( 67 ) with ( 71 ) results in

$$\dot{Q}_0' + b_1 Q_0' = b_0 u_0' \qquad (72)$$

where $b_0 = kr_d$, and $b_1 = \frac{2S_u}{r_d}$.

If the flame is anchored on a perforated flame holder (see [ 4 ] ), the heat release per unit area is given by

( 73 ) $$\dot{q}_0' = \frac{4n_f}{\pi D^2} Q_0'$$

where D is the diameter of the flame holder and $n_f$ is the number of holes. The resulting relationship between the heat release rate per unit area and the instantaneous flow velocity at x = $x_0$ is

( 74 ) $$\dot{q}_0' + b_0 q_0' = b_2 u_0',$$

where

( 75 ) $$b_2 = \frac{2S_u}{r_d}\left(\frac{d}{D}\right)^2 n_f \rho \Delta h_r.$$

**Feedback Model**

By developing an effective velocity, $\tilde{u}_0'$, that contributes to the heat release rate where

( 76 ) $$\tilde{u}_0' = u_0' - \theta a_0 b_2 = \left(\frac{1}{\gamma k^2}\frac{d\psi}{dx}(x_0)\right)\dot{\eta}$$

we obtain

( 77 ) $$\dot{q}_0' + b_3 q_0' = b_2 \tilde{u}_0'$$

where $b_3 = b_1 - \theta a_0 b_2$, and θ = 0.5 from [ 6 ]. The following two equations and Figure B-2 describe the feedback model.

( 78 ) $$\eta(t) = \left[\frac{S}{S^2+\omega^2}\right]\left(\frac{\gamma a_0}{E}\psi(x_0) q_0'(t)\right) \text{ or } \ddot{\eta} = -\omega^2\eta + \frac{\gamma a_0}{E}\psi(x_0)\dot{q}_0'$$

( 79 ) $$q_0'(t) = \left[\frac{Sb_2}{S+b_3}\right]\left(\frac{1}{k^2}\frac{d\psi}{dx}(x_0)\eta(t)\right) \text{ or } \dot{q}_0' = -b_3 q_0' + b_2\left(\frac{1}{k^2}\frac{d\psi}{dx}(x_0)\dot{\eta}\right)$$

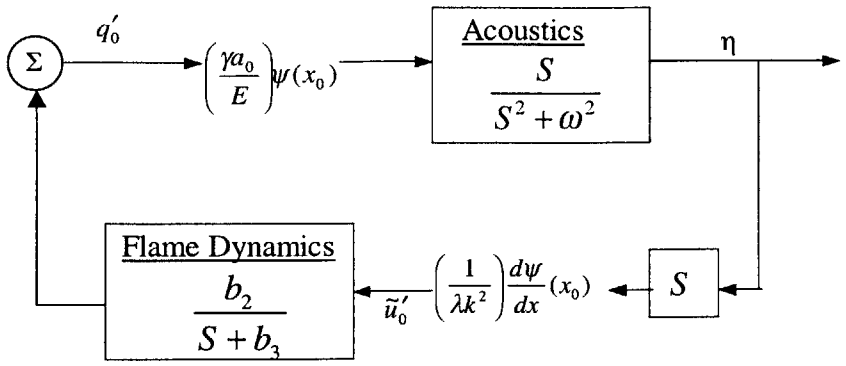



Figure B-2: The combustion feedback system

**Input-Output Model**

By considering the effect of mean flow velocity, mean heat additions, coupled acoustic modes, damping (see [ 10 ]), and the addition of fuel perturbation due to the injector yields

( 80 ) $$\dot{q}_f' = -b_3 q_f' + b_2\left[\bar{\phi}\tilde{u}' + \bar{u}\phi'\right]$$

( 81 ) $$\ddot{\eta}_i = -\omega_i\eta_i - 2\zeta\omega_i\dot{\eta}_i + bt_i\dot{q}_f'$$

( 82 ) $$\tilde{u}' = \sum_{i=1}^{n}(ct_i\dot{\eta}_i)$$

where $\phi'$ is the equivalence ratio perturbation, and $\bar{\phi}$ is the mean equivalence ratio.

