

Adaptive Feedback Technique for Unmodeled Disturbance Rejection

by

Daniel H. Miller

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

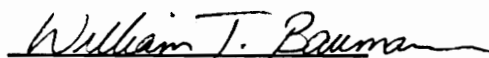
in partial fulfillment of the requirements for the degree of

Master of Science

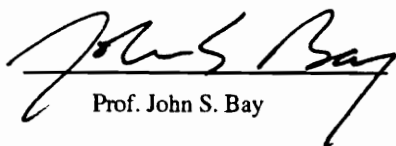
in

Electrical Engineering

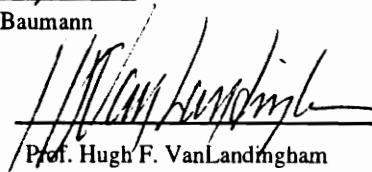
APPROVED:



Prof. William T. Baumann



Prof. John S. Bay



Prof. Hugh F. VanLandingham

April, 1995

Blacksburg, Virginia

c.7

LD
5655
V855
1995
M555
C.2

Adaptive Feedback Technique for Unmodeled Disturbance Rejection

by

Daniel H. Miller

Committee Chairman: William T. Baumann

Electrical Engineering

(ABSTRACT)

This work presents experimental results of a hybrid control technique that blends traditional fixed feedback control with adaptive feedback techniques. The objective is to provide damping for transient disturbances via the fixed controller and rejection of steady disturbances via the adaptive controller.

This technique differs from other control techniques in that it does not require a measurement of an external signal coherent with the disturbance, nor does it require knowledge of how the disturbance enters the system. As with all feedback techniques, stability is an issue. The adaptive controller is implemented to minimize stability problems and in the case of perfect system identification it is guaranteed stable.

The experimental results are from a cantilever beam with piezoceramic actuators and sensors. The fixed feedback controller results are compared to the adaptive feedback control results. The comparison will include results from disturbances resulting from a single harmonic, narrow band signal, and broad band signal.

Acknowledgments

I would like to take this opportunity to thank all those people whose contributions made this research become a reality. First, I would like to thank Dr. Baumann for serving as my advisor and committee chair. On more than one occasion, significant problems were overcome through his patience. I still want to know if the whiteboard in his office has some magical properties. I would also like to thank Dr. VanLandingham and Dr. Bay for their suggestions and serving on my committee.

Special thanks go to Dr. Harry Robertshaw for the use of his lab and for taking me paddling when it was obvious I had been in the lab too long. I'll really miss Friday afternoon's at the river. I would also like to thank Dan Cole and Dr. Will Saunders who explained to me how things actually worked. Additional thanks go to the guys in the lab (Darrell McAlister, Ben Fong, Thomas Snyder, and Peter Tappert) for their humor and ability to put up with me. Hey, we all made it.

I would also like to thank my family. To my brother/best friend David, thanks for everything and I'll see you on the river very soon. To my mother and father, I know I'll never be able to express how much your support and encouragement mean.

Contents

1 Introduction	1
1.1 Motivation	1
2 Adaptive-Q Theoretical Development	6
2.1 Overview	6
2.2 Control Development.....	6
2.3 Fixed Feedback Controller	13
2.4 Q_p Block	15
2.5 Adaptation Algorithm	16
2.6 Adaptive-Q Model	18
3 Experimental Setup and System Identification	19
3.1 Overview	19
3.2 Testbed	19
3.2.1 Beam Design	20
3.2.2 Sensor Amplifier.....	24
3.2.3 Power Amplifiers.....	27
3.2.4 Anti-Aliasing Filters.....	27
3.2.5 Smoothing Filters	29
3.2.6 DSP Input Protection	29
3.2.7 Digital Signal Processing Board.....	30
3.3 System Identification.....	31

4 Experimental Results	38
4.1 Overview	38
4.2 LQG Fixed Feedback	39
4.3 Disturbance Rejection Overview	42
4.3.1 Harmonic Disturbance	42
4.3.1.1 Experimental Results Harmonic Disturbance	43
4.3.1.2 Harmonic Disturbance Simulation	49
4.3.1.3 Conclusions Harmonic Disturbance	49
4.3.2 Optimal Disturbance Rejection Controller	51
4.3.3 Narrow Band Disturbance Rejection	55
4.3.3.1 Experimental Results Narrow Band Disturbance.....	55
4.3.3.2 Narrow Band Disturbance Simulation.....	60
4.3.3.3 Conclusions Narrow Band Disturbance	61
4.3.4 Broad Band Disturbance.....	63
4.3.4.1 Experimental Results Broad Band Disturbance	63
4.3.4.2 Broad Band Disturbance Simulation	65
4.3.4.3 Conclusions Broad Band Disturbance.....	66
4.4 Robustness.....	67
5 Conclusions	73
5. Conclusions	73

List of Figures

1.1.1 Overhead View Active Control in a Car	3
2.2.1 Transfer Function Plant Model.....	7
2.2.2 Two-Port Plant Model.....	8
2.2.3 Overview Control Block Diagram.....	11
2.2.4 Adaptive-Q Control Block Diagram.....	13
2.3.1 LQG Block Diagram.....	14
2.6.1 Complete Adaptive-Q System Model	18
3.2.1 Testbed Diagram	20
3.2.2 Strain Distributions Cantilever Beam.....	23
3.2.3 Beam Diagram.....	24
3.2.4 Sensor Amplifier.....	25
3.2.5 Sensor Amplifier Frequency Response	26
3.2.6 Anti-Aliasing Filter	28
3.2.7 Frequency Response Anti-Aliasing Filter.....	29
3.2.8 DSP Input Protection.....	30
3.3.1 Frequency Response (Model and Data) Control Input to System Output.....	33
3.3.2 Frequency Response (Model and Data) Disturbance Input to System Output	34
4.2.1 Output Noise (No Input)	41
4.2.2 Open and Closed Loop Frequency Response (Simulation).....	41
4.2.3 Frequency Response Experimental and Simulation	42

4.3.1 Disturbance 12.8 Hz No Control Input	44
4.3.2 12.8 Hz Disturbance LQG Controller.....	44
4.3.3 System Output No Control Compared to Converged LMS.....	45
4.3.4 3rd Order LMS 12.8 Hz Disturbance Converging and Converged	45
4.3.5 3rd Order RLS 12.8 Hz Disturbance Converging and Converged	46
4.3.6 100th Order LMS 12.8 Hz Disturbance Converging and Converged	46
4.3.7 8th Order RLS 12.8 Hz Disturbance Converging and Converged	47
4.3.8 100th Order LMS 70 Hz Disturbance Converging and Converged	47
4.3.9 8th Order RLS 70 Hz Disturbance Converging and Converged	48
4.3.10 100th Order LMS 195 Hz Disturbance Converging and Converged.....	48
4.3.11 Simulation 100th Order LMS No control Versus Converged LMS.....	49
4.3.12 Plant and Disturbance Model	52
4.3.13 Augmented Plant	53
4.3.14 LQGDR Block Diagram	54
4.3.15 Narrow Band Disturbance 70 Hz Center Freq.	56
4.3.16 Narrow Band Disturbance 70 Hz Center LQG Control	56
4.3.17 LMS 100th Order 70 Hz Center Converging and Converged.....	57
4.3.18 8th Order RLS 70 Hz Center Converging and Converged	57
4.3.19 100th Order LMS 195 Hz Center Converging and Converged.....	58
4.3.20 Reduced Control Authority LQG Control	59
4.3.21 100th Order LMS 70 Hz Center (Reduced LQG) Converging and Converged.....	59
4.3.22 70 Hz Center Open and Closed Loop LQGDR Simulation	60

4.3.23 195 Hz Center Open and Closed Loop LQGDR Simulation	61
4.3.24 70 Hz Center LQGDR Simulation Versus Experimental Adaptive-Q.....	62
4.3.25 195 Hz Center LQGDR Simulation Versus Experimental Adaptive-Q.....	62
4.3.26 Broad Band Disturbance (No Control).....	64
4.3.27 Broad Band LQG	64
4.3.28 100th Order LMS Broad Band Converging and Converged.....	65
4.3.29 LQGDR Simulation Broad Band Open and Closed Loop	66
4.3.30 Broad Band LQGDR Simulation Versus Experimental Adaptive-Q.....	67
4.4.1 Exact Model and Model with Zeros Moved -10 %	68
4.4.2 Exact Model and Poles Moved -10%	69
4.4.3 Exact Model and Poles and Zeros Moved -10%.....	69
4.4.4 Exact Model and Poles and Zeros Moved +10%	70
4.4.5 Control Output True Plant and Zeros Moved -10%.....	71
4.4.6 Control Output Poles Moved -10% and Poles and Zeros Moved -10%	71
4.4.7 Control Output Poles and Zeros Moved +10%.....	72

List of Tables

- 2.2.1 Definitions Transfer Function Plant Model..... 7
- 2.5.1 RLS Algorythm Summary..... 16
- 2.5.2 LMS Algorythm Summary..... 17
- 3.2.1 Definitions Natrual Freq. and Mode Shape Formulas..... 21
- 3.2.2 Natural Frequencies (Computed) 22
- 3.2.3 Strain Nodes..... 23
- 3.2.4 PZT Locations..... 24
- 3.2.5 Component Values Sensor Amplifier..... 26
- 3.2.6 Component Values Anti-Aliasing Filter 28
- 3.2.7 Component Values DSP Protection Circuit..... 30
- 3.3.1 Experimentally Determined Natural Frequencies 34
- 3.3.2 Model A-Matrix 35
- 3.3.3 Model B-Matrix..... 36
- 3.3.4 Model E-Matrix..... 37
- 3.3.5 Model C-Matrix..... 37
- 4.3.1 Convergence Times 50

Chapter 1

Introduction

1.1 Motivation

Active Structures consist of a structure with sensors, actuators and a control system. The control system, through the sensors and actuators, can change structural characteristics based on environmental conditions. This ability to alter the system response is commonly called active control. Structural characteristics of interest include radiated noise and vibrations. In this work, disturbances are the name given these characteristics. Thus, a disturbance is any unwanted signal on the structure.

Control techniques that address the disturbance rejection problem take numerous forms. Two broad classes are feedback and feedforward techniques. Both of these techniques have seen applications, feedforward being the most successful. However, they all bring with them certain requirements. The requirements of the different approaches are best illustrated with an example. The example is not a review of all the possible control techniques. The objective is to show how the proposed control technique compares with popular techniques.

Consider an active noise suppression system for a car. In this case we are going to take a simplistic view of the problem. More realistic noise control systems for car and airplane applications have been reported in the literature[10][11]. Our problem addresses a single noise source and a single cancellation point. The noise cancellation point is the driver's seat. The objective is to cancel the noise that the driver hears. The example's flaw is the driver would want to hear certain noises, such as a police siren, but would not want to hear others like road noise. A sensor (e.g. microphone) measures the noise at the driver's seat. Next, the system requires a method to alter the sound content in the car. A natural choice is the car audio system with the speakers being the control actuators. A controller in the car radio will generate the control signal. The next step is to identify a noise source. The car engine is a good source. The engine compartment contains a sensor to measure this noise. A diagram of the car system is shown in figure 1.1.1.

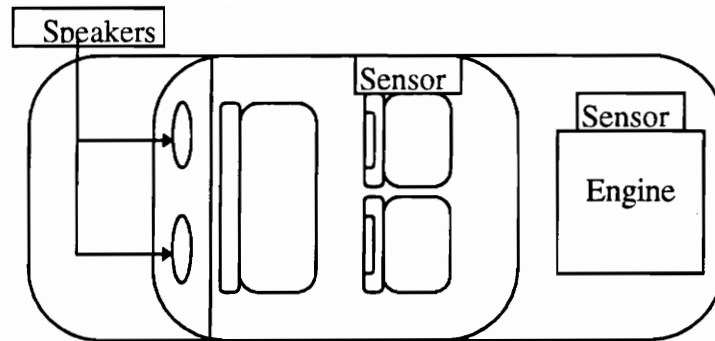


Figure 1.1.1 Overhead View Active Control in a Car

Adaptive feedforward control is one method to solve the noise problem. The idea is to measure a signal that is similar (coherent) with the disturbance. In the example, the sensor on the engine serves this purpose. The disturbance signal allows the controller to adapt, on-line, to the disturbance and feed a signal forward to cancel the noise. The approach works well and has been applied to a similar problem in an aircraft cabin[11]. However, the controller needs a signal coherent to the disturbance. In the above example, it must have the engine sensor.

Adaptive feedback is another approach that has seen some application. The controller adapts the control system (the feedback gains) on-line in an attempt to determine the correct feedback mechanism to attenuate the noise. The problem is that traditional fully adaptive feedback controllers experience stability problems. These stability issues arise from the non-linear nature of adaptive controllers[2].

Another technique is a fixed feedback control. A system model is used to design, off-line, a feedback controller to attenuate the noise output. The problem is that to achieve acceptable fixed feedback performance requires a precise mathematical model[8][17]. The model specifies how the control signal and the disturbance (engine noise) enter and affect the car. The modeling process introduces errors. The deviations from the true system cause the fixed feedback controller to sacrifice performance to overall controller robustness. Additionally, a model from disturbance input to system output is not always available. These effects, in addition to the fact that the system's response will change over time, yield a fixed feedback controller that rarely meets the performance of the feedforward technique.

The feedforward and feedback control techniques have a similar problem. They must know something about the disturbance signal. Either how it enters the plant or a signal coherent with it. The purely adaptive feedback approach has stability problems. The work that follows presents a technique that blends a fixed feedback controller with an adaptive controller into one unified approach. The approach seeks to divorce the system from the unwanted effects of fixed feedback, feedforward and adaptive feedback, while realizing the benefits of each. In the proposed technique, the only requirement is a model of how the control signal affects the plant. The disturbance model is not a requirement. The disturbance and exactly how it affects the system is determined on-line by the adaptive control section. Thus, our approach requires no more modeling than the standard feedforward approach, but more modeling than a fully adaptive approach.

The work that follows presents simulation and experimental results for the unified control technique. Adaptive-Q disturbance estimate(Adaptive-Q) feedback is the name given the unified approach. The technique uses a fixed feedback controller to provide damping for transient disturbances while an adaptive controller provides damping for steady disturbances. Experimental results from a cantilever beam with bonded piezo-ceramic(PZT) actuators provide experimental validation of the technique.

The contribution of this work is the validation of an as yet experimentally unproven control technique(Adaptive-Q). The technique is validated by comparing simulations of the control technique with the actual experimental results. Moreover, an experimental testbed, with well defined characteristics and control code, is now available for future work.

Chapter 2

Adaptive-Q Theoretical Development

2.1 Overview

Chapter 2 presents a theoretical outline of disturbance rejection via the Adaptive-Q technique. The technique is explored in detail in a paper by Tay and Moore[19]. The specific development in this section relies heavily on Baumann's work[3].

2.2 Control Development

The objective of a disturbance rejection control is to make the transfer function from disturbance input to system output have a desirable frequency response. In most cases, the controller seeks to attenuate or eliminate the disturbance. Therefore, to begin the theoretical discussion, a plant model concerning these transfer functions is shown(Figure 2.2.1). It should be noted, that in many cases the output and y shown in

figure 2.2.1 are in many cases the same signal. The signals differ in cases when the observed output signal and the feedback signal are not the same signal.

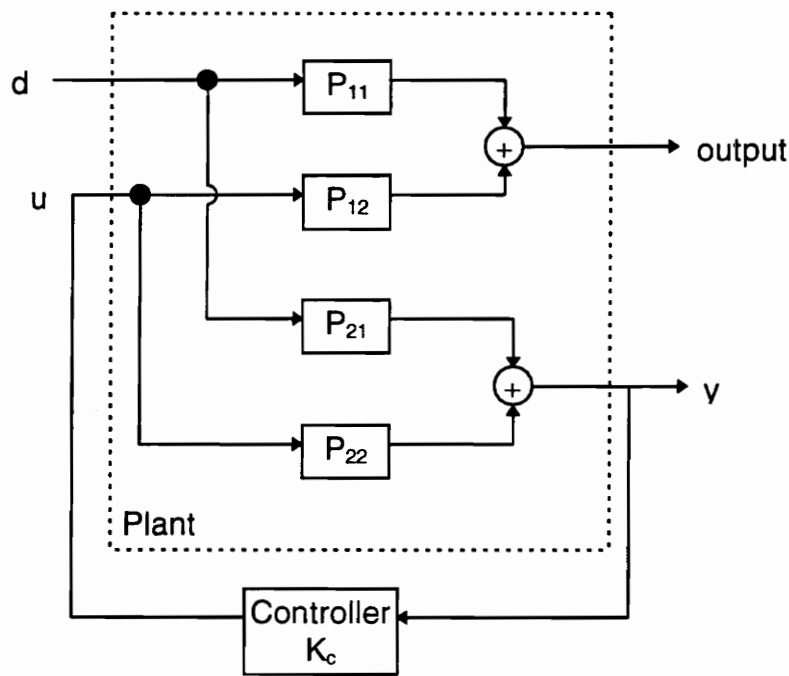


Figure 2.2.1 Transfer Function Plant Model

The signal and block definitions are:

Table 2.2.1 Definitions Transfer Function Plant Model

u	Control input
d	Disturbance input
y	Feedback output
K _c	Controller
output	Error output
P ₁₁	Transfer function from disturbance input to system output
P ₂₁	Transfer function from disturbance to feedback output
P ₁₂	Transfer function from control input to system output
P ₂₂	Transfer function from control input to feedback output

The plant model leads to the following system input/output equations.

$$\begin{aligned}\text{output} &= P_{11}d + P_{12}u \\ y &= P_{21}d + P_{22}u \\ u &= K_c y\end{aligned}$$

The transfer function from disturbance input to system output is:

$$\text{output} = P_{11}d + P_{12}K_c(I + P_{22}K_c)^{-1}P_{21}d$$

Inspection of the output equation shows that the equation, assuming an open loop stable plant, has a component that can go unstable. The component is:

$$K_c(I + P_{22}K_c)^{-1}$$

Therefore, a stable control must maintain the stability of this component. The controllers that have these characteristics are the stabilizing controllers.

To define the set of stabilizing controllers the plant model is redrawn.(figure 2.2.2)
The model structure is common to H_∞ control design. A quick reference concerning this model structure is contained in Shahian and Hassul [18].

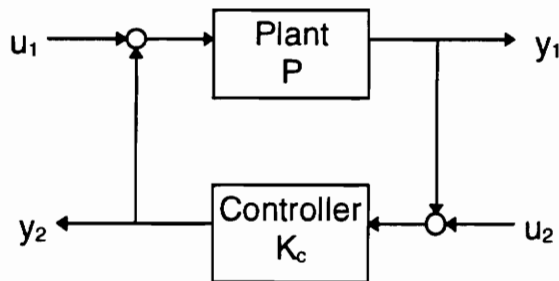


Figure 2.2.2 Two-Port Plant Model

In the model, we have added a dummy input and output (u_2 and y_2) to define controller internal signals. From figure 2.2.2, stability is defined as the transfer functions from

$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$ to $\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ all being stable this guarantees internal as well as external stability. The

transfer functions are:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} PK_c(I + K_cP)^{-1} & -PK_c(I + PK_c)^{-1} \\ K_cP(I + K_cP)^{-1} & K_c(I + PK_c)^{-1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

Define:

$$Q_p = K_c(I + PK_c)^{-1}$$

Substituting this value, the equations are simplified to yield:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} I - PQ_p & PQ_p \\ Q_pP & Q_p \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

Therefore, if Q_p is stable then stability as defined above will result. The transfer function from disturbance input to system output can be written as:

$$\text{output} = P_{11}d + P_{12}Q_pP_{21}d$$

Thus, as proven by Moore and Tay[19], as long as Q_p is bounded-input bounded-output (BIBO) stable then the resulting transfer function between the input and the output will be BIBO stable. Additionally, Moore and Tay[19] show that by varying the Q_p parameter the overall controller can sweep over all stabilizing controllers for the plant. The ability to vary Q_p to sweep the stabilizing controllers defines the adaptive sections purpose. Inherent is that the adaptive block must be BIBO stable to ensure the resulting

system stability. The adaptive scheme must also change parameters to sweep the set of stabilizing controllers to find the desired controller to eliminate or attenuate the disturbance signal. An example of a Q_p structure, having these stability characteristics, is a finite impulse response(FIR) filter. The FIR filter due to its finite length will be unable to sweep all stabilizing controllers. Simulations have shown that this subset is more than adequate[3]. The next task is to apply the Adaptive-Q approach to the beam problem.

An approach to the beam control is to adaptively vary Q_p until the control objective is achieved. This approach would be a purely adaptive feedback. One difficulty with this approach is determining a rule to vary Q_p using no a priori knowledge about the plant. Additionally, a purely adaptive control would provide little rejection to transient disturbances since it would not be able to adapt fast enough. A better method is to use the system knowledge to design a fixed feedback controller to provide a fixed amount of system damping. The application of Adaptive-Q theory allows the overall control to gain back performance lost due to modeling uncertainties. An additional benefit is that the combined control structure is able to adapt to unknown disturbances that may enter the plant. To proceed with this idea a fixed feedback control structure that can use the adaptive-Q theory is explored.

One of the many candidate control structures is a state feedback control in combination with a state estimator(figure 2.2.3).

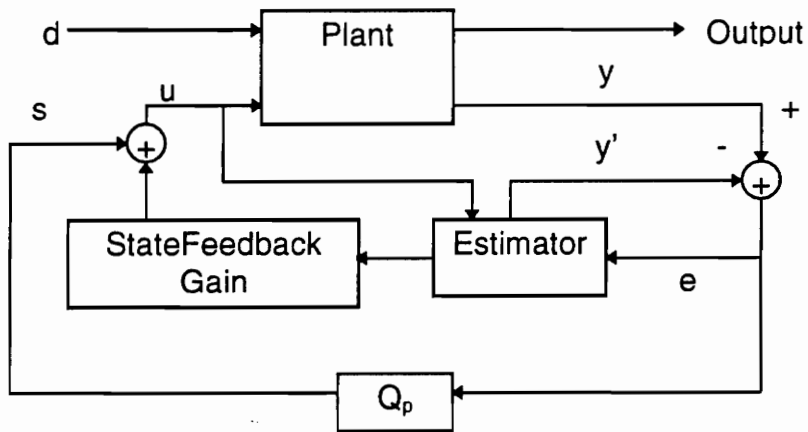


Figure 2.2.3 Overview Control Block Diagram

In the structure, the control input affects the estimator (assuming a perfect system identification) in the same manner as the plant. This causes the estimation error to remain constant. Thus, the transfer function from s to e is zero and Q_p cannot drive the system unstable since it does not create a feedback loop. However, by varying the transfer function of the Q_p block the system sweeps out all controllers that stabilize the plant. A linear quadratic gaussian (LQG) control is the choice for the state feedback/state estimator design.

The next problem is how to adjust Q_p to give the desired system output. The system output is broken down as follows:

$$\text{output} = P_{11}d + P_{12}Q_p e$$

If we assume that y, u, e and s are scalars then we can write:

$$\text{output} = P_{11}d + Q_p P_{12}e$$

or in standard output error format:

$$P_{11}d = \text{output} - Q_p P_{12}e$$

A restatement of the problem is to find the Q_p that minimizes system output. In other words, the implemented algorithm is the filtered-X LMS[21]. The fact that Q_p sweeps all the stabilizing controllers yields a system where stability is guaranteed assuming perfect system identification.

