AN EXPERIMENT IN STATE-SPACE VIBRATION CONTROL OF STEADY DISTURBANCES ON A SIMPLY-SUPPORTED PLATE

by

Stephen P. Rubenstein

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

MASTERS OF SCIENCE

in

Mechanical Engineering

APPROVED:

H. H. Robertshaw, Chairman

R. G. Leonard

W. T. Baumann

May, 1991

Blacksburg, Virginia

AN EXPERIMENT IN STATE-SPACE VIBRATION CONTROL OF STEADY DISTURBANCES ON A SIMPLY-SUPPORTED PLATE

by

Stephen P. Rubenstein

Committee Chairman: Harry H. Robertshaw
Mechanical Engineering

(ABSTRACT)

The formulation and implementation of a low-order vibration controller using Linear Quadratic (LQ) modern control theory is demonstrated. The controller is implemented to reject persistent disturbances of the fundamental mode of a simply-supported plate. The plate is excited by a 60 Hz harmonic excitation, a narrowband excitation centered at 60 Hz, and an impulse excitation caused by an impact hammer. The extraction of the eigen-properties of the first four modes of the plate is presented. The eigen-properties form the basis of the modal model of the plate. Also, a brief discussion of the control law is provided. Simulated and experimental results are presented for the transputer-based, parallel-processing control sysytem. The steady-state fundamental modal amplitude is attenuated by more than an order of magnitude at the disturbance frequency, and the modal damping is doubled. Additionally, the implementation issues of applying the LQG controller are addressed including temporal and spatial aliasing considerations.

Acknowledgements

I'd like to begin this work by recognizing some of the people who helped me. First, I want to thank my advisor, Harry Robertshaw. His guidance and support in the lab helped drive the project to this point, and his assistance outside the lab helped me stay at Tech to see it. Bill Baumann's assistance was indespensible when it came to both understanding and applying the controller. I'd also like to acknowledge Chris Fuller for his inspiration and Robert Leonard for his guidance.

The work was sponsored under an ONR grant N0014-88-K-0721.

The guys in the lab, Robert Wynn, Buddy Clark, and Ken Ellis, were always willing to add their knowledge, assistance and experience. To Will Saunders, all I can say is thank you. Thank you very much.

My parents and siblings were supportive and patient during my ordeal. They provided me with both a haven to rest and an avenue to vent my frustration.

Finally, I'd like to acknowledge my steadying influence, Michele Avra Leshan. Michele was my cheerleader, coach, psychiatrist, typist and photographer. Her love and support inspired me, and G-d willing, we'll be wed in April, 1992. I love you, dear, and thank you.

Contents

List of Figures

List of Tables

Nomenclature

Symbol	Description
A	discrete-time augmented system matrix.
A^r_{jk}	r^{th} modal residual at the j location due to a force at k.
a, b, h	dimensions of the simply-supported plate.
a, b	dimension of finite elements for spatial filtering (chapter 3).
B	discrete-time augmented system control input matrix.
C	damping matrix Newtonian coordinate system (chapter 3).
	plate output matrix (chapter 4).
$\mathcal{C}$	damping matrix generalized coordinate system.
C_a	augmented system output matrix.
C_d	disturbance model output matrix.
C_{sf}	smoothing filter output matrix.
D	bending stiffness.
D_u	control input feedthrough matrix.
D_w	disturbance input feedthrough matrix.
F	force input (chapter 3).
	plate system matrix (chapter 4).
F_a	augmented system matrix.
F_d	disturbance model system matrix.
F_{sf}	smoothing filter system matrix.
f_k	input force at location k.
G	plate control input matrix.
G_a	augmented system control input matrix.
G_d	disturbance model Gaussian mean-zero input matrix.
G_{sf}	smoothing filter control input matrix.
H_a	controlled state output matrix.
$H(\omega)$	response function matrix.
I	spatial portion of analytical modal filter (chapter 3).
	identity matrix.

K	stiffness matrix Newtonian coordinate system.
$\mathcal{K}$	stiffness matrix generalized coordinate system.
K_c	LQR feedback gains.
K_f	Kalman filter gains.
K_{fd}	discrete-time Kalman filter gains.
k_b	flexural wave number.
$k_{n,p}$	flexural wave number of the (n,p)th mode.
k_x, k_y	x- and y-direction wave number.
L	plate disturbance input matrix.
L_a	augmented system disturbance input matrix.
L_x, L_y	length and width of the plate.
M	mass matrix Newtonian coordinate system.
$\mathcal{M}$	mass matrix generalized coordinate system.
m	mass per unit area.
Q	augmented system state vector.
$\hat{Q}$	augmented system estimated state vector.
Q_{fc}	spectral density matrix of the disturbance.
Q_{fd}	covariance matrix of the disturbance.
q	generalized coordinate.
R_c	control input weighting matrix.
R_{fc}	spectral density matrix of the measurement noise.
R_{fd}	covariance matrix of the measurement noise.
r	plate state vector.
s	smoothing filter state vector.
u	control input.
$\tilde{u}$	control signal at the input to the smoothing filter.
$\ddot{u}$	plate acceleration.
v	disturbance model state vector.
$W_{n,p}$, W_o	modal contribution factor.
w	sensor output analytical modal filter (chapter 3). disturbance input.
w_p	out-of-plane displacement of the plate.
x	generalized coordinate.
y	measured output, modal accelerations.

Greek Letters

Symbol	Description
ζ	damping ratio.
η, ξ	natural coordinate system.
η	Gaussian mean-zero disturbance model input.
μ	measurement noise.
Φ	modal matrix.
ϕ	eigenvector.
Ψ	interpolation functions for the analytical modal filters. reciprocal modal vector.
ω_r	natural frequency of the r^{th} mode.

Chapter 1

Introduction

Active vibration control has been the focus of extensive research in recent years. One of the approaches is the modern control, feedback paradigm. This model-based approach has controllability, observability, and stability criteria established. Further, there are both steady and transient suppression methods available. For complicated problems the model may be quite large, however the modern control toolbox includes model and controller order reduction approaches. The reduced-order descriptions allow very large controllers to be experimentally implementable. Also the robustness of a controller can be analyzed and improved. The modern control approach includes all of the above named features as well the tools of linear systems theory when we apply linear controllers.

In this thesis, we will discuss the construction of a testbed for the application of modern controllers to reject persistent disturbances and the implementation of a Linear Quadratic Gaussian, LQG, vibration controller to compensate for the effects of harmonic and narrowband persistent disturbances on a simply-supported plate. The LQG controller marries the optimal control approach of the Linear Quadratic Regulator (LQR) with the optimal stochastic estimator known as the Kalman Fil-

ter. The model-based, feedback controller minimizes a quadratic cost functional that trades off between controlled variable performance and controller capability while the optimal estimator combines a stochastic, dynamic state estimator with a static, recursive least-squares estimator.

Disturbance inputs arise from a number of different sources. The engine in a car can be viewed as a disturbance input that causes the frame of the car to vibrate which may excite structural modes in the auditory range [1]. Thus, the isolation of the engine vibrations would provide a quieter ride. Further, in problems involving the automatic guidance and control of vehicles, the action of waves, wind gusts, updrafts, currents and terrain can be viewed as external disturbance forces. Disturbance compensation seeks to lessen the effects of these external forces on the structure's desired response.

The relative ease of designing modern control laws using linear systems theory has led to a number of analytical investigations of disturbance compensation using state-space controllers. However, implementing the controllers on real structures requires significant effort in the areas of data acquisition hardware and controller performance. This work demonstrates the effectiveness of a low-order LQG vibration controller to reject the disturbance inputs to a simply-supported plate with an experiment. Further, the work establishes a procedure for the experimental application of these controllers.

The LQG controller is flexible such that the change to an acoustic controller simply involves a change of the cost function. Further, by implementing a model of the

radiating field we can effectively control the radiating noise while only measuring the response of the ;structure if we use an acoustic cost function. Baumann [17] has formulated the acoustic control problem in this type of framework. First, the testbed needed to be verified with vibration control since it involves many fewer states than the acoustic controller. There is ongoing work in implementing the acoustic controller.

In the next section, we will review the current and most recent literature available on disturbance compensation. We are especially interested in the application of LQ controllers.

1.1 Literature Review

A number of disturbance compensation techniques have been proposed and some have been experimentally verified. Sievers and von Flotow [2] provide an overview and comparison of disturbance rejection methods. They discuss both feedforward and feedback compensators. The feedforward compensators include single-input-single-outpput, SISO, Least-Mean-Square, LMS, adaptive filtering, and the multi-input-multi-output, MIMO, multiple error LMS algorithm. They present SISO and MIMO feedback controllers in the classical, higher harmonic, and modern control framework. They showed that all of the methods placed pairs of complex poles at the disturbance frequency with a departure angle of 180^o on a root locus plot. The departure angle sends the poles into the left-hand plane, thus guaranteeing stability for low gains. They proposed improvements for two LQ-based techniques: [1]frequency shaped cost functional method, FSCF, [3], and [2]disturbance modeling

method, DM, [4]. FSCF shapes the weighting matrices in the frequency domain which can be viewed as filtering the output of the system with a narrowband filter. DM includes a colored noise model of the disturbance in the augmented state equations, and then the LQG gains are calculated.

Feedfoward techniques have been experimentally implemented for both vibration and acoustic control [5, 6, 7, 8]. Fuller, et al. [8] apply the adaptive LMS algorithm to control the sound transmission through a plate excited by a sinusoidal input. They compare the effectiveness of the controller when microphones and accelerometers are used as the error transducer. They found that the microphone is a better transducer for acoustic control since it contains the structural acoustic interaction information. Elliot, et al. [9] present a discussion of the important properties of the adaptive LMS control scheme which is not a model-based controller. The ability of the LMS algorithm to perform transient suppression has not been shown yet.

Both Meirovitch [10] and Mace [11] have developed feedback techniques which formulate transient compensators. Meirovitch, et al. discuss coupled control and independent modal space control, IMSC. They develop IMSC to address the control spillover into the unmodeled modes when coupled control is implemented. While a coupled control law can control multiple modes with a single actuator, the IMSC requires a single actuator per controlled mode, however it has fewer computational requirements. Mace developed a feedback controller to create 'disturbance-free' regions on a thin beam. He used a waveform approach to control the flexural vibrations caused by a transient disturbance propagating along a waveguide. The above two papers analytically formulate feedback, transient compensators. Next, we will look

at techniques to compensate for persistent disturbances.

Johnson [12] formulated a fundamental approach to disturbance compensation with a feedback controller. This method assumes that the disturbance is some kind of waveform that can be modeled with a state-space description. The control is broken into two parts. First, the feedback of the plant's states, or estimated states, is used to regulate the undisturbed plant. Second, the feedback of the disturbance states, or the estimated states, compensates for the disturbance effects. The gains for each of the two parts of the controller are calculated independently. For example, the plant state feedback gains can be calculated using a pole-placement technique while the disturbance state feedback gains are always a function of the disturbance model. Sorrells [13] extended Johnson's disturbance accomodating controller, DAC, to account for a stochastic input. He modeled the disturbance as the sum of a structured waveform and a Gaussian, random process. He demonstrated with a simulation that the Johnson method would compensate for the effects of a non-mean-zero, noisy sinusoidal input on a second order system. He compared this result with a LQG compensator for the same situation and found that the LQG compensator did not reject the structured, i.e., the sine-wave, portion of the disturbance. Possibly, he did not augment the LQG paradigm with a disturbance model of the sine-wave which is what we do to compensate for a non-Gaussian disturbance. Two other feedback techniques available for disturbance compensation are higher harmonic control [14] and unknown-but-bounded control [15].

Now, we would like to discuss some of the work that has been done to formulate the LQG disturbance compensation problem. Unlike the Johnson method above,

the LQG method assumes the knowledge of the statistics of the disturbance input, not the structured form of the input. The form of the disturbance only is important when the input is colored noise. Colored noise is white noise that has somehow been filtered, i.e., band limited. Specifically, we are interested in techniques to account for a structure driven by colored noise disturbance. As Sievers and von Flotow presented, there are two basic techniques for dealing with colored noise. The first is using the frequency-shaped cost functionals to determine the feedback gains. Gupta [3] shapes weighting matrices in the cost functional to force the solution to compensate for the colored noise input. The second method is augmenting the state-space model of the plant with some type of disturbance model which represents the filtering of a Gaussian, white noise sequence to form the colored noise input. The augmented system is in the standard LQG form where the augmented plant is driven by a Gaussian, white noise disturbance and the measurement noise is also assumed to be a Gaussian, white noise sequence. Only the statistical properties of the two random sequences are known, and they are used to formulate the compensator feedback gains for the estimate's augmented state vector [4]. Next, we would like to note a couple of applications of the LQG controller.

Sievers and von Flotow [1] formulated a LQG controller to actively isolate the vibrations of an engine from the flexible structure on which it is mounted. They simulated the response of the structure when the LQG compensator was formulated with both the FSCF method and the DM method. They assumed that the transmission of the vibrations of the engine can be modeled as a narrowband disturbance, and found that both techniques place a notch in the transmission curve at the disturbance frequency.

Rao [16] formulated a LQG controller for a turret-gun system where the turret must follow a command path to be aimed and the firing of the ordinance is a transient disturbance that excites the structural modes. He simulated both a full-order and a reduced-order disturbance compensator, and he demonstrated the controller's effectiveness as the turret followed both a step and a ramp command signal.

Baumann, et al. [17, 18] reported on an LQG framework to control the acoustic energy radiated from a flexible structure subject to transient and/or persistent disturbances of nearly arbitrary waveforms. They modeled the low-frequency radiated field with radiation filters which describe the far-field acoustic pressure as a function of the surface velocity of the structure, and augmented the plant with these states. Further, they formulated a state-space model of the disturbance and augmented the plant with these states. They examined the response of a clamped-clamped beam to transient and persistent harmonic, narrowband, and broadband disturbances for both a vibration and an acoustic cost function. They demonstrated that the LQG controller is effective as both a vibration and an acoustic controller. The experimental verification of this controller is one of the stated goals of this research.

We have reviewed the formulation and simulation of a few feedback control techniques, and now we will look at some experimental applications. Baz, et al. [19] implemented a modified independent modal space controller, MIMSC, to reject the transient excitation of a cantilevered rectangular beam. They controlled the beam with a piezoelectric actuator mounted at the base of the structure and measured the response with three displacement sensors. The outputs were numerically differen-

tiated to formulate a twelfth-order model. The modal displacements and velocities were calculated from the physical states and these were fed back to the controller. They found that the MIMSC provided better control than the IMSC when the beam was excited with a second-mode disturbance. They accounted for the better performance of the MIMSC due to its ability to time-share the single actuator to control both the first and second mode based on the highest instantaneous modal energy. The computational requirements limited the controller effectiveness to a bandwidth that included only the first two modes, both less than 11 Hz.

Wynn [20] implemented a MIMO, linear quadratic regulator to suppress the transient vibrations in a flexible continuum attached to an adaptive truss. The adaptive truss is used as the actuator to suppress the vibrations in the beam. This work highlights the experimental application of an optimal controller.

Finally, Rubenstein, et al. [21] have reported the experimental results of the application of a low-order LQG vibration controller to reject persistent harmonic and narrowband disturbances of a simply-supported plate. These results are also presented in this thesis.

1.2 Outline of the Thesis

First, we describe the design process of the simply-supported plate. In this chapter, we also discuss the instrumentation and controller hardware. Next, we set about to develop the modal model for the plate. Both analytical and experimental methods are examined. Then, we present the formulation of the LQG vibration controller.

We include both the continuous-time and discrete-time descriptions of the controller. Next, we look at the simulated and experimental results for the implementation of the low-order LQ vibration controller. Finally, we end with some concluding remarks.