The output equations to get pressure are

( 83 ) $$P = \bar{P}\sum cct_i\eta_i$$

# Appendix C: Evaluation Model parameter script for MATLAB® and Simulink® model

The script file establishing the evaluation model parameters is given below, and the simulation model that is used with this script file is shown in Figure C-1 through Figure C-3.

```
%=====================================================================
% filename: Thesis\final\1model\comb_inj_nl_ns.m
%
% Purpose: Create model for:
%          1.) combustor - acoustics, heat release,
%                          The acoustics have 2 modes and are coupled
%          2.) fuel injector - with nonlinear saturation
%          3.) nonlinear combustor pressure saturation
%          4.) noise - nonlinear noise model
%
% ??-Feb-99 JP Hathout Created
% 15-Mar-99 BJ Brunell Change b1 to b3
%                      Added damping for computational stability
%                      Separated acoustic and heat release S.S. matrices
% 16-Mar-99 BJ Brunell Copied from 1comb\comb.m
%                      Added c1 and c2 for limit cycles
% 31-Oct-99 BJ Brunell Copied from
%              Thesis\old_inj_model\4system_id\open_loop\comb_id_nl.m
%
%
% Note: see thesis\final\model\ol_cl_rep.m for O.L.,C.L., and transfer funct
%       represenation of the combustor
%=====================================================================
disp(' final\1model\comb_inj_nl_ns.m ');
format short e
v=3;
% C=1 % 1=modes are coupled, 0=modes are uncoupled - linear C.L.
% Poinsot exp., no meanflow, 2 modes

% Acoustic parameters
gamma=1.4;                    % (-) Specific heat ratio
c=350;                        % (m/s) Speed of sound
pbar=1E+05;                   % (bar) Static pressure
rho=gamma*pbar/c^2;           % (kg/m^3) Density
L  =.49;                      % (m) Acoustic Length
                              % for gulati Leff=0.5, Poinsot Leff=0.49
a0=(gamma-1)/(rho*c^2);       % Acoustic Constant

% Flame parameters
eps=1.0;                      % (-) Correction factor for flame radius
x0 = 0.24;                    % (m) Flame location Gulati =.25, Poinsot =.24
phibar=.7;                    % (-) Average equivalence ratio
ubar=2;                       % (m/s) Average velocity
theta=0.5;                    % (-) Correction factor for velocity
dhr=50*(10^6)/15.6;           % Heat of reaction for Propane w/o phi
su=0.3;                       % (m/s) Laminar Burning Velocity
R=.02;                        % (m) =0.025 gul.. 0.02 poi
Rf=eps*.75E-03;               % (m) Radius of flame
nfl=80;                       % Number of holes in flame holder

% Constants to simplify flame equations
b1=2*su/Rf;                   % =w_f
b2=b1*rho*dhr*nfl*Rf^2/(R^2);% w/o theta effect.
b3=b1-theta*a0*b2;            % b1(=w_f) with effect of theta.

% Passive damping for computational stability
zetcl2 = .014 ;  % was 0.012
```

```
% Fuel Injector Parameters
K_i=200;                        % Injector Gain: slow=20, med,fast=200
omg_i=3000;                     % Injector bw: slow=360, med=3000, fast=1200*10
uppersat=.06;                   % Injector upper saturation: s=.09, m=.08, f=.2
lowersat=-.01;                  % Injector lower saturation: s=-.01,m=-.01, f=-
.125
tmdly = .0005;                  % Injector time delay

% create the noise model parameters
wh = 1500*2*pi;                  % (rad/s) High frequency cut-off
wl = 60*2*pi;                   % (rad/s) Low frequency cut-off
[nh,dh]=butter(2,wh,'high','s'); % high pass butterworth filter
[nl,dl]=butter(2,wl,'s');       % low pass butterworth filter
nh=nh.*1.1;                     % Change high frequency to be ~1% full scale
nl=nl.*25;                      % Change low frequency to be ~4% full scale
nn = conv(nh,dl)+conv(nl,dh);% bandstop filter numerator
nd = conv(dh,dl);               % bandstop filter denominator
cz = 0.3;                       % constant multiplier
mx_sat=15;                      % max saturation
mn_sat=-15;                     % min saturation

% Define sensor (xs) and actuator (xa) position
if v==1, disp('D/D'); xs=.17*L;  xa=0.17*L; end
if v==2, disp('A/D'); xs=.51*L;  xa=0.17*L; end
if v==3, disp('A/C'); xs=0.51*L; xa=.25*L;  end
if v==4, disp('A/B'); xs=.51*L;  xa=.45*L;  end
if v==5, disp('A/A'); xs=.51*L;  xa=.51*L;  end

% Assumed acoustic mode solutions - Closed-Open case
k1=pi/(2*L);                    % Wave number for 1st acoustic mode
w1=c*k1;                        % Frequency of 1st acoustic mode
k2=3*pi/(2*L);                  % Wave number for 2nd acoustic mode
w2=c*k2;                        % Frequency of 2nd acoustic mode
E=L*0.5;                        % Energy in the modes

% Flame Feedback
bt=gamma*a0/E*[cos(k1*x0);
               cos(k2*x0)];
ct=(1/gamma)*[-sin(k1*x0)/k1 -sin(k2*x0)/k2];

% Actuator and sensor
cct=[cos(k1*xs) cos(k2*xs)];

bt1=bt(1);      bt2=bt(2);
ct1=ct(1);      ct2=ct(2);
cct1=cct(1);    cct2=cct(2);

% Define describing function parameters
c1 = 1; c2 = -6.25;

% ========================================================================
%                          Define State Space Models
% ------------------------------------------------------------------------
% =====  Define Acoustic Model ======
App = [  0         1         0          0      ;
       -w1^2 -2*zetcl2*w1    0          0      ;
         0         0         0          1      ;
         0         0      -w2^2   -2*zetcl2*w2];

B1 = 0;
B2 = bt1;
B3 = 0;
```

```
B4 = bt2;
Bpp = [B1;B2;B3;B4];
Cpp=pbar*[cct1 0 cct2 0 ];
Dpp = [0];
% ==== Acoustic Initial conditions for simulation ===
acoust_ic = 5e-1*[0 1 0 1];
% =====  Define Heat Dynamics Model ======
Cti = [0 ct1 0 ct2];
Ah = [-b3];


%=========================================================================
```