The actual beam implementation is simply the implementation of the filtered-X LMS algorithm. In our case, the estimation error is fed into the plant model. The resulting signal is the input to the adaptive algorithm. This value, along with the present system output value, is used to adjust the weights of Q_p , a FIR filter, to try to minimize the system output. A block diagram of the system is shown in fig(2.2.4).

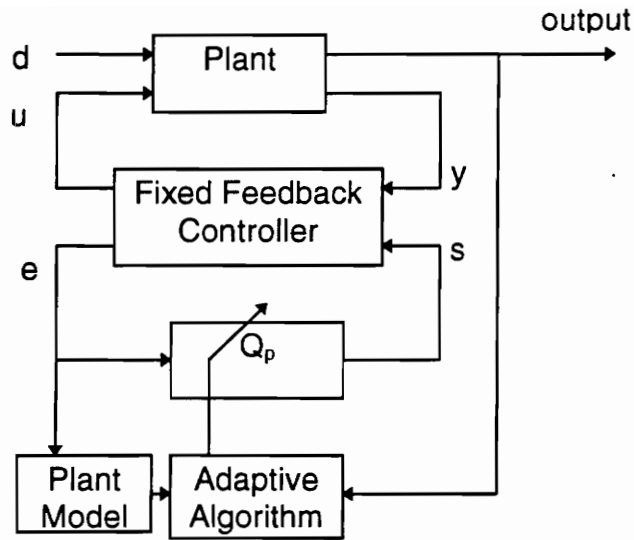


Figure 2.2.4 Adaptive-Q Control Block Diagram

The next sections discuss the design of the individual blocks shown in figure 2.2.4 for the cantilever beam.

2.3 Fixed Feedback Controller

The control structure for the fixed feedback controller is a linear quadratic gaussian (LQG) controller. Many reference books exist that explain how to design an LQG controller. One good reference is by Kirk[15]. This section relies heavily on his work.

The LQG controller is an implementation of linear optimal control techniques. The controller contains a linear quadratic regulator(LQR) controller in combination with a linear quadratic estimator (LQE). The LQE section is a Kalman filter. The LQG design is based on a cost function. The cost function allows the designer to weigh the control

output verses the system states. The resulting controller will be optimal in the sense of minimizing the cost function. A detailed discussion of the cost function and the associated equations is in Anderson and Moore[1]. Additional work on LQG control designs is in Rubenstein[17].

A block diagram of the LQG control structure is shown in figure 2.3.1.

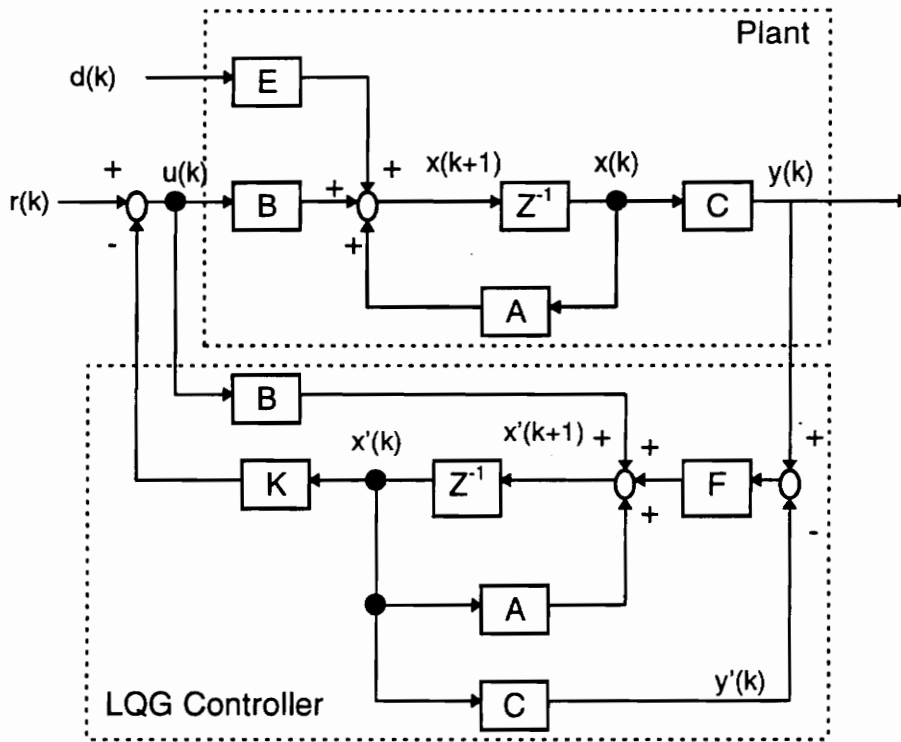


Figure 2.3.1 LQG Block Diagram

The discrete time (D-T) controller state equations are as follows.

$$x(k+1) = Ax(k) + Bu(k) + Ed(k)$$

$$y(k) = Cx(k)$$

$$u(k) = -Kx'(k) + r(k)$$

The D-T controller state equations are:

$$\begin{aligned}x'(k+1) &= Ax'(k) + Bu(k) + F(y(k) - y'(k)) \\y'(k) &= Cx'(k)\end{aligned}$$

Combination of above equations results in a single set of D-T state equations.

$$\begin{aligned}\begin{bmatrix} x(k+1) \\ x'(k+1) \end{bmatrix} &= \begin{bmatrix} A & -BK \\ FC & A - BK - FC \end{bmatrix} \begin{bmatrix} x(k) \\ x'(k) \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} r(k) + \begin{bmatrix} E \\ 0 \end{bmatrix} d(k) \\ \begin{bmatrix} y(k) \\ y'(k) \end{bmatrix} &= \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} x(k) \\ x'(k) \end{bmatrix}\end{aligned}$$

The state equations shown above include the method the disturbance (E matrix) enters the plant. This is done for completeness. The fixed feedback controller used in the Adaptive-Q control will not use this model. The solution of the discrete time version of the Ricatti equation determines the values for F and K. Specifically, the MATLAB commands DLQR and DLQE perform these calculations. Chapter 3 will discuss the actual implementation of the LQG controller.

2.4 Q_p Block

The Q_p block is a function that needs to change its frequency response. In other words, the block is an adaptive filter. Varying the filter coefficients allows the filter to sweep the stabilizing control structures. The Q_p block also must remain stable with changing filter coefficients. A FIR filter structure has these properties and is the Q_p block used in the experiment and simulations. The choice of an FIR filter and the corresponding number of coefficients causes the control structure to be able to only sweep a subset of the stabilizing controllers. In simulation, this subset has proven to be more than adequate[3].

2.5 Adaptive Algorithm

The adaptive algorithm is responsible for adjusting the Q_p filter weights(FIR filter coefficients). The algorithm to perform this task is a choice for the system designer.

Haykin's [13] adaptive filter book is a good reference. In this book, numerous algorithms are explored in detail. The algorithms used for our design will be the least mean squares (LMS) algorithm and the recursive least squares (RLS) algorithm. From Haykin [13] the RLS and LMS algorithm summaries are:

Table 2.5.1 RLS Algorithm Summary

RLS Algorithm
Define: $\hat{w}(n)$ = tap weight vector $k(n)$ = gain vector $\alpha(n)$ = priori estimation error $P(n)$ = correlaion matrix inverse Initialization: $P(0) = \delta^{-1}I$ δ = small positive number $\hat{w}(0) = 0$ for each time step compute: $k(n) = \frac{\lambda^{-1}P(n-1)u(n)}{1 + \lambda^{-1}u^T(n)P(n-1)u(n)}$ $\alpha(n) = d(n) - \hat{w}^t(n-1)u(n)$ $\hat{w}(n) = \hat{w}(n-1) + k(n)\alpha^*(n)$ $P(n) = \lambda^{-1}P(n-1) - \lambda^{-1}k(n)u^T(n)P(n-1)$

Table 2.5.2 LMS Algorithm Summary

LMS Algorithm

define:

M = number of taps

μ = step - size

$u(n)$ = M by 1 input vector

$d(n)$ = desired response

$\hat{w}(n+1)$ = estimate of weights

for each time step compute:

$$e(n) = d(n) - \hat{w}^T u(n)$$

$$\hat{w}(n+1) = \hat{w}(n) + \mu u(n) e^*(n)$$

The LMS algorithm has the advantages of being relatively simple and numerically efficient to compute. The disadvantage of the LMS algorithm is that it has a fixed step size. The RLS algorithm has the advantage that it uses a variable step size. The disadvantage is that it is numerically intensive to compute.

2.6 Adaptive-Q Model

The final completed model is shown in figure 2.6.1. The control simulation and the beam control follow this model. Use this model as a guide when studying the simulation code contained in Appendix A.

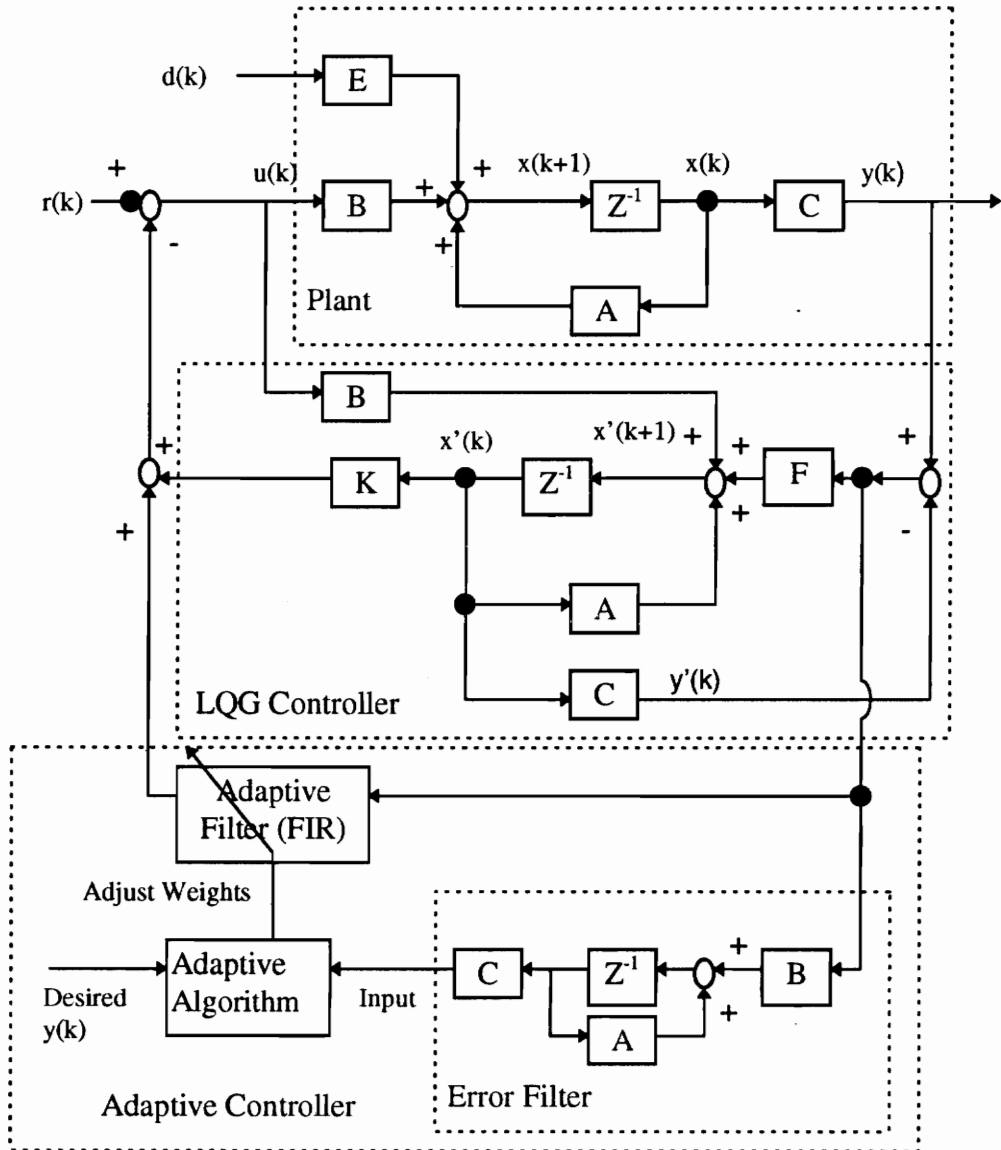


Figure 2.6.1 Complete Adaptive-Q System Model

Chapter 3

Experimental Setup and System Identification

3.1 Overview

Chapter 3 presents the testbed design and implementation. The chapter includes the cantilever beam design, testbed electronics design, and the mathematical system identification.

3.2 Testbed

To experimentally test the Adaptive-Q theory requires a testbed. The testbed consists of a cantilever beam with PZT sensor and actuator pairs, signal conditioning

circuits, and a digital signal processing (DSP) board to implement the control algorithm.

The testbed block diagram is shown in figure 3.2.1.

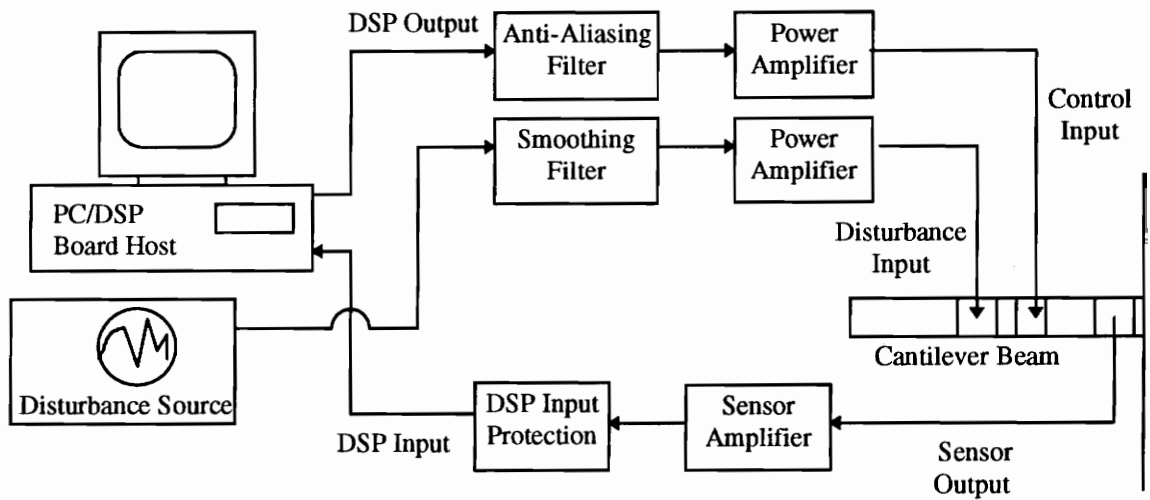


Figure 3.2.1 Testbed Diagram

A discussion of the blocks follows in the next sections.

3.2.1 Beam Design

The experimental structure is a cantilever beam with piezoceramic(PZT) actuators and sensors. The cantilever beam is chosen for the structures well defined characteristics. The desirable characteristics include mathematical formulas for determining the beam natural frequencies and the mode shapes. The natural frequency is the frequency at which the beam will tend to vibrate once excited[5]. A mode shape is the relative displacement of a point along a beam when vibrating at a specific natural frequency[5]. The actuators and sensor, however, change these natural frequencies. The change is small and models do exist to perform the calculation including these devices[12]. However, the purpose of

the experiment is to validate a control theory and thus an exact cantilever beam model is not a requirement. A simpler design is done without sensors and actuators. The cantilever beam mode shape and natural frequency formulas are taken from Blevins[5].

The formulas are:

Natural Frequency:

$$f_i = \frac{\lambda_i^2}{2\pi L^2} \left(\frac{EI}{m} \right)^{1/2} \quad i = 1, 2, 3, \dots$$

Mode Shapes:

$$y_i \left(\frac{x}{L} \right) = \cosh \left(\frac{\lambda_i x}{L} \right) + \cos \left(\frac{\lambda_i x}{L} \right) - \sigma_i \left(\sin \left(\frac{\lambda_i x}{L} \right) - \sinh \left(\frac{\lambda_i x}{L} \right) \right)$$

The variable definitions are:

Table 3.2.1 Definitions Natural Freq. and Mode Shape Formulas

Symbol	Definition
E	modules of elasticity
I	area moment of inertia
L	span of beam
λ	constant ref. Blevins[5]
σ	constant ref. Blevins[5]
x	distance along span of beam
m	mass per unit length

The beam design goal is to have four natural frequencies within the Nyquist rate for the discrete time (DT) control. It is desirable to have the fifth modes position away from the Nyquist rate to avoid aliasing (frequency fold-over) in the DT controller. The system sampling rate is 1000 Hz. This places the Nyquist rate at 500 Hz. The beam material is aluminum due to availability. The beam's width and thickness are 1" and 1/16"

respectively. The width and thickness correspond to available stock. This leaves length the only design parameter. Using the natural frequency formula (reference NFREQCF.M Appendix A), the beam length is iteratively calculated to be 12''. The length corresponds to the following beam natural frequencies.

Table 3.2.2 Natural Frequencies (Computed)

Mode Number	Natural Frequency (Hz)
1	13.94
2	87.40
3	244.72
4	479.56
5	792.7552

The values are just initial design goals. System identification techniques, on the completed system, determine the actual values.

An additional beam design consideration is the location of the sensors and actuators. PZT sensors and actuators perform best when located in areas of high strain[9]. The second derivative of the mode shape formula determines the strain distribution[9]. Figure 3.2.2 shows the first four beam strain distributions. The point where a strain distribution crosses zero is a location of zero strain. Therefore, the actuator and sensor placement should avoid the strain nodes whenever possible. Additional information concerning the use of PZT as sensors and actuators is contained in Cole[8], Hagood[12] and Crawley[9]. The strain distributions and strain nodes are calculated using the MATLAB m_files STRAINCF.M and STRAINZ.M contained in Appendix A.

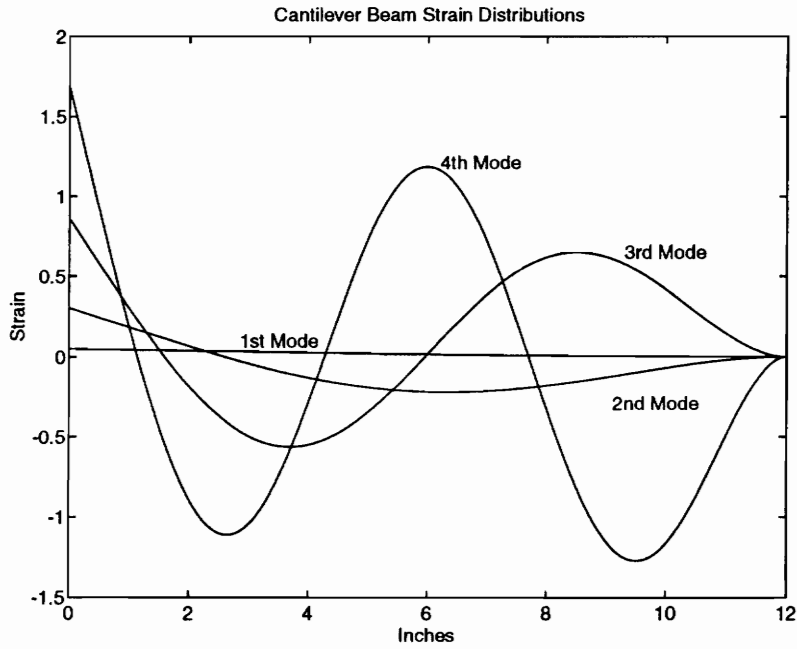


Figure 3.2.2 Strain Distributions Cantilever Beam

The zero crossings are:

Table 3.2.3 Strain Nodes

Mode #	Zero Crossing		
	#1	#2	#3
1	N/A	N/A	N/A
2	2.592	N/A	N/A
3	1.5876	5.958	N/A
4	1.1328	4.2708	7.7004

The PZT locations correspond to avoiding as many strain nodes as possible.

Additionally, the PZT locations are weighted towards the areas of highest strain. The

PZT material is 1''x1''x1/100''. The PZT placement is:

Table 3.2.4 PZT Locations

PZT #	PZT Beam Location (inches)	
	Start	End
1	.1	1.1
2	3.35	4.45
3	4.85	5.85

Figure 3.2.3 shows the completed beam with the PZT actuators and sensors.

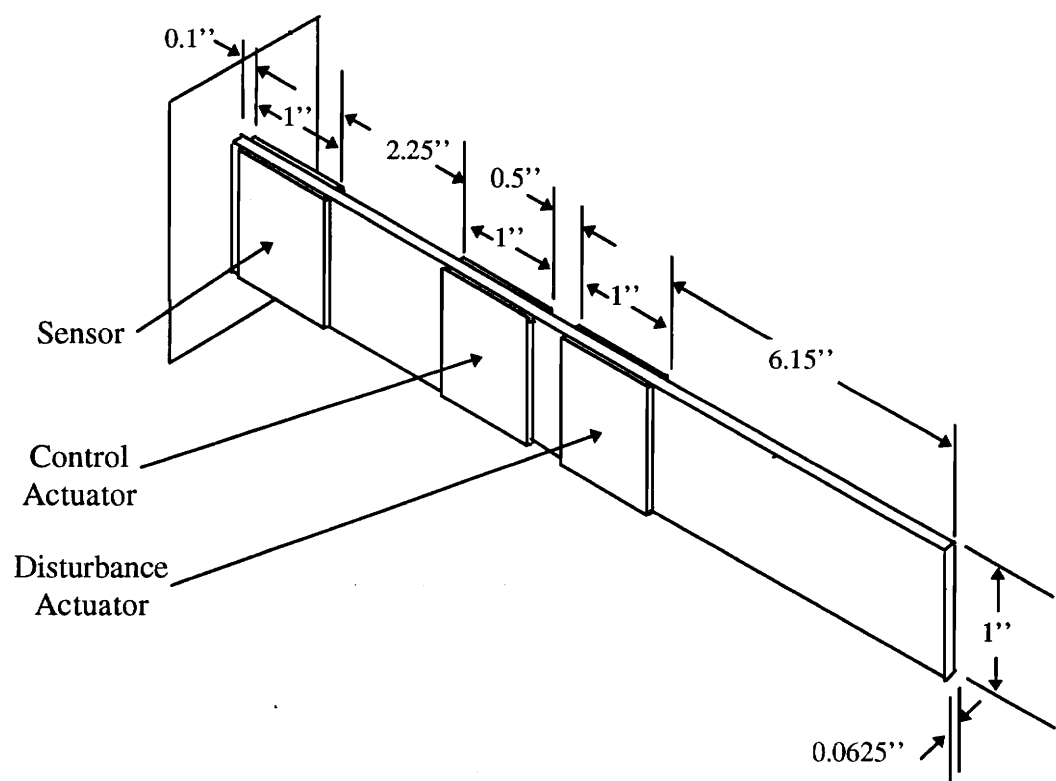


Figure 3.2.3 Beam Diagram

3.2.2 Sensor Amplifier

The sensor amplifier design undertaken in this section is from Cole [8]. A simplified PZT model is a voltage source with a capacitor in series. The series capacitance

is typically small on the order of 25 nF. The capacitance in conjunction with the measuring device input impedance creates a high pass filter with cutoff frequency

$\frac{1}{2\pi RC_p}$ [8]. Thus, if the input impedance is not very large, the sensors low frequency

performance degrades. To alleviate this problem a junction field-effect transistor(JFET)

input operational amplifier (Op-Amp) circuit is constructed. The JFET Op-Amp choice is

due to the high input impedance of the device. Specifically, the LF411 is the device used.

The next challenge is that the PZT sensors are extremely sensitive to low frequency

vibrations[8]. This can cause DC shifts in the sensor signal which are undesirable.