Chapter 2

Experimentation

The experimental objectives are to build a testbed for Active Structural Acoustic Controller (ASAC) verification and to implement a Linear Quadratic (LQ) vibration controller on a simple structure. In the chapter, the design of the simple structure, the simply-supported plate, the instrumenting of the testbed, the controller hardware, and the experimental procedure are presented.

2.1 Plant Design

The testbed specifications require a structure with a simple model, i.e., a structure with well known behaviour to limit the emphasis of our problems to the implementation of the control law as opposed to modeling the structure. The structure must also be adaptable to perform both vibration control and acoustic control experiments in an LQG framework, and must be simple to fabricate. In this section, we will describe the process we used to design the simply-supported plate, and the result of the design process are the dimensions of the testbed.

We chose a simply-supported plate as the testbed. The simply-supported plate has

a closed-form solution for its two-dimensional bending motion. Wallace [22] determined the radiation resistance of the vibrational modes of a simply-supported plate. Therefore, the plate is both a structure with well-defined behaviour and easy to integrate into the acoustic control framework.

The desired plant should have low natural frequencies, should meet the thin plate assumption, have distinct eigenvalues and is easy to fabricate. The twelfth natural frequency is specified to be below 500 Hz to specify the minimum temporal sampling requirement of the data acquisition system. Based on the Nyquist sampling requirement, we established a minimum sample frequency of 1000 Hz for a twelve-mode compensator. The thin plate assumption allows the effects of shear deformation to be ignored in the solution of the bending-wave equation. The final two characteristics lead to the choice of a rectangular plate that is an $\frac{1}{8}$ in. thick. Since all of the eigenvalues are distinct on a rectangular plate, the mode shapes are also distinct. Candidate designs for the plate are analyzed by solving the bending-wave equation.

The two-dimensional bending wavefield in a thin plate can be described by [23]

$$D\left(\frac{\delta^4 w_p}{\delta x^4} + \frac{\delta^4 w_p}{\delta x^2 \delta y^2} + \frac{\delta^4 w_p}{\delta y^4}\right) = -m\frac{\delta^2 w_p}{\delta t^2} \tag{2.1}$$

where the plate dimension and coordinate system are shown in Fig 2.1. D and m are the bending stiffness and the mass per unit area of the plate respectively, and w_p is the out-of-plane displacement of the bending wave. Assume a simple harmonic, plane wave solution of the form

$$w_p(x, y, t) = W_o\, e^{i\omega t + ik_x x + ik_y y} \tag{2.2}$$

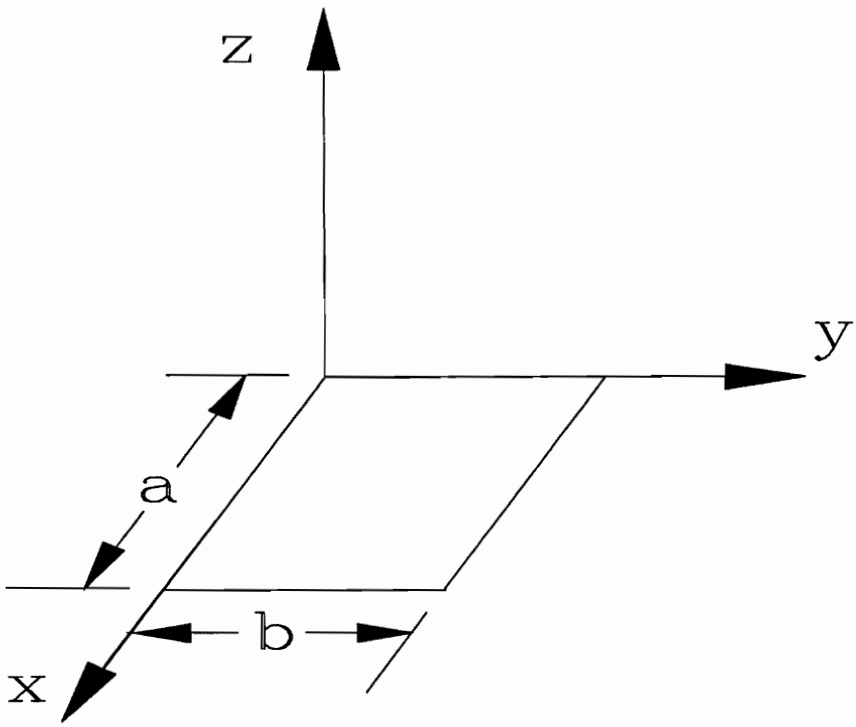


Figure 2.1: Rectangular plate dimensions and coordinate system

where k_x and k_y are the wave numbers in the x- and y-directions, respectively. The wave numbers are defined as

$$k_x = \frac{n\pi}{a}, \qquad n = 1, 2, \ldots \tag{2.3}$$

$$k_y = \frac{p\pi}{b}, \qquad p = 1, 2, \ldots \tag{2.4}$$

and the flexural wave number is

$$k_b = \sqrt{k_x^2 + k_y^2}. \tag{2.5}$$

Equation 2.2 is substituted into equation 2.1, and making use of the relation in equation 2.5, the flexural wave number is also

$$k_b = (\frac{m\omega^2}{D})^{\frac{1}{4}} \tag{2.6}$$

where k_b is a function of the response frequency, ω. Next the simply-supported boundary conditions are applied, and the surface displacement, in the (n,p)th mode at any time, is given by

$$w_{n,p} = W_{n,p} \sin\frac{n\pi x}{a} \sin\frac{p\pi y}{b} \tag{2.7}$$

where $W_{n,p}$ is the modal amplitude of the (n,p)th mode. Notice, equation 2.5 can be used to find the flexural wave number for the (n,p)th mode, $k_{n,p}$. Thus, by manipulating equation 2.6, the natural frequencies of the plate are

$$\omega_{n,p} = \sqrt{\frac{k_{n,p}^4 D}{m}}. \tag{2.8}$$

Since the flexural wave number is a function of the length and width of the plate and the bending stifness is a function of the thickness of the plate, a variety of combinations of a, b, h were analyzed based on the criteria presented at the beginning of this section. The chosen design was a cold-rolled steel plate, 0.5 m by 0.6 m long and 2.9 mm thick. The twelth natural frequency of this plate is below 500 Hz, the rectangular plate meets the thin plate assumption, all of the eigenvalues are distinct, and the VPI&SU had fabricated a similar plate previously and had little difficulty fabricating a plate with the above dimensions.

Duplicating the fixture fabricated by the Mechanical Engineering shop at VPI&SU for the acoustic lab, the plate is attached along its edges to thin steel shim stock, and the opposite edge of the shim stock is bolted into a steel frame. The shim stock is joined to the plate through machine screws and adhesive in order to emulate the simply-supported boundary condition. RTV silicone rubber adhesive is applied along both edges of the shim stock to reduce the transmission of the plate excitation to the frame. The testbed is isolated from the surroundings by placing it on a large granite block.

The next step of the experimental design is to instrument the structure.

2.2 Instrumentation and Control Hardware

The experimental setup is comprised of three major components: [1]the experimental plant, [2]the sensors and actuators, and [3]the data acquisition and control law hardware. The design and construction of the plant was discussed in the previous

section, and the specifications of the sensors, actuators, and the data acquisition and control law hardware are presented below.

The response of the plate is measured by twelve piezoelectric accelerometers arranged in a nonsymmetric array. We specified that the accelerometers be lightweight to reduce their mass loading effects and be adaptable for a variety of applications such as system identification and closed-loop experiments. PCB model number 336A04 accelerometers were chosen. These sensors have a nominal sensitivity of 100 $\frac{mV}{g}$ and weigh 4 grams. The sensors are mounted with wax and are located to ensure the observability of the first twelve plate vibration modes. The coordinates of each of the accelerometers is given in Table 2.1 where the origin is located at the lower left corner of the plate. A PCB model 483B07 twelve-channel charge amplifier with zero to 100 continuous gain dial on each channel is implemented to power the accelerometers and amplify the response signal. The output of the charge amplifier is connected to the data acquisition system.

Two Ling Dynamic Systems model V203 electromagnetic shakers were chosen as the disturbance and the control inputs, and they were mounted at accelerometer locations 5 and 8, respectively. These locations are chosen so the point force actuators have most of their authority on modes one and two. They are located near a node line of modes three and four, and, in the closed-loop experiments, this will reduce the excitation of these modes by the disturbance input and reduce the possibility of control spillover into these modes. We want to begin our experiments with a low-order compensator as this limits the number of states to track. The input command voltages are amplified through a current amplifier, designed by Robert

Table 2.1: Accelerometer locations

location	serial number	x (m)	y (m)
1	1093	0.12	0.38
2	1095	0.12	0.26
3	1096	0.12	0.11
4	1097	0.25	0.38
5	1098	0.25	0.26
6	1099	0.25	0.11
7	1100	0.35	0.38
8	1101	0.35	0.26
9	1102	0.35	0.11
10	1103	0.47	0.38
11	1104	0.47	0.26
12	1105	0.47	0.11

Salerno and shown in Fig. 2.2, which converted the voltage signal to a current signal. The current driven shakers apply a force output that is a constant proportion to the command voltage, for all values of the command voltage, within the frequency range of interest. The shakers do not exhibit this linear behaviour when driven by a voltage signal because of the electromagnetic design. Figure 2.3 shows the characteristic force response of the shakers to an input voltage passed through the current amplifiers. The force measurement dropouts, the ripples in the curves, occur at the natural frequencies of the plate where the load cell senses a small force that causes a large response. The gains for the disturbance and the control input are 0.53 N/V and 0.55 N/V, respectively.

The shakers are attached to the plate through a stinger rod assembly. The stinger rods are designed with a low flexural stiffness to isolate the plate from the rotary inertia effects of the shaker [24], and they must have a high axial compressive stiffness in order to transmit the force and prevent buckling [25]. The stinger assemblies are 1.125 in. lengths of 0.04 in. diameter piano wire.

The final piece of instrumentation is the data acquisition and control law hardware. In order to avoid the additional states introduced by time delays, the data acquisition hardware was required to sample all the input channels of data simultaneously. Also, we required a data acquisition system that could accomodate all the measurements necessary to identify the higher order plate vibrations. Therefore, Ellis [26] designed, developed and implemented a transputer-based data acquisition board (TransDAC) consisting of sixteen simultaneously sampled channels of input and sixteen simultaneous channels of output. The TransDAC along with a con-

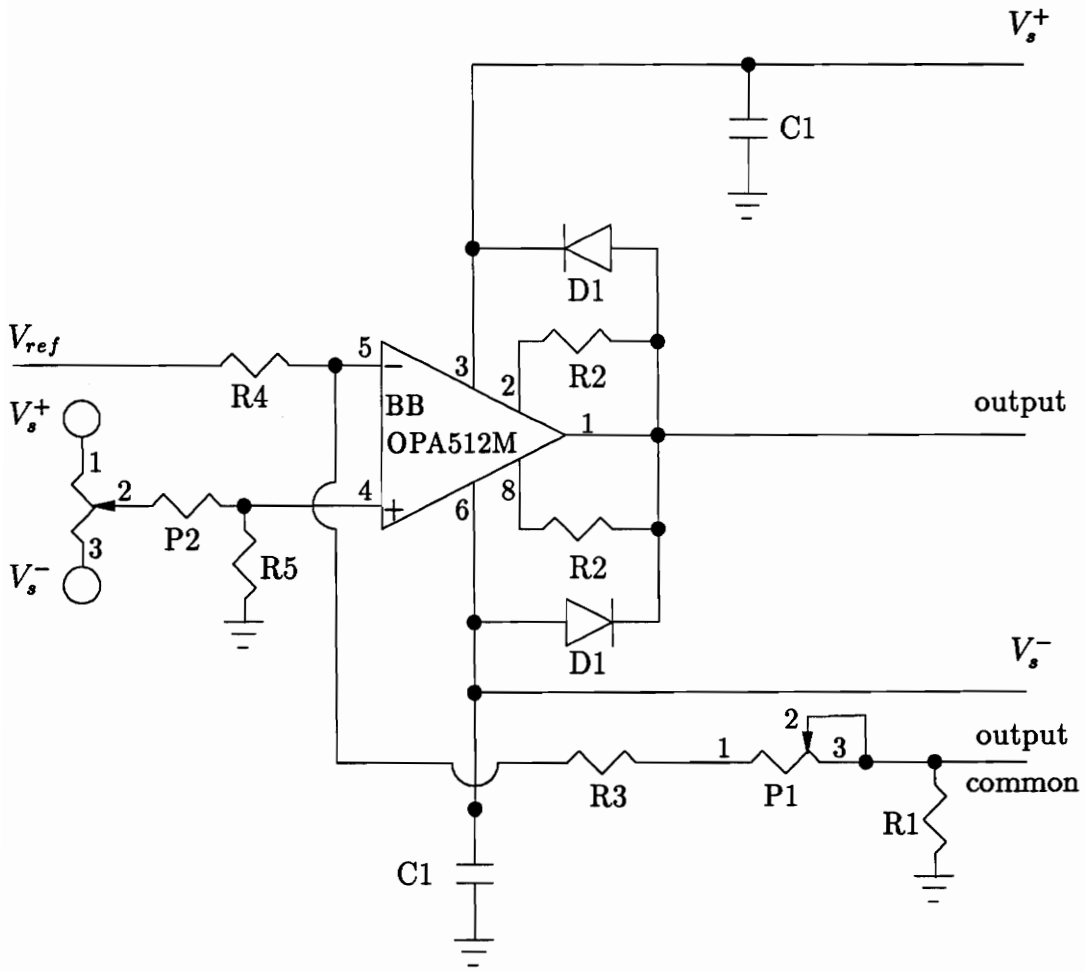


Figure 2.2: Current amplifier circuit

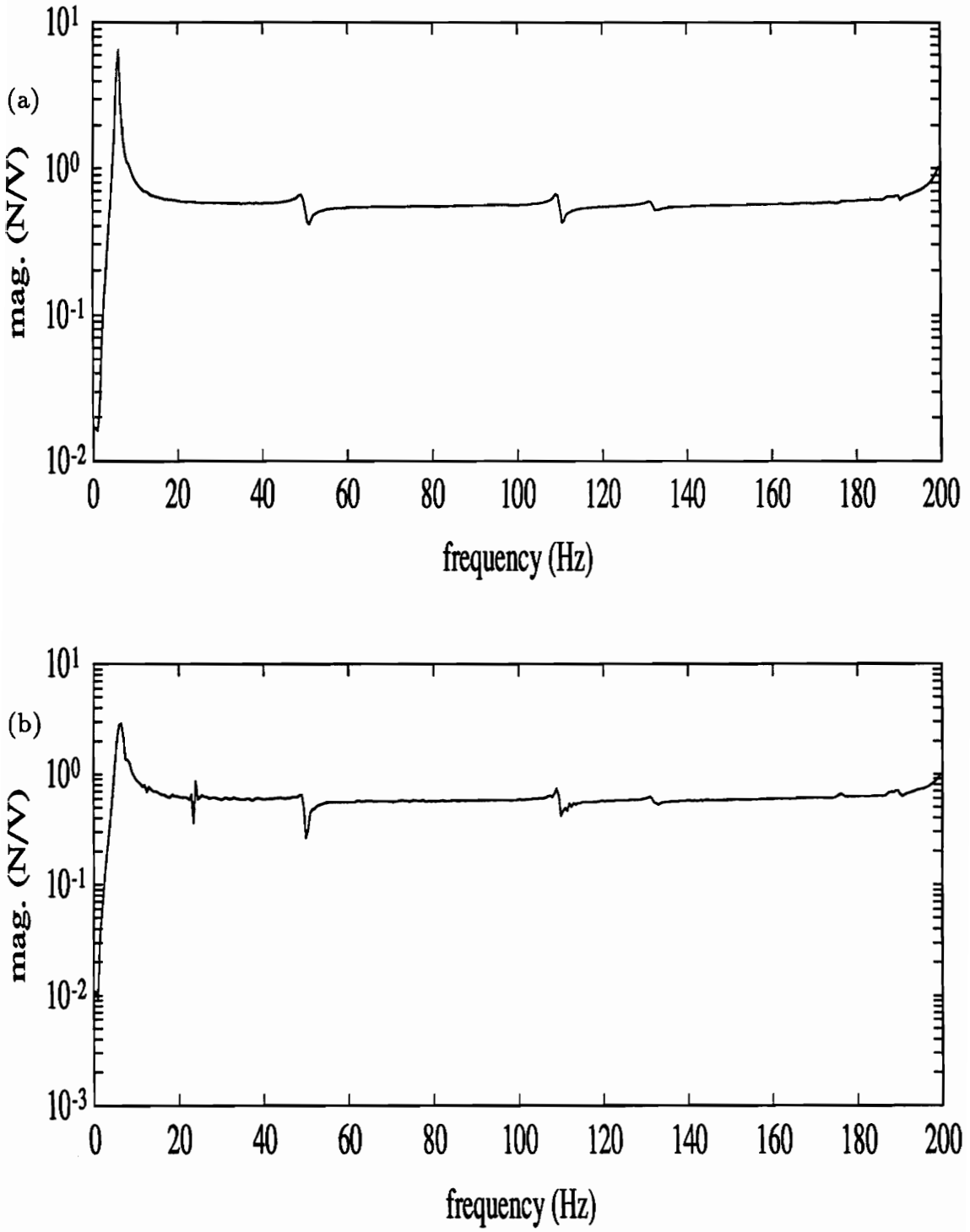


Figure 2.3: Shaker responses: (a)disturbance input, (b)control input

trol law transputer formed the transputer-based, parallel-processing control system which was installed in a 386/387 computer. The multitasking ability of the control system architecture, shown in Fig. 2.4, allowed the control law to be calculated on a single, 17 MHz T800 transputer while the TransDAC, with a T222 transputer, performed the data acquisition. The two parallel processes only communicated when the control law needed the input or when it was ready to send the control.