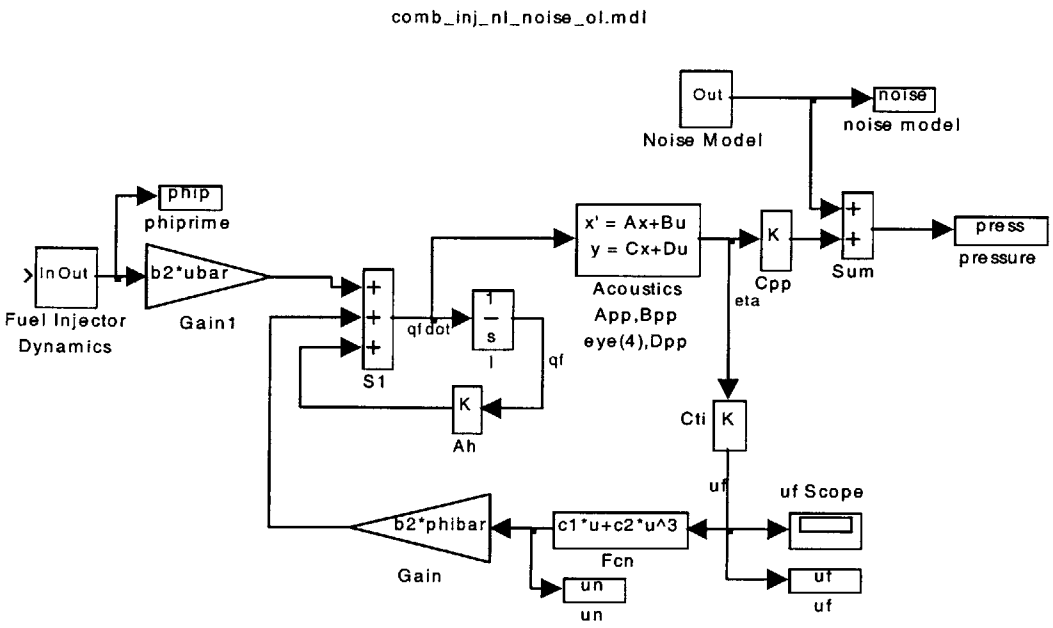



Figure C-1: Complete nonlinear simulation model with fuel injector and noise addition.