Additionally, the voltage produced by the PZT sensor is relatively small compared with

the sensitivity of the DSP board's Analog to Digital (A/D) inputs. Therefore, the Op-Amp

circuit must also perform as a high pass filter and add gain to the system. The Op-Amp

circuit design chosen is a Sallen-Key high pass circuit(figure 3.2.4).

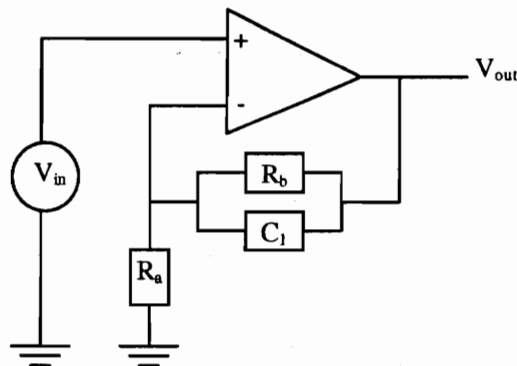


Figure 3.2.4 Sensor Amplifier

The transfer function for the Sensor Amplifier is:

$$\frac{V_{out}}{V_{in}} = \frac{s^2 C_1 R_a R_b + s(R_a + R_b)}{s^2 C_1 R_a R_b + s(R_a + C_1 R_a R_b) + R_a}$$

The component values are:

Table 3.2.5 Component Values Sensor Amplifier

Component	Value
R_a	1.0 K Ω
R_b	10.0 K Ω
C_1	0.41 nF

Figure 3.2.5 shows the sensor amplifier frequency response.

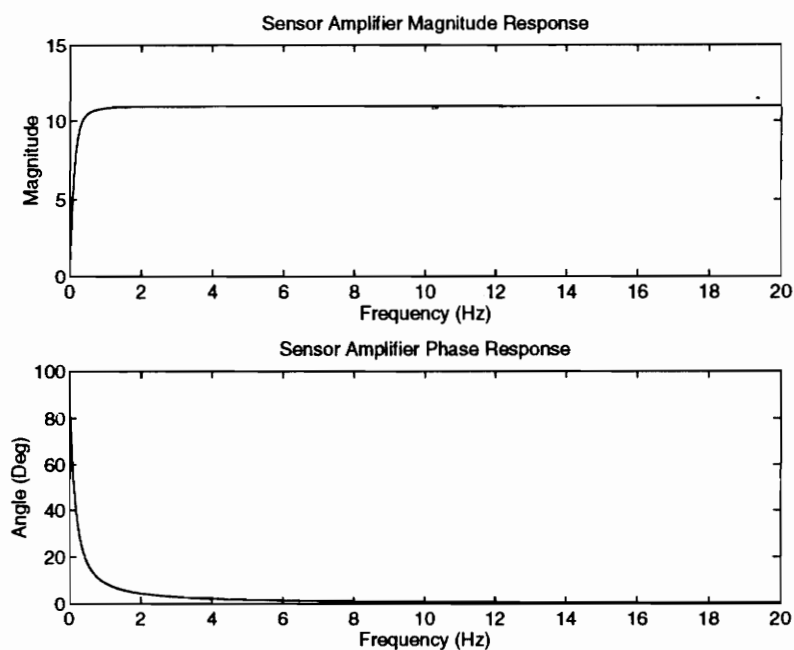


Figure 3.2.5 Sensor Amplifier Frequency Response

3.2.3 Power Amplifiers

The DSP board's digital to analog (D/A) outputs are relatively low output power and voltage devices. The power amplifier serves as a buffer and amplifier for the DSP outputs. The amplifier provides a gain of 2.5V/V to the system.

3.2.4 Anti-Aliasing Filters

A digital control system requires anti-aliasing filters to avoid aliasing in the discrete time controller. Digital signal processing books explain this in detail. Oppenheim and Schaffer [16] is a recommendation. The anti-aliasing filter is a low pass filter. The cutoff frequency is approximately half the sample rate. The problem is that a low pass filter that has an abrupt cutoff is a higher order filter. Remembering that the final system contains the anti-aliasing filter model as well as the cantilever beam model, the lower the anti-aliasing filter order the lower the system model order. Additionally, the high order filter introduces phase delay into the system that is hard to model. A trade-off between the performance of the anti-aliasing filter and the order of the system model must be reached. In an effort to keep the system model order small, the anti-aliasing filter is a second order lowpass filter. The cut-off frequency is moved back from the half sample rate in an attempt to give the resulting system good performance albeit over a reduced frequency range. The anti-aliasing filter design implemented is a unity gain Sallen-Key Op-Amp. see figure 3.2.6

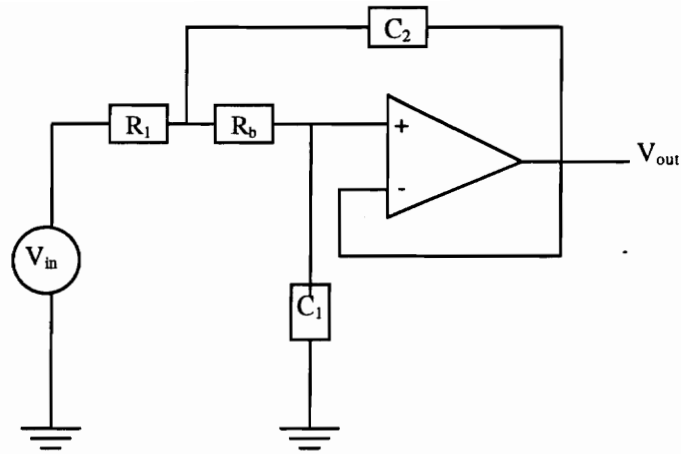


Figure 3.2.6 Anti-Aliasing Filter

The transfer function for the sensor amplifier is:

$$\frac{V_{out}}{V_{in}} = \frac{1}{s^2 R_1 R_2 C_1 C_2 + s C_1 (R_1 + R_2) + 1}$$

The component values are:

Table 3.2.6 Component Values Anti-Alias Filter

Component	Value
R_1	$1.0 \text{ M}\Omega$
R_2	$1.0 \text{ M}\Omega$
C_1	220 pF
C_2	470 pF

Figure 3.2.7 shows the sensor amplifier frequency response.

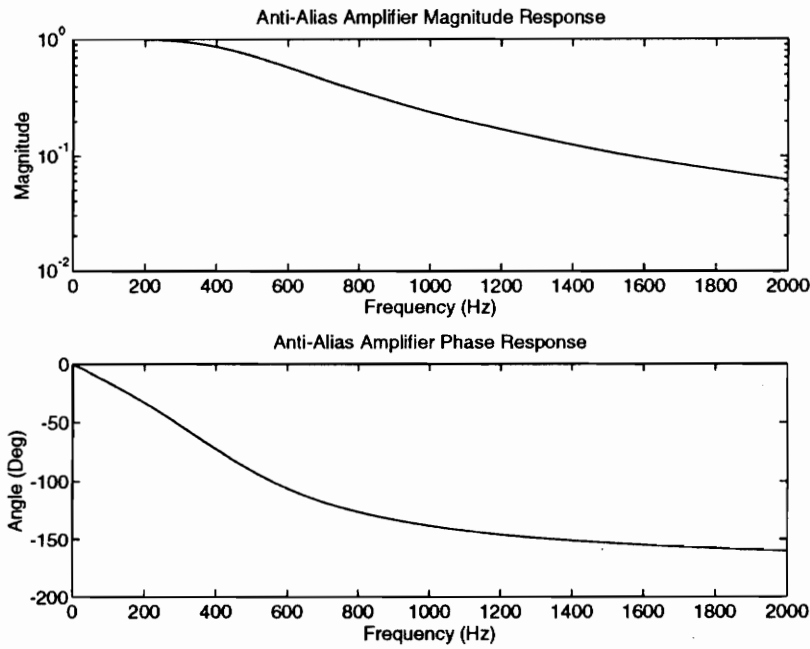


Figure 3.2.7 Frequency Response Anti-Aliasing Filter

3.2.5 Smoothing Filters

The smoothing filter is a low pass filter with a 3 dB cutoff at 500 Hz. The 500 Hz cut off frequency corresponds to the Nyquist rate. The filters are responsible for band limiting the disturbance input when the disturbance is generated digitally. The noise filters are commercially manufactured (Frequency Devices Model Number 9002) 8th order butterworth filters.

3.2.6 DSP Input Protection

The DSP board used in the experiment is limited to input values in the range of -3 volts to +3 volts. Additionally, the input voltages higher than these values can cause the input section to fail. Therefore, a diode circuit limits the linear input voltage range on the

order of ± 1.0 volts. These limits are very conservative. However, the system output is ultimately limited by the control inputs. Thus, I sought to be over protective of the DSP board inputs.

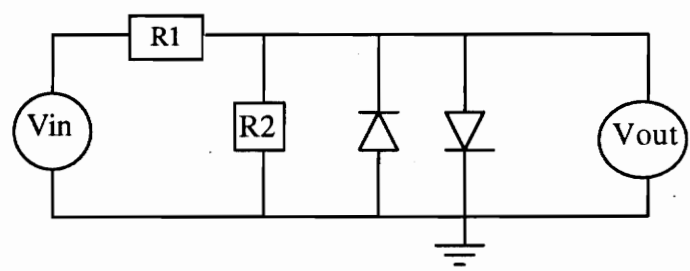


Figure 3.2.8 DSP Input Protection

Table 3.2.7 Component Values DSP Protection Circuit

r1	39 K Ω
Diode	50 V Max
r2	2.2 K Ω

3.2.7 Digital Signal Processing Board

The DSP board is a TMS320C30 based DSP board manufactured by Spectrum Signal Processing. The TMS320C30 chip is a 32-bit floating point DSP microprocessor. The floating point board eliminates the problems associated with a fixed point arithmetic. The board has two analog inputs and two analog outputs. The maximum voltages associated with these input output pairs is ± 3 volts. The TMS320C30 board plugs into an expansion slot on a PC AT bus and contains the DSP microprocessor and associated memory. The PC microprocessor and the DSP microprocessor communicate through dual-port memory on the DSP board.

The software tools supplied with the board allow the DSP microprocessor code to be written in C or assembly language. For the experiment, the majority of the software is written in C while only certain time critical functions are written in assembly language.

The DSP board requires a program that runs on the PC microprocessor to download, initialize and communicate with the DSP microprocessor. The program that performs this function is written in C and uses libraries supplied by Spectrum to communicate with the board.

3.3 System Identification

Many modern control techniques use a mathematical system model during the design phase. The model describes how the inputs influence the outputs. Generating the system model is a system identification problem. The technique, in our case, is to inject a pseudo-random sequence into the system. The input and the resulting system output are stored for frequency analysis. An instrumental variables technique, implemented in MATLAB, performs the curve fit to the experimental data. Ideally, the system model consists of poles and zeros associated with each system component. The system components that determine these poles and zeros are the anti-aliasing filter, the sensor amplifier and the beam itself. Therefore, the minimum system model order is 2 poles and zeros for each amplifier and 8 poles and zeros for the beam. The resulting system model is 12th order. Inherent in the beam data collection method are additional effects other than just the aforementioned components. Examples of these error sources are environmental noise and aliased modes. Thus, the data needs to be overfit. Additional poles and zeros model these error sources. Some of these error sources require modeling while others do

not. The end result is that the curve fitting process needs to begin with a higher order model than required. In our specific case, good results are obtained with 25th order models. A 25 order model significantly increases system complexity. The added complexity results in a control algorithm that takes longer to execute. Since the DSP board implements the algorithms in real time, a smaller model is desirable. The system complexity reduction is achieved using model reduction techniques. The MATLAB function DMODRED and DBALREAL perform the model reduction. These techniques consist of reordering the system so that the poles and zeros that have the least effect on the system model can be eliminated. The method chosen to reduce the model is an iterative approach. The poles and zeros identified as having the least effect on the model are removed. The frequency response is then redone. A comparison is then made between the data and the model. This process is repeated until the model has the necessary accuracy and model order. The final order chosen for the beam is 15th order. The 15th order model gave good magnitude and phase response. Figure 3.3.1 shows the frequency response from control input to system output for the beam (experimental data) compared with the model(simulation). A lesson learned during the system identification process is that phase response is very important. Attention to this match is as important if not more important than the magnitude match.

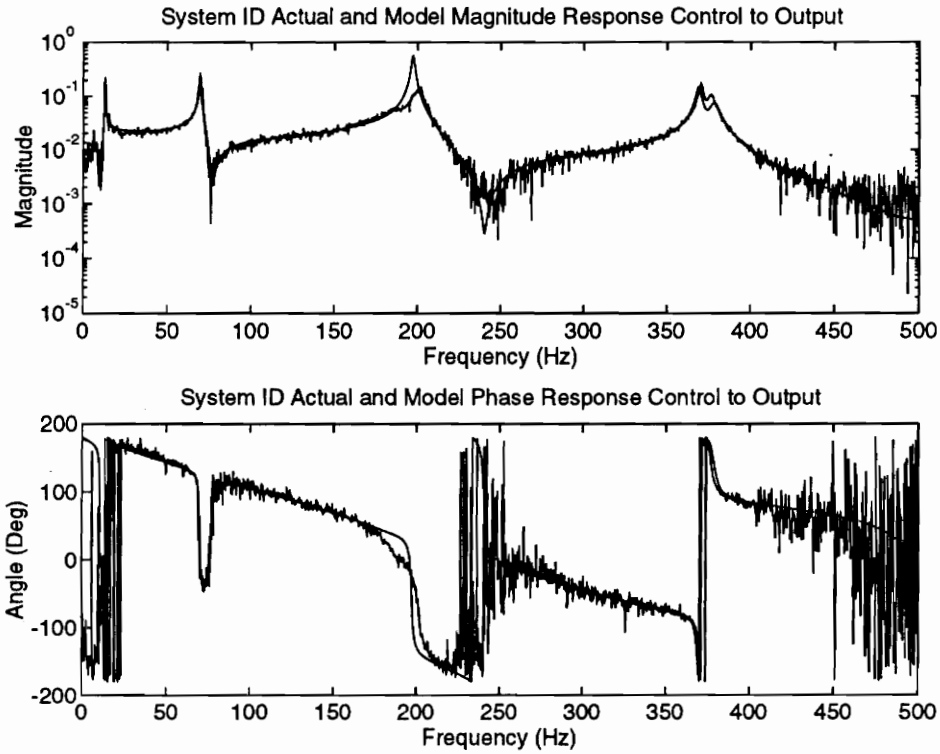


Figure 3.3.1 Frequency Response (Model and Data) Control Input to System Output

A system model from disturbance input to system output is not a requirement for the Adaptive-Q technique. However, some of the simulations do require the disturbance model. The frequency response of disturbance input to system output with the model (simulation) compared with the beam (data) is shown in figure 3.3.2.

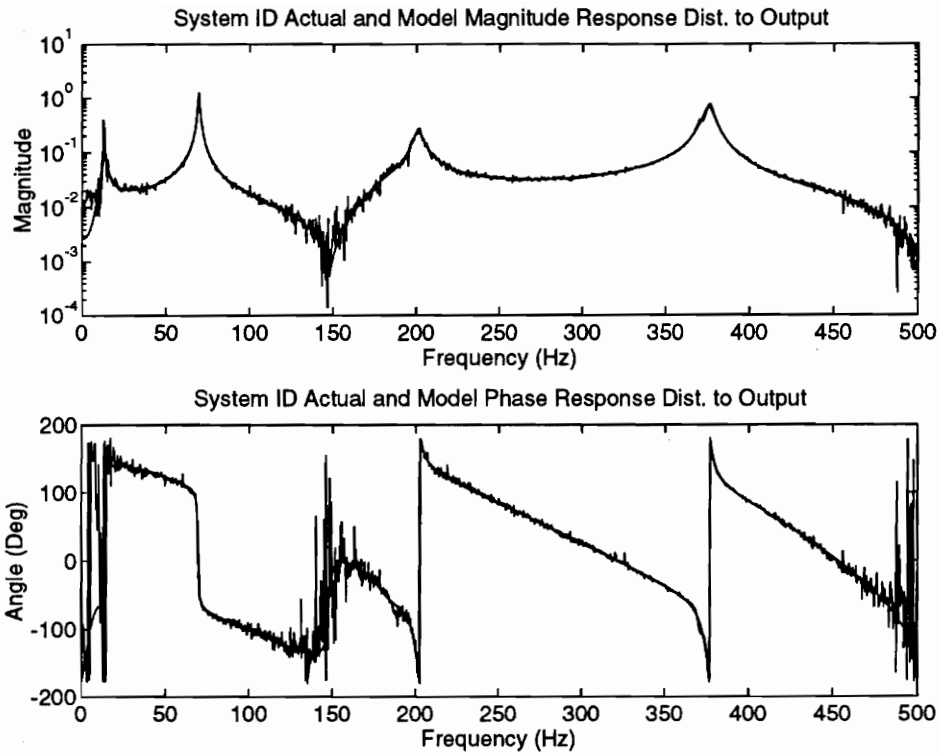


Figure 3.3.2 Frequency Response (Model and Data) Disturbance Input to System Output

The system identification process yields the beam natural frequencies.

Table 3.3.1 Experimentally Determined Natural Frequencies

Mode Number	Natural Frequency (Hz)
1	12.8
2	70.0
3	195
4	377
5	631

As expected, the natural frequencies shifted from the original calculation. The shift does not cause a problem since the resulting natural frequencies are within the design goal. This also points to the fact that even in the simple structure (cantilever beam), system identification techniques are invaluable for determining the system model.

The cantilever beam model generated during the system identification is

represented in state space as:

$$x(k+1) = Ax(k) + Bu(k) + Ed(k)$$

$$y(k) = Cx(k)$$

The matrices that make up the model are:

A Matrix:

Table 3.3.2 Model A-Matrix

	x1	x2	x3	x4	x5	x6
x1	4.2009e-001	9.0000e-001	-4.7070e-002	-1.5618e-002	3.4288e-003	6.8916e-003
x2	-8.4154e-003	-4.6169e-002	-6.8514e-001	-6.9496e-001	-1.2526e-002	-6.1474e-003
x3	4.4264e-003	1.6709e-002	6.9395e-001	-7.1219e-001	9.2481e-003	8.8383e-003
x4	9.0685e-001	-4.2033e-001	7.1261e-003	3.8867e-003	1.0954e-004	-8.6391e-004
x5	-8.3017e-004	-3.7553e-004	-1.0725e-002	-1.8760e-003	9.9538e-001	-8.1274e-002
x6	-8.4544e-004	-6.7112e-003	4.6099e-003	7.5796e-003	8.0310e-002	9.9359e-001
x7	2.6210e-004	-5.3280e-003	1.4086e-002	7.3810e-003	8.4142e-003	-1.3591e-002
x8	-2.4791e-003	-1.4707e-002	1.3880e-002	1.8969e-002	8.3134e-003	1.6714e-004
x9	-6.4765e-004	-2.0035e-002	5.3023e-002	3.2950e-002	3.7816e-003	-2.6823e-002
x10	-6.8716e-004	2.6093e-003	-1.0916e-002	-3.8382e-003	-3.6934e-003	8.1589e-003
x11	1.2096e-003	2.5294e-003	-5.9446e-003	-7.4379e-003	7.7741e-003	-1.1731e-003
x12	-1.1369e-004	5.3305e-004	-3.2140e-003	-1.3772e-003	1.3786e-004	2.0106e-003
x13	1.8735e-004	-3.9664e-004	1.3540e-003	1.5735e-004	1.4081e-003	-1.3982e-003
x14	2.3452e-004	1.5896e-003	-3.0764e-003	-2.6254e-003	3.5709e-004	8.9802e-004
x15	-4.7199e-004	-2.2221e-003	3.8324e-003	3.8978e-003	-1.2911e-003	-5.1001e-004

	x7	x8	x9	x10	x11	x12
x1	9.7011e-004	-9.6074e-004	-3.6421e-004	1.5888e-004	-2.5544e-004	-5.6237e-007
x2	-1.5529e-002	7.3134e-003	2.1207e-002	3.2711e-003	2.5856e-003	1.6895e-004
x3	2.5479e-002	-5.0645e-003	-5.4487e-002	-1.2573e-002	-2.8966e-003	-2.1361e-004
x4	-2.3808e-002	7.6072e-003	3.7129e-002	7.2923e-003	3.0039e-003	2.7584e-004
x5	1.2732e-002	-1.1026e-002	-1.5963e-002	-2.6118e-003	-2.1399e-003	1.4476e-004
x6	2.0335e-002	8.8043e-004	-2.8947e-002	-7.0729e-003	-8.5121e-004	-2.4925e-004
x7	3.0181e-001	9.4713e-001	-1.0440e-002	-6.6180e-003	4.9702e-003	-9.6715e-004
x8	-9.2140e-001	2.8265e-001	-1.3113e-001	-3.4499e-002	-3.2818e-003	6.3404e-004
x9	7.3405e-002	-7.6672e-002	3.9449e-002	-5.1745e-001	3.6765e-001	6.4678e-003
x10	-7.4718e-003	2.3926e-002	4.0402e-001	-6.6230e-001	-4.7600e-001	6.1210e-002
x11	-3.7085e-002	-6.0023e-003	9.8089e-002	-1.7204e-001	7.6077e-002	-6.3908e-001
x12	-7.1597e-003	5.5163e-003	1.5952e-001	5.1667e-002	4.6426e-001	3.4456e-001
x13	-2.7177e-003	-4.3103e-003	-1.6413e-003	1.8724e-002	-1.9253e-001	3.6245e-001
x14	-5.1512e-003	2.2406e-003	-8.1088e-002	-1.3520e-001	1.2178e-001	1.9827e-001
x15	8.9288e-003	-8.0169e-004	1.3769e-001	2.0165e-001	-6.5109e-002	-4.0309e-001

	x13	x14	x15
x1	-6.2826e-005	2.6867e-005	6.7460e-005
x2	1.3545e-003	-3.4541e-004	-8.2939e-004
x3	-2.6427e-003	5.4904e-004	1.1907e-003
x4	2.0609e-003	-4.5737e-004	-1.0670e-003
x5	-9.1485e-004	3.1291e-004	6.7671e-004
x6	-1.3176e-003	2.0364e-004	4.5111e-004
x7	1.3275e-004	-5.5822e-004	-1.1871e-003
x8	-4.8598e-003	1.0762e-003	1.9075e-003
x9	6.8023e-002	-3.1972e-002	-8.4110e-002
x10	-7.3127e-002	4.5900e-002	9.4516e-002
x11	-2.7712e-001	-5.7367e-002	-3.9680e-003
x12	-6.3399e-001	1.6766e-001	3.4383e-001
x13	-4.2941e-001	-5.5477e-001	-5.4370e-001
x14	2.7121e-001	-6.3139e-001	5.5330e-001
x15	-1.3236e-001	-4.6900e-001	2.2353e-001

B Matrix:

Table 3.3.3 Model B-Matrix

	u
x1	-1.3811e-002
x2	1.1602e-002
x3	2.2920e-002
x4	-2.0748e-002
x5	-2.3321e-002
x6	1.8335e-002
x7	4.7110e-002
x8	-1.6733e-002
x9	4.9462e-002
x10	-4.8934e-002
x11	7.4513e-002
x12	-4.4745e-003
x13	1.4666e-002
x14	5.6036e-003
x15	-1.6804e-002

E Matrix:

Table 3.3.4 Model E-Matrix

	d
x1	-2.9494e-002
x2	8.2029e-002
x3	1.3796e-001
x4	-2.4435e-002
x5	-1.2761e-003
x6	2.9151e-002
x7	2.8824e-002
x8	8.6875e-002
x9	1.4883e-001
x10	-1.4914e-002
x11	-3.5200e-002
x12	-5.3311e-003
x13	3.9563e-004
x14	-1.2629e-002
x15	1.9035e-002

C Matrix:

Table 3.3.5 Model C-Matrix

	x1	x2	x3	x4	x5	x6
y1	3.4277e-002	8.2278e-002	1.4180e-001	2.4095e-002	-2.3539e-002	-3.4238e-002

	x7	x8	x9	x10	x11	x12
y1	9.5912e-002	-3.8804e-002	-1.6356e-001	-2.9947e-002	-1.5886e-002	-1.049e-003

	x13	x14	x15
y1	-9.5635e-003	2.3046e-003	5.3879e-003

Chapter 4

Experimental Results

4.1 Overview

Chapter 4 presents the cantilever beam experimental results. The exploration of several control situations is undertaken to determine overall controller performance. The disturbances investigated are simple harmonic disturbances, narrow band disturbances, and a broad band disturbance. Each of the cases compares simulation results with actual beam results. MATLAB on a 486 computer performs the simulations and all the simulation code is included in appendix A. Additionally, a brief discussion concerning control robustness is undertaken.