While the transputer-based control system was being built, a DT2818 data acquisition board was installed in the 386/387 computer. The DT2818 has four simultaneous analog-to-digital conversion (A/D) channels and two simultaneous digital-to-analog conversion (D/A) channels. The two-mode compensator, which will be discussed in more detail later, consists of eight difference equations for the state-space representation of the system and includes the conversion from the measurement to the modal domain. The DT2818, sampling four channels of input, had a sample speed of 455 Hz for the two-mode compensator while the transputer-based control system had a sample speed of 3300 Hz in the same configuration. The first two modes of the plate are approximately 50 Hz and 110 Hz, and empirical results had shown that good modal identification requires sample frequencies about ten times faster than the plate motions of interest. Baz, et al. [19] also found the same smpling requirement. The DT2818, in this configuration, could not even provide good identification of the second mode since it could not meet the temporal oversampling requirement. Additionally, the large number of inputs available on the TransDAC allowed spatial oversampling, i.e., more sensors than identified modes. This increased the expected accuracy of the modal acceleration estimates required for control. Therefore the transputer-based control system is used for all controller

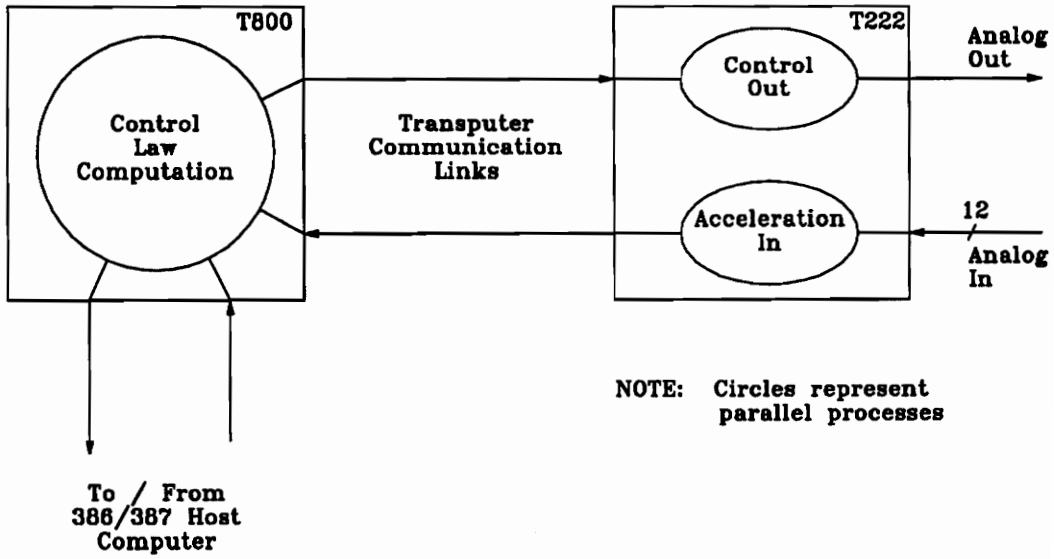


Figure 2.4: Control system architecture

results presented unless otherwise noted.

A schematic of the experimental setup is shown in Fig. 2.5. The final pieces of instrumentation needed to close the control loop are the smoothing filter and the disturbance signal generator. The staircase nature of the control signal is smoothed by a second-order, low-pass filter before driving the control shaker. The smoothing filter also limits the frequency content of the control signal above the bandwidth of the compensator. The disturbance shaker was driven by a signal generator for harmonic inputs or by a 286 AT computer configured with a transputer-based D/A system for narrowband inputs.

In concluding this chapter, the general experimental procedure is discussed.

2.3 Experimental Procedure

The experimental loop begins with the disturbance generation. In the persistent vibration suppression experiment, the signal is sent to the disturbance shaker and the plate response is allowed to reach its steady-state value. The response is measured by the twelve accelerometer array which are sampled by the TransDAC hardware discussed above. The sampled measurements are linearly transformed through a spatial filter into the modal domain. For the two-mode controller, the first two modal accelerations are used by the control law to estimate the states of the system which are used to calculate the control signal. The control signal is sent to the control shaker through the smoothing filter to close the loop.

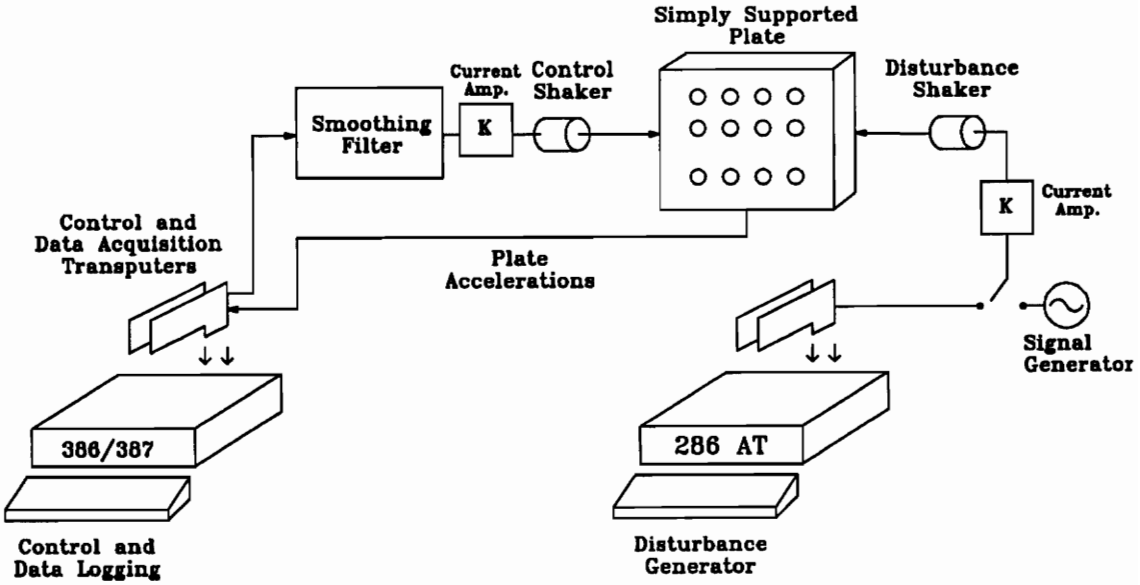


Figure 2.5: Experimental setup

The only variation for the transient vibration suppression is the disturbance is generated by an impact hammer alone without the contribution of the steady disturbance. The response of the plate is measured both when the controller is off and on to generate the open- and closed-loop responses.

In this chapter, the design of the experimental fixture was addressed, including hardware and procedural issues. In the next chapter, the real structure is modeled which will lead to the formulation of the control law.

Chapter 3

Modeling of the Simply-Supported Plate

Modeling is the chain that links the physical system, described in the previous chapter, with the mathematical formulation, i.e., the state-space description, presented in the next chapter. In this chapter, we are interested in modeling the simply-supported plate which forms the foundation of the system matrix portion of the state-space description. We wish to describe the motion of the simply-supported plate in a finite-dimensional manner to facilitate the implementation of the LQG control law. Therefore, a standard modal transformation is used to express the equations of motion in terms of the generalized set of coordinates as opposed to a Newtonian basis, and the modal description is truncated. The transformation requires the knowledge of the plant's eigen-properties.

Analytical and experimental structural modeling approaches are available. The tools we use to model the plate are: [1]the exact solution to the motion of a simply-supported plate, [2]a finite-element approach, and [3]an experimental modal method. The closed-form solution was described in the previous chapter, and it applies to an ideal simply-supported plate without any mass loading effects due to both sensors

and actuators. It was used to design the test fixture as well as a verification of both the finite-element and the experimental results. The finite-element model was implemented on NASTRAN. We can analyze the mass loading effects as well as simulate the steady response of the structure on NASTRAN. Finally, an experimental modal test was performed to formulate the eigen-properties used in the implementation of the control law. The experimental results are used since they include the effects of the physical boundaries which we will see are close to being simply-supported however they are slightly stiffer. The modal test is also an experimental verification of the NASTRAN results.

In this chapter, we will first develop the equations of motion of the plate and transform them to the modal domain in order to identify the necessary eigen-properties. Next, the finite-element and the experimental results will be presented and compared to the exact solution obtained in the previous chapter. Finally, we will discuss a few methods of transforming from the Newtonian to the modal basis including the one used in the implementation of the control law.

3.1 Equations of Motion

The motion of the plate can be described in the Newtonian basis by a set of p second order differential equation

$$M\ddot{x}(t) + C\dot{x}(t) + Kx(t) = F(t) \tag{3.1}$$

where M , C , and K are the mass, damping, and stiffness matrices. The response, $x(t)$, is a p-dimensional vector described at p discrete locations on the structure. The p locations represent the p sensors used to measure some form of the response. The structure is assumed to be lightly damped and the system matrices are p-dimensional and symmetric. $F(t)$ are the external forces such as the control and disturbance inputs which total m point forces.

The above set of simultaneous differential equations are coupled through one or more of the system matrices. The solution of the above matrices require the knowledge of the mass and stiffness matrices at some location. One method of extracting these matrices is a finite-elment approach. An accurate description of the physical mass and stiffness matrices requires a dense finite-element model which has a high order. Therefore, we would like to transform the coordinates to a basis which transform the coupled set of simultaneous equations into an independent set of equations which would require a smaller-order description. If we choose the transformation matrix correctly, we still have a physically significant description of the motion of the plate. Let us rewrite the response as

$$x(t) = \Phi\, q(t) \tag{3.2}$$

where Φ is a constant, nonsingular, p-dimensional matrix and $q(t)$ is a vector of generalized coordinates. Substituting equation 3.2 into equation 3.1, we find

$$M\,\Phi\ddot{q}(t) + C\,\Phi\dot{q}(t) + K\,\Phi q(t) = F(t). \tag{3.3}$$

Next, we premultiply equation 3.3 by Φ^T, and obtain

$$\mathcal{M}\ddot{q}(t) + \mathcal{C}\dot{q}(t) + \mathcal{K}q(t) = \mathcal{F}(t) \tag{3.4}$$

where the matrices

$$\mathcal{M} = \Phi^T M \Phi \quad \mathcal{C} = \Phi^T C \Phi \quad \mathcal{K} = \Phi^T K \Phi \tag{3.5}$$

are symmetric, and

$$\mathcal{F}(t) = \Phi^T F(t) \tag{3.6}$$

is a p-dimensional vector of the generalized forces associated with the generalized coordinates $q(t)$. Equation 3.4 describes the motion of the plate in the generalized coordinate domain, however this simultaneous set of equations may still be coupled unless the transformation matrix Φ is chosen properly.

When Φ is the modal matrix, whose columns are the solution to the eigenvalue problem of the system, equation 3.4 becomes a set of p independent, second-order differential equations. Further, we can normalize the eigenvectors to the mass of the plate such that

$$\mathcal{M} = I_p$$

$$\mathcal{C} = \begin{bmatrix} 2\zeta_1\omega_1 & 0 & \dots & & 0 \\ 0 & \ddots & 0 & \dots & 0 \\ 0 & 0 & 2\zeta_r\omega_r & 0 & 0 \\ 0 & \dots & 0 & \ddots & 0 \\ 0 & & \dots & 0 & 2\zeta_p\omega_p \end{bmatrix} \quad \mathcal{K} = \begin{bmatrix} \omega_1^2 & 0 & \dots & & 0 \\ 0 & \ddots & 0 & \dots & 0 \\ 0 & 0 & \omega_r^2 & 0 & 0 \\ 0 & \dots & 0 & \ddots & 0 \\ 0 & & \dots & 0 & \omega_p^2 \end{bmatrix} \tag{3.7}$$

where I_p is the p-dimensional identity matrix, ω_r and ζ_r are the modal frequency and damping of the r^{th} mode of the plate. We have chosen a rectangular plate and all of the eigenvalues are distinct. Equation 3.2 through equation 3.7 represent the standard modal transformation of the equations of motion. This representation requires the knowledge of the eigen-properties of the plate, specifically, the natural frequencies, modal damping and the mass-normalized eigenvectors. There are both analytical and empirical approaches to find the eigen-properties of a structure and they fall in the area of modal analysis. We used the closed-form solution discussed in the last chapter, a finite-element model, and an experimental modal method to formulate the modal model of the plate. The results of the latter two techniques are presented in the next section, and the results are compared with each other as well as with the closed-form solution previously discussed.

3.2 Modal Analysis

We wish to model the dynamic response of the simply-supported plate using the description in equation 3.4 where the system matrices are described in equation 3.7. Modal analysis is the means to find the required eigen-properties of the modal model, and we implemented a finite-element and an experimental approach to find the natural frequencies, modal damping and eigenvectors of the plate.

A finite-element model of the plate was built and implemented on MSC NASTRAN to account for the mass loading of the plate by the accelerometers and the shakers. Further, we can use the finite-element model to simulate the forced and natural response of the plate. This feature helped us size the sensors and actuators and

provides an analytical tool to predict the experimental behaviour. Also, the model supports the fabrication of a testbed for controller implementation as it is flexible to add different types of sensors and actuators. Finally, we built the finite-element model with more complicated problems in mind. For example, for a more complicated structure an experimental modal approach may not be feasible, so some type analytical tool would be needed.