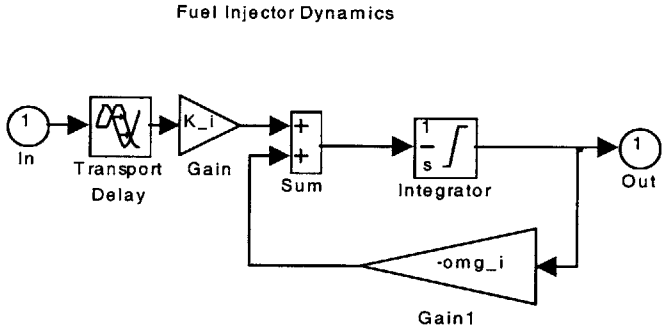



Figure C-2: Fuel injector model used in complete evalution model

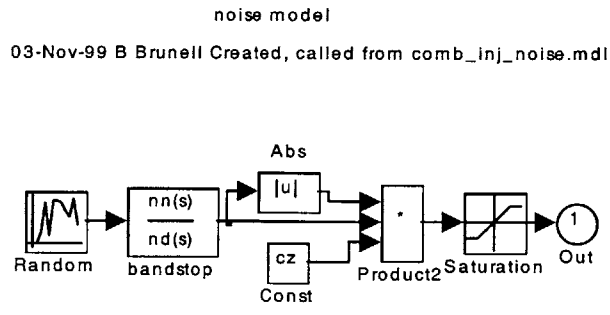



Figure C-3: Noise model used in complete evalution model

# Appendix D: System Identification and LQG control script for evaluation model

```
%=============================================================
% filename:\thesis\final\7closed_loop\sysid_cl.m
%
% Purpose 1.) Create a system id model of nonlinear
%          evaluation model
%         2.) Create LQG control based on the system id model
%         3.) Set the model conditions (see Appendix C: Evaluation Model parameter
script for MATLAB® and Simulink® model)
%
% 12-Dec-99 B Brunell - Created
%
%
% Note: to be run with (simulink nonlinear model)
%      \final\7closed_loop\comb_inj_nl_noise_lqg_cl.mdl
%
% Note: run validation.m to test for model goodness
%=============================================================
i=1;
% Choose the system ID method and other scipt choices
set_model       = 'y';  % setup the model parameters
set_control     = 'n';  % setup the open-loop based LQG controller
run_sim_id      = 'n';  % Run the simulink model for id data
load_data       = 'y';  % Load simulation or experiment data
volts_2_press   = 'y';  % system id from volts 2 pressure
                        %   - or 'n' = from phiprime 2 pressure
pre_filter      = 'y';  % prefilter the data before system id
ss_id           = 'n';  % Create 6th order sub-space model
armax_id        = 'n';  % Create 6th order prediction-error model
oe_id           = 'y';  % Create Bang-Bang control 2nd order id model
cl_lqg_control  = 'n';  % Build controller


% input the max simulation time
tmax = .4;
% input the time to start control
t_start_ctrl = 0.20;
%-------------------------------------------------------------
% Set model parameters
if set_model == 'y',
   cd ../1model
   comb_inj_nl_ns
   cd ../7closed_loop
end

%-------------------------------------------------------------
% Set the open-loop system id based controller
if set_control == 'y',
   cd ../5open_loop
   sysid_ol
   cd ../7closed_loop

   % Create bandpass butterworth filter noise input
   w_lo = 80*2*pi; % (rad/s) low corner frequency
   w_hi = 1500*2*pi; % (rad/s) high corner frequency
   [n_in,d_in] = butter(3,[w_lo w_hi],'s');
end

%-------------------------------------------------------------
% Run the simulation to get id data
if run_sim_id == 'y',
   disp('Running comb_inj_nl_noise_lqg_cl.mdl ....');
```

```
   sim('comb_inj_nl_noise_lqg_cl',[0 0.5])
end

%-------------------------------------------------------
% load data for system id
if load_data == 'y',
   cd id
   if volts_2_press == 'y',
      load cl_volts.mat % get input signal
      input = volts;
   else
      load cl_phip.mat  % get input signal
      input = phip;
   end
   load cl_press.mat % get output signal
   cd ..
      % Create 2 column matrix of [out in]
   data = [[press(2,1000:6000)]',[ input(2,1000:6000)]'];
   % initialize bandpass filter to just pass input to output
   Afd = [0]; Bfd = [0]; Cfd = [0]; Dfd = [1];
   if pre_filter == 'y',
      samp_rate=10000;      % Hertz - 10 KHz used in the lab
      samp_t = 1/samp_rate; % Seconds
      % set corners of bandpass filter
      lc = 100.0;   % (Hz) lower corner
      hc = 1000.0;  % (HZ) upper corner
      % convert to fraction of discrete frequency
      lcd = lc*(2*samp_t);
      hcd = hc*(2*samp_t);
      % Create butterworth filter
      [Afd,Bfd,Cfd,Dfd] = butter(3,[lcd hcd]); filt=ss(Afd,Bfd,Cfd,Dfd,samp_t);
      % filter the pressure data
      data(1:5001,1)=lsim(filt,data(1:5001,1));
   end
   % Do preprocessing of the data
   % data = dtrend(data,0);
   % Split data for correlation
   datae = data([1:2501],:);    % Data for estimation
   datav = data([2501:5000],:); % Data for validation
end

%-------------------------------------------------------
% Do system id estimation of parameters
if ss_id == 'y',
   disp(' Warning!! using Sub-Space Methods ')
   % Estimates a state-space model using a sub-space method.
   % TH=N4SID(Z,ORDER,NY,AUXORD,DKX,MAXSIZE,TSAMP)
   % try #1 (fail) th = n4sid(datae,2,1,5,[1,1,1],[],0.0001);
   th = n4sid(datae,6,1,15,[0,1,1],[],0.0001)
   titl = 'State-Space model using sub-space method';mu=5e-5; rho=5e-0;
   % convert from theta model to state-space model
   [aid,bid,cid,did,kid,x0id] = th2ss(th);
   % Present information on the model
   present(th)
end

if armax_id == 'y',
   disp(' Warning!! using ARMAX prediction-error Methods ')
   % Computes the prediction error estimate of an ARMAX model.
   % TH=ARMAX(Z,NN,maxiter,tol,lim,maxsize,T,'trace')
   th = armax(datae,[6 5 2 1],[],[],[],[],0.0001); mu=5e-2; rho=5e1;
   %th = armax(datae,[6 6 4 5],[],[],[],[],0.0001);
```

```
   titl = 'Prediction error estimate of ARMAX Model';
   % convert from theta model to state-space model
   [aid,bid,cid,did,kid,x0id] = th2ss(th);
   x0id = 1e0*ones(1,size(aid,1));
   % Present information on the model
   present(th)
end


if oe_id == 'y',
   disp(' Warning!! using output-error model ')
   % Computes the prediction error estimate of an ARMAX model.
   % TH=ARMAX(Z,NN,maxiter,tol,lim,maxsize,T,'trace')
   th = oe(datae,[2 2 1],[],[],[],[],0.0001);
   titl = 'Prediction error estimate of OE Model';
   % convert from theta model to state-space model
   [aid,bid,cid,did,kid,x0id] = th2ss(th);
   x0id = 1e0*ones(1,size(aid,1));
   % Present information on the model
   present(th)
end

%-------------------------------------------------------
%-- Build Controller ------
if cl_lqg_control == 'y',
   [as,bs,cs,ds] = d2cm(aid,bid,cid,did,.0001,'zoh');

   % FARE - Observer Design
   % mu smaller is faster
   % mu=5e-2; % 1e-3 for Jenn's
   Q=bs*bs';
   Sigma=are(as',cs'*cs/mu,Q);
   H=1/mu*Sigma*cs';

   % CARE - Controller Design
   % rho bigger less control effort
   % rho=5e1; % 100 for Jenns
   Q=cs'*cs;
   R=are(as,1/rho*bs*bs',Q);
   K=1/rho*bs'*R; % control linear gain matrix full state feedback

   % build input - output s.s. matrices for
   % observer compensator
   aid_ctl=as-bs*K-H*cs;
   bid_ctl=H;
   cid_ctl=-K;
   did_ctl=zeros(size(cid_ctl,1),size(bid_ctl,2));
   titl2=['LQR on system id: mu=',num2str(mu),'rho=',num2str(rho)];
   % note: use sysid_lqg_checkout.mdl to evaluate mu and rho
end
```