4.2 LQG Fixed Feedback

The Adaptive-Q technique, discussed in chapter 2, requires a fixed feedback controller. The fixed feedback controller is a LQG design. Therefore, before the Adaptive-Q testing can begin a fixed feedback LQG controller requires implementation.

The system model, calculated in chapter 3, is used to design a fixed feedback controller. The fixed feedback control provides a fixed amount of system damping. The fixed damping is important for controlling transient (random) disturbances. These types of disturbances may occur in such a manner that the adaptive section is not given enough time to converge to solution. Thus, the fixed feedback control does provide some measure of control. Additionally, other hybrid controllers have shown that the fixed damping can lead to faster convergence for the adaptive section[3].

The LQG controller with a complete and accurate system model, including the way the disturbance enters the plant, can provide all the damping realizable for the plant. However, the LQG design for Adaptive-Q assumes that how the disturbance enters the plant is unknown. The LQG design still needs to know that a disturbance is present so we model it being a white noise source entering through the control input (B-matrix). The result is that LQG control provides only a portion of the required damping. The performance sacrifice gives the controller the ability to handle the unknowns in the plant model and remain stable. Because the system model is incomplete and contains inaccuracies, an iterative approach is followed to get the LQG controller functioning. The method consists of varying the weighting matrices for the LQR and LQE portions of the LQG controller. The weighting matrixes affect the amount of damping provided by the

controller. Generally, the less damping the controller provides the more robust the LQG control. In our case, the control design parameters are varied until an instability point is reached. At that point, the parameters were backed off so the resulting system is stable. The m_file BMSIMID5.M in appendix A contains the specific commands that generate the LQG design. As part of the LQE design, the system output noise needs measurement. A plot of the system output noise is shown in figure 4.2.1. The output noise is the system noise floor. The LQG simulation results, for the open and closed loop case, are shown in figure 4.2.2. Once the LQG control is functioning on the beam, we must verify the controller is performing as designed. System identification techniques for the closed loop system are the verification technique. The closed loop data is converted to the frequency domain for comparison purposes. The comparison of the simulation and actual closed loop frequency response is shown in figure 4.2.3. Analysis of the actual closed loop frequency response versus the simulated closed loop frequency response shows good agreement. However, the responses are not identical. To be identical the plant model used to design the controller would have to be exact. As previously discussed, the plant model is not exact. Additionally, the fixed feedback controller is not providing uniform damping throughout the frequency band of interest. This clearly points to the need for an additional control.

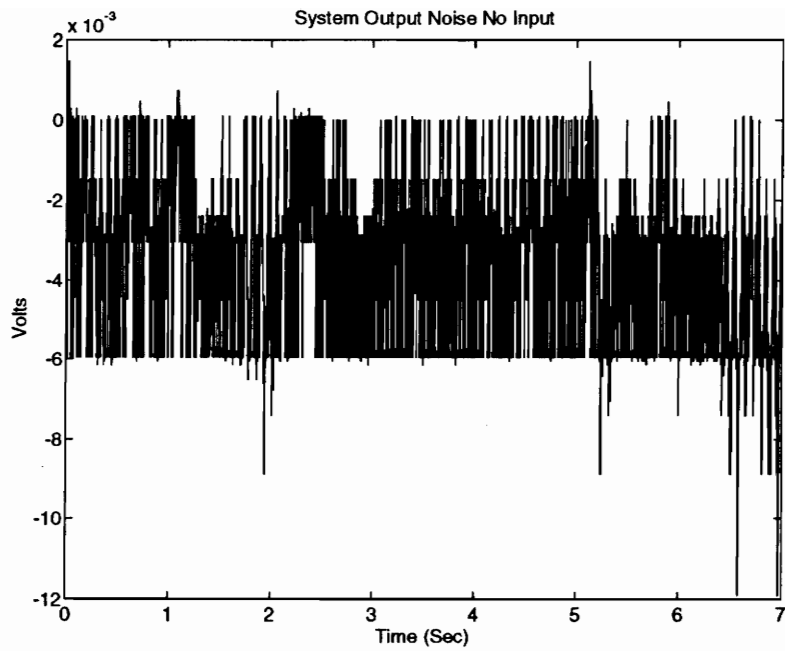


Figure 4.2.1 Output Noise (No Input)

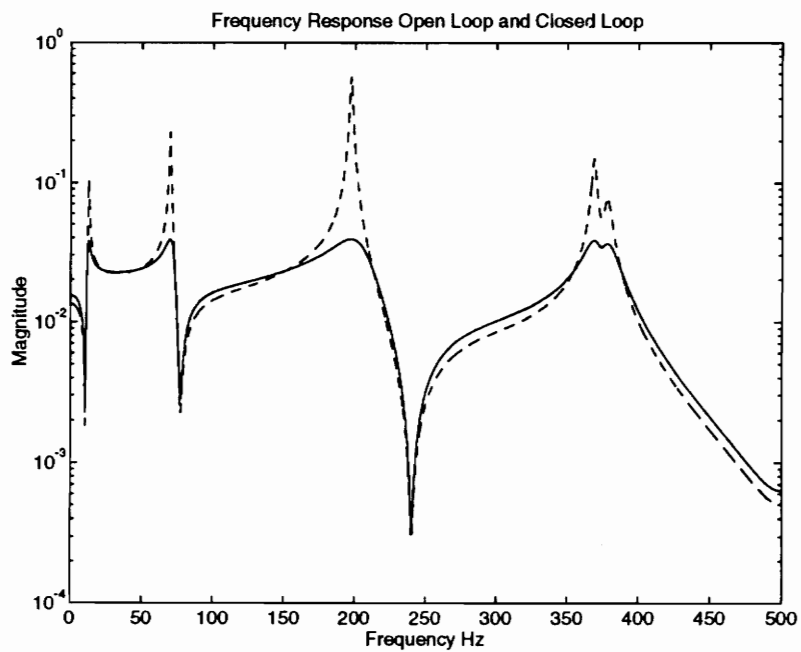


Figure 4.2.2 Open and Closed Loop Frequency Response (Simulation)

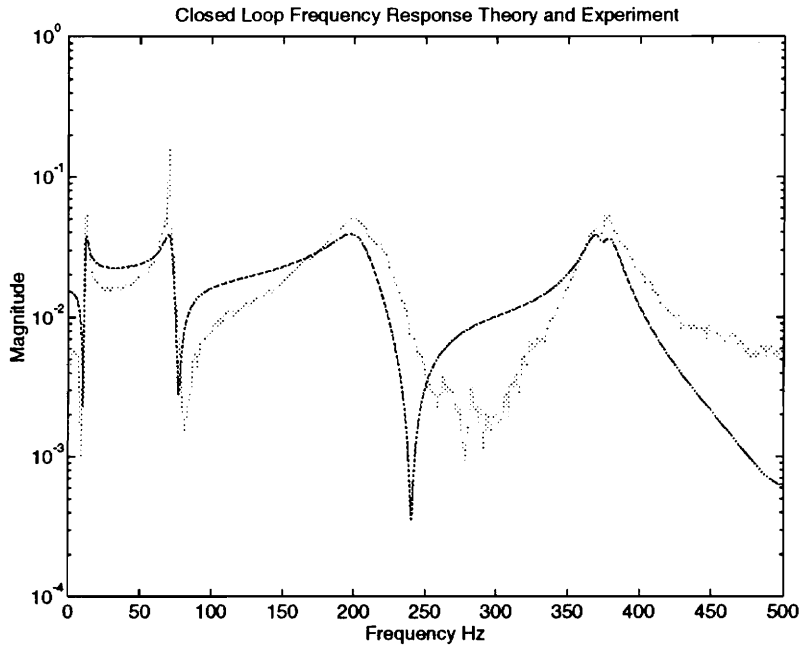


Figure 4.2.3 Frequency Response Experimental and Simulation

4.3 Disturbance Rejection Overview

To test the theory developed in chapter 2, three types of disturbances are considered.

- Harmonic Disturbance
- Narrow Band Disturbance
- Broad Band Disturbance

The experimental and simulation results for each of these cases follow in the next sections.

4.3.1 Harmonic Disturbance

The experimental harmonic disturbances are sine waves. The sine wave frequencies are 12.8 Hz, 70 Hz, and 195 Hz. The three frequencies correspond to the first

three beam modes. Ideally, to cancel the sine wave disturbance the adaptive FIR filter section need only be second order, since a sine wave is uniquely determined by magnitude and phase. However, in the experiment the minimum filter order is third order. The extra coefficient takes care of any DC shifts that may occur due to modeling errors. The DC modeling error is expected because the system model around DC is known to contain inaccuracies. These inaccuracies are best seen by placing a DC bias on a control signal in the experimental setup. The experimental setup will not reflect this DC bias in its output. However, the system model does pass a portion of the DC bias.

The harmonic test case uses two adaptation algorithms. The recursive least squares(RLS) algorithm and the least mean squares (LMS) algorithm. Variations are made to the number of filter coefficients to see if this affects adaptation time. The LMS algorithm tests use a third order filter and a one hundredth order filter. The RLS algorithm tests use a third order filter and an eighth order filter. The RLS algorithm is limited to eighth order due to DSP board memory constraints.

The harmonic disturbance is generated by a Hewlett Packard function generator model number 3325a.

4.3.1.1 Experimental Results Harmonic Disturbance

The plots that follow present the experimental results for the harmonic disturbances. The 12.8 Hz case is examined in detail while only the final results are presented for the other two cases, the 12.8 Hz case being representative of the results obtained for the other two.