The experimental modal method is an experimental verification of the analytical model. We can use the results to adjust the analytical model's characteristics to better meet the real structure. Also, the experimental data adds to the foundation of the plate as a testbed for controller implementation, and qualifies the integrity of the structure, i.e., the assumed simply-supported boundary conditions are shown to be close to the ideal based on the plate's response. For example, the experimental data may be used to design a robust controller.

In the next two sections, we will discuss the implementation of the analytical and experimental modeling methods. This discussion will be followed by the comparison of the results of the two methods with each other as well as with the closed-form solution.

3.2.1 Finite-Element Model

The finite-element model is used to establish an analytical spatial description of the plate, to formulate a modal model, and to analyze the forced response of the plate. The PC-based MSC-PAL2 was used as a preprocessor to build the 360-node model

of the plate. The model has 343 quadrilateral plate elements, 2.9 mm thick, with bending stiffness. The nodes are arranged to include the locations of the accelerometer array and the actuator locations. The mass of the sensors, 4 g each, and the mass of the stinger pad, 2 g each, are added to the model as nodal masses. The stinger pads are the aluminum nuts that are glued to the plate, and the stingers are screwed into the pads. We assume that the inertial effects of the shakers are isolated from the plate by the stingers. The automatic node generation and connectivity are used. The plate has a Young's modulus of elasticity of 207 GPa, Poisson's ratio of 0.292, and a density of 7798.6 $\frac{kg}{m^3}$. The in-plane translation and the rotation about the axis perpendicular to the surface of the plate at all of the nodes is fixed to zero. Along the edges of the model, the out-of-plane translation is fixed as well as the rotation about the in-plane axis perpendicular to that edge. The MSC-PAL2 model generates the MSC NASTRAN input file. We add to this file a proportional damping of 1% critical across the bandwidth of the modal analysis. NASTRAN implements a modified Givens approach to extract the natural frequencies and the eigenvectors of the structure. The results of the eigenvalue analysis is given after we discuss the experimental modal approach.

3.2.2 Experimental Modal Method

The empirical model is built as an experimental verification of the finite-element model and will be used in the controller formulation. The experimental modal method encompasses the extraction of the eigen-properties of the structure from a group of frequency response functions, FRF, in some sense that limits the mean-squared error between the curve-fit and the measured response. Therefore, we can

separate this method into two tasks: [1]the collection of the response functions such that they contain all of the eigen-property information, and [2]the curve-fitting of the data. We measured the FRF of the simply-supported plate for both an impulse and a burst-random excitation. The burst-random test responses were curve fit with MODHAN to extract the eigen-properties of the plate. The tests and the curve fitting are discussed below.

Impact Test of the Simply-Supported Plate

The impact testing was used to initially verify the the analytical models. We wanted to compare the natural frequencies in the zero-to-500 Hz bandwidth obtained from the finite-element approach with those noted from the response functions. These were also compared with the ideal solutions. The plate was struck with a modally tuned hammer equipped with a load cell, and the response was measured with a colocated accelerometer. The data acquistition and FRF caculation were performed with the PC AT-based Transfer Function Analysis System, TFAS, written by Robert H. Wynn and William R. Saunders. TFAS is a one-channel pair analyzer that can be run in either an impulse or a random mode. Three different tips on the hammer were used to yield a flat input spectrum up to 500 Hz. Fifty ensembles were averaged during each test, and the resulting FRF were scanned to compare the natural frequencies with the analytical set.

When we impact tested the first plate that was fabricated we could not find a consistent set of empirical natural frequencies. We wanted to eliminate the simply-supported boundary conditions, so we hung the plate to emulate free-free conditions

and impact tested the plate. We compared the results with the empirical models for free-free plates determined by Leissa [27], and they did not agree. We noticed that the plate was slightly curved and this accounted for the discrepancy. Therefore, a second plate was fabricated from cold-rolled steel since it is flatter than the hot-rolled steel of the first plate. Once the second plate was set in the fixture, we used an impact test to verify the analytical model, specifically, the material properties. We used the impact test approach to verify the finite-element model and used a burst-random approach to establish the experimental modal model.

Burst-Random Modal Test

We wanted to extract the modal properties of the plate from empirical data. Specifically, we wanted the data to contain the natural frequencies, the modal damping, and the eigenvectors. A burst-random input drives the plate with random noise for a part of the data-acquisition window and then the input is off for the rest of the window. This input has the feature of imparting input energy in all of the bins with a random input while also containing the natural response in the data window.

The burst random input was generated on the 386 SX by a program written by William R. Saunders. We configured the plate in the control experiment mode, i.e., the shakers and the accelerometers were arranged in their experimental locations. The input was used to drive the disturbance input shaker. The input was measured with a load cell while the response was measured at the twelve accelerometer locations and a roving accelerometer was used to take an additional 32 responses. The locations of all of the responses are marked on the plate with the blue numbers.

Once again, TFAS was used to perform the data-acquisition and FRF calculation. Seventy-five averages per channel pair were collected and the data was analyzed up to 500 Hz. Once we collected the FRF, we calibrated them into physical units and uploaded the results to the VAX system for the curve fitting routine.

Curve Fitting of the Empirical Data

MODHAN was used to extract the eigen-properties from the FRF collected during the burst random test. In this section, we will briefly discuss how we used the MODHAN results to extract the modal properties. The accelerance plots can be described by [28]

$$\frac{\ddot{x}_j(\omega)}{f_k(\omega)} = H_{jk}(\omega) = \sum_{r=1}^{N} \frac{\omega^2 \phi_j^r \phi_k^r}{(\omega_r^2 - \omega^2)^2 + i2\zeta_r \omega \omega_r} \tag{3.8}$$

where we assume that the structure is proportionally damped. $\ddot{x}_j(\omega)$ is the acceleration at location j due to the input, $f_k(\omega)$, at k. ω_r and ζ_r are the natural frequency and damping of the r^{th} mode while ϕ_j^r is the component of the associated mass-normalized eigenvector at location j. We can rewrite the above equation as

$$\begin{aligned} H_{jk}(\omega) &= \sum_{r=1}^{N} \frac{\omega^2 \phi_j^r \phi_k^r}{(i\omega - p_r)(i\omega - p_r^*)} \\ p_r &= -\zeta_r \omega_r + i\omega_r \sqrt{1 - \zeta_r^2} \\ &= -\delta_r + i\omega_{dr} \end{aligned} \tag{3.9}$$

where ω_{dr} is the damped natural frequency. Next, we take the partial fraction expansion of equation 3.9

$$H_{jk}(\omega) = \sum_{r=1}^{N} \left[\frac{A_{jk}^{r}}{(i\omega - p_r)} + \frac{A_{jk}^{r*}}{(i\omega - p_r^*)} \right] \tag{3.10}$$

where

$$A_{jk}^{r} = \frac{\phi_j^r \phi_k^r}{2i\omega_{dr}} \tag{3.11}$$

is the residual of the FRF in the r^{th} mode. MODHAN extracts the the natural frequency, damping ratio, and the residual for each mode. We can use equation 3.11 to find the mass-normalized eigenvectors.

Since, the disturbance excitation for the closed-loop experiments was planned to be between the first and second natural frequencies, we used MODHAN to extract the eigen-properties for the first four modes, from 40 Hz to 200 Hz, from the FRF at the twelve accelerometer array locations. The curve fits for the fifth, colocated with the disturbance input, and the eighth, colocated with the control input, accelerometer locations are shown in figure 3.1 and figure 3.2, respectively.

Now, we have extracted the eigen-properties from the experimental data, and we will next compare the results with analytical model.

3.2.3 Analytical and Empirical Modal Models

The natural frequencies are shown in table 3.1. Notice that the natural frequencies do not vary by more than 7% in the first four modes. Th experimentally extracted damping ratios, less than 1.5% critical in the first four modes, validates the lightly-

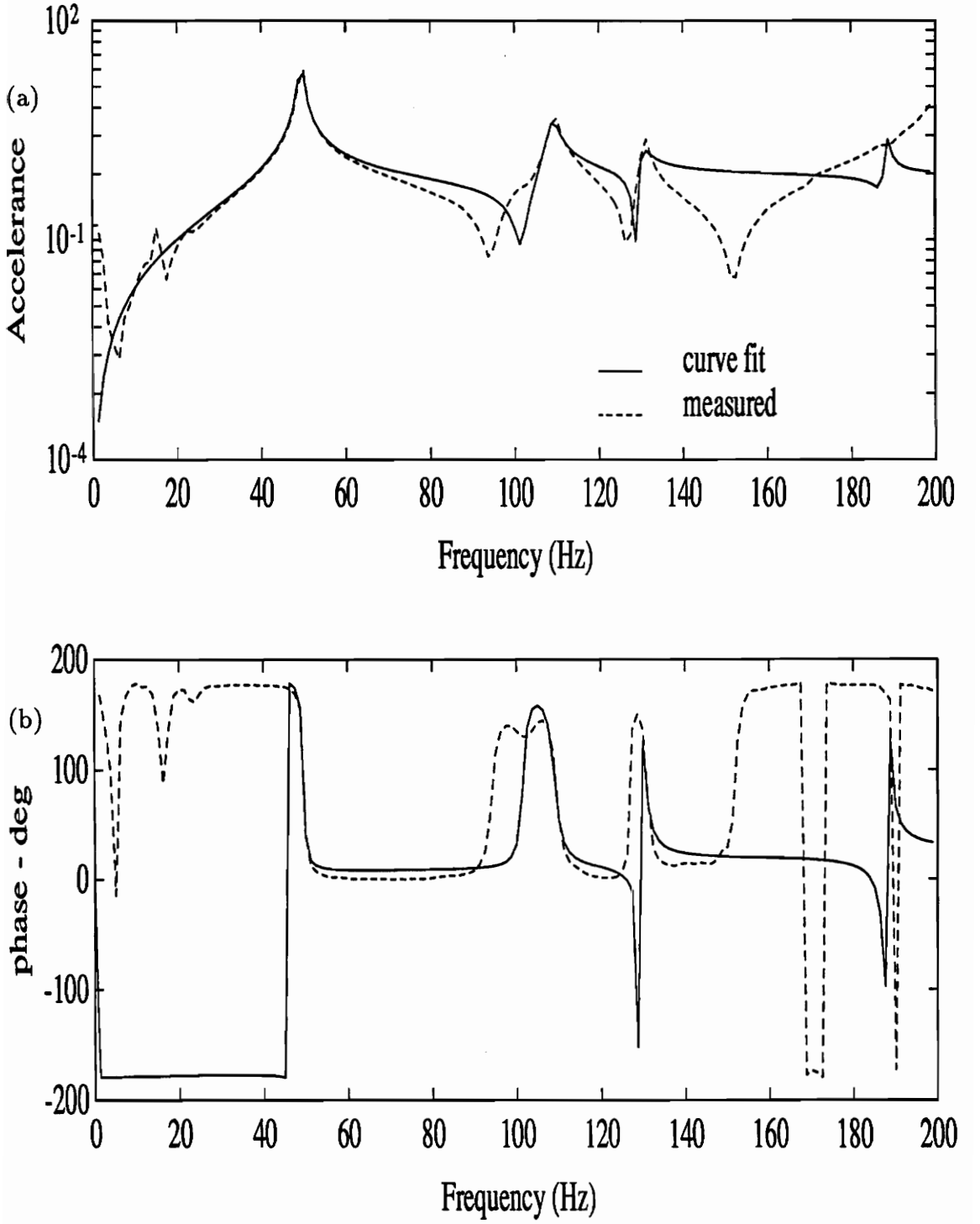


Figure 3.1: Frequency Response Function for Accelerometer 5: a) Magnitude, b) Phase

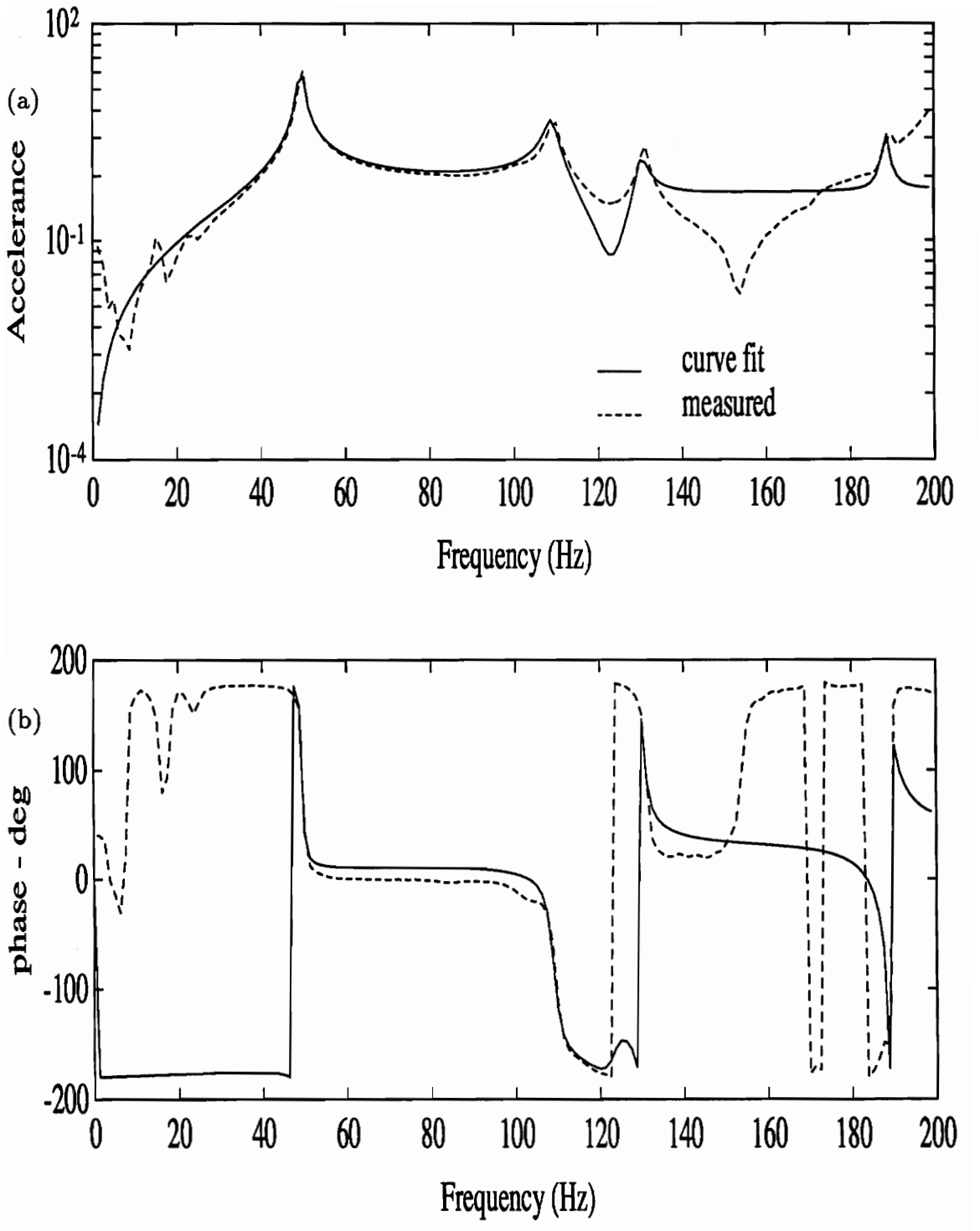


Figure 3.2: Frequency Response Function for Accelerometer 8: a) Magnitude, b) Phase

Table 3.1: Natural Frequency Comparison

Mode		Analytical	FEM	Experimental	
n	type	ω_n Hz	ω_n Hz	ω_n Hz	ζ_n % crit.
1	(1,1)	47.9	46.4	49.5	0.8
2	(2,1)	106.9	104.1	108.9	1.2
3	(1,2)	132.8	129.6	130.3	0.8
4	(2,2)	191.8	187.3	188.5	0.3

damped structure assumption. We also realize that we no longer have an ideal situation due to the mass loading effects, and the closed-form solution is not an adequate description of the plate. The variance between the finite-element and the experimental model at low frequencies, i.e., the first two modes, is expected since we realize that the real boundary conditions are probably a bit stiffer than the ideal simply-supported boundary conditions. The ideal mode shapes are given in figure 3.3. The mode shapes are used to transform the problem from the Newtonian domain to the modal domain, and in the next section we will discuss three techniques for performing this transformation. Table 3.2 contains the first four eigenvectors at the accelerometers extracted by the finite-element model and table 3.3 contains the same information for the experimental model. The nodal line that is parallel to the long direction of the plate for modes three and four is pulled up towards the actuators due to the asymmetric mass loading. Further, we realize that the excitation for the burst-random test is very close to this nodal line, and therefore does not excite these modes as well as it does modes one and two. Hence the extracted eigenvectors for modes three and four could be improved by conducting a second modal test at a different excitation location. From the closed-loop perspective, the location of the actuators was chosen to limit the authority to the first two modes; therefore, being near the nodal line of modes three and four is attractive. Finally, figure 3.1 shows that we have fit the poles of the FRF with little error, however there is significant error in the model where it is trying to fit the zeros. This signifies that future work can include better modeling approaches or robust controller design. When we applied the LQG controller, we found that the fit of the zeros is just as important as the fit of the poles since the misfit of the zeros is a modeling error. Modeling errors can cause the closed-loop system to go unstable.