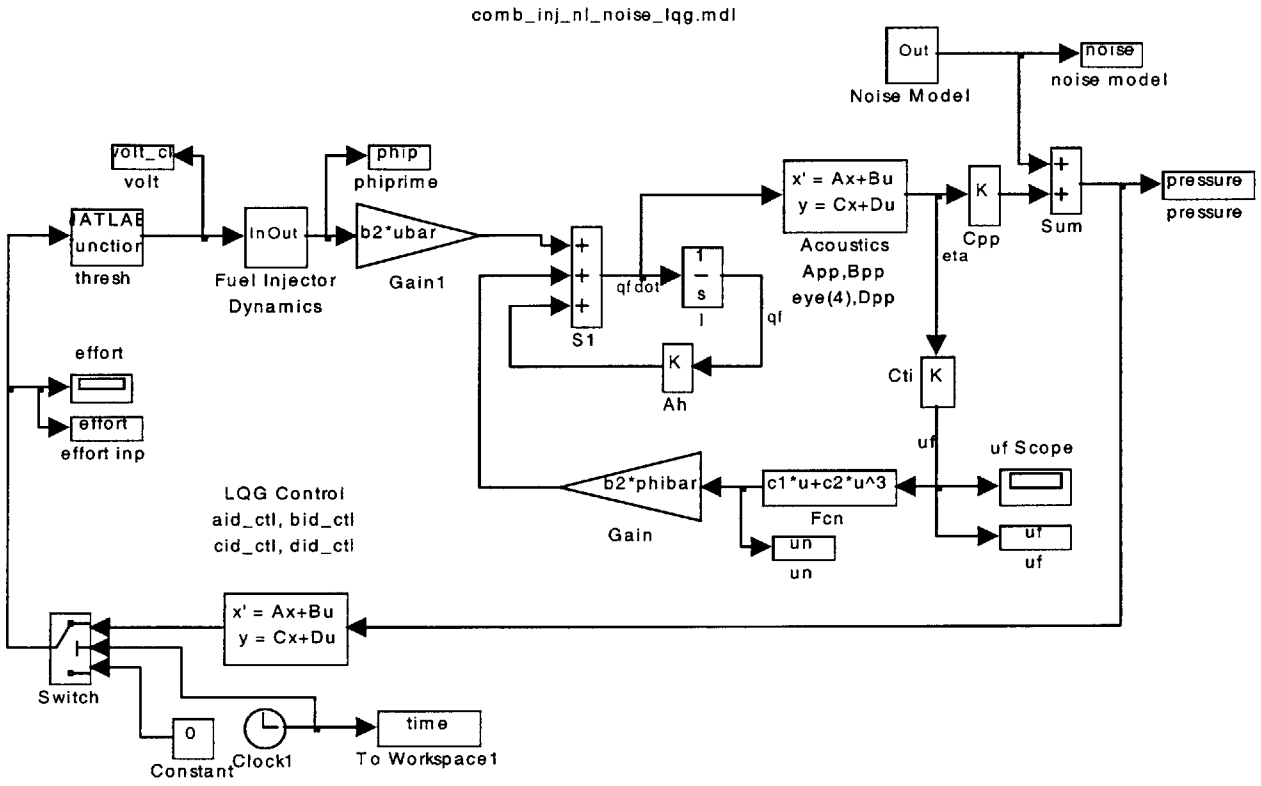



Figure D-1: Evaluation model combustor in Simulink® with LQG control

# Appendix E: 'Bang-Bang' Control Development, Algorithm, and Simulation block diagram for the Evaluation Model

The 'Bang-Bang' control development is taken from [ 17 ] through [ 21 ]. The goal is to take an n dimensional linear system with a constant input,

( 84 ) $$\dot{x} = Ax(t) + Bu(t)$$

where $-U_{\max} \le u(t) \le U_{\max}$, and transfer it to the origin in the minimum amount of time. The performance index J will be:

( 85 ) $$J = \int_{t_0}^{t_f} L dt = \int_{t_0}^{t_f} 1 dt = (t_f - t_0).$$

The Hamiltonian for the current problem is

( 86 ) $$H = L + \lambda^T f = 1 + \lambda^T (Ax + Bu).$$

Then according to Pontryagin's minimum principle, $\{H(x^*, u^*, \lambda^*, t) \le H(x^*, \delta u, \lambda^*, t)\}$ for all admissable $\delta u$, where $^*$ = optimal. Therefore the optimum control $u^*(t)$ must satisfy

( 87 ) $$1 + (\lambda^*)^T (Ax^* + Bu^*) \le 1 + (\lambda^*)^T (Ax^* + Bu).$$

For optimality then: $(\lambda^*)^T Bu^* \le (\lambda^*)^T Bu$ for all admissable u. The problem then is to minimize (take on a value as close to -∞ as possible) $\lambda^T(t)Bu(t)$. This implies that if;

$\lambda^T B$ is positive $\rightarrow$ u = -$U_{\max}$,

$\lambda^T B$ is negative $\rightarrow$ u = $U_{\max}$,

$\lambda^T B$ is called the switching function. This can be more compactly stated as

( 88 ) $$u(t) = -\text{sgn}(\lambda^T B)$$

The canonic equations are

( 89 ) $$\dot{x} = \frac{\partial H}{\partial \lambda} = Ax(t) + bu(t) = Ax(t) - b\,\text{sgn}(\lambda^T B),$$

( 90 ) $$\dot{\lambda} = -\frac{\partial H}{\partial x} = -A^T \lambda(t).$$

The solution to equation ( 90 ) is

( 91 ) $$\lambda(t) = e^{-A^T (t - t_f)} \lambda(t_f).$$

Rewrite the linear system equations in terms of time to go $\tau$, by letting

( 92 ) $$\tau = t_f - t,$$

and introducing $\xi$,

( 93 ) $$\frac{d\xi(\tau)}{d\tau} = -A\xi(\tau) - Bu(t_f - \tau)$$

Substituting u from equation ( 88 )

( 94 ) $$\frac{d\xi(\tau)}{d\tau} = -A\xi(\tau) + B\,\mathrm{sgn}\left(\lambda^T(t_f - \tau)B\right)$$

Substituting ( 92 ) into ( 91 ) results in,

( 95 ) $$\lambda(t_f - \tau) = e^{-A^T(t_f - \tau - t_f)}\lambda(t_f).$$

Substituting ( 95 ) into ( 94 ),

( 96 ) $$\frac{d\xi(\tau)}{d\tau} = -A\xi(\tau) + B\,\mathrm{sgn}\left(\lambda(t_f)e^{A^T(\tau)}B\right).$$