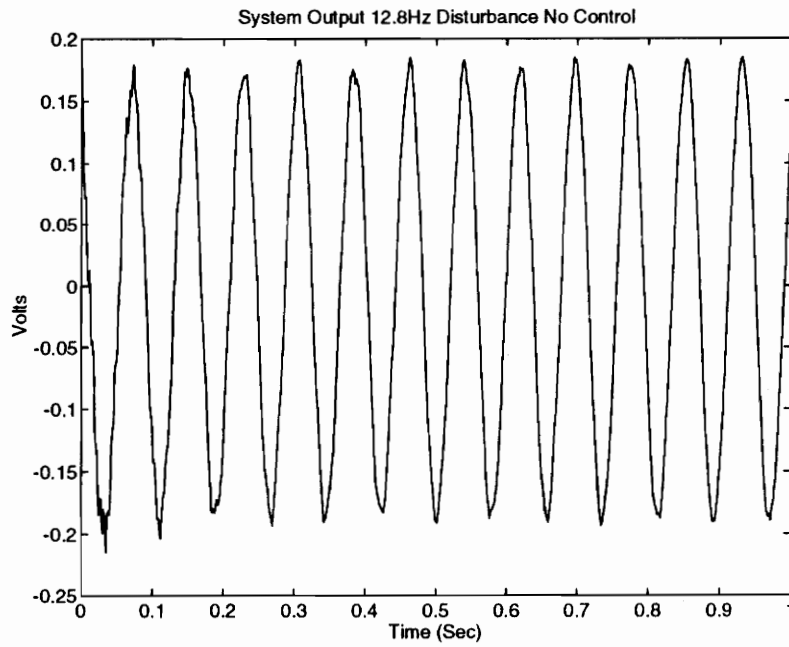


Figure 4.3.1 Disturbance 12.8 Hz No Control Input

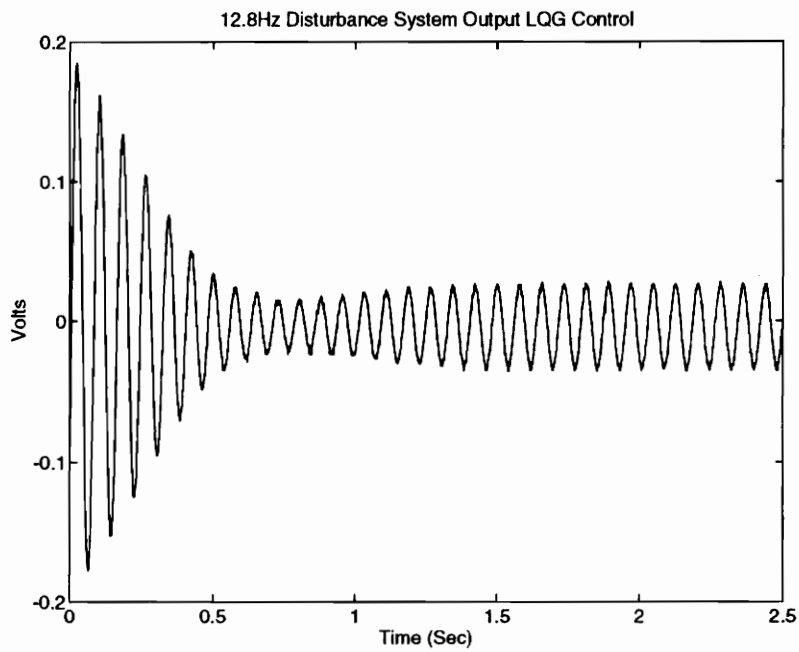


Figure 4.3.2 12.8 Hz Disturbance LQG Controller

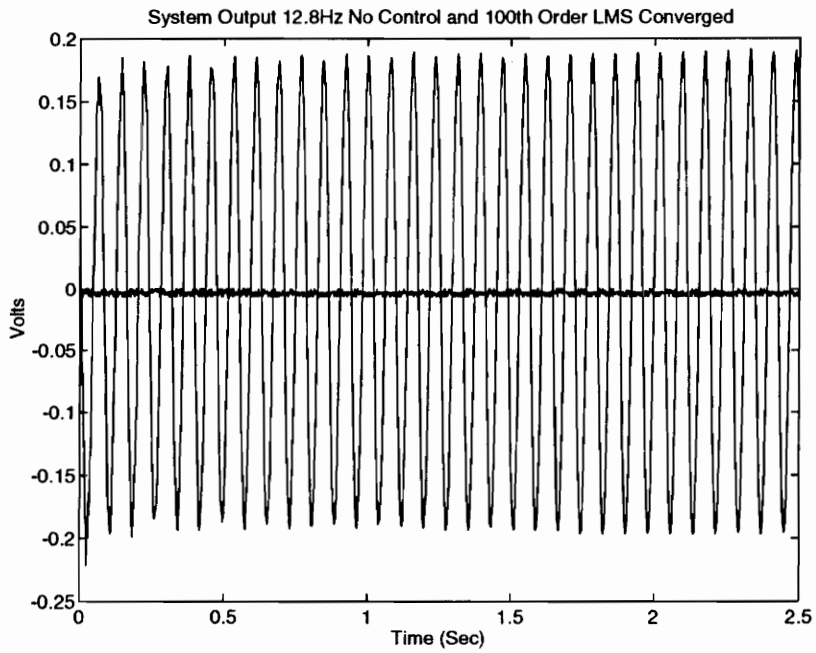


Figure 4.3.3 System Output No Control Compared to Converged LMS

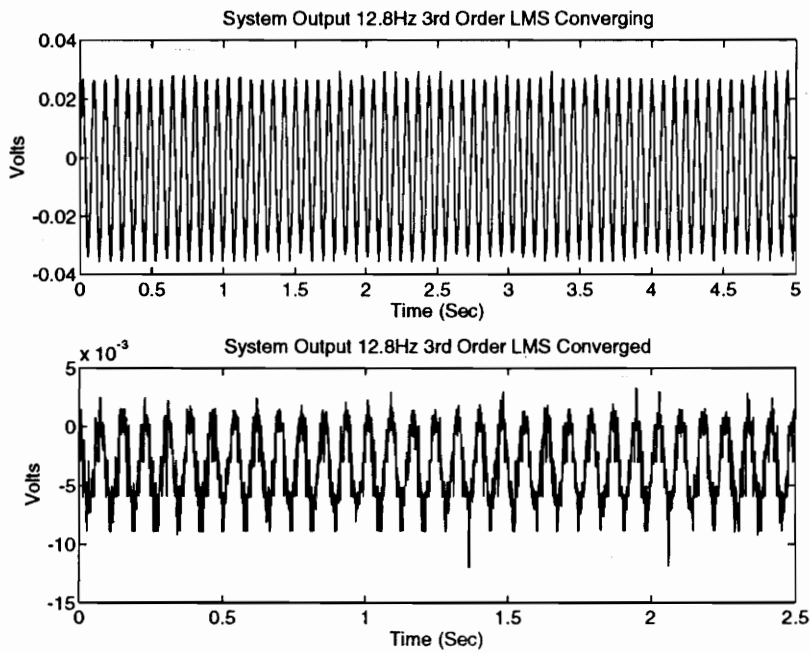


Figure 4.3.4 3rd Order LMS 12.8 Hz Disturbance Converging and Converged

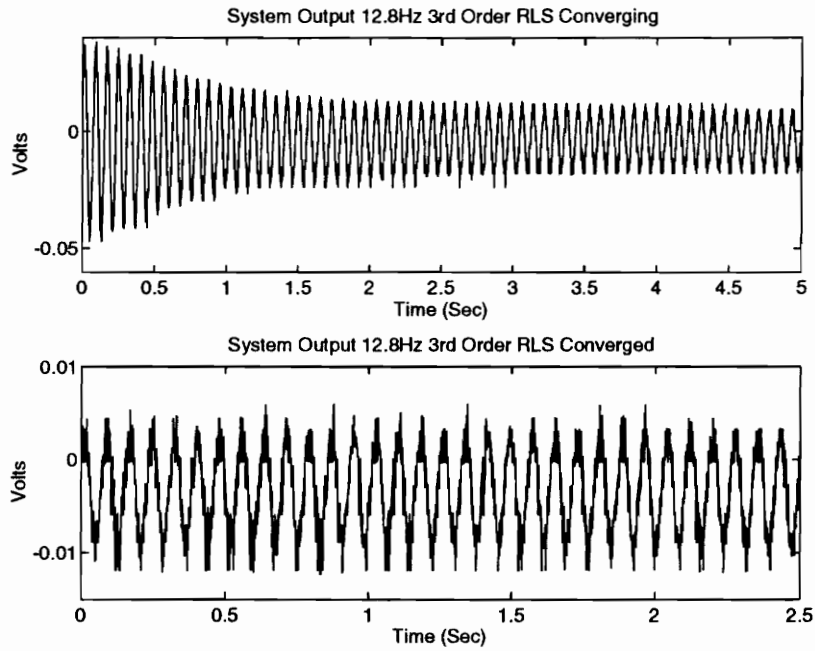


Figure 4.3.5 3rd Order RLS 12.8 Hz Disturbance Converging and Converged

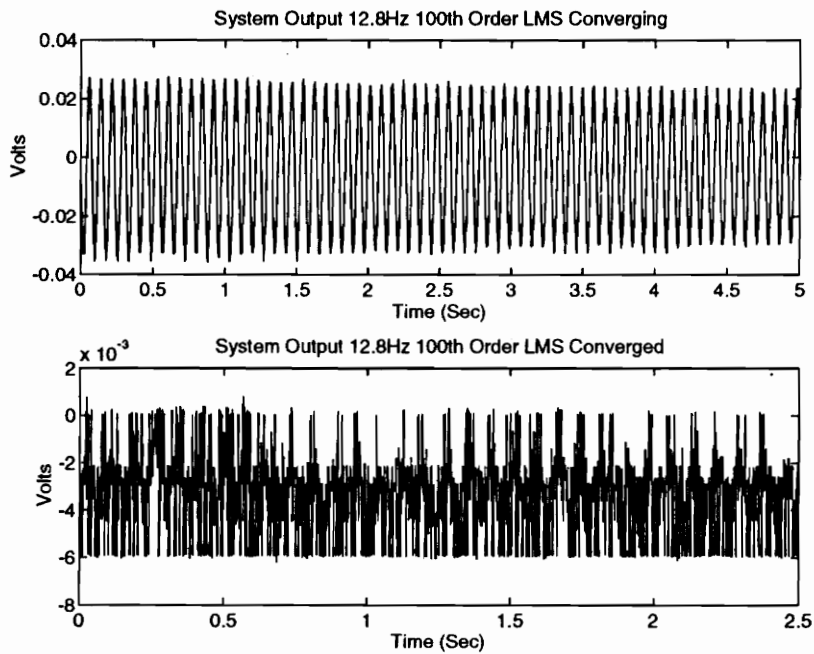


Figure 4.3.6 100th Order LMS 12.8 Hz Disturbance Converging and Converge

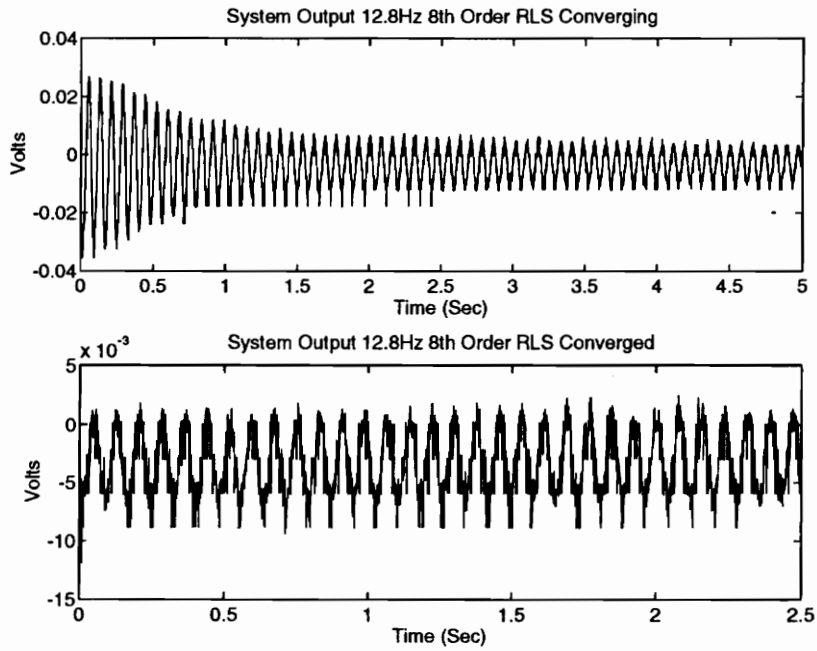


Figure 4.3.7 8th Order RLS 12.8 Hz Disturbance Converging and Converged

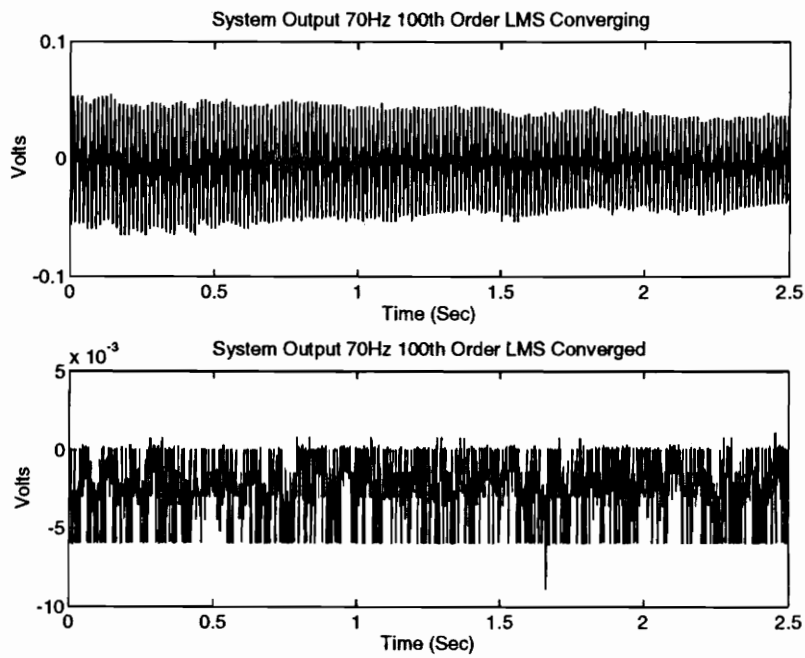


Figure 4.3.8 100th Order LMS 70 Hz Disturbance Converging and Converged

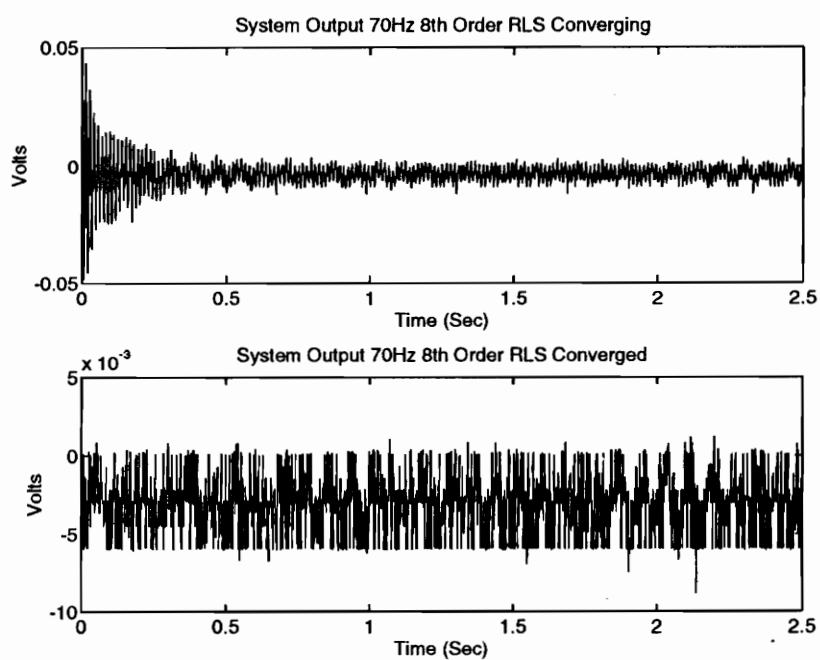


Figure 4.3.9 8th Order RLS 70 Hz Disturbance Converging and Converged

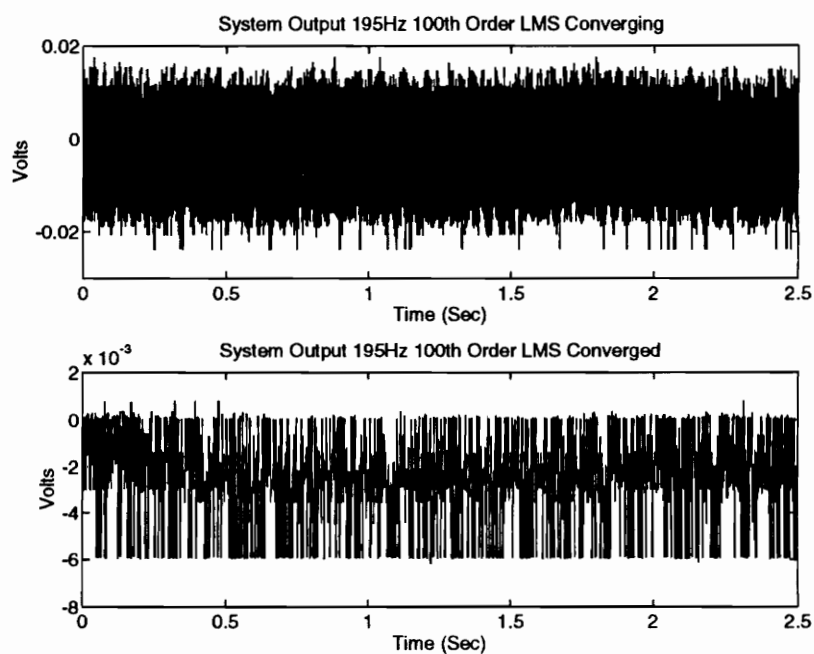


Figure 4.3.10 100th Order LMS 195 Hz Disturbance Converging and Converged

4.3.1.2 Harmonic Disturbance Simulation

The simulation results for the Adaptive-Q controller for the three harmonic disturbances considered are similar. In every case, the controller reduces the output to practically zero. Therefore, only simulation results for the 12.8 Hz disturbance are shown.

The simulations are done in MATLAB using the m_file SSADAP3.M contained in appendix A.

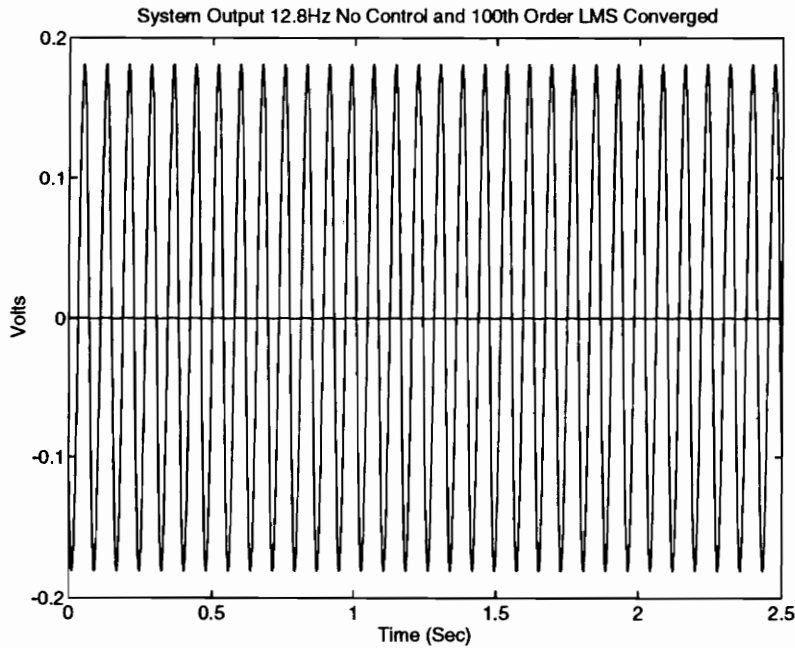


Figure 4.3.11 Simulation 100th Order LMS No Control Versus Converged LMS

4.3.1.3 Conclusions Harmonic Disturbance

Analysis of the harmonic disturbance results leads to the following conclusions.

The LQG controller alone, as expected, was unable to completely cancel the disturbance.

In fact, the LQG control used in the experiment achieves the highest damping achievable while still remaining stable. Reference appendix A m_file BMSIMID5.M for the values used to construct the LQG controller. The Adaptive-Q controller, in all of the cases, was able to bring the disturbance output to the noise floor(Reference Fig. 4.2.1). This result is in complete agreement with the simulation. The adaptive algorithm convergence speed did show significant differences. Convergence time is the time required to reach the noise floor or in some of the RLS cases very close to the noise floor. It should be noted, that significant damping is achieved much faster than these times reflect. In the RLS case, reference figure 4.3.9, significant damping is added in less than 0.5 seconds.

Table 4.3.1 Convergence Times

Algorithm	Convergence Time (Minutes)
LMS 3rd order	10
LMS 100th order	2
RLS 3rd order	1
RLS 8th order	1

The RLS and LMS algorithms require a step size parameter. The step size parameter is either fixed (LMS) or variable (RLS). In both algorithm's this parameter must be set by the system designer and effects the convergence speed. In general, the higher the parameter the faster the convergence time. However, if the step size parameter is too large the adaptation algorithm is unstable[13]. The method to determine the step size value is inexact and theoretical formulas are hard to compute. The technique used in the experiment was to find a set of step size variables that worked for all the disturbance cases considered. Simply stated, the step size variable was increased until the system went unstable. With system stability in mind, the values where then significantly reduced.

4.3.2 Optimal Disturbance Rejection Controller

The harmonic case results in a system output going to either zero (simulation), or the noise floor(experimental). However, the narrow band and broad band cases yield simulation results that are non-zero. Thus, to make performance comparisons require a baseline. The control structure for the metric is an LQG disturbance rejection (LQGDR) controller. The LQGDR controller differs in two major ways from the LQG control structure in chapters 2 and 3. The first is in the Kalman filter (LQE) design. The estimator has information concerning the disturbance frequency content. The control design also includes a model from disturbance input to system output. Since the LQGDR controller is the performance metric, control energy is not a limitation. The LQE portion uses the same information concerning state and output noise as the prior LQG design used for the Adaptive-Q control. The derivation of the LQGDR begins with augmenting the existing plant state equations to include a model of the disturbance frequency spectrum. The disturbance is modeled with the state equations:

$$\begin{aligned}x_d(k+1) &= A_d x(k) + B_d d(k) \\ y_d(k) &= C_d x_d(k)\end{aligned}$$

where the feedthrough term D_d is assumed to be zero.

The disturbance state equations augment the plant disturbance input to create colored noise for the Kalman filter as shown in figure 4.3.12. It is worth noting that the disturbance model in the above equation is a D-T model. The disturbance is a C-T signal.

This is a possible error source, however, at a D-T system sampling rate of 1000 Hz these effect are negligible.

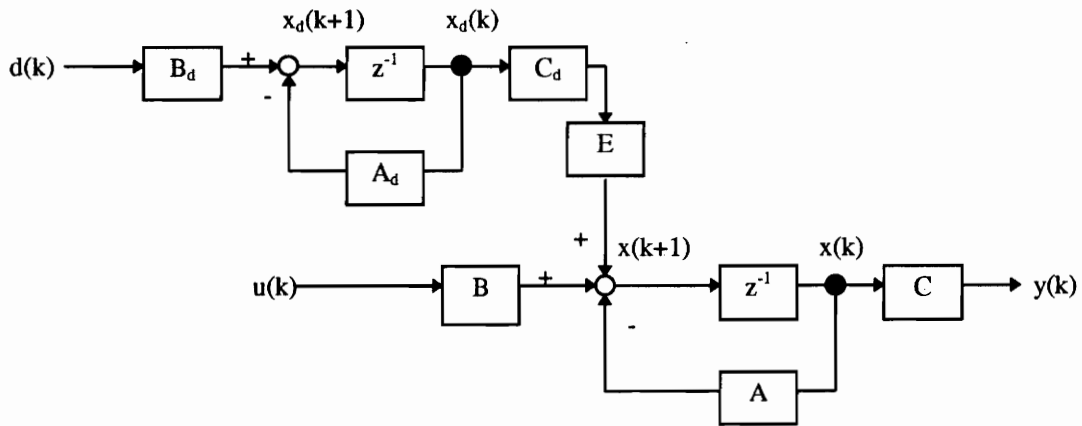


Figure 4.3.12 Plant and Disturbance Model

The augmented plant model state equations are:

$$\begin{bmatrix} x(k+1) \\ x_d(k+1) \end{bmatrix} = \begin{bmatrix} A & EC_d \\ 0 & A_d \end{bmatrix} \begin{bmatrix} x(k) \\ x_d(k) \end{bmatrix} + \begin{bmatrix} B & 0 \\ 0 & B_d \end{bmatrix} \begin{bmatrix} u \\ d \end{bmatrix}$$

$$y(k) = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ x_d(k) \end{bmatrix}$$

Next define the system matrices as:

$$A_{aug} = \begin{bmatrix} A & EC_d \\ 0 & A_d \end{bmatrix} \quad B_{aug} = \begin{bmatrix} B \\ 0 \end{bmatrix} \quad E_{aug} = \begin{bmatrix} 0 \\ B_d \end{bmatrix}$$

$$C_{aug} = \begin{bmatrix} C & 0 \end{bmatrix}$$

The augmented system is then redrawn as shown in figure 4.3.13.

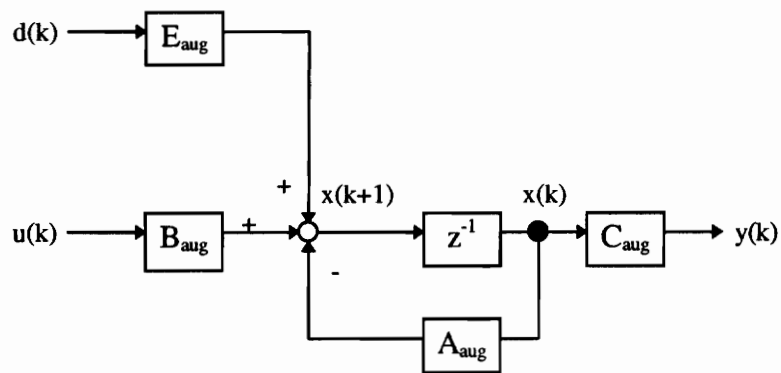


Figure 4.3.13 Augmented Plant

The augmented system model (figure 4.3.13) is used to design the LQE and LQG sections. The resulting control structure including the LQGDR controller is shown in figure 4.3.14. In redrawing the control system, the plant model replaces the augmented model. The augmented model is only needed for the control design.

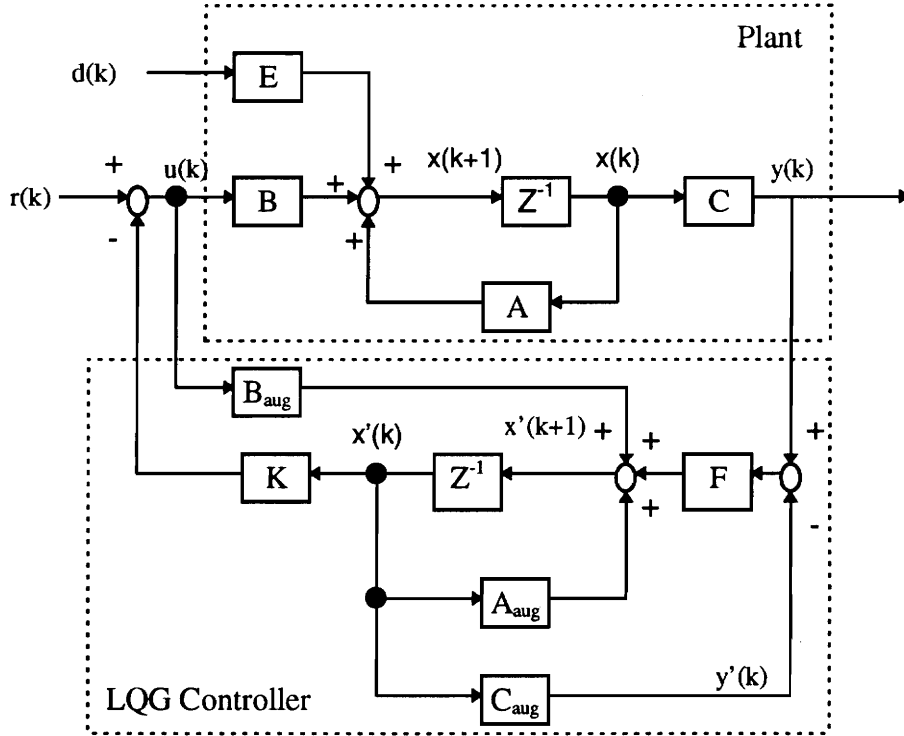


Figure 4.3.14 LQGDR Block Diagram

The state equations for the LQGDR controller are:

$$\begin{bmatrix} x(k+1) \\ x'(k+1) \end{bmatrix} = \begin{bmatrix} A & -BK \\ FC & A_{aug} - B_{aug}K - FC_{aug} \end{bmatrix} \begin{bmatrix} x(k) \\ x'(k) \end{bmatrix} + \begin{bmatrix} B \\ B_{aug} \end{bmatrix} r(k) + \begin{bmatrix} E \\ 0 \end{bmatrix} d(k)$$

$$y(k) = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ x'(k) \end{bmatrix}$$

The optimal disturbance rejection controller, in simulation, has a perfect system model.

The reality is that this controller will not be physically realizable on the beam due to errors in the system modeling process. Thus, the performance metric is a MATLAB simulation

and no experimental results are available. The MATLAB code, in appendix A, to perform the simulation is WBDIST2.M

4.3.3 Narrow Band Disturbance Rejection

Two cases are considered for the narrow band disturbance rejection experiment. The first is a 20 Hz bandwidth disturbance centered around 70 Hz. The second case is a 20 Hz bandwidth disturbance centered around 195 Hz. The disturbance is generated by filtering a pseudo-random sequence with a discrete time band pass filter. The LQGDR control structure, as previously stated, is the comparison method. The discrete time filters are:

70 Hz Center

$$h(z) = \frac{5.885e - 4z^6 - 1.7654e - 3z^4 - 1.161e - 14z^3 + 1.7654e - 3z^2 - 2.903e - 15z}{z^6 - 5.209z^5 + 11.81z^4 - 14.83z^3 + 10.89z^2 - 4.428z + 0.7839}$$

195 Hz Center

$$h(z) = \frac{2.9639e - 2z^6 - 4.0027e - 2z^5 + 0.0426z^4 + 4.663e - 15z^3 - 4.2602e - 2z^2 + 4.0027e - 2z - 2.9639e - 2}{z^6 - 1.995z^5 + 4.204z^4 - 4.112z^3 + 4.035z^2 - 1.838 + 0.8844}$$

4.3.3.1 Experimental Results Narrow Band Disturbance

The plots that follow present the experimental results obtained for the narrow band disturbances. The 70 Hz case is examined in detail while only the final results will be presented for the other cases; 70 Hz case being representative of the results obtained for the other case.

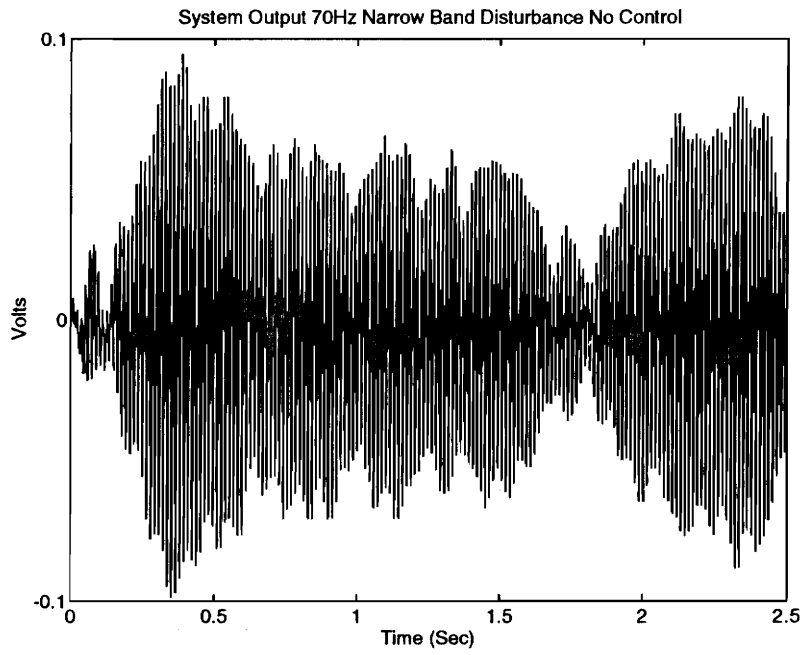


Figure 4.3.15 Narrow Band Disturbance 70 Hz Center Freq.