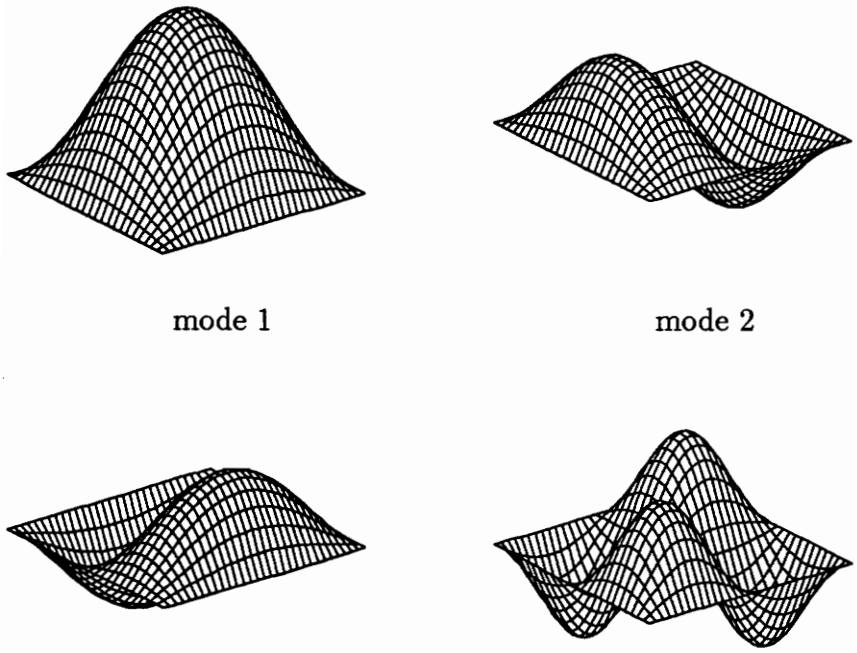


Figure 3.3: Ideal Mode Shapes of the Simply-Supported Plate

Table 3.2: NASTRAN Generated Eigenvectors

accelerometer location	Mode			
	1	2	3	4
1	0.3165	0.5043	0.4722	0.7546
2	0.4690	0.7470	0.0962	0.1551
3	0.3152	0.5022	-0.4736	-0.7553
4	0.4998	0.2473	0.7468	0.3696
5	0.7406	0.3670	0.1523	0.0752
6	0.4976	0.2470	-0.7486	-0.3721
7	0.4999	-0.2482	0.7472	-0.3701
8	0.7407	-0.3667	0.1517	-0.0757
9	0.4976	-0.2457	-0.7494	0.3693
10	0.3166	-0.5046	0.4718	-0.7514
11	0.4691	-0.7467	0.0946	-0.1516
12	0.3152	-0.5014	-0.4753	0.7556

Table 3.3: Experimentally Extracted Eigenvectors

accelerometer location	Mode			
	1	2	3	4
1	0.2068	0.5350	0.4716	0.6245
2	0.4290	0.7181	0.1104	0.1516
3	0.3072	0.4771	-0.5241	-0.6138
4	0.4001	0.2333	0.8233	0.1619
5	0.7033	0.3062	0.1353	0.0554
6	0.4829	0.1404	-0.9120	-0.4227
7	0.4468	-0.2215	0.7428	-0.6210
8	0.7033	-0.3664	0.1091	-0.2185
9	0.4480	-0.2853	-0.8733	0.4771
10	0.2803	-0.4538	0.5085	-1.0755
11	0.4365	-0.7018	0.0766	-0.2210
12	0.1884	-0.4729	-0.5331	1.0680

3.3 Spatial Filtering

Spatial filters, also refered to as modal filters, are one method to extract modal information from a discrete set of measurements. A second method involves the implementation of a modal observer. Meirovitch and Baruh [29] discuss the formulation and the characteristics of these observers which are subject to spillover effects. Modal filters are not subject to observation spillover effects provided enough discrete measurements are taken. Observation spillover is the inclusion of higher-order modal information in the distributed measurements that corrupt the modal estimation. The LQG controller includes a full-state observer to approximate the modal displacements and velocities from measured modal accelerations. Therefore, we chose to modally filter the measured accelerations.

The three candidate modal filters were a finite-element approach suggested by Meirovitch [29], an empirical approach developed by Zhang [30], and a least-squares approach that follows directly from the discrete formulation of the expansion theorem. Below, we will briefly discuss each method and explain the choice of the third method as the implemented filter.

3.3.1 Finite-Element Approach

The basis for this method relies on the second part of the expansion theorem which expresses the modal filter as

$$\ddot{q}_r(t) = \int_D M(P)\,\phi_r(P)\,\ddot{u}(P,t)\,dD, \tag{3.12}$$

where $\ddot{q}_r$ and $\phi_r(P)$ are the modal acceleration and the eigenvector of the r^{th} mode, respectively, $M(P)$ is the mass of the structure, and $\ddot{u}(P,t)$ is the measured acceleration. The above integral implies that the mass and the acceleration of the structure must be known at every point P in the domain D. However, we are interested in forming the modal coordinates from a discrete number of measurements on the structure.

Let's consider a two-dimensional plate structure with K sensors, none of which are located along the boundaries of the finite dimensional structure. The orthonormal coordinate basis of the plate is (x,y) where L_x and L_y are the length of the plate in the x- and y- directions, respectively. Further, there are m columns of sensors placed in the x- direction and n rows of sensors placed in the y- direction such that $n * m = K$. Therefore, the rectangular domain is divided into $(m+1) * (n+1)$ elements.

Denoting the acceleration in the $(j,l)^{th}$ element by $\ddot{u}_{j,l}(P,t)$, the modal filter expression, equation 3.12, becomes

$$\ddot{q}_r(t) = \sum_{l=1}^{n+1} \sum_{j=1}^{m+1} \int_{D_{j,l}} M(P)\, \phi_r(P)\, \ddot{u}_{j,l}(P,t)\, dD, \tag{3.13}$$

where $D_{j,l}$ is the domain of the $(j,l)^{th}$ element. For the sake of convenience, we choose the natural coordinate system (ξ,η) for the $(j,l)^{th}$ element as

$$\xi = j - \frac{x}{a}, \qquad j = 1,2,\ldots,m+1, \tag{3.14}$$

$$\eta = l - \frac{y}{b}, \qquad l = 1, 2, \ldots, n+1, \tag{3.15}$$

where

$$\begin{aligned} a &= \frac{L_x}{m+1} \\ b &= \frac{L_y}{n+1} \end{aligned} \tag{3.16}$$

are the length and width of each element. Borrowing from the finite-element ideas, we can approximate the acceleration of the $(j,l)^{th}$ element as

$$\ddot{u}_{j,l}(\xi, \eta, t) = \Psi(\xi, \eta) * w_{j,l}(t), \qquad 0 \leq \xi \leq 1, \;\; 0 \leq \eta \leq 1, \tag{3.17}$$

where

$$\Psi = \left[\begin{array}{cccc} \psi_1 & \psi_2 & \psi_3 & \psi_4 \end{array} \right] \tag{3.18}$$

$$w_{j,l}(t) = \left[\begin{array}{cccc} \ddot{u}(x_{j-1}, y_{l-1}, t) & \ddot{u}(x_j, y_{l-1}, t) & \ddot{u}(x_j, y_l, t) & \ddot{u}(x_{j-1}, y_l, t) \end{array} \right]^T, \tag{3.19}$$

are the interpolation functions and the sensor outputs, respectively. Reddy presents a set of linear interpolation functions for a four-node rectangular element [31].

Equation 3.17 separates the element acceleration into the product of a spatial and a temporal function. We substitute this form of the element acceleration into equation 3.13 and replace the global coordinates with the natural coordinate system, and

the spatial filter for the r^{th} mode is

$$\ddot{q}_r(t) = \sum_{l=1}^{n+1} \sum_{j=1}^{m+1} ab \int_0^1 \int_0^1 M(aj - a\xi, bl - b\eta)\, \phi_r(aj - a\xi, bl - b\eta)\, \Psi(\xi, \eta)\, w_{j,l}(t)\, d\xi\, d\eta. \tag{3.20}$$

The separable nature of the above expression allows the spatial integration to be evaluated off-line since the sensor output $w_{j,l}(t)$ is the only time-dependent variable. Let the spatial component of the modal filter for the r^{th} mode be

$$\begin{aligned} I_{j,l\,r} &= ab \int_0^1 \int_0^1 M(aj - a\xi, bl - b\eta)\, \phi_r(aj - a\xi, bl - b\eta)\, \Psi(\xi, \eta)\, d\xi\, d\eta \\ & \quad j = 1, 2, \ldots, m+1, \\ & \quad l = 1, 2, \ldots, n+1. \end{aligned} \tag{3.21}$$

The implementation of the finite-element modal filter is the vector multiplication of $I_{j,l\,r}$ and $w_{j,l}(t)$ where

$$\ddot{q}_r(t) = \sum_{l=1}^{n+1} \sum_{j=1}^{m+1} I_{j,l\,r}\, w_{j,l}(t). \tag{3.22}$$

Next, the calculation requirements per mode of this method will be examined.

Let's add a four-by-three array of accelerometers to the plate. This arrangement matches the actual experimental fixture. The plate is then discretized into twenty rectangular elements. For each element equation 3.21 must be evaluated and results in 20 four-by-one vectors. The spatial filter in equation 3.22 requires 80 multiplications and 79 additions per mode. Since we assume that there are no sensors on the boundaries, and that the boundaries are simply-supported, the calculation requirement per mode is reduced to 62 multiplications and 61 additions. Further, equation 3.21 needs the mass of the system to be known as a continuous function in

each element or at least at certain discrete points if the integral is evaluated numerically. The discretization of the mass into elements removes some of the physical significance of the transformation, and there are more direct techniques which are discussed below. This requirement limits the suitability of this method for a real structure.

3.3.2 Empirical Modal Filters

Zhang [32] introduced a new type of modal parameter, the reciprocal modal vector, that is defined based on the orthogonality requirement of the eigenvectors. When the eigenvectors, Φ are normalized by the mass matrix, M, the reciprocal modal vector, Ψ, is given as

$$\Psi^T M \Phi = I_p \tag{3.23}$$

where I_p is the p-dimensional identity matrix. Zhang describes methods for calculating Ψ that guarantee the orthogonality of Ψ to Φ. The calculation of Ψ requires the extraction of the eigenvalues and eigenvectors as well as the measured transfer function [32]. The motivation for implementing this new modal parameter to modal filters is the abilty to use Ψ to improve the orthogonality of the modal matrix and the empirical nature of the parameter does not add approximation error to the modal filter as the finite-element approach described in the prevous section [30].

This method does show promise as a technique to improve the quality of the modal filters, however as a first approach to the spatial filtering requirements, we chose a more direct approach that follows from the modal expansion theorem. The least-

square approach retains the information of the modal test and is the direct application of its results.

3.3.3 Least-Squared Modal Filters

The square modal matrix that is used to transform the model from the p-dimensional Newtonian domain to the modal domain has p independent column vectors that form the basis of the modal domain. Therefore, any response of the plate can be described as a linear combination of these eigenvectors. This is the first part of the expansion theorem and is given by

$$x(t) = \sum_{r=1}^{p} \phi_r q_r(t) \tag{3.24}$$

where ϕ_r is the eigenvector of the r^{th} mode and $q_r(t)$ is the r^{th} modal coordinate. When both $x(t)$ and $q(t)$ are the same dimension, then the modal filter is simply the inverse of the modal matrix such that

$$q(t) = \Phi^{-1} x(t). \tag{3.25}$$

However, as Meirovitch [29] has shown in developing the analytical modal filters, we often need to spatially oversample a mode in order to extract its modal coordinate. Therefore, the modal filter is used to map from the p-dimensional Newtonian domain to the n-dimensional modal domain where $n \leq p$. We can truncate equation 3.24 to n modes, which limits the accuracy of the model of the plate response. However, if the n-mode model of the response is sufficiently accurate, then

$$x(t) = \Phi q(t) \tag{3.26}$$

where Φ is the pxn truncated modal matrix. The modal filter is then the pseudo-inverse such that

$$q(t) = \left[\Phi^T \Phi\right]^{-1} \Phi^T x(t). \tag{3.27}$$

For the experiments discussed in this work, the 12 acceleration measurements were transformed into 4 modal accelerations through the implementation of the above modal filtering technique where the columns of Φ were the experimentally extracted eigenvectors. Through this transformation, the number of calculations per mode is only twelve multiplications and eleven additions, a significant reduction from the analytical modal filters discussed previously. For the best estimation of the plate's response, the modal model should retain as many modes as the computational requirement will allow.

3.4 Summary

In this chapter we have transformed the description of the plate's response from a Newtonian to a modal basis. The modal domain allows us to describe the response in a reduced-dimensional description. We formulated the modal model with both analytical and experimental techniques in keeping with our goal of establishing a testbed for controller implementation. The resulting modal models for a four-mode description were discussed and we found that the analytical and experimental models agreed to within 7% in natural frequencies and the mode shapes are the same, i.e., the nodal lines for the fourth mode, a (2,2) plate mode, are predicted to be in nearly the same location for each of the methods, and the finite-element model eigenvectors and the experimentally extracted eigenvectors are the same order of

magnitude for the first four modes of the plate. The experimental model more completely describes the real boundary conditions and structural damping, and we chose to use it to transform the problem to the modal domain. Finally, we looked at three techniques to accomplish the modal transformation, one analytical, and two empirical. The second experimental method is used in the discrete controller since it has fewer calculation requirements than the analytical method, and it uses the experimentally extracted eigenvectors directly which keeps the problem physically significant.

Chapter 4

Controller Design

The Linear Quadratic Gaussian (LQG) controller is used to reject the persistent disturbances exciting the flexural vibrations of the simply-supported plate. The LQG controller marries the optimal control approach of the Linear Quadratic Regulator (LQR) with the optimal stochastic estimator known as the Kalman Filter. The model-based, feedback controller minimizes a quadratic cost functional that trades off between controlled variable performance and controller capability while the optimal estimator combines a stochastic, dynamic state estimator with a static, recursive least-squares estimator. The plate is assumed to exhibit linear, time invariant dynamic behaviour where the random inputs and the measurement error are Gaussian.