Taking the Laplace

( 97 ) $$SI\xi(s) - \xi(0) = -A\xi(s) + BU(s)$$

Solve for $\xi(s)$

( 98 ) $$\xi(s) = [SI + A]^{-1}\xi(0) + [SI + A]^{-1}BU(s)$$

Let $\Phi = [SI + A]^{-1}$ be the resolvent matrix, then

( 99 ) $$\xi(s) = \Phi\xi(0) + \Phi BU(s).$$

The time response is found by taking the inverse Laplace

( 100 ) $$\xi(t) = \mathrm{L}^{-1}[\xi(s)].$$

The inverse Laplace of the resolvent matrix is the matrix exponential or the transition matrix,

( 101 ) $$\Phi(t) = L^{-1}[\Phi(s)] = L^{-1}\left[(SI + A)^{-1}\right] = e^{-At}.$$

Since the second term in equation ( 101 ) is the product of two function in the Laplace domain, the inverse Laplace is the convolution integral of 2 functions.

( 102 ) $$L^{-1}[\Phi(s)BU(s)] = L^{-1}\left[(SI + A)^{-1}BU(s)\right] = \int_0^t e^{-A(t-p)}Bu(p)dp.$$

Thus the time to go response of equation ( 96 ) with initial conditions of $\xi(0) = x(t_f)$

( 103 ) $$\xi(\tau) = \int_0^{\tau} e^{-A(\tau-p)} Bu(p)dp = \int_0^{\tau} e^{-A(\tau-p)} B \operatorname{sgn}\left[\lambda^T(t_f) e^{Ap} B\right] dp$$

By allowing the sgn function to be either $U_{min}$ or $U_{max}$ in ( 103 ) the time to go response and the phase plane trajectories for both values of sgn can be determined either optimally by solving the equations or sub-optimally by fitting curves to the switching functions. In this thesis the sub-optimal approach was taken. The basic sub-optimal 'Bang-Bang' control algorithm is taken from [ 18 ]. It is modified for the 2nd order s.s. system id model dynamcs. The control algorithm is used as a sub routine that is called from the Simulink® model comb_inj_nl_noise_bang.mdl presented in

Figure E-1.

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

function [u]=sub_id_2nd(x)
% [u] = sub_id_2nd(x)
%
% Simulation of Bang-Bang sub-Optimal Control
% The switching function is based on a 2nd order
%
% INPUTS:
% x     - state vector
% OUPUTS:
% u     - control

% 08-May-99 B Brunell Copied from sub_2nd changed switching
%                     function

%=========================================================================
% Note: switching function for this id model is
% based off of less than one circle in the x1, x2 plane.
% after the switching function hits its max, then just continue
% a straight line after that point
%=========================================================================
% Least squares fit from id_2nd_sw_fun.m
c = [ -9.4715e-001
       2.49241e+002
      -1.880689e+005
       7.3234769e+007];
% Setup Parameters for this model type
epsilon=5e-3;
% Designate max and min control effort
umin = 0; umax = 5;
% Designate the min and max x1 that the curve fit is used
x1min = -0.04; x1max = -x1min;
% Compute the corresponding x2
x2max = c(1)*x1min + c(2)*x1min^3 + c(3)*x1min^5 + c(4)*x1min^7;
x2min = -x2max;
%-----------------------------------------
% compute Switching Function and Control
%-----------------------------------------
% first see if outside of curve fit
if (x(2,1)>=x2max)      sw = 0; u = umin;
elseif (x(2,1)<=x2min) sw = 0; u = umax;
```