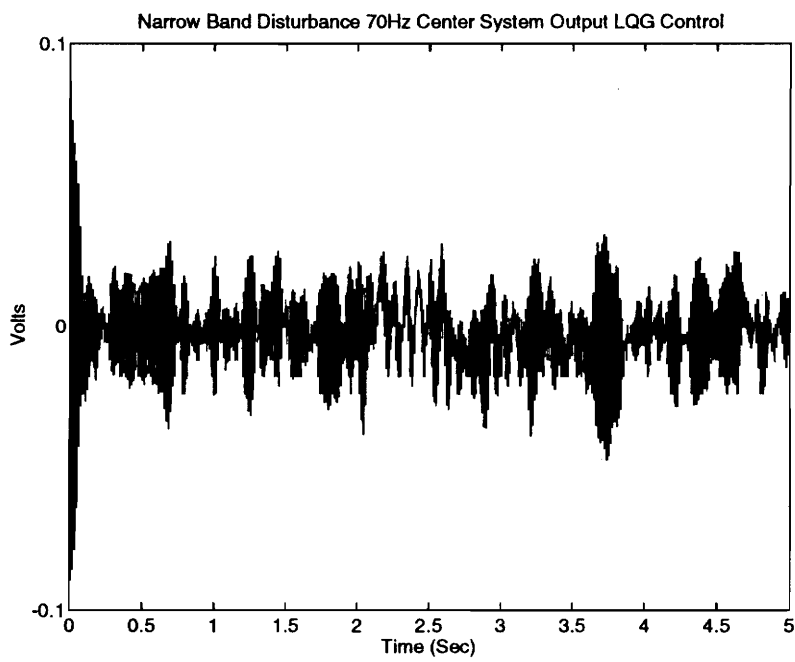


Figure 4.3.16 Narrow Band Disturbance 70 Hz Center LQG Control

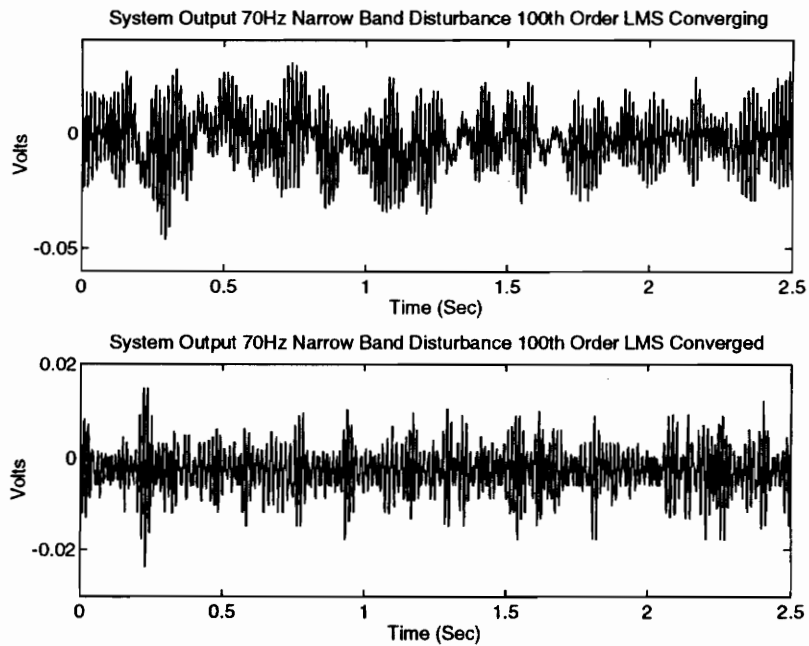


Figure 4.3.17 100th Order LMS 70 Hz Center Converging and Converged

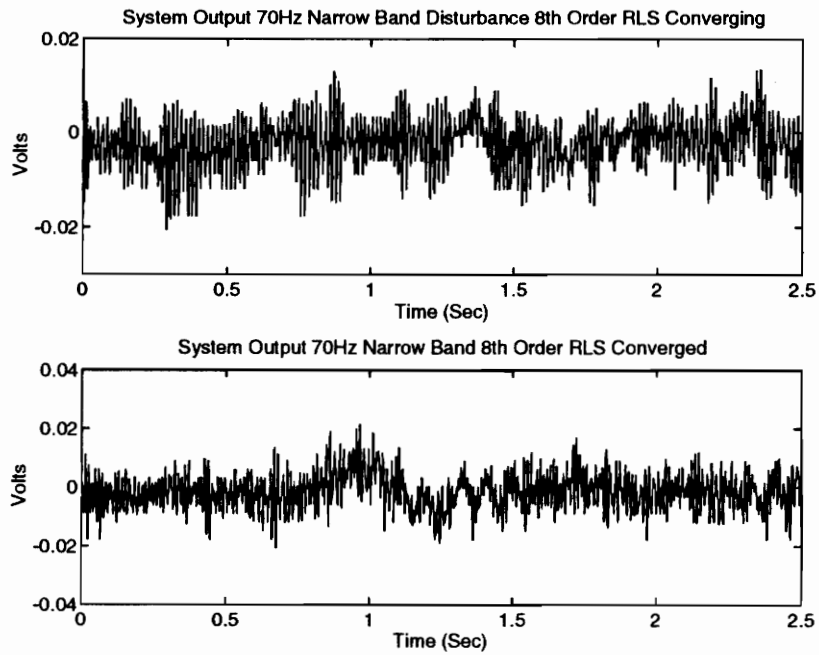


Figure 4.3.18 8th Order RLS 70 Hz Center Converging and Converged

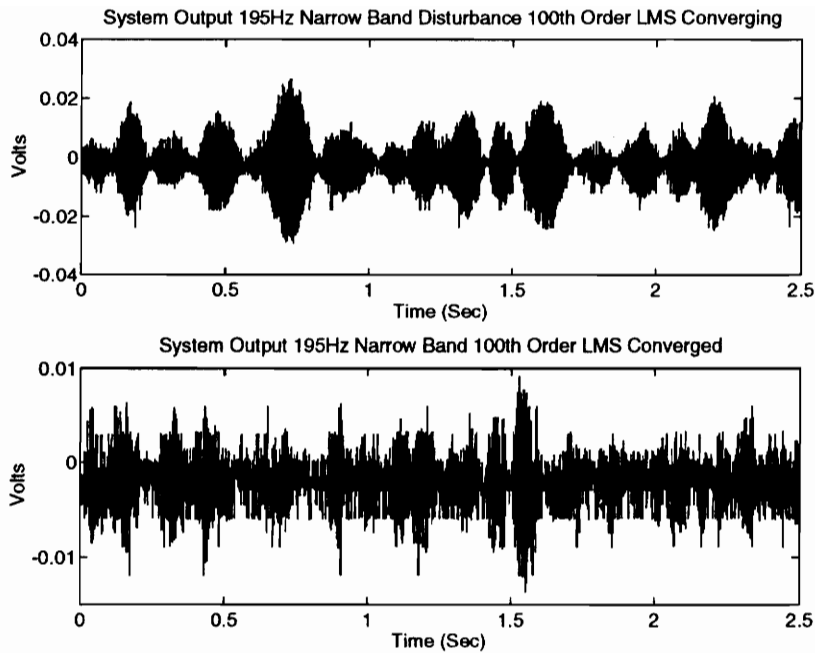


Figure 4.3.19 100th Order LMS 195 Hz Center Converging and Converged

The experimental results for the narrow band disturbances use the same LQG controller as in the harmonic case. Evaluation of the narrow band results for the LQG control show that the LQG control is providing a significant amount of the overall damping. Although this is very acceptable, we are also interested in how well the adaptive section performs. To perform this check the LQG controller is redone with a higher cost associated with control input. This causes the LQG control to provide significantly less damping. The exact numbers used for the Q and R parameters are contained in the MATLAB m_file BMSIMID5.M contained in appendix A. The results for the control are shown in the next two plots.

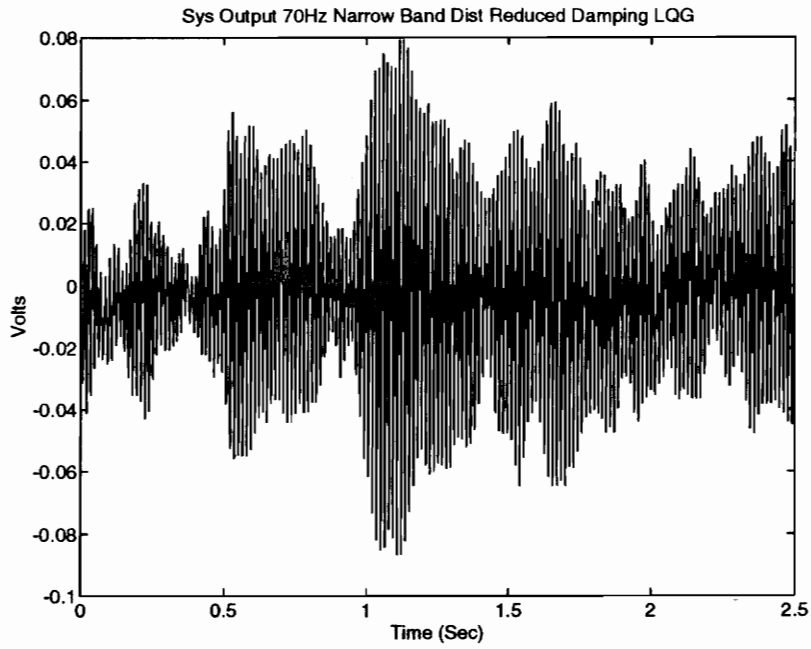


Figure 4.3.20 Reduced Control Authority LQG Control

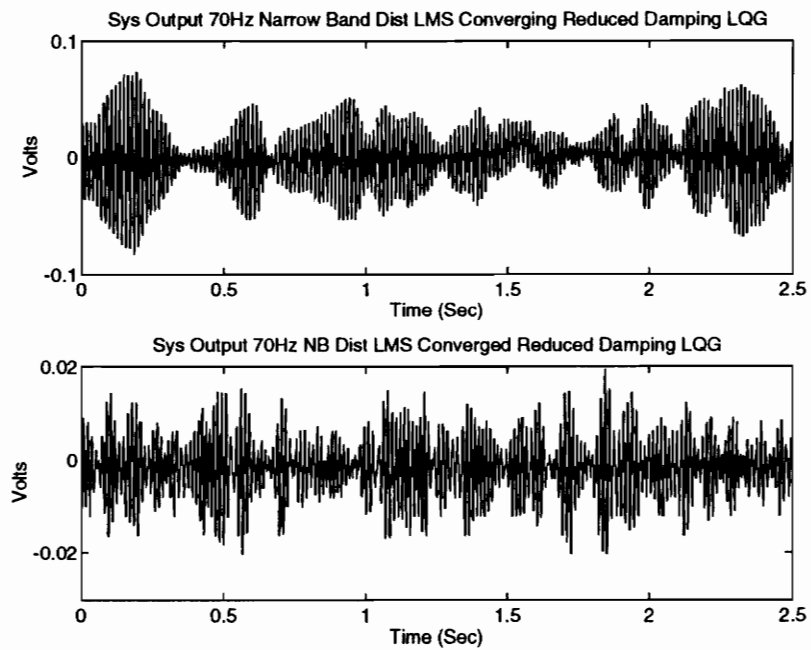


Figure 4.3.21 100th Order LMS 70 Hz Center (Reduced LQG) Converging and Converged

4.3.3.2 Narrowband Disturbance Simulation

The LQGDR simulation results are shown in the figures that follow. The simulation is performed for each case. Once again, the simulations are done in MATLAB using the m_file WBDIST2.M in appendix A.

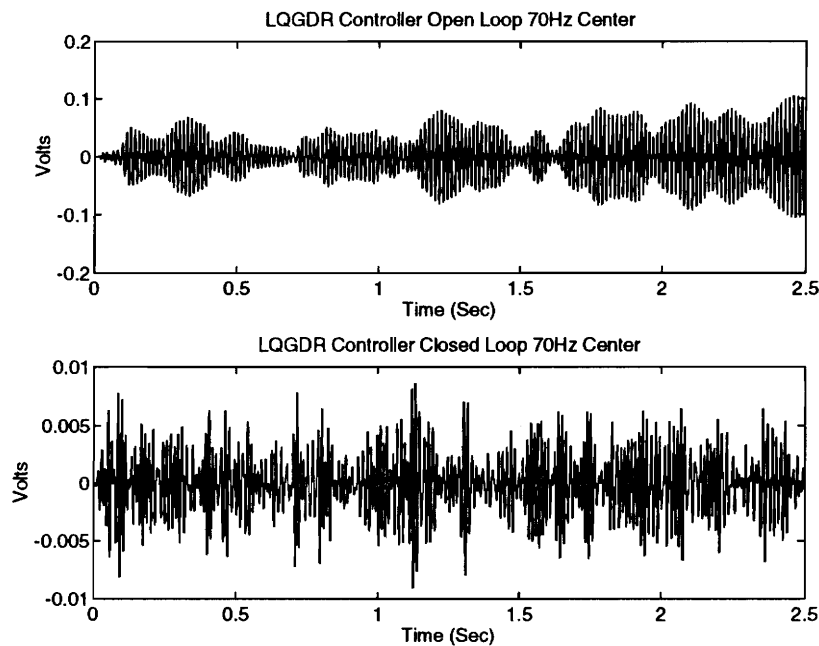


Figure 4.3.22 70 Hz Center Open and Closed LQGDR Simulation

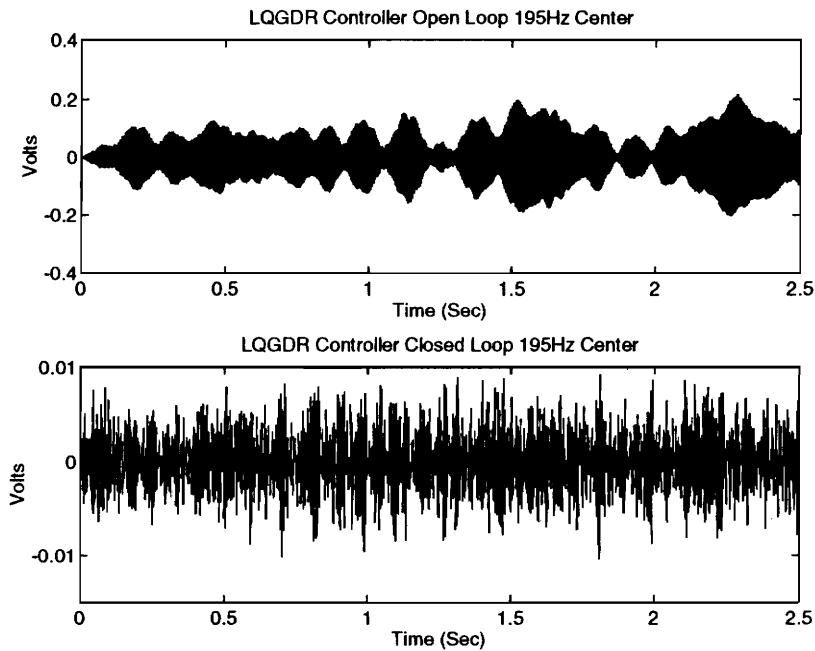


Figure 4.3.23 195 Hz Center Open and Closed LQGDR Simulation

4.3.3.3 Conclusions Narrow Band Disturbance

The narrow band disturbance rejection results compare well with LQGDR simulation. A comparison of the Adaptive-Q control results and the LQGDR control results is in the following two plots. The results show the Adaptive-Q controller performed within a factor of two of the baseline. The rejection provided in the 70 Hz case is 15 dB. The damping for the 195 Hz case is 18 dB.

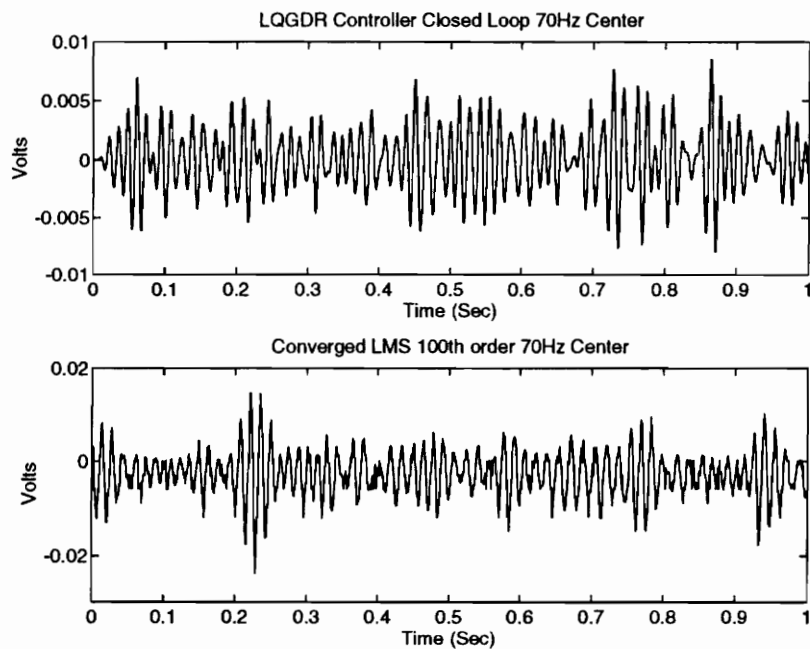


Figure 4.3.24 70 Hz Center LQGDR Simulation Versus Experimental Adaptive-Q

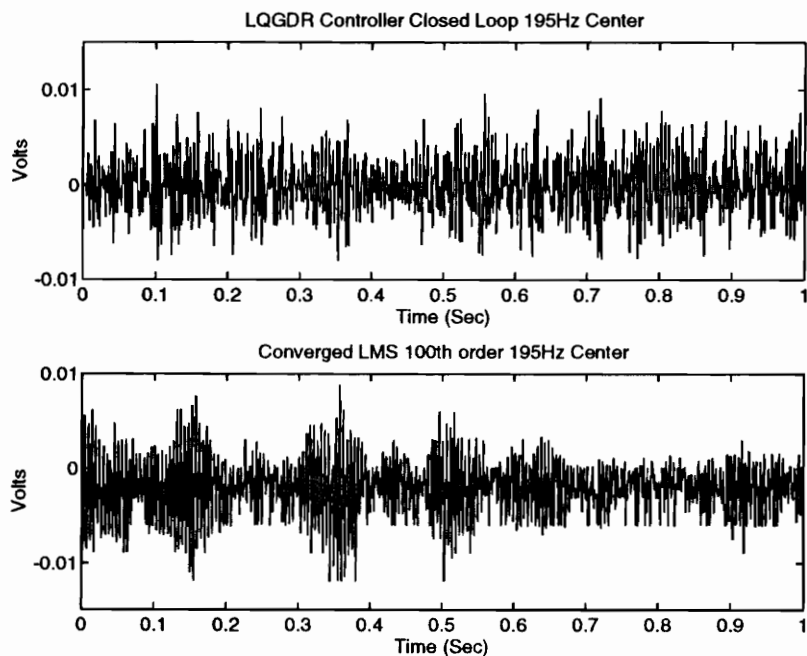


Figure 4.3.25 195 Hz Center LQGDR Simulation versus Experimental Adaptive-Q

4.3.4 Broad Band Disturbance

The broad band disturbance considered is a 150 Hz bandwidth signal centered at 75 Hz. The disturbance is used because it spans two modes. The disturbance is generated by feeding a pseudo-random sequence through a lowpass filter with 3 dB point set at 150 Hz. The filter is manufactured by Frequency Devices model number 9002.

4.3.4.1 Experimental Results Broad Band Disturbance

The plots that follow present the experimental results obtained for the broad band disturbance. The RLS algorithm was not used in this experiment due to its low order. The filter order determines the bandwidth of the disturbance the control can adaptively suppress. As mentioned in the harmonic case, the minimum order for a single harmonic is a second order filter. For disturbances that have a richer frequency content additional coefficients are required to model this content. Experimentally, it was determined that the 8th order RLS adaptive filter was insufficient for the broad band case and as such was not used.

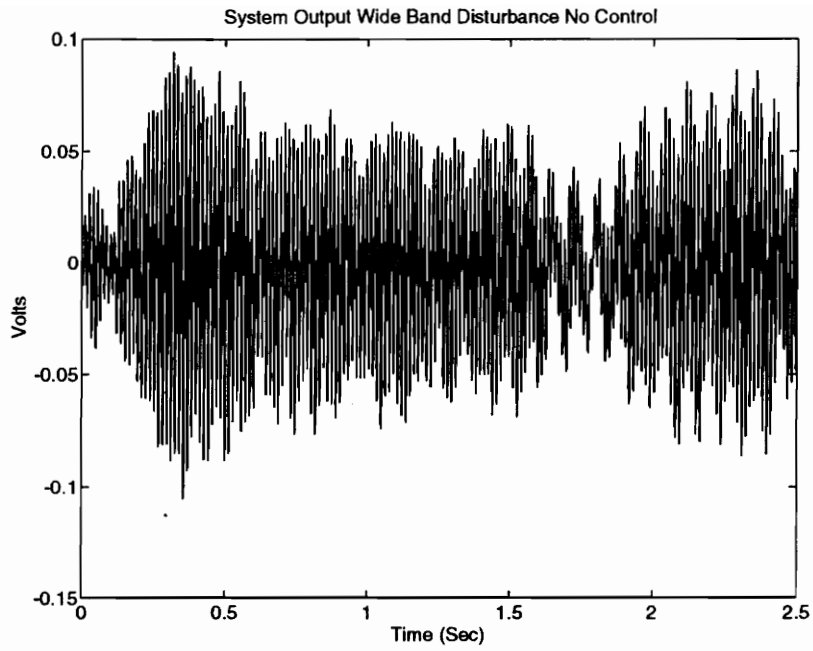


Figure 4.3.26 Broad Band Disturbance (No Control)

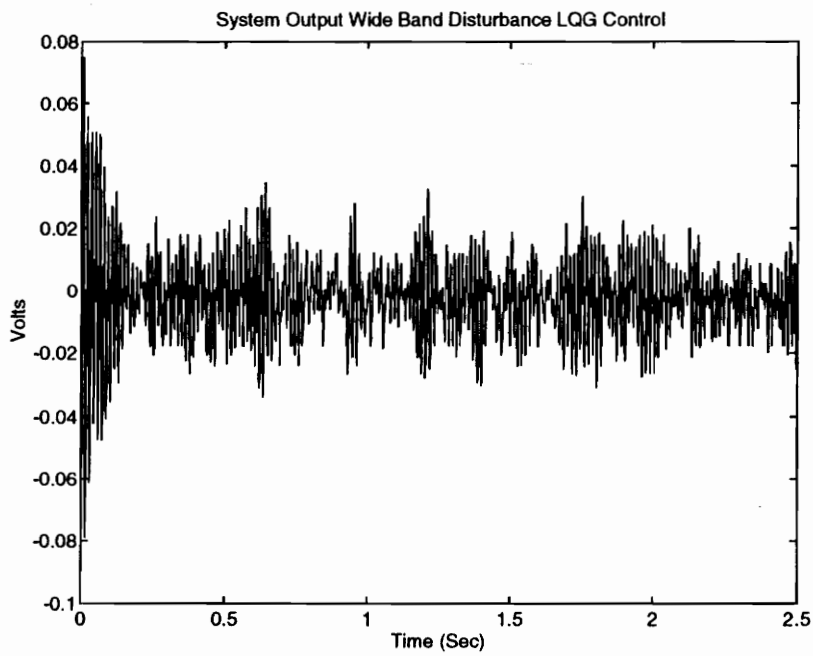


Figure 4.3.27 Broad Band LQG

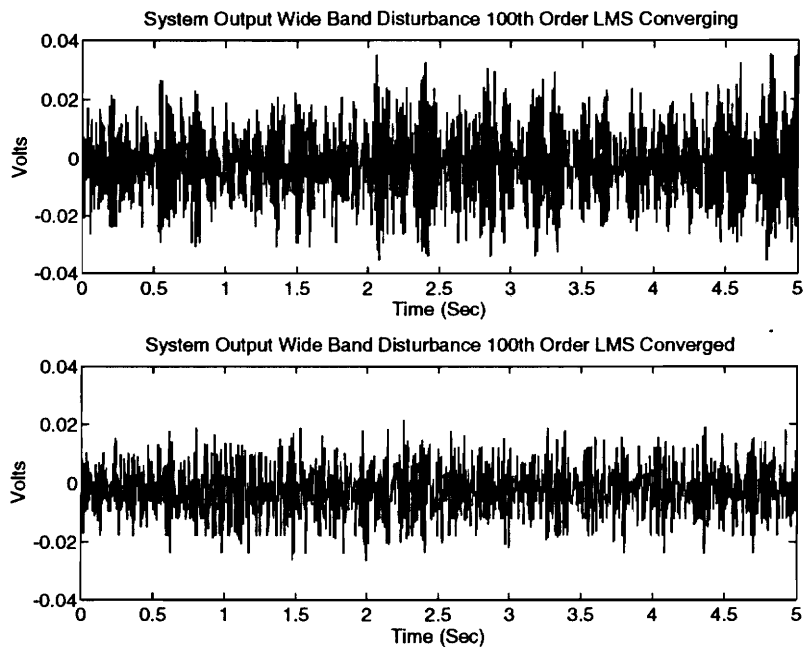


Figure 4.3.28 100th Order LMS Broad Band Converging and Converged

4.3.4.2 Broad Band Disturbance Simulation

The broad band simulation for LQGDR controller is shown in figure 4.3.29. The simulation is performed in MATLAB using the m_file WBDIST2.M contained in appendix A.

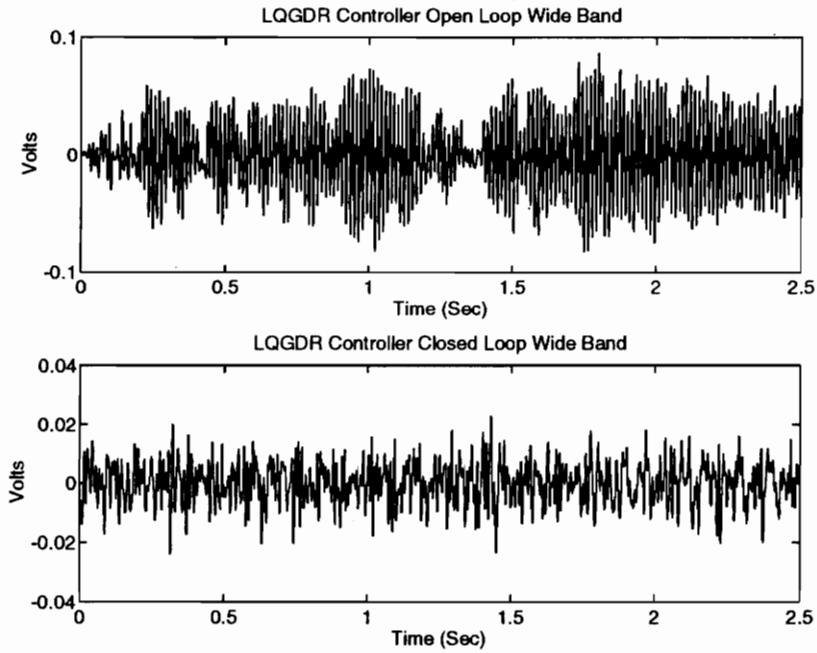


Figure 4.3.29 LQGDR Simulation Broad Band Open and Closed Loop

4.3.4.3 Conclusions Broad Band Disturbance

The broad band disturbance rejection results compare well with the LQGDR result. The Adaptive-Q controller result and the LQGDR controller result are compared in figure 4.3.30. The results show the Adaptive-Q controller performed within a factor of two of the baseline. The rejection provided by the control structure is 15 dB.