So far in this work, we have built our experimental fixture and modeled its modal characteristics. In this section we will present the physics that drive our control experiments and establish a framework to attack other more complicated problems such as multi-input and multi-output controllers, and the acoustic controller. First, the continuous-time description of the LQG controller is discussed. Note that the disturbance dynamics are augmented to the state-space description of the plate and

all of these states are estimated. Next, the state-space equations are discretized to formulate the difference equations used in the digital controller. Finally, we will end this section by looking at the compensator transfer function and identify those characteristics that are attractive. We will only briefly discuss the controller physics and a more complete description of the optimal controller can be found in Kirk [33], and the stochastic estimator is discussed by Stengel [4]. Both of these references were used to develop this chapter.

4.1 Continuous-Time Description

The digital implementation of the controller clearly requires a sampled data description of the system, however the system is a linkage of three distinct analog components, the simply-supported plate, the smoothing filter, and the disturbance input. The state-space model of the plate is augmented in the continuous-time domain to reflect this physical reality. Further, the time-domain description demonstrates the significance of the state matrices in terms of physical characteristics of the system, i.e., the plate's eigen-properties. In this section we will start with equation 3.4 and develop the augmented state-space matrices for the system. Next, we will incorporate the state matrices in the LQG control law and note the necessary equations for the optimal control solution.

4.1.1 State-Space Description

The first subsystem which we model is the simply-supported plate. The response of the plate in the modal domain is described by the n simultaneous, second-order

differential equations given in equation 3.4. We solve for the modal acceleration

$$\ddot{q} = -\mathcal{K}q - \mathcal{C}\dot{q} + \Phi^T(u + w) \tag{4.1}$$

where the mass matrix, $\mathcal{M}$, is the n-dimensional identity matrix. The persistent disturbance force and the control input force are $w \epsilon R^t$ and $u \epsilon R^m$, respectively. The state-space description is

$$\begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} 0 & I_n \\ -\mathcal{K} & -\mathcal{C} \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} 0 \\ \phi^u \end{bmatrix} u + \begin{bmatrix} 0 \\ \phi^w \end{bmatrix} w \tag{4.2}$$

where ϕ^u and ϕ^w are the eigenvector elements for each of the n modes at the locations of the control input and the disturbance input forces, respectively. The output is

$$y = \begin{bmatrix} -\mathcal{K} & -\mathcal{C} \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} \phi^u \end{bmatrix} u + \begin{bmatrix} \phi^w \end{bmatrix} w + \mu \tag{4.3}$$

where y is the modal acceleration and μ is the measurement noise which is assumed to be mean-zero and Gaussian. The above two equations can be rewritten as

$$\begin{aligned} \dot{r} &= Fr + Gu + Lw \\ y &= Cr + D_u u + D_w w + \mu \end{aligned} \tag{4.4}$$

where $r \epsilon R^{2n}$ is the state of the plate in modal coordinates. We augment the state-space model of the plate, equation 4.4, with the smoothing filter and the disturbance model dynamics. We want the state-space model to represent all of the dynamics of the structure and its disturbance.

The second subsystem is the smoothing filter used to reduce the effects of the stair-

case nature of the control on the higher order modes of the plate. The smoothing filter is built using a second-order Sallen-Key design with a transfer function of

$$\frac{u(s)}{\tilde{u}(s)} = \frac{\omega_{sf}^2}{s^2 + 2\zeta_{sf}\omega_{sf}s + \omega_{sf}^2} \tag{4.5}$$

where $\tilde{u}(s)$ is the Laplace domain output of the digital-to-analog converter channel. ω_{sf} and ζ_{sf} are the smoothing filter cut-off frequency and damping ratio, respectively. The state space description of the filter is

$$\begin{aligned} \begin{bmatrix} \dot{s}_1 \\ \dot{s}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\omega_{sf}^2 & -2\zeta_{sf}\omega_{sf} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_{sf}^2 \end{bmatrix} \tilde{u} \\ u &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \end{aligned} \tag{4.6}$$

which can be rewritten as

$$\begin{aligned} \dot{s} &= F_{sf}s + G_{sf}\tilde{u} \\ u &= C_{sf}s \end{aligned} \tag{4.7}$$

where $s \in R^{2m}$ is the state of the smoothing filter. In the multi-input case, there may be a smoothing filter on each control channel.

There are other methods of disturbance rejection which are mentioned in the introduction, however this modern control technique involves building the model of the disturbance into the augmented state-space description of the structure. Further, the LQG paradigm assumes that the process noise is mean-zero and Gaussian. Often times, we can identify the noise as being harmonic or band-limited which is the case in these experiments. Therefore, we model the disturbance as the colored noise

output of a filter driven with Gaussian and mean-zero noise, η. We implement a narrowband filter model with the transfer function

$$\frac{w(s)}{\eta(s)} = \frac{s}{s^2 + 2\zeta_d\omega_d s + \omega_d^2} \tag{4.8}$$

where ω_d is the center frequency of the filter and ζ_d is the damping ratio of the filter. Once again, we want the state-space description which is

$$\begin{aligned} \begin{bmatrix} \dot{v}_1 \\ v_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\omega_d^2 & -2\zeta_d\omega_d \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \eta \\ w &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \end{aligned} \tag{4.9}$$

which we rewrite as

$$\begin{aligned} \dot{v} &= F_d s + G_d \eta \\ w &= C_d v. \end{aligned} \tag{4.10}$$

$v \epsilon R^{2t}$ is the state of the disturbance input model.

The augmented system is formed by substituting equation 4.7 and equation 4.10 into equation 4.4 to yield

$$\begin{aligned} \begin{bmatrix} \dot{r} \\ s \\ v \end{bmatrix} &= \begin{bmatrix} F & GC_{sf} & LC_d \\ 0 & F_{sf} & 0 \\ 0 & 0 & F_d \end{bmatrix} \begin{bmatrix} r \\ s \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ G_{sf} \\ 0 \end{bmatrix} \tilde{u} + \begin{bmatrix} 0 \\ 0 \\ G_d \end{bmatrix} \eta \\ y &= \begin{bmatrix} C & D_u C_{sf} & D_w C_d \end{bmatrix} \begin{bmatrix} r \\ s \\ v \end{bmatrix} + 0\tilde{u} + 0\eta + \mu \end{aligned} \tag{4.11}$$

which can be rewritten as

$$\begin{aligned} \dot{Q} &= F_a Q + G_a \tilde{u} + L_a \eta \\ y &= C_a Q + 0\tilde{u} + 0\eta + \mu \end{aligned} \tag{4.12}$$

where $Q \epsilon R^{2n+2m+2t}$ is the augmented state vector. Notice in equation 4.12 that the output is still y not y_a as we measure the plate's modal accelerations which are not states of the plate. The plate's states are modal displacement and velocity, and full-state feedback requires the implementation of a full-state observer. Also, the process noise, η, and the measurement noise, μ, are mean-zero and Gaussian. Further, the feedthrough of the control and disturbance inputs that appear in the representation in equation 4.4 no longer appears in equation 4.12 due to the dynamics of the smoothing filter on the control input and the disturbance input model. We have formulated the state-space representation of the system and we would like to design a controller to reject the persistent disturbance.

4.1.2 LQG Formulation

As stated in the introduction to this chapter, the LQG design approach combines a LQR controller with the state estimation of the Kalman-Bucy filter, a continuous time equivalent of the Kalman filter, which is the optimal stochastic observor. First, we identify the states that we want to control. The control states are

$$z = H_a Q \tag{4.13}$$

where we choose to control the modal coordinates of the plate. The quadratic cost to be minimized is

$$J = \int_{-\infty}^{\infty} (Q^T H_a^T H_a Q + \tilde{u}^T R_c \tilde{u})\, dt \tag{4.14}$$

where $H_a^T H_a$ is a positive semi-definite weighting matrix and R_c is a positive definite weighting matrix which specifies the amount of control effort expended. The optimal compensator is state feedback, with the form

$$\tilde{u} = -K_c Q \tag{4.15}$$

where $K_c \epsilon R^{mx(2n+2m+2t)}$ is a function of the solution of the Ricatti equation [33]. So far, we have identified an optimal controller that requires the measurement of all of the states, however we measure only the modal accelerations.

The stochastic, optimal estimator of $Q(t)$ in $[0, t_f]$ is based on the conditional probability function in $[0, t_f]$. The state estimate $\hat{Q}(t)$ is the mean value of the distribution. The estimator minimizes the error covariance for all t in $[0, t_f]$ by using all of the prior information and the current measurement. Given the statistics of the initial conditions, the inputs, and the uncertainties, we minimize the mean-squared value of the measurement residual with respect to a filter gain matrix. The minimization of this error makes $\hat{Q}(t)$ approach $Q(t)$. Therefore, the Kalman-Bucy filter formulates the optimal estimate of the state vector, and its implementation requires the calculation of the optimal covariance matrix, $P(t)$, and the filter gain matrix, $K_f(t)$. Note, we assume that the system has reached steady conditions due to the persistent disturbance and the time dependence of the covariance and filter gain matrices will be dropped.

The linear dynamic model was described in equation 4.12. The expected value of the intial state and covariance are

$$E[Q(0)] = \hat{Q}(0) \tag{4.16}$$

$$E[(Q(0) - \hat{Q}(0))(Q(0) - \hat{Q}(0))^T] = P_o. \tag{4.17}$$

The disturbance is white, zero-mean Gaussian random process

$$E[\eta(t)] = 0 \tag{4.18}$$

$$E[\eta(t)\eta^T(\tau)] = Q_{fc}\delta(t-\tau) \tag{4.19}$$

where Q_{fc} is the spectral density matrix and $\delta(t)$ is the Dirac delta function. The measurement error is also a white, zero-mean Gaussian process

$$E[\mu(t)] = 0 \tag{4.20}$$

$$E[\mu(t)\mu^T(\tau)] = R_{fc}\delta(t-\tau) \tag{4.21}$$

where the measurement uncertainty is given by its spectral density matrix. R_{fc}. We assume that the disturbance input and the measurement noise are uncorrelated.

We combine the dynamic information given in equation 4.12 with the recursive least-squares estimator which makes use of the most current measurement to update the estimate, and the Kalman-Bucy filter is

$$\dot{\hat{Q}} = F_a\hat{Q} + G_a\tilde{u} + K_f(y - \hat{y})$$

$$\hat{y} = C_a \hat{Q} \tag{4.22}$$

with

$$K_f = P C_a^T R_{fc}^{-1} \tag{4.23}$$

where K_f is a function of the solution to the covariance equation which is a dual to the Ricatti equation solution used to find the optimal control gains [4]. We rewrite the optimal control in equation 4.15 as

$$\tilde{u} = -K_c \hat{Q} \tag{4.24}$$

where K_c and K_f can be computed independently since the separation principle applies.

The LQG controller design for a discrete-time system essentially follows the development presented above. The form of the difference equations will be presented in the next section.

4.2 Discrete-Time System and Controller

The sampled-data description is needed to implement the digital controller. We assume that the control input is zero-order-held during the sample time, T, while the disturbance input may be acting continuously during the interval. The sampled-data representation of the system is

$$Q_{k+1} = AQ_k + B\tilde{u}_k + \eta_k$$

$$y_k = C_a Q_k + \mu_k \tag{4.25}$$

with

$$A = e^{F_a T} \quad B = \int_0^T e^{F_a v} dv \tag{4.26}$$

and η_k is the discrete sequence of numbers that represent the continuous process noise, $\eta(t)$. The quadratic cost is similar to equation 4.14 except the integral is replaced by a summation of all the samples. The optimal control to minimize the cost function is state feedback

$$\tilde{u}_k = -K_c Q_k \tag{4.27}$$

where once again we need to estimate all of the states.

The discrete-time optimal estimator is the Kalman filter which minimizes the mean-squared value of the measurement residual in order to make the estimated state, $\hat{Q}_k$, approach the actual state. We follow the same procedure as in the last section, however we must account for the disturbance dynamics that occur during the sample interval in some statistical sense, and we also must distinguish between the estimate that occurs before the measurement update, $\hat{Q}_{k,k-1}$ and the estimate that includes the information from the current measurement, $\hat{Q}_{k,k}$.

The expected value of the state and its covariance are known,

$$E[Q_o] = \hat{Q}_o \tag{4.28}$$

$$E[(Q_o - \hat{Q}_o)(Q_o - \hat{Q}_o)^T] = P_o. \tag{4.29}$$

We assume that the disturbance is a white, zero-mean, Gaussian, continuous process as given in equation 4.19. Further, we can identify the covariance of the random sequence η_k as

$$Q_{fd} = \int_0^T e^{F_a v} L_a Q_{fc} L_a^T e^{T F_a v} dv \tag{4.30}$$

and the sequence is assumed to be mean zero and Gaussian with the covariance given above. The measurement error is also a white, zero-mean Gaussian sequence that is uncorrelated with the disturbance input,

$$E[\mu_k] = 0 \tag{4.31}$$

$$E[\mu_k \mu_k^T] = R_{fd}. \tag{4.32}$$

The discrete optimal estimator is

$$\begin{aligned} \hat{Q}_{k,k-1} &= A\hat{Q}_{k-1,k-1} + B\tilde{u}_{k-1} \\ \hat{y}_k &= C_a \hat{Q}_{k,k-1} \\ \hat{Q}_{k,k} &= \hat{Q}_{k,k-1} + K_{fd}[y_k - \hat{y}_k] \end{aligned} \tag{4.33}$$

where K_{fd} is computed using the sampled-data equivalents of the matrices in equation 4.23. The optimal compensator is the feedback of the estimated states

$$\tilde{u}_k = -K_c \hat{Q}_{k,k-1} \tag{4.34}$$

Equation 4.33 and equation 4.34 are the difference equations needed to implement the digital LQG controller. In the next section we will discuss the characteristics of the compensator as well as the optimal gain calculation.

4.3 Discrete Compensator Analysis

In this section we will discuss some of the characteristics of the discrete LQG compensator. The discussion is used to understand how the LQG controller rejects the persistent disturbances. First, we need to identify the transfer function of the compensator. Therefore we want to write the input into the plant, $\tilde{u}_k$, as a function of the output of the plant, y_k. We can rewrite equation 4.33 as

$$\hat{Q}_{k+1,k} = [A - AK_{fd}C_a]\,\hat{Q}_{k,k-1} + B\tilde{u}_k + AK_{fd}y_k. \tag{4.35}$$

Next, we substitute equation 4.34 into the above equation,

$$\hat{Q}_{k+1,k} = [A - AK_{fd}C_a - BK_c]\,\hat{Q}_{k,k-1} + AK_{fd}y_k \tag{4.36}$$

and we take the z-transform of the above difference equation and use the left shifting theorem [34]

$$\mathcal{Z}(\hat{Q}_{k+1,k}) = z\mathcal{Z}(\hat{Q}_{k,k-1}) = [A - AK_{fd}C_a - BK_c]\,\mathcal{Z}(\hat{Q}_{k,k-1}) + AK_{fd}\mathcal{Z}(y_k). \tag{4.37}$$

We solve for $\mathcal{Z}(\hat{Q}_{k,k-1})$, $\hat{Q}(z)$, in terms of $\mathcal{Z}(y_k)$, $y(z)$, and find

$$\hat{Q}(z) = [zI - [A - AK_{fd}C_a - BK_c]]^{-1}\,AK_{fd}y(z) \tag{4.38}$$

where I is the identity matrix. Again we use the state feedback relationship in equation 4.34, the compensator transfer function is

$$\tilde{u}(z) = -K_c \left[zI - [A - AK_{fd}C_a - BK_c]\right]^{-1} AK_{fd}y(z). \tag{4.39}$$

The output of the system is driven by the persistent disturbance at the frequency of the disturbance. The compensator in equation 4.39 is fed back and appears in the denominator of the closed-loop transfer function of the system. Therefore, we expect the compensator to have large gain at the disturbance frequency, while remaining relatively flat ouside of the disturbance bandwidth.