```
% if in curve fit region then calc the switching function and control
else
   x2sw = c(1)*x(1,1) + c(2)*x(1,1)^3 + c(3)*x(1,1)^5 + c(4)*x(1,1)^7;
   sw=x(2,1)-x2sw;
   % check to see if close to the switching curve
   if (abs(sw)<epsilon)
      if (x(1,1) > 0) u=umax; end
      if (x(1,1) < 0) u=umin; end
   else
      if (sw < 0) u=umax; end
      if (sw > 0) u=umin; end
   end
end
% Check to see if close to the origin
if(x(1,1)^2+x(2,1)^2 < epsilon) u=0; end
```

The Simulink® model of the combustor with 'Bang-Bang' control is shown below in Figure E-1.

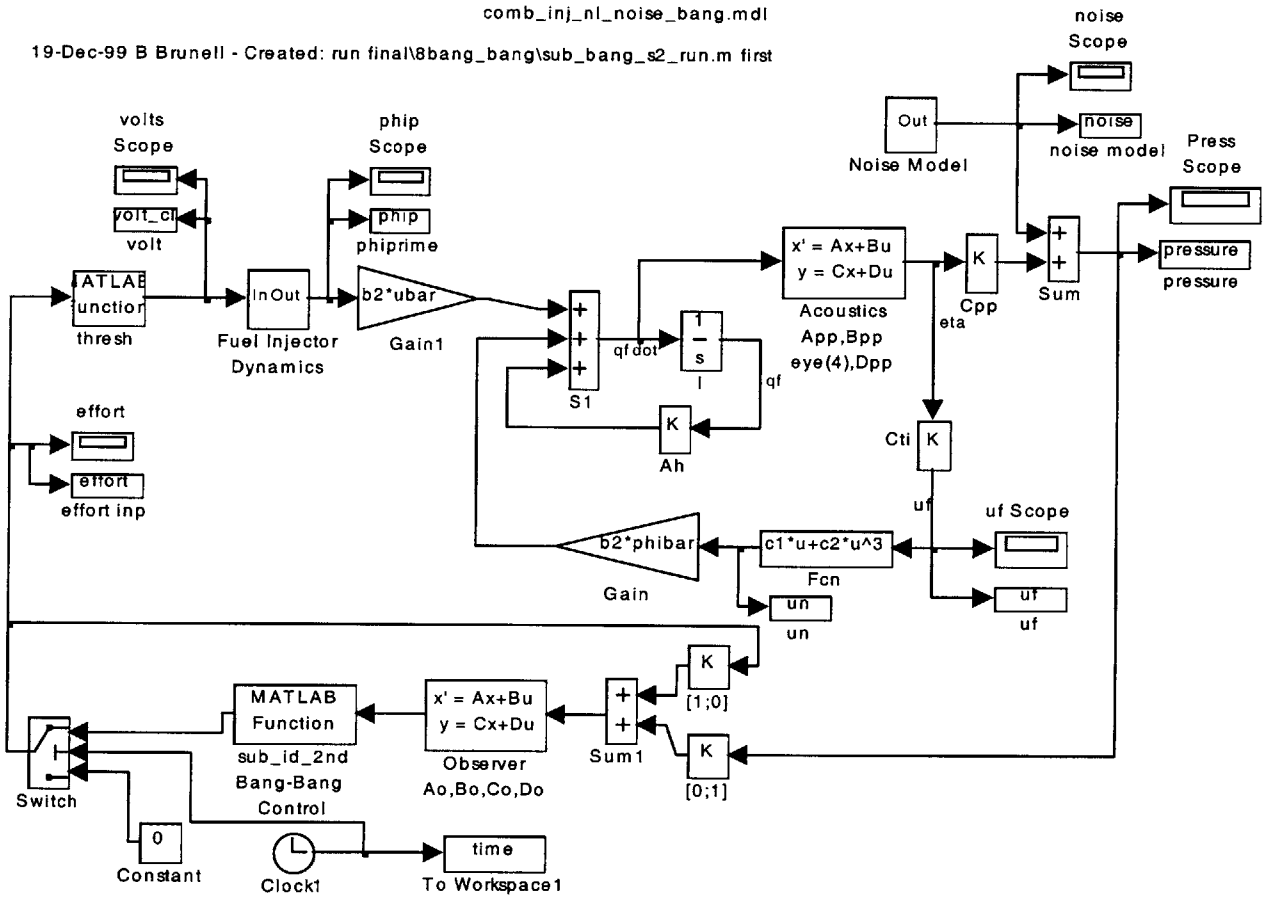



Figure E-1: Evaluation model combustor in Simulink® with 'Bang-Bang' control

# Appendix F: Lead-lag control algorithm

The MATLAB® to create the adjustable lead-lag is shown below.

```
==========================================================================
function [u]=ldlg6_fct(x)
% [u] = ldlg6_fct(x)
%
% Purpose: Create a tunable lead lag control that can
%          range from 0 to 360 degree lead.
%          This is accomplished by using a series of
%          6 lead lags.
%
% INPUTS:
% x(1)   = reference
% x(2)   = Time
% OUPUTS:
% u      = phase lead at same amplitude as reference

% 11-Nov-99 B Brunell - Created
% can use 6control\ldlg_fct_test.mdl for evaluation

% Choose the unstable frequency and the phase
global phase_i;         % declare as global so can read in from script
% phase_i   = 180;        % (deg) phase lead
omega     = 500;        % (Hz) input unstable frequency in HZ
omega     = omega*2*pi; % (rad/sec) convert to rad/s
num_ldlg = 6.0;         % Number of lead lags
T         = 0.0001;     % Update rate
% ++++++++++++++++++++++++++++++++++++++++++++++++++++++++
% Note: must define state as global every time fct called
global state
% initialize states
if x(2) == 0,
   state(1:num_ldlg)=0;
   disp('Initialized');
end
% set intermediate variable equal to the input
p = x(1);  % This puts a minus sign before lead lag
% divide phase over 6 lead lags
phase = phase_i/num_ldlg;  %
% compute alpha
alpha = 0.8007*exp(0.0474*phase);
% compute the lead and lag poles
zero = omega/(sqrt(alpha));
pole = alpha*zero;
lead = 1/zero;
lag = 1/pole;
% compute the gain to keep all phases at unity gain
k = 1/(0.511029*exp(0.023832*phase*num_ldlg));

% Create discrete lead lag controller
%------------------------------
% loox through lead lag num_ldlg times
for j=1:num_ldlg,
   % start Tustin lead lag control algorithm
   tmp_real_2 = p - state(j);
   tmp_real_3 = p - (tmp_real_2*(2*lag - T))/(2*lag+T);
   tmp_real_2 = (tmp_real_2*(2*lead+T))/(2*lag+T);
   tmp_real_4 = tmp_real_2;
   tmp_real_1 = tmp_real_4 + state(j);
   state(j)   = tmp_real_3;
   %
```

```
   p = tmp_real_1; % pass the output as input to next ld-lg
end % end of multiple lead lags
% ------------------------------
u = tmp_real_1*k; % set the function output
% ++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

# References


[ 1 ] J.W.S. Rayleigh. *The Theory of Sound,* volume 2. Dover, New York, 1945.

[ 2 ] K.R. McManus, T. Poinsot, and S.M. Candel. "A review of active control of combustion instabilities, 1992.

[ 3 ] G.J. Bloxsidge, A.P. Dowling, N. Hooper, and P.J. Langhorne. "Active control of an acoustically driven combustion instability". *Journal of Theoretical and Applied Mechanics*, supplement to vol. 6, 1987.

[ 4 ] W. Lang, T. Poinsot, and S. Candel. "Active control of nonlinear pressure oscillations in combustion chambers". *Journal of Propulsion and Power*, Vol. 8, No. 6:1282-1289, 1992.