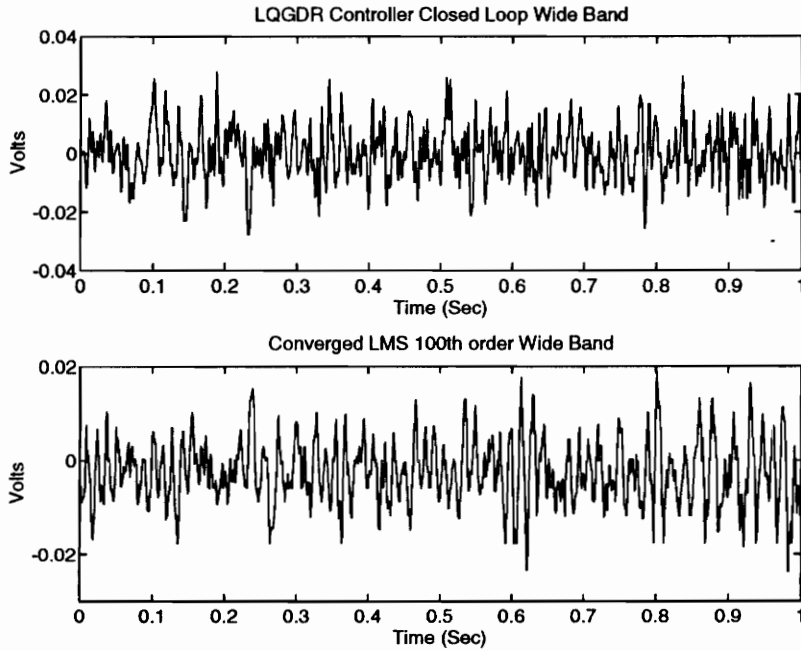


Figure 4.3.30 Broad Band LQGDR Simulation Versus Experimental Adaptive-Q

4.4 Robustness

The Adaptive-Q control technique uses the experimentally determined system identification. System identification techniques introduce modeling errors. Therefore, the ability of the control algorithm to adapt to these errors is important. Generically, control robustness describes this ability. The following discussion is a brief look into the robustness of this control structure. An in depth study is beyond the scope of the paper.

The results presented in this section are not from the beam itself. The results are simulations. The simulation allows the definition of an exact model. This would be impossible on the beam itself. Variations to the exact model simulate system modeling errors. Specifically, the poles and zeros of the system are moved by 10 percent in both the positive and negative directions(i.e. perturbed pole/zero= $(1 \pm .1)$ *original pole/zero). The

MATLAB m_file IDVAR.M performs this function and is contained in appendix A. The frequency responses for exact plant and the varied plants are shown in the following figures.

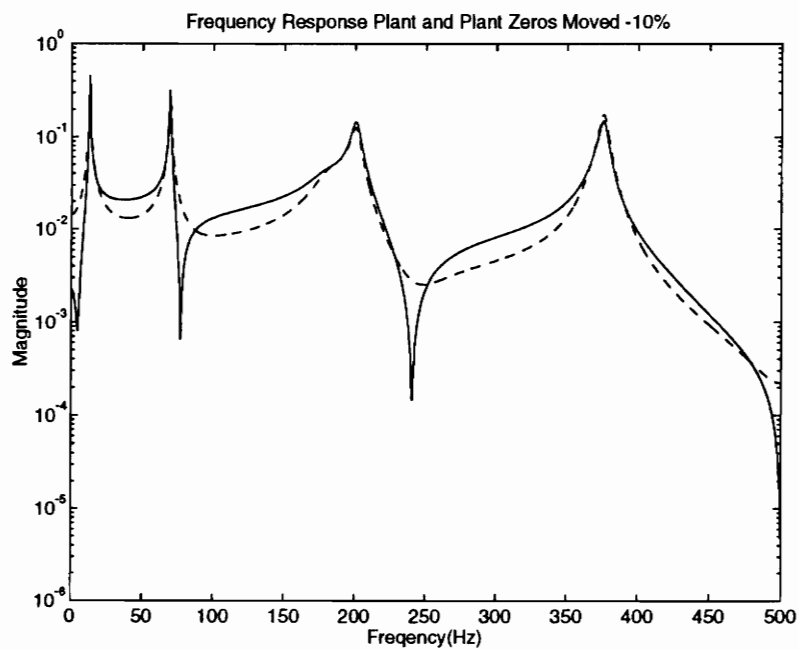


Figure 4.4.1 Exact Model (-) and Model with Zeros Moved -10% (--)

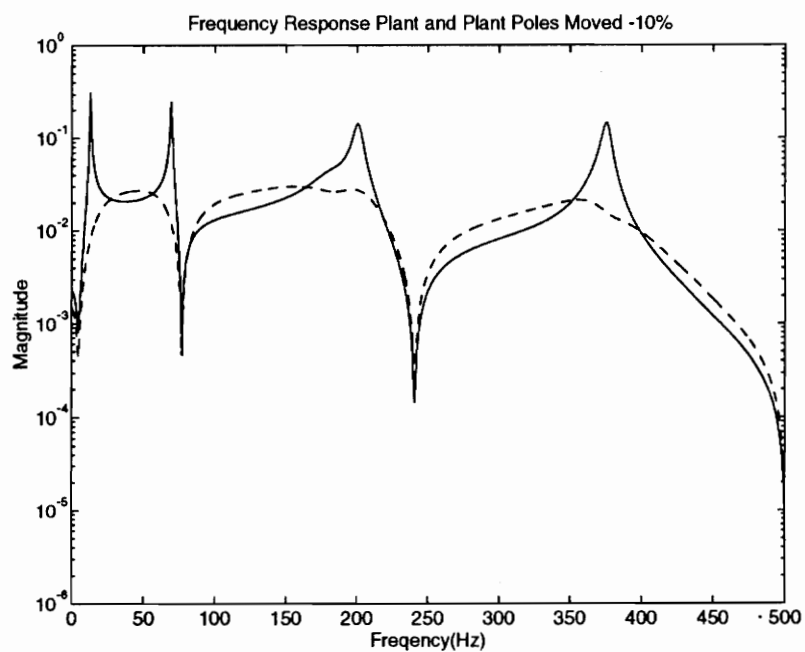


Figure 4.4.2 Exact Model (-) and Poles Moved -10% (--)

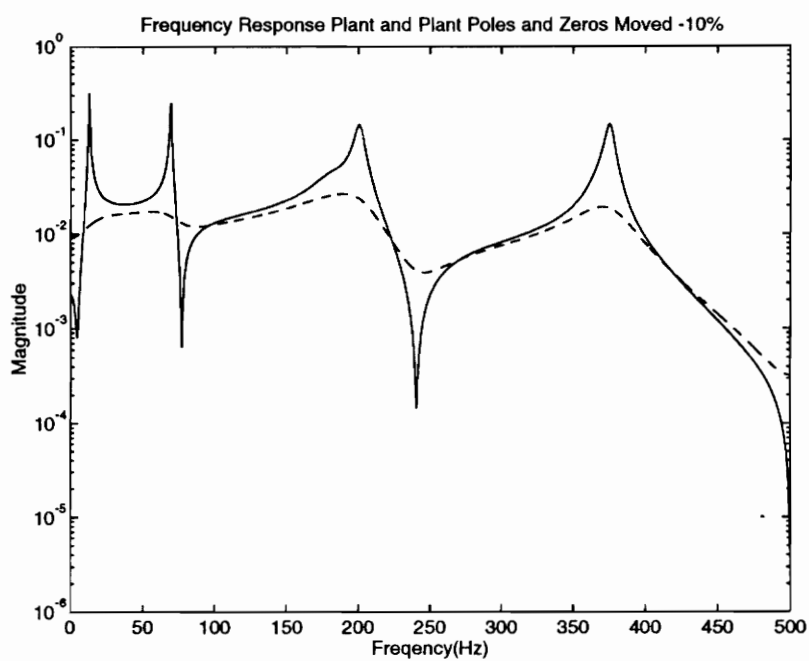


Figure 4.4.3 Exact Model (-) and Poles and Zeros Moved -10% (--)

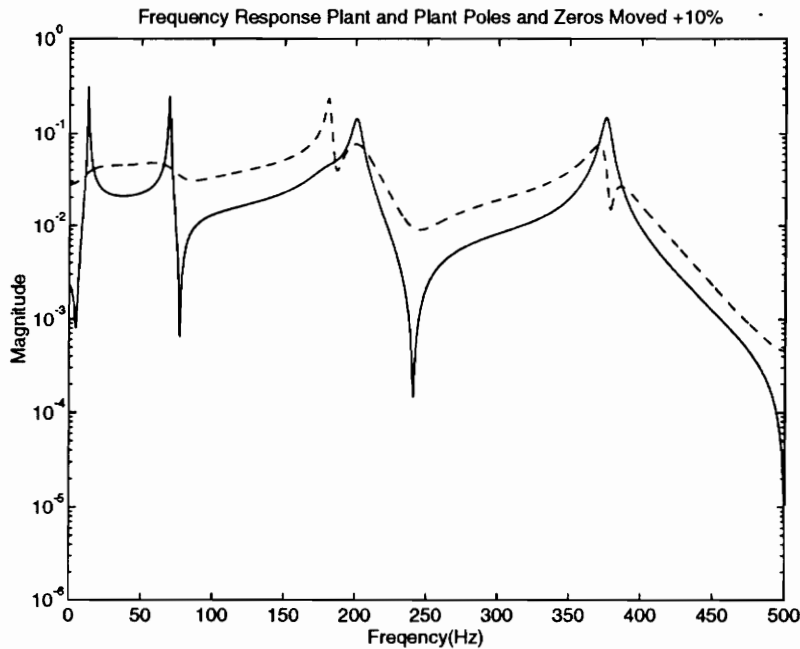


Figure 4.4.4 Exact Model (-) and Poles and Zeros Moved +10% (--)

The system simulation uses the true plant to act as the beam and the varied plants for the controller design. The disturbance input for the simulation is a 70 Hz sine wave. The results shown are the output for the closed loop Adaptive-Q controller. The outputs shown are for the first 7 seconds of operation because this period shows the transient introduced by the pole/zero variation. The final results converge to zero as in the prior simulations with sine waves. The MATLAB code to perform the simulation is ADVAR.M and is in appendix A. The results are encouraging for the systems. The system output, after an initial upset, recovers and have performance that rivals the true system.

The simulation varied the poles and zeros in a somewhat contrived manner. Further investigation into this area, with more real world system identification errors, needs to be undertaken. The simulation results are shown below.

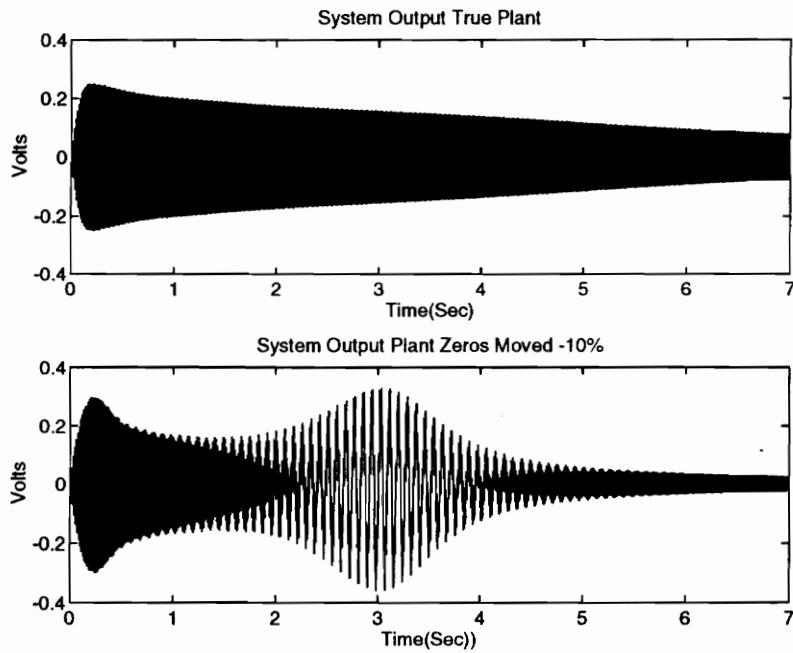


Figure 4.4.5 Control Output True Plant and Zeros Moved - 10%

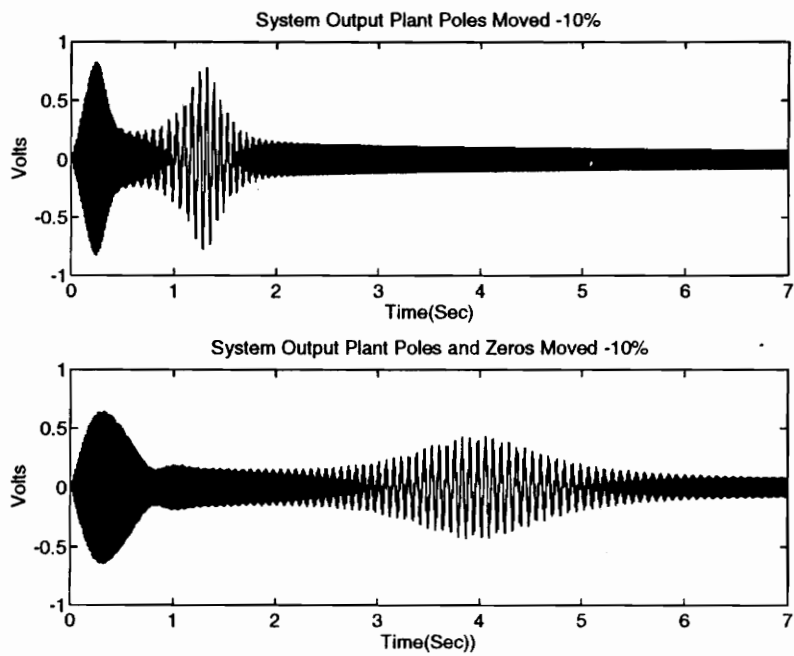


Figure 4.4.6 Control Output Poles Moved -10% and Poles and Zeros Moved -10%

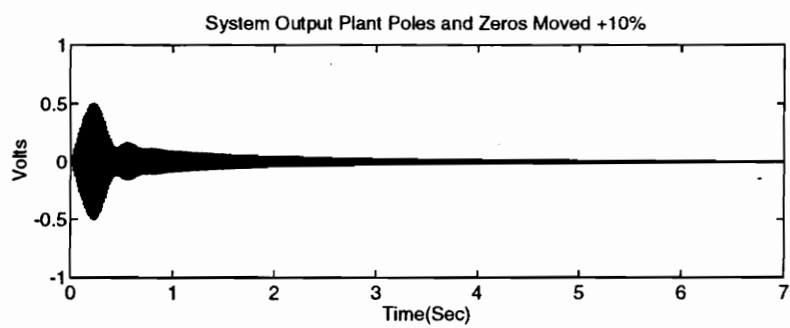


Figure 4.4.7 Control Output Poles and Zeros +10%

Chapter 5

5. Conclusions

Active control of vibrations in structures is an evolving field. Specifically, applications are being developed to do everything from make the interior of airplanes quieter to making buildings resilient to earthquakes. Further research will yield many more practical applications.

This work discusses the theoretical and experimental results obtained for an adaptive disturbance rejection controller. Unlike many other control schemes, the control structure does not need information concerning the disturbance signal. The previously untested control theory is validated with experimental results for various disturbances.

The disturbance results compare very well with simulations. Included is the design and implementation of a testbed to evaluate the control structure. The testbed consists of a cantilever beam with piezoceramic actuators and sensors. The control algorithm runs on a TMS320C30 DSP board by Spectrum Signal Processing.

The work answers many questions concerning the control structure. It also has also brought forth many new questions. Thus, I wanted to outline areas that need additional research. The first area is controller scalability. The experimental system is a multi-input single-output system(MISO), a multi-input multi-output (MIMO) system poses unique computational problems and this area needs investigation. As implemented, the testbed is limited by the control signal amplitude it can produce. This is due mainly to the power amplifiers. A new system with higher gain low noise amplifiers will allow the system to control higher magnitude disturbances. The system robustness to plant modeling errors is only briefly discussed in the work. It would be interesting to do further work into overall system robustness. This could include the use of fixed feedback control structures, such as H_∞ controllers, that have better robustness characteristics. The adaptive filter structure considered in the work is an FIR filter. A FIR filter is limited in terms of the response it can produce for a given filter order. IIR filters can usually emulate the same response with lower orders. Thus, stable IIR filter structures need to be considered. In addition to a new filter structure, other adaptive algorithms could be considered to see if the adaptation time could be decreased. The work in [13] addresses many of these techniques. It would be interesting to apply these techniques.

All of the aforementioned research areas point in one simple direction. The control technique needs to solve a real world problem. A real world application will be the ultimate test of the control algorithm.

References

- 1) Anderson, Brian D.O. and John B. Moore, *Optimal Control Linear Quadratic Methods*, Prentice Hall, 1990.
- 2) Åström, Karl Johan and Björn Wittenmark, *Adaptive Control*, Addison-Wesley Publishing Company, 1989.
- 3) Baumann, William T., "An Adaptive Feedback Approach to Structural Vibration Suppression", (Unpublished at writing date)
- 4) Beards, C.F. *Vibration Analysis and Control System Dynamics*, Ellis Horwood Limited, 1981.
- 5) Blevins, Robert D., *Formulas for Natural Frequency and Mode Shape*, Van Nostrand Reinhold, 1979.
- 6) Chassaing, Rulph, and Darrell W. Horning, *Digital Signal Processing with the TMS320C25*, John Wiley & Sons, 1990.
- 7) Chattopadhyay, Aditi and Charles E. Seeley, 1993, "Application of Design Optimization Techniques for Vibration Control of Structures", *SPIE Smart Structures and Intelligent Systems*, Vol.1917, pp 748-759.
- 8) Cole, Daniel G. 1992, "Design of, and Initial Experiments with, a MIMO Plate Control Testbed", Master's Thesis, VPI&SU.

- 9) Crawley, Edward F. and Javier de Luis, 1987, "Use of Piezoelectric Actuators as Elements of Intelligent Structures", *AIAA Journal*, vol. 25, no. 10, pp 1373-1385.
- 10) Dehandschutter, W., P.Sas, C.Bao, 1994, "Active Acoustic and Structural Control of Car Cabin Noise - Experimental Study on a Scale Model", *Proceedings of National Conference on Noise Control Engineering*, Fort Lauderdale, Fl, pp 409-441.
- 11) Fuller, C.R. and Gary P. Gibbs, 1994, "Active Control of Interior Noise in a Business Jet Using Piezoceramic Actuators", *Proceedings of National Conference on Noise Control Engineering*, Fort Lauderdale, Fl, pp 389-394.
- 12) Hagood, Nesbitt W., Walter H. Chung, Andreas von Flotow, 1990, "Modeling of Piezoelectric Actuator Dynamics for Active Structural Control", *AIAA-90-1087-CP*.
- 13) Haykin, Simon, *Adaptive Filter Theory*, Prentice Hall, 1986.
- 14) Inman, Daniel J, *Vibration with Control, Measurement and Stability*, Prentice Hall, 1989.
- 15) Kirk, Donald E., *Optimal Control Theory an Introduction*, Prentice Hall, 1970.
- 16) Oppenheim, Alan V. and Ronald W. Schafer, *Discrete-Time Signal Processing*, Prentice Hall, 1989.
- 17) Rubenstein, Stephan P. 1991, "An Experiment in State-Space Vibration Control of Steady Disturbances on a Simply-Supported Plate", Master's Thesis, VPI&SU.
- 18) Shahian, Bahram and Michael Hassul, *Control System Design Using MATLAB*, Prentice Hall, 1993.

- 19) Tay, T.T. and J.B. Moore, 1991, "Enhancement of Fixed Controllers via Adaptive-Q Disturbance Estimate Feedback", *Automatica*, Vol. 27, No.1 pp.39-53.
- 20) Viperman, Jeffery S, "Adaptive Feedforward Control of Broadband Structural Vibration", Master's Thesis, VPI&SU.
- 21) Widrow, Benard, and Samuel D. Sterns, *Adaptive Signal Processing*, Prentice Hall, 1985.