We realize that the controller can only optimally reject as many modes as there are control actuators. For our system, we have a single control actuator and chose to optimally reject mode 1 since it is a more efficient radiator at low frequencies than mode 2. In order to calculate the state feedback gains that yield the optimal rejection, we borrow from the disturbance accomodation method formulated by Johnson [12]. Recall that Johnson breaks the control into two parts. The first part is the feedback of the plant states, r in equation 4.4, to regulate the behaviour of the undisturbed plant while the second part is the feedback of the disturbance states, v in equation 4.10, to reject its effects. The Johnson method can be represented as

$$\tilde{u}_k = -\begin{bmatrix} K_r & K_s \end{bmatrix} \begin{bmatrix} \hat{r}_{k,k-1} \\ \hat{s}_{k,k-1} \end{bmatrix} - K_v \hat{v}_{k,k-1} \tag{4.40}$$

where K_r, K_s, and K_v are the feedback gains onthe estimated plate, smoothing filter, and disturbance states, respectively. We want to find the K_v that will optimally reject the effects of a 60 Hz excitation in the first modal response. Further, we want

to assume that the continuous-time feedback gains on the estimated disturbance staes nearly approximate their discrete-time equivalents. The necessity for this assumption will be apparent below.

K_r and K_s result from the LQR solution which provides the optimal regulation of the undisturbed plate. We rewrite equation 4.40 in the continuous-time domain

$$u = -[K_r r + K_v v] \tag{4.41}$$

where the smoothing filter is being ignored for the moment. For a single-input-single-output system, when we substitute equation 4.41 into eqaution 4.4 we find that the disturbance is exactly cancelled when

$$\frac{u}{w} = -\frac{\phi^w}{\phi^u}. \tag{4.42}$$

The exact cancellation is simply the ratio of the eigenvectors at the disturbance input and the control input locations. If we had tried to to analyze the exact cancellation in the discrete-time domain, we would have lost the physical significance of the above expression. Therefore, any mode of the plate can be optimally rejected as long as we know the ratio of the eigenvectors for that mode. Note, equation 4.42 does not include the effects of the smoothing filter which adds both magnitude and phase to the controller, and while u is the force applied to the plate, we calculate $\tilde{u}$. For our application, the smoothing filter only adds a deterministic amount of phase to the system. By feeding back both of the states of the disturbance model we generate a vector that is 180° out-of-phase with the disturbance. The gains on each of the states is dependent upon the center-frequency of the disturbance, the

cut-off frequency of the smoothing filter, and accuracy of the eigenvectors. For example, the extracted eigenvectors in table 3.3 predict that a 1:1 magnitude ratio between the control and the disturbance input would yield an exact cancellation of the first mode. However, we found that the ratio needed to be 1.2:1 when we manually cancelled the disturbance signal by sending the gained, inverse of the signal to the controller. We adjusted the gain until the first modal output was reduced to a minimum.

In this section we have determined a transfer function for the compensator which can be used to evaluate the effectiveness of candidate controllers based on their gain both inside and outside the disturbance bandwidth. We want a controller that has high gain inside of the bandwidth and small gain outside of the bandwidth. Further, we discussed a method to optimally reject a single mode of the plate with a single actuator.

In this chapter we have generated the state-space description of the simply-supported plate and augmented it with the dynamics of the smoothing filter and the model of the disturbance. We formulated the LQG controller for the simply-supported plate that will reject narrowband persistant disturbances. Finally, we generated the necessary difference equations to implement the controller in both simulation and on the real structure. In the next chapter we discuss the results of this work.

Chapter 5

Simulation and Experimental Results

In this section we apply the discrete LQG controller to reject persistent disturbances of the simply-supported plate. The establishment of the plate as a testbed for controller verification is not complete without an initial controller application. Specifically, we want to be able to apply state-space controllers. First, we implemented a two-mode controller to reject a 60 Hz sinusoidal disturbance. Next, the controller was used to compensate for a narrowband input, centered at 60 Hz. Finally, the inherent transient suppression provided by a feedback controller was shown through an impact test.

We placed the actuators near the nodal line for modes three and four to limit their authority to the first two modes. Although, the LQG framework is well suited for multi-input, multi-output (MIMO) systems with large system matrices, we wanted to keep the order of the controller low to improve our physical understanding of the control process. Further, we wanted to avoid temporal aliasing and our empirical work had shown that good modal identification, i.e., a smooth, sinusoidal sampled signal when the plate was driven by a sinusoidal input, requires sample frequencies

about ten times faster than the plate motion of interest. The 60 Hz frequency is between the first and second natural frequencies, and from figure 3.1, we note that the curve fit of the measured FRF is closer at 60 Hz as opposed to 75 Hz. We chose to place the excitation frequency closer to the fundamental mode, a volumetric mode, since it is a more efficient radiator than the second mode. The vibration control of the plate can be heard since the response is dominated by the first modal response.

In this chapter, we will first discuss the simulation and implementation of the control law. Next, we will present the simulation and experimental results for the three LQG applications. First, we will demonstrate the effectiveness of the controller to reject a sinusoidal disturbance. Next narrowband disturbance compensation will be shown. Finally, the transient suppression of the controller is shown in the time histories of a open-loop impact test and a closed-loop impact test.

5.1 Simulation and Implementation of the Contoller

This section briefly discusses the controller simulation code in Appendix B and the discrete controller software which can be found in Appendix C. Before we discuss the codes, let's review the compensator model parameters. The plate is modeled with the experimental modal model which has $2n$ states where n is the number of modes in the model. A second-order narrowband filter, equation 4.8, models the disturbance input. The center frequency of the filter is 60 Hz while the damping ratio is 1% critical for a sinusoidal input, and 10% critical for a narrowband. The disturbance model contributes $2t$ states to the augmented system where t is the number

of inputs. A second-order low pass filter, equation 4.5, models the smoothing filter. The cut-off frequency is set at 120 Hz and the damping ratio is 70.7% critical, and this model contributes $2m$ states where m is the number of control inputs. For the simulations and experiments in this chapter a two-mode controller is implemented with one disturbance force driving the plate, and a single control input. Therefore, an eighth-order compensator is implemented. Since the only measurements are accelerations, a full-order Kalman filter is implemnted in real time to estimate all of the eight states. Next, we will discuss the simulation of the system, the plate and the estimator, as well as the implementation of the digital controller.

The simulation data were created using MATLAB software on a 386 computer. The simulation code is used to obtain a real-time representation of both the plant and the estimator's response in the open- and closed-loop. The plant is simulated by forming the discrete version of the plate dynamics, using the experimental sample frequency, augmented with the smoothing filter dynamics. The control input is assumed to be zero-order-held, while the disturbance input is first-order-held. The disturbance input is modeled as a first-order-hold since it acts continuously during the sampling interval. Therefore, the plant simulation is driven by the disturbance time sequence that is the output of the disturbance model in equation 4.10. Equation 4.33 is calculated to simulate the estimator. The loop is closed with equation 4.34. MATLAB routines are used to calculate the feedback and the estimator gains, and the system matrices are stored for use in the digital controller.

The discrete controller uses the TransDAC system where the control is calculated on one transputer while the data acquisition is performed by a second transputer.

Each transputer is downloaded its own set of C language instruction that were written by Ellis. The first program, filthost.c, contains the instructions that implent the discrete LQG controller. The control loop begins with the estimator prediction equation, the first expression in equation 4.33. Next, the control is calculated and sent to the data acquisition transputer which in turn sends it through the smoothing filter to the plate. The estimated modal acceleration outputs are calculated just before the acceleration measurements are read in and the modal accelerations are calculated in the control code. The calculation of the estimated output serves as a timing device to insure that the measured output is correlated to the control without a time delay. Next, we use the modal accelerations to update the state estimate and the control loop begins again. The second program, filt_t2.c, samples the twelve accelerometer channels simultaneously and performs the digital-to-analog conversion of the controller output with a zero-order-hold. The system is capable of sampling 16 channels simultaneously while executing 16 DAC operations simultaneously. The parallel nature of the architecture allows the control code and the data acquisition code to run independently. For the two-mode controller, the TransDAC samples at 2085 Hz, but with software and hardware improvements this system has no temporal bound, i.e., there is no maximum sample frequency for this system.

In the following sections, the simulation and experimental results from the three types of configurations will be presented.

5.2 Harmonic Disturbance Compensation

Simulations of the LQG compensator were performed using the identical controller structure as that implemented in the experiment, i.e., an eighth-order controller as discussed previously. For the simulation, the plant model was truncated to four modes. This provided a two-mode controller operating on a four-mode plant with optimal disturbance compensation targeted for the first mode only. An additional control actuator would have allowed optimal rejection of both modes.

Simulated time histories of the first two plate modes are shown in figure 5.1. The estimator and feedback gains are displayed in table 5.1. The figures show the open-loop steady-state response to a 5 N peak disturbance followed by the closed-loop response. The controller was turned on at 0.19 secs. The controller reduced the first-mode response by more than an order of magnitude, and the amplitude of mode two was driven slightly higher because of the relative phasing of the two modes at the control input location. The controller was designed to optimally reject the first mode. One controller can only optimally reject a single mode.

Experimental time histories for the first two-plate modes are illustrated in figure 5.2. The experimental time histories are nearly identical to the simulated time histories. This highlights an important issue for controller design. Since the bandwidth of the disturbance and the compensator is substantially within the bandwidth of the lower order plate modes, it is possible to design a low order controller without any degradation in performance. This is an important consideration given the amount of computational performance required for higher order control systems.

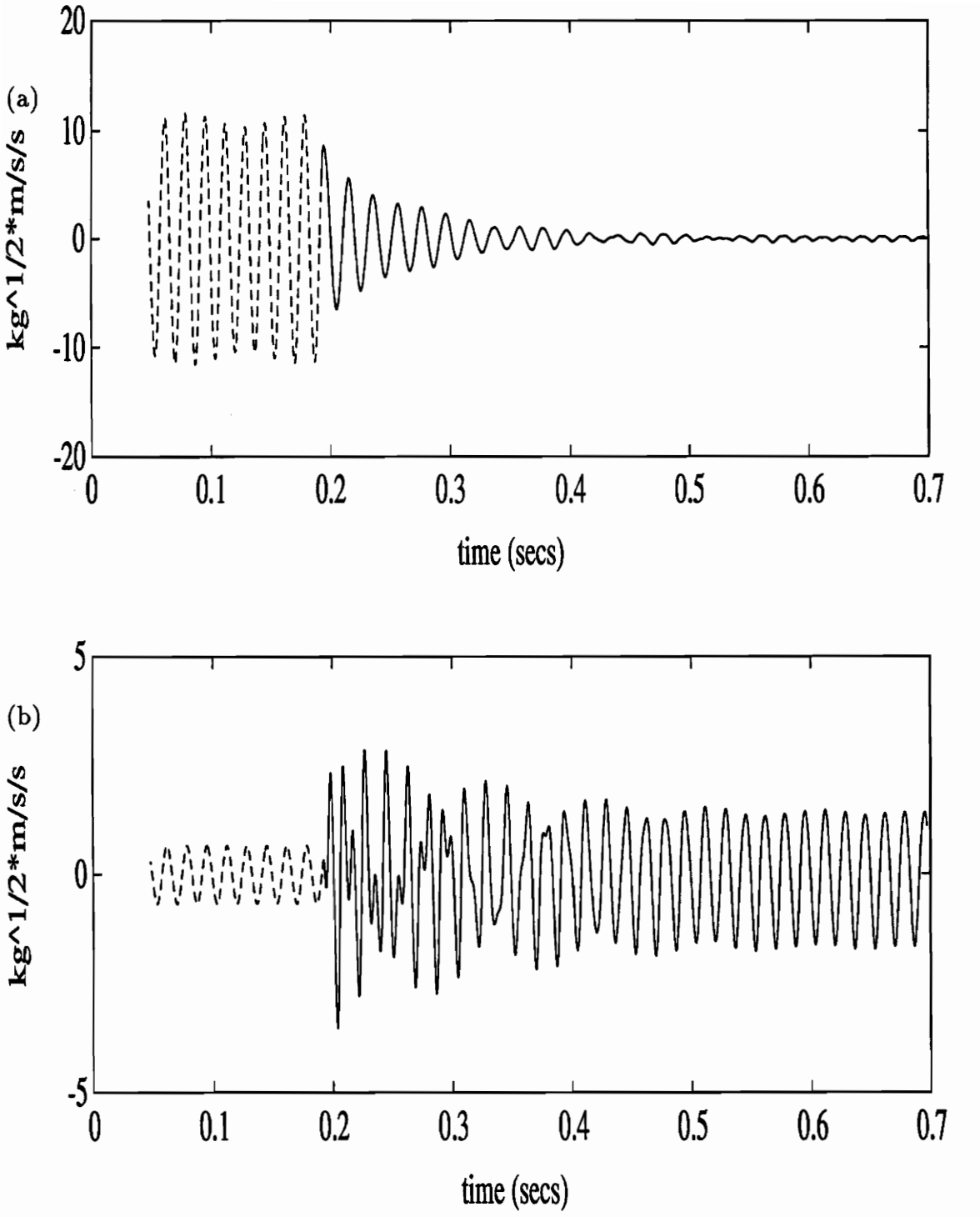


Figure 5.1: Simulated Harmonic Disturbance Rejection: a) Mode 1, b) Mode 2. The dotted line is the open-loop response and the solid line is the closed-loop response.