[ 5 ] M. Fleifil and A.F. Ghoniem. "A dynamic model for unsteady heat release in a premixed flame in an oscillating flow". Technical report, Reacting Gas Dynamics Laboratory, MIT, Cambridge, MA, 1995.

[ 6 ] A.M. Annaswamy, M. Fleifil, Z. Ghoniem, and A.F. Ghoniem. Feedback Model of Thermoacoustic Instability in Combustion Processes. Technical Report 9502, Adaptive Control Laboratory, Department of Mechanical Engineering, MIT., May 1995.

[ 7 ] M. Fleifil, A.M. Annaswamy, J.P. Hathout, and A.F. Ghoniem. A model-based active control design for thermoacoustic instability. Technical report, Adaptive Control Laboratory, MIT, Cambridge, MA, 1996.

[ 8 ] J. Rumsey, M. Fleifil, A.M. Annaswamy, J.P. Hathout, and A.F. Ghoniem. "The role of active control in suppressing thermoacoustic instability". In *Proceedings of the American Control Conference*, Albuquerque, New Mexico, June 1997.

[ 9 ] A.M. Annaswamy, M. Fleifil, J. Rumsey, J.P. Hathout, and A.F. Ghoniem. An input-output model of thermoacoustic instability and active control design. Technical Report 9705, Adaptive Control Laboratory, Department of Mechanical Engineering, MIT., July 1997.

[ 10 ] A.M. Annaswamy, M. Fleifil, J.W. Rumsey, R. Prasanth, J.P. Hathout, and A.F. Ghoniem. Thermoacoustic Instability: Model-based Optimal Control Designs and Experimental Validation. Technical Report, Adaptive Control Laboratory, Department of Mechanical Engineering, MIT.

[ 11 ] J.P. Hathout, A.M. Annaswamy, M. Fleifil, and A.F. Ghoniem. "A model-based active control design for thermoacoustic instability". In *ASME International Mechanical Engineering Congress and Exposition*, Dallas, Texas, November 1997.

[ 12 ] A.M. Annaswamy, M. Fleifil, J.P. Hathout, and A.F. Ghoniem. "Impact of linear coupling on the design of active controllers for thermoacoustic instability". *Combustion Science and Technology*, 128:131-180, December 1997.

[ 13 ] J.W. Rumsey. *Model-based Active Control of Thermoacoustic Instability in Continuous Combustion Processes*. MIT Mechanical Engineering Thesis. Cambridge, MA, 1998.

[ 14 ] F.A. Williams. *Combustion Theory*. Addison-Wesley Co., Reading, MA, 1965.

[ 15 ] T. Myint-u and L. Debnath. *Partial Differential Equations for Scientists and Engineers*. Elsevier Science Publishing Co., New York, NY, 1987.

[ 16 ] G. Stein and M. Athans. The LQG/LTR procedure for multivariable feedback control design. *IEEE Transactions on Automatic Control*, 32:105-114, February 1987.

[ 17 ] A.E. Bryson, Y. Ho. *Applied Optimal Control*. Hemisphere Publishing Co., 1975.

[ 18 ] F.L. Lewis, V.L. Syrmos. *Optimal Control*. Wiley-Interscience, New York, 1995.

[ 19 ] C.B. Speedy, R.F. Brown, G.C. Goodwin. *Control Theory: Identification and Optimal Control*. Oliver & Boyd, Edinburg, 1970.

[ 20 ] A.P. Sage. *Optimum systems Control*. Prentice-Hall Inc., Englewood Clifs, N.J., 1968.

[ 21 ] J.P. LaSalle. "The Bang-Bang Principle". *Optimal and Self-optimizing control*. Edited by Rufus Oldenburger, 258-261, The MIT. Press, Cambridge, MA, 1966.

[ 22 ] D. McDonald, "Nonlinear Techniques for Improving Servo Performance," *National Electronics Conference,* VI (1950), 400-421.

[ 23 ] K. Youcef-Toumi. "Modeling and Control of Engineering Systems". Course number 2.151 notes, MIT, 1998.

[ 24 ] Lee Company. *Electro-fluidic Systems Technical Handbook*. 6$^{th}$ edition. The Lee Company Technical Center, Westbrook, CT, 1994.

[ 25 ] J.E. Slotine, W. Li. *Applied Nonlinear Control*. Prentice-Hall Inc., Englewood Cliffs, N.J., 1991.

[ 26 ] L. Ljung. *System Identification: Theory for the User*. 2$^{nd}$ edition. Prentice-Hall Inc., Upper Saddle River, N.J., 1999.

[ 27 ] Parr, T.P., Gutmark, E.J., Hanson-Parr, D.M., and Yu, K., *Proc. Of 8$^{th}$ ONR Prop. Mtg.*: 215-224 (1995).

[ 28 ] Yu, K., Wilson, K.J., and Schadow, K.C., "Scale-Up Experiments on Liquid-Fueled Active Combustion Control" AIAA Paper 98-3211, 1998.

[ 29 ] J.E. Tierno and J.C. Doyle. "Multimode active stabilization of a Rijke tube". In *DSC-Vol. 38*. ASME Winter Annual Meeting, 1992.

[ 30 ] Y.-T. Fung. V. Yang, and A. Sinha. Active control of combustion instabilities with distributed actuators. *Combust. Sci. And Tech.*, 78:217-245, 1991.

[ 31 ] R.K. Prasanth, R.K. Mehra, A.M., Annaswamy. "A System Identification Model of the MIT Laminar Combustor and Model Based Control". Technical Report, Adaptive Control Laboratory, Department of Mechanical Engineering, MIT, 1999.