Appendix A MATLAB M-Files

BMSIMID5.M

```
%-----
% Function [f,k]=bmsimid5(a,b,c,d,sim)
% Purpose:   Calculate the state feedback and estimator gain
%            matrices.
% Inputs:    a = A State Matrix
%            b = B State Matrix
%            c = C State Matrix
%            d = D State Matrix
%            sim = Flag (1=Plot) (0=No Plot)
% Outputs:   f = Estimator Gain
%            k = State feedback gain
% Author:    Daniel H. Miller
%-----
function [f,k]=bmsimid5(a,b,c,d,sim);

ts=1/1000;           % Sampling Frequency
d=0;                 % Skip D Matrix
e=b;                 % Assume Disturbance enters like control
%-----
% LQR Design
%-----
q=c'*c;              % Q for LQR
r=1e-3;              % R for LQR Best Case LQG Design
% r=0.5e-3;          % R for reduced Gain LQG Design
k = dlqr(a,b,q,r);   % Feedback Gain

%-----
% LQG Design
%-----
g=b;                 % How Noise Enters Plant
sn = 1;              % State Noise LQG
on = 1e-4;           % Output Noise LQG
f=dlqe(a,g,c,sn,on); % Estimator Gain
f=a*f;               % Use Latest Estimate See Dr. V's Book

if (sim==1)
    %-----
    % Form Closed Loop System
```

```

% Note :Augmented System Only Used in Estimator
%-----
acl=[a -b*k;f*c a-b*k-f*c];    % A Matrix Closed Loop
bcl = [b;b];                    % B Matrix Closed Loop
ccl = [c zeros(1,15)];          % C Matrix Closed Loop
ecl = [e ;zeros(15,1)];          % Disturbance Closed Loop
dcl=0;                           % D- Matrix Closed Loop

%-----
% System Simulation
%-----
endtime=1;                       % End Time For Simulation
t=linspace(0,endtime,endtime*1/ts); % Samples
freq=70;                          % Freq to inject Nominally 20 Hz
in=sin(2*pi*freq*t);              % Input Signal

[yopn]=dlsim(a,e,c,0,in);          % Open Loop Simulation
[ycl]=dlsim(acl,ecl,ccl,0,in);      % Closed Loop Simulation

plot(t,yopn,t,ycl);                % Plot Open Loop and Closed Loop
title('Closed Loop and Open Loop Response');
%-----
% Frequency Response of System
%-----
disp('Press a Key ');
pause
[nump,denp]=ss2tf(a,b,c,d);
[numcl,dencl]=ss2tf(acl,bcl,ccl,dcl);
[hcl,wcl]=freqz(numcl,dencl,4096);
[hp,wp]=freqz(nump,denp,4096);
semilogx(wcl/(2*pi)*1000,20*log10(abs(hcl)),wp/(2*pi)*1000,20*log10(abs(hp)))
end;

```

SSADAP3.M

```
% -----
% [out]=ssadap3(a,b,c,d,e,ac,bc,cc,dc,f,k,in)
% Purpose:  ssadap3 simulates the Q-parameterization controller.
%           Two system are provided one for the controller model
%           and one for the plant model. Normally, these are the
%           same. They would differ to allow testing concerning
%           Robustness
% Inputs:   a = A Matrix true plant
%           b = B Matrix true plant
%           c = c Matrix true plant
%           d = d matrix true plant
%           e = e matrix true plant
%           numpc = Numerator of Controller Plant
%           denpc = Denominator of Controller Plant
%           in = disturbance input
%           f = estimator gain
%           k = state feedback gain
% Outputs:  out = System Output
% Author:   Daniel H. Miller
% -----
function [out]=ssadap3(a,b,c,d,e,ac,bc,cc,dc,f,k,in);

order=25;                                % Order of Adaptive Filter
p=1e5*eye(order);                        % RLS Algo
w=zeros(order,1);                        % weights
ua=zeros(order,1);                       % input adaptive
uf=zeros(order,1);                       % input filter

% Initialization Stuff
[n,m]=size(in);
y=zeros(1,1);
x=zeros(15,1);
xp=zeros(15,1);
yp=zeros(1,1);
yd=zeros(1,1);
xd=zeros(15,1);
yh=zeros(1,1);
xh=zeros(15,1);
out=zeros(1,1);
u=zeros(1,1);
filtout=0;
ah=ac-bc*k-f*cc;
```

```

for i=1:1:n-1
    filtout=w'*uf;          % Compute Adaptive Input

    %-----
    % Make Disturbance Input
    %-----
    yd=c*xd;
    xd=a*xd+e*in(i,:);

    if i ~= 1
        u=-k*xh;
    end;
    %-----
    % Compute Plant
    %-----
    y=c*x;
    x=a*x+b*u-b*filtout;
    out(:,i)=y+yd;

    %-----
    % Controller/Estimator States
    %-----
    yh=cc*xh;
    xh=ah*xh+f*out(:,i)-bc*filtout;

    adaptin=(out(:,i)-yh);    % Adaptive Filter Input
    %-----
    % Error Filter
    %-----
    yp=cc*xp;
    xp=ac*xp+bc*adaptin;

    %-----
    % Compute Adaptive Filter Weights
    % Use either RLS or LMS Algo
    %-----
    %[w,ua,p]=rlsalgo1(-1*yp,0,out(:,i),w,ua,p,order);
    [w,ua]=lmsalgo1(-1*yp,0,out(:,i),w,ua,order);

    %-----
    % Update Filter Inputs
    %-----
    for j=order:-1:2
        uf(j,1)=uf((j-1),1);
    end;

```

```
uf(1,1)=adaptin;  
end;
```

RLSALGO1.M

```
%-----
% function [w,x,P]=rlsalgo1(in,desired,w,u,P,order)
% Purpose:   Function to perform RLS Adaptation
%            "Adaptive Filter Theory" Simon Haykin pg 389
% Inputs:    in = Algo Input
%            actual = Actual Value of Signal to be adapted
%            desired = Signal to make into
%            w = matrix of filter weights
%            u = matrix of filter inputs
%            P = weights to determine adaptation speed
%            order = adaptive filter order
% Outputs:   w = Matrix of Algo weights
%            u = Matrix of Algo inputs
%            P = New weighting Matrix
% Author:    Daniel H. Miller
%-----
function [w,u,P]=rlsalgo1(in,desired,actual,w,u,P,order)

lamda=1.0;          % forgetting factor

%-----
% Algorithm per Haykin See Reference
%-----
k = (inv(lamda)*P*u)/(1+inv(lamda)*u'*P*u);
n = desired - actual;
w = w + n*k;
P = inv(lamda)*P - inv(lamda)*k*u'*P;
for i=order:-1:2
    u(i,1)=u((i-1),1);
end;
u(1,1) = in;
```

LMSALGO1.M

```
%-----
% function [w,x,P]=lmsalgo(in,desired,w,u,order)
% Purpose:    Function to perform LMS Adaptation
%             "Adaptive Filter Theory" Simon Haykin
% Inputs:     in = Adaptive Algo Input
%             actual = Actual Value of Signal to be adapted
%             desired = Signal to make into
%             w = matrix of filter weights
%             u = matrix of filter inputs
% Outputs:    w = Matrix of Algo weights
%             u = Matrix of Algo inputs
%-----
function [w,u]=lmsalgo(in,desired,actual,w,u,order)

beta=0.9;      % Fixed Step Size
%-----
% Algo Per Haykin See Reference
%-----
err=actual-desired;

for i=order:-1:1
    w(i,1)=w(i,1)-2*beta*err*u(i,1);
end;

for i=order:-1:2
    u(i,1)=u((i-1),1);
end;
u(1,1) = in;
```

•

WBDIST2.M

```
%-----
% Function [yopn,ycl,u1,t,out,in]=wbdist1(a,b,c,d,e,numfc,denfc);
% Purpose:   Design the LQGDR controller for the
%            system. Disturbance Input known from Sysid
% Inputs:    a = A Matrix
%            b = B Matrix
%            c = C Matrix
%            d = D Matrix
%            e = E Matrix
%            numfc = disturbance model numerator
%            denfc = disturbance model denominator
% Outputs:   yopn = Open loop response
%            ycl = Closed loop response
%            u1 = Control Input
%            t = simulation time for plot purposes
%            out = system output
%            in = Disturbance Input
% Author:    Daniel H. Miller
%-----
function [yopn,ycl,u1,t,out,in]=wbdist1(a,b,c,d,e,numfc,denfc);
%-----
% Optimal Control Design For Cantilever Beam
% Optimal Controller designed to reject specific frequency
%-----
ts=1/1000;           % Sampling Frequency

%-----
% Disturbance Frequency Model
%-----
[ad,bd,cd,dd]=tf2ss(numfc,denfc);   % State Space D-T

%-----
% Augment Plant With Narrow Band Filter
%-----
aaug=[a e*cd;zeros(6,15) ad];      % Augmented A Matrix A + Noise Model
baug = [b;0 ;0;0;0;0;0];           % Augmented B Matrix
caug = [c 0 0 0 0 0 0];            % Augmented C Matrix
daug = 0;                           % Augmented D Matrix
eaug = [e*dd;bd];                   % Augmented E Matrix

%-----
% LQR Design
```

```

%-----
q=caug'*caug;           % Q for LQR
r = 1e-10;              % R for LQR
k = dlqr(aaug,baug,q,r); % Feedback Gain

%-----
% LQG Design
%-----
gaug=eaug;              % How Noise Enters Plant
sn = 1;                 % State Noise LQG
on = 1e-4;              % Output Noise LQG
f=dlqe(aaug,gaug,caug,sn,on); % Estimator Gain
f=aaug*f;               % Use Latest Estimate

%-----
% Form Closed Loop System
% Note :Augmented System Only Used in Estimator
%-----
acl=[a -b*k;f*c aaug-baug*k-f*caug]; % A Matrix Closed Loop
bcl = [e;baug];           % B Matrix Closed Loop
ccl = [c zeros(1,21)];    % C Matrix Closed Loop
ecl = [e ;zeros(21,1)];   % Disturbance Closed Loop
dcl=0;                   % D- Matrix Closed Loop

%-----
% System Simulation
%-----
endtime=7;               % End Time For Simulation
t=linspace(0,endtime,endtime*1/ts); % Samples
noise=2.5*(rand(1,endtime*1/ts)-0.5); % White Noise Source
in=filter(numfc,denfc,noise); % Input Disturbance

[yopn]=dlsim(a,ep,c,0,in); % Open Loop Simulation
[ycl,xcl]=dlsim(acl,ecl,ccl,0,in); % Closed Loop Simulation
[u1]=-[zeros(1,15) k]*xcl'; % Control Signal

[nump,denp]=ss2tf(a,e,c,d); % Form transfer function plant
[numcl,dencl]=ss2tf(acl,ecl,ccl,dcl); % Form transfer function closed loop
[hcl,wcl]=freqz(numcl,dencl,4096); % Freq Response Closed Loop
[hp,wp]=freqz(nump,denp,4096); % Freq Response Open Loop

% Plot Plant/System
semilogx(wcl/(2*pi)*1000,20*log10(abs(hcl)),wp/(2*pi)*1000,20*log10(abs(hp)));

```

ADVAR.M

```
%-----
% Function [out,outz10,outp10,out10,out10p]=adapvar(a,b,c,d,e)
% Purpose:   The function computes the response of the adaptive
%            controller when the state space model has been altered
%            by +-10%
% Inputs:    a = A Matrix
%            b = B Matrix
%            c = C Matrix
%            d = d Matrix
%            e = e Matrix
% Outputs:   out = Normal System Output (No Change in Poles and Zeros)
%            outz10 = System Output Zeros moved -10%
%            outp10 = System Output Poles moved -10%
%            out10 = System Output Poles and Zeros moved -10%
%            out10p = System Output Poles and Zeros moved +10%
% Author:    Daniel H. Miller
%-----

function [out,outz10,outp10,out10,out10p]=adapvar(a,b,c,d,e);
%-----
% Vary Plant by +-10 %
%-----
[numz10,denz10,nump10,denp10,num10,den10,num10p,den10p]=idvar(a,b,c,d,0);

[az10,bz10,cz10,dz10]=tf2ss(numz10,denz10);    %Change to State Space
[ap10,bp10,cp10,dp10]=tf2ss(nump10,denp10);    %Change to State Space
[a10,b10,c10,d10]=tf2ss(num10,den10);          %Change to State Space
[a10p,b10p,c10p,d10p]=tf2ss(num10p,den10p);    %Change to State Space

[f,k]=bmsimid5(a,b,c,d,0);                    % Get Controller Gains
[fz10,kz10]=bmsimid5(az10,bz10,cz10,dz10,0);  % Get Controller Gains
[fp10,kp10]=bmsimid5(ap10,bp10,cp10,dp10,0);  % Get Controller Gains
[f10,k10]=bmsimid5(a10,b10,c10,d10,0);        % Get Controller Gains
[f10p,k10p]=bmsimid5(a10p,b10p,c10p,d10p,0);  % Get Controller Gains

ts=1/1000;                                     % Sample Rate
freq = 70;                                     % Disturbance freq
endtime=7;                                     % Endtime for Sim
t=linspace(0,endtime,endtime*1/ts);
t=t';
in=sin(2*pi*freq*t);                          % make Disturbance
%-----
```

```

% Compute Adaptive Controller Response to systems
%-----
[out]=ssadap3(a,b,c,d,e,a,b,c,d,f,k,in);
[outz10]=ssadap3(a,b,c,d,e,az10,bz10,cz10,dz10,fz10,kz10,in);
[outp10]=ssadap3(a,b,c,d,e,ap10,bp10,cp10,dp10,fp10,kp10,in);
[out10]=ssadap3(a,b,c,d,e,a10,b10,c10,d10,f10,k10,in);
[out10p]=ssadap3(a,b,c,d,e,a10p,b10p,c10p,d10p,f10p,k10p,in);

```

IDVAR.M

```
% -----
% function
[numz10,denz10,nump10,denp10,num10,den10,num10p,den10p]=idvar(a,b,c,d,plot);
% idvar moves the poles and zeros of the passed system model by
% plus and minus 10 and plots the results. In the +10 percent case the poles go unstable
% the polystab function is used to reflect these poles inside the unit circle.
% Inputs:      a = System A Matrix
%              b = system B Matrix
%              c = system C Matrix
%              d = System D Matrix
%              plot = Flag (1=Plot 0=NoPlot)
% Outputs:     numz10 = numerator zeros moved by -10%
%              denz10 = denominator zeros moved by -10%
%              nump10 = numerator poles moved by -10%
%              denp10 = denominator poles moved by -10%
%              num10 = numerator poles and zeros moved by -10%
%              den10 = denominator poles and zeros moved by -10%
%              num10p = denominator poles and zeros moved by +10%
%              den10p = denominator poles and zeros moved by +10%
% Author:      Daniel H. Miller
% -----
function[numz10,denz10,nump10,denp10,num10,den10,num10p,den10p]=idvar(a,b,c,d,plot)

[nump,denp]=ss2tf(a,b,c,d);
[zp,pp,kp]=tf2zp(nump,denp);

zp10=.90*zp;           % Move Zero by 10%
pp10=.90*pp;           % Move Poles by 10%
zp10p=1.1*zp;
pp10p=1.1*pp;
[numz10,denz10]=zp2tf(zp10,pp,kp); % System Zeros Moved
[nump10,denp10]=zp2tf(zp,pp10,kp); % System Poles Moved
[num10,den10]=zp2tf(zp10,pp10,kp); % System Both Moved
[num10p,den10p]=zp2tf(zp10p,pp10p,kp); % System Both Moved
% -----
% denominator unstable reflect poles back inside unit circle
% -----
den10p=polystab(den10p);

[hp,wp]=freqz(nump,denp,2048); % Freq Response
[hz10,wz10]=freqz(numz10,denz10,2048); % Freq Response
```

```

[hp10,wp10]=freqz(num10,den10,2048); % Freq Response
[h10,w10]=freqz(num10,den10,2048); % Freq Response
[h10p,w10p]=freqz(num10p,den10p,2048); % Freq Response

if (plot ==1)

    semilogy(wp/(2*pi)*1000,abs(hp),wz10/(2*pi)*1000,abs(hz10)); % Plot It
    title('Frequency Response Plant and Plant Zeros Moved -10%');
    ylabel('Magnitude');
    xlabel('Frequency(Hz)');
    print -deps c:\dhm\thesis\figures\idvarz10.eps;
    disp('Press a Key');
    pause;

    semilogy(wp/(2*pi)*1000,abs(hp),wp10/(2*pi)*1000,abs(hp10)); % Plot It
    title('Frequency Response Plant and Plant Poles Moved -10%');
    ylabel('Magnitude');
    xlabel('Frequency(Hz)');
    print -deps c:\dhm\thesis\figures\idvarp10.eps;
    disp('Press a Key');
    pause;

    semilogy(wp/(2*pi)*1000,abs(hp),w10/(2*pi)*1000,abs(h10)); % Plot It
    title('Frequency Response Plant and Plant Poles and Zeros Moved -10%');
    ylabel('Magnitude');
    xlabel('Frequency(Hz)');
    print -deps c:\dhm\thesis\figures\idvar10.eps;
    disp('Press a Key');
    pause;

    semilogy(wp/(2*pi)*1000,abs(hp),w10p/(2*pi)*1000,abs(h10p)); % Plot It
    title('Frequency Response Plant and Plant Poles and Zeros Moved +10%');
    ylabel('Magnitude');
    xlabel('Frequency(Hz)');
    print -deps c:\dhm\thesis\figures\idvar10p.eps;

end;

```

NFREQCF.M

```

%*****
%* Function nfreqcf
%* [nfres]=nfreqcf(hs,ts,l,n);
%* Purpose: Calculate the n Natural Freq. for a
%*          Clamped Free beam
%*          Alunimum Beam Assumed
%* Inputs
%* hs= height of beam (m)
%* ts = thickness of the beam (m)
%* l = Length Beam (m)
%* n = Number of Natural Frequencies to Compute
%* Outputs
%* nfreqs = Natural Frequencies
%* Author Daniel H. Miller
%* References : "Formulas for Natural Frequency and Mode
%*              Shapes" Robert D. Blevins Pg. 108
%* Ackowlegements : Darrell S. McAlister
%*              Aluminum Specs/Formulas
%*****
function [nfreqs]=nfreqcf(hs,ts,l,n);

nfreqs=zeros(1,n);
lam=zeros(1,n);
lam(1,1)=1.87510407;
lam(1,2)=4.69409113;
lam(1,3)=7.85475744;
lam(1,4)=10.99554073;
lam(1,5)=14.13716839;
if n > 5
    for i=6:1:n
        lam(1,i)=(2*i-1)*pi/2;
    end;
end;
Es = 1.65e9/12^2*6895;
Is = (1/12)*hs*ts^3;
ws = (0.1*12^3)*16.02;
m = ts*hs*ws;
for i=1:1:n
    nfreqs(1,i)=lam(1,i)^2/(2*pi*l^2)*sqrt(Es*Is/m);
end;

```

% Init natural Frequency var

% weighted wave Numbers see ref.

% Solutions to equation

% $\cos(\beta(i)*l)*\cosh(\beta(i)*l)=-1$

% where $\beta(i)=\lambda(1,i)/l$

% Modules of Elasticity (pa)

% Area moment of inertia (m^4)

% Density kg/m^3

% Mass per unit Length

STRAINCF.M

```

%*****
%* Function straincf
%* [x,sshapes]=straincf(n,l,ssize);
%* Purpose: Calculate the Strain Distribution for a Clamped
%*      Free beam
%* Note: Max Number of Modes = 5
%* Inputs
%* n = Number of Modes
%* l = length of the Beam
%* ssize = Increment between mshape calculations
%* Outputs
%* x      = Points where Strain Shapes were calculated
%* sshapes = Strain Shape Matrix
%* Author Daniel H. Miller
%* References : "Formulas for Natural Frequency and Mode
%*      Shapes" Robert D. Blevins Pg. 108
%*      "Use of Piezoelectric Actuators as
%*      Elements of Intelligent Struc."
%*      E.F. Crawley
%*****
function [x,sshapes]=straincf(n,l,ssize)
x=0:ssize:l;          % Set x points to calc Mode Shapes
[mr,nc]=size(x);      % Size of X
sshapes=zeros(n,nc);  % Init sshapes var
lam=zeros(1,5);
lam(1,1)=1.87510407;   % weighted wave Numbers see ref.
lam(1,2)=4.69409113;   % Solutions to equation
lam(1,3)=7.85475744;   % cos(beta(i)*l)*cosh(beta(i)*l)=-1
lam(1,4)=10.99554073;  % where beta(i)=lam(1,i)/l
lam(1,5)=14.13716839;
phi=zeros(1,5);
phi(1,1)=0.734095514;  % see ref
phi(1,2)=1.018467319;
phi(1,3)=0.999224497;
phi(1,4)=1.000033553;
phi(1,5)=0.99999855;
if n <= 5
    for i=1:1:n
        % Strain Shape 2nd derivative of Mode Shape See Ref #2
        sshapes(i,:)= (lam(1,i)^2/l^2)*cosh(lam(1,i)*x/l) ...
            +(lam(1,i)^2/l^2)*cos(lam(1,i)*x/l) ...
            -phi(1,i)*(lam(1,i)^2/l^2)*sinh(lam(1,i)*x/l) ...

```

```
        -phi(1,i)*(lam(1,i)^2/l^2)*sin(lam(1,i)*x/l) ;  
    end;  
end;  
if n > 5  
    echo 'M-File Only calculates 1st 5 mode shapes';  
end;•
```

STRAINZ.M

```

%*****
%* Function strainz *
%* [zeropnt,err]=strainz(n,start,stop,tol,maxit); *
%* Purpose: Calculate the Zero Crossing for Strain *
%*      Distribution for a Clamped Free Beam *
%* Note: Max Number of Modes = 5 *
%* Inputs *
%* n = Mode Number *
%* start = Start of Interval to search *
%* stop = End of Interval to search *
%* tol = Tolerance for Error *
%* maxit = Max. Iterations *
%* Outputs *
%* zeropnt = Points where Strain Shape crosses zero *
%* err = Error *
%* Author Daniel H. Miller *
%* References : "Formulas for Natural Frequency and Mode *
%*      Shapes" Robert D. Blevins Pg. 108 *
%*      "Use of Piezoelectric Actuators as *
%*      Elements of Intelligent Struc." *
%*      E.F. Crawley *
%*****
function [zeropnt,err]=strainz(n,start,stop,tol,maxit);
l = 1;
lam=zeros(1,5);
lam(1,1)=1.87510407; % weighted wave Numbers see ref.
lam(1,2)=4.69409113; % Solutions to equation
lam(1,3)=7.85475744; % cos(beta(i)*l)*cosh(beta(i)*l)=-1
lam(1,4)=10.99554073; % where beta(i)=lam(1,i)/l
lam(1,5)=14.13716839;
phi=zeros(1,5);
phi(1,1)=0.734095514; % see ref
phi(1,2)=1.018467319;
phi(1,3)=0.999224497;
phi(1,4)=1.000033553;
phi(1,5)=0.99999855;
count = 0;
savstart = start;
savstop = stop;
startfp = (lam(1,n)^2/l^2)*cosh(lam(1,n)*start/l) ...
+(lam(1,n)^2/l^2)*cos(lam(1,n)*start/l) ...
-phi(1,n)*(lam(1,n)^2/l^2)*sinh(lam(1,n)*start/l) ...

```

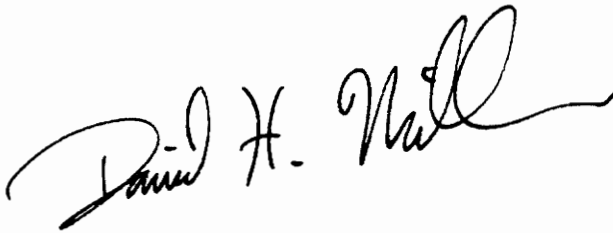
```

    -phi(1,n)*(lam(1,n)^2/l^2)*sin(lam(1,n)*start/l)
stopfp = (lam(1,n)^2/l^2)*cosh(lam(1,n)*stop/l) ...
    +(lam(1,n)^2/l^2)*cos(lam(1,n)*stop/l) ...
    -phi(1,n)*(lam(1,n)^2/l^2)*sinh(lam(1,n)*stop/l) ...
    -phi(1,n)*(lam(1,n)^2/l^2)*sin(lam(1,n)*stop/l)
if startfp > 0
    stop = savstart;
    start = savstop;
end;
fp = 1;
if n <= 5
%-----
% Bisection Method to find zero
%-----
    while ((abs(fp) >= tol) & (count < maxit))
        x=(start+stop)/2;
        % Strain Shape 2nd derivative of Mode Shape See Ref #2
        fp = (lam(1,n)^2/l^2)*cosh(lam(1,n)*x/l) ...
            +(lam(1,n)^2/l^2)*cos(lam(1,n)*x/l) ...
            -phi(1,n)*(lam(1,n)^2/l^2)*sinh(lam(1,n)*x/l) ...
            -phi(1,n)*(lam(1,n)^2/l^2)*sin(lam(1,n)*x/l)
        if (fp > 0)
            stop = x;
        end;
        if (fp < 0)
            start = x;
        end;
        error = 0 - fp;
        count = count+1;
    end;
end;
if n > 5
    echo 'M-File Only calculates 1st 5 mode shapes';
end;
zeropnt = x;
err = 0-fp;•

```

Vita

Dan Miller was born in Roanoke, Virginia on January 24, 1966. The majority of his childhood was spent in Waynesboro Virginia. Having always torn things apart as a child, deciding to begin an Electrical Engineering degree in 1984 seemed a natural. After receiving the degree in June of 1989, he began work for General Electric in Salem Virginia. After four years of work, he asked for and received a leave of absence to pursue a Masters degree. After graduation, he plans to work for GE Fanuc in Charlottesville, Virginia.

A handwritten signature in black ink, reading "Daniel H. Miller". The signature is written in a cursive, flowing style with a long horizontal stroke at the end.