Table 5.1: The Estimator and Feedback Gains for Harmonic Disturbance Rejection Optimally Tuned to Reject the Fundamental Mode

$$
\begin{aligned}
R_c &= 5.0000e-04 \\
R_{fd} &= 25 \\
Q_{fc} &= 500 \\
K_c &= \begin{bmatrix} -7.2453e+03 \\ 1.0313e+04 \\ 2.8210e+01 \\ -1.6924e-01 \\ 4.6005e-02 \\ 4.4272e-05 \\ -3.5000e+02 \\ 8.0000e-01 \end{bmatrix}^T \\
K_{fd} &= \begin{bmatrix} -1.0651e-06 & -7.3961e-08 \\ 2.6749e-08 & -7.5769e-08 \\ 2.5351e-06 & -4.0545e-05 \\ 3.8290e-05 & 1.7172e-06 \\ 1.1773e-22 & -3.4462e-23 \\ 2.0195e-20 & -9.6003e-21 \\ -1.8284e-04 & -3.6860e-05 \\ 5.1900e-02 & -6.5285e-03 \end{bmatrix}
\end{aligned}
$$

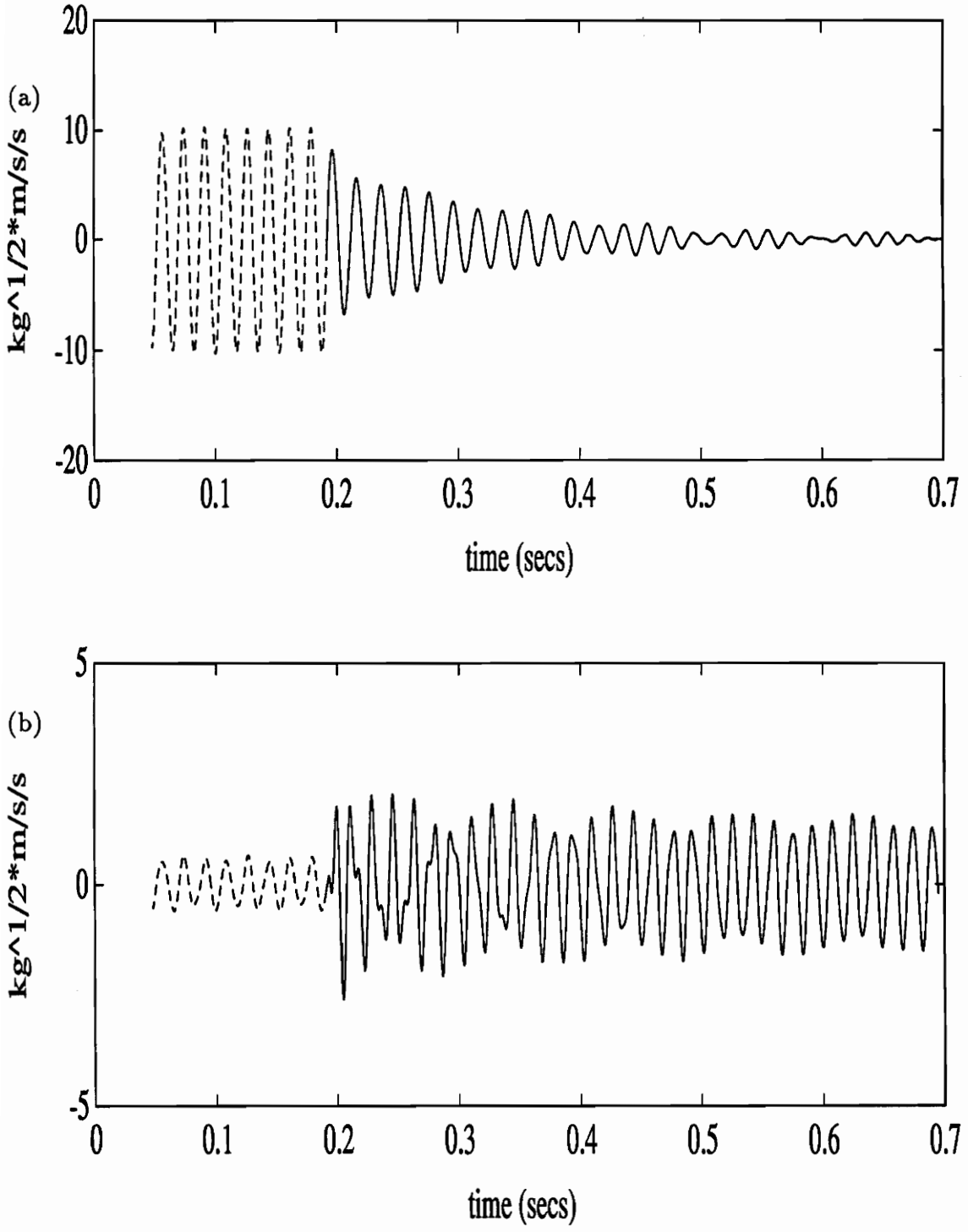


Figure 5.2: Experimental Harmonic Disturbance Rejection: a) Mode 1, b) Mode 2. The dotted line is the open-loop response and the solid line is the closed-loop response.

The disturbance model can be modified to emulate a narrowband disturbance input which is a group of closely spaced harmonics. We next modify the controller to reject the narrowband disturbance input.

5.3 Narrowband Disturbance Compensation

The narrowband signal was generated by a 286 AT equiped with a transputer system. The center frequency of the signal was 60 Hz with 10% critical damping. The measured disturbance spectrum is shown in figure 5.3. The simulated time history for the first two modes is shown in figure 5.4, and the experimental time history is shown in figure 5.5. The estimator and feedback gains are displayed in table 5.2. Once again, the controller was turned on at 0.19 seconds and we notice that the first mode is rejected while the second mode is driven slightly higher. The simulated results predicted the experimental behavior well. It is difficult to see the controller performance in the time domain. A narroband disturbance is a modulated sine wave. The signal is the filtered output of a Gaussian, random sequence which caused the narrowband disturbance to have a varying amplitude. If the control is turned on when the disturbance input has a low amplitude, the control appears to be ineffective.

Figure 5.6 and figure 5.7 are the autospectra of the first two modal accelerations in the open- and closed-loop. Notice how the first modal acceleration is rejected by 15 dB power at the disturbance center frequency. We also see that the second modal acceleration is driven higher at the disturbance center frequency. The disturbance

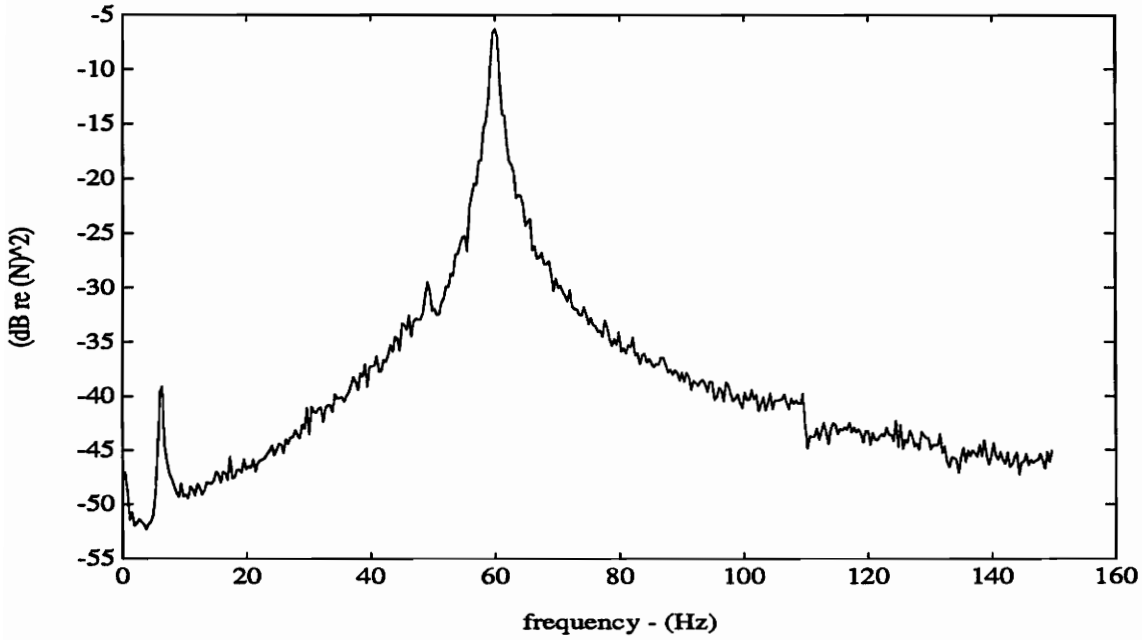


Figure 5.3: Measured Disturbance Spectrum

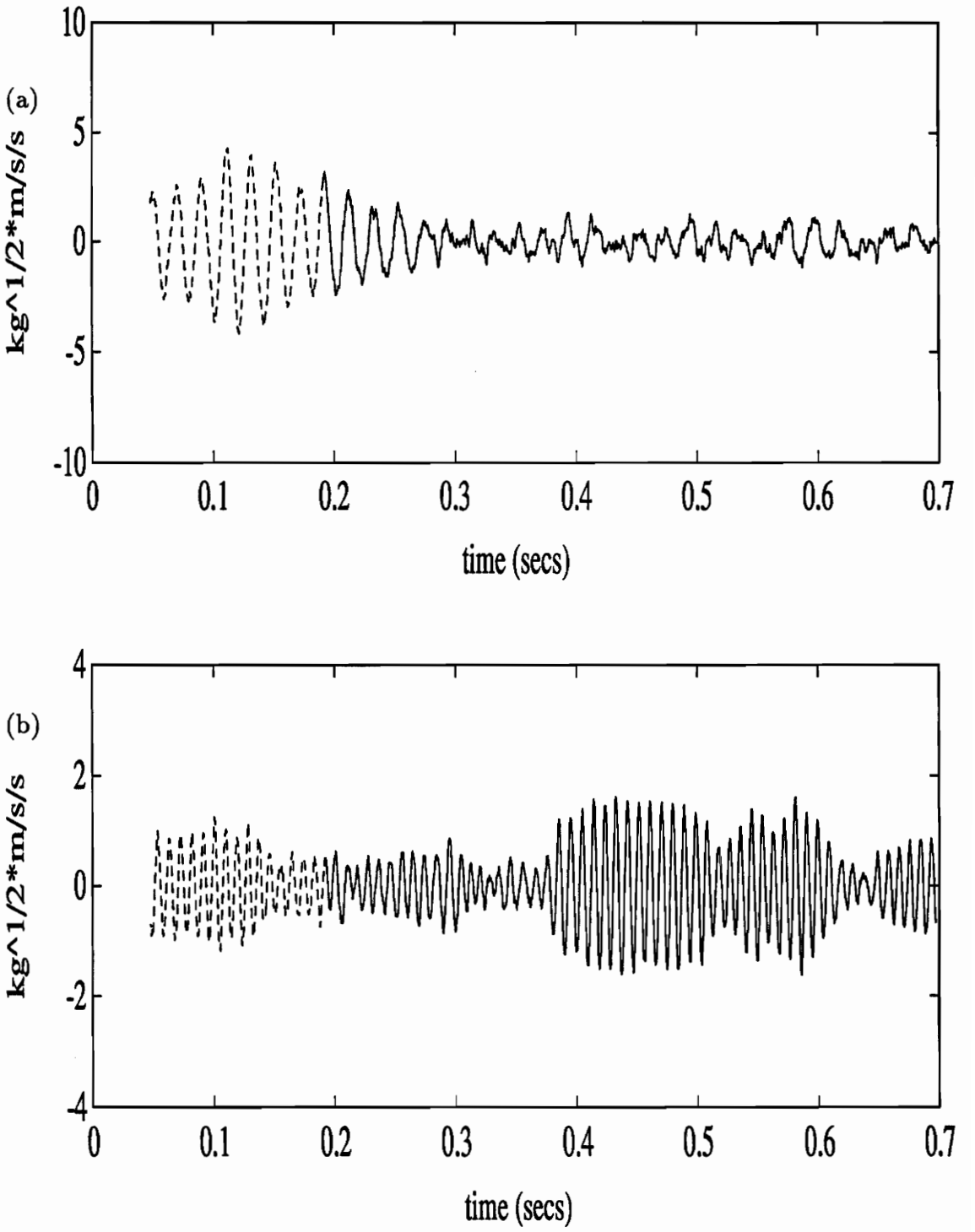


Figure 5.4: Simulated Narrowband Disturbance Rejection: a) Mode 1, b) Mode 2. The dotted line is the open-loop response and the solid line is the closed-loop response.

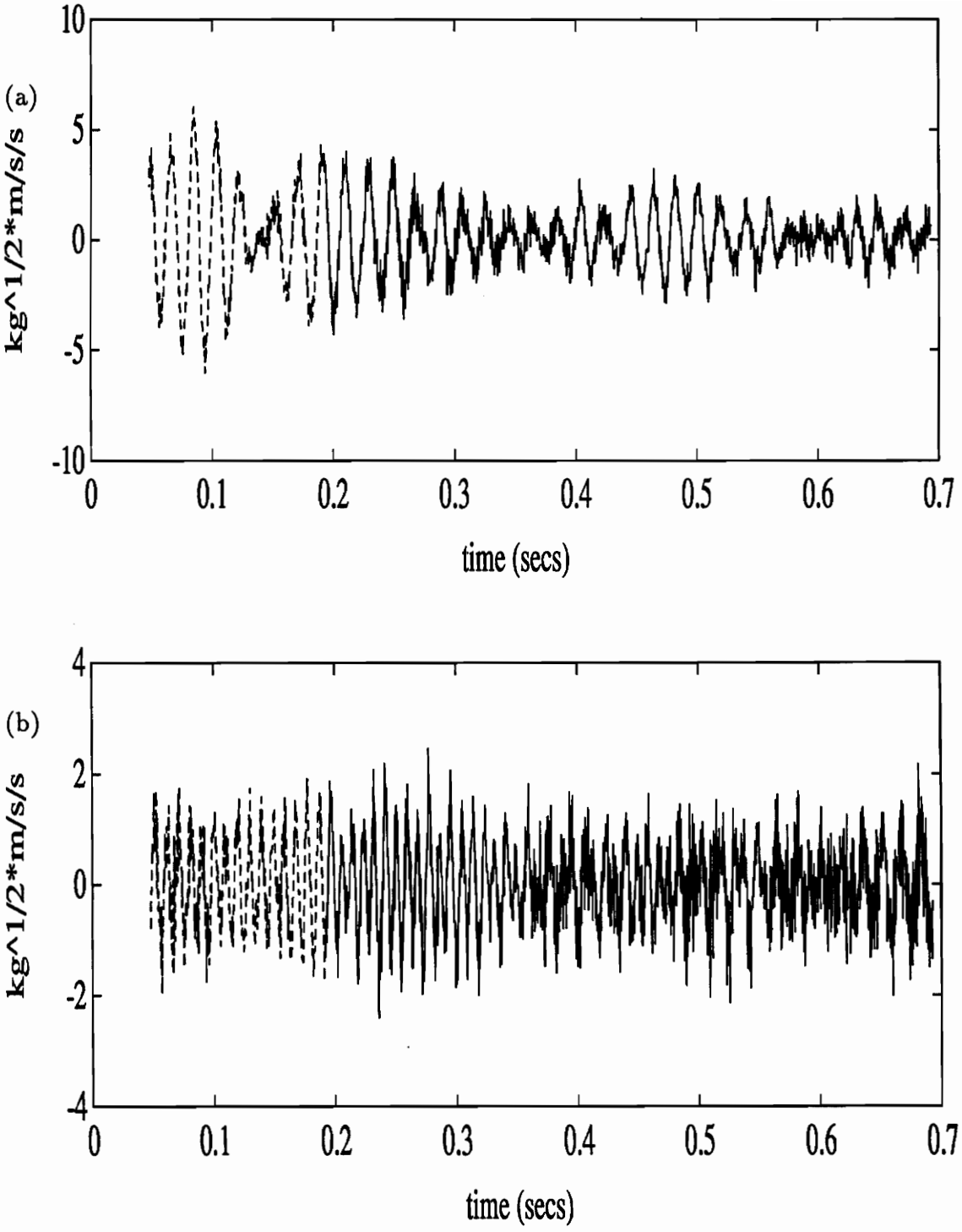


Figure 5.5: Experimental Narrowband Disturbance Rejection: a) Mode 1, b) Mode 2. The dotted line is the open-loop response and the solid line is the closed-loop response.

Table 5.2: The Estimator and Feedback Gains for Narrowband Disturbance Rejection Optimally Tuned to Reject the Fundamental Mode

$$
\begin{aligned}
R_c &= 5.0000e-04 \\
R_{fd} &= 25 \\
Q_{fc} &= 500 \\
K_c &= \begin{bmatrix} -7.2453e+03 \\ 1.0313e+04 \\ 2.8210e+01 \\ -1.6924e-01 \\ 4.6005e-02 \\ 4.4272e-05 \\ -3.5000e+02 \\ 8.0000e-01 \end{bmatrix}^T \\
K_{fd} &= \begin{bmatrix} -8.9521e-07 & -4.0323e-08 \\ 2.2789e-08 & -7.1731e-08 \\ 1.5123e-06 & -3.0317e-05 \\ 2.9004e-05 & 9.8571e-07 \\ 3.9815e-23 & 5.9338e-24 \\ -3.5806e-20 & 2.3916e-20 \\ -1.2818e-04 & -2.4815e-05 \\ 4.0818e-02 & -3.8168e-03 \end{bmatrix}
\end{aligned}
$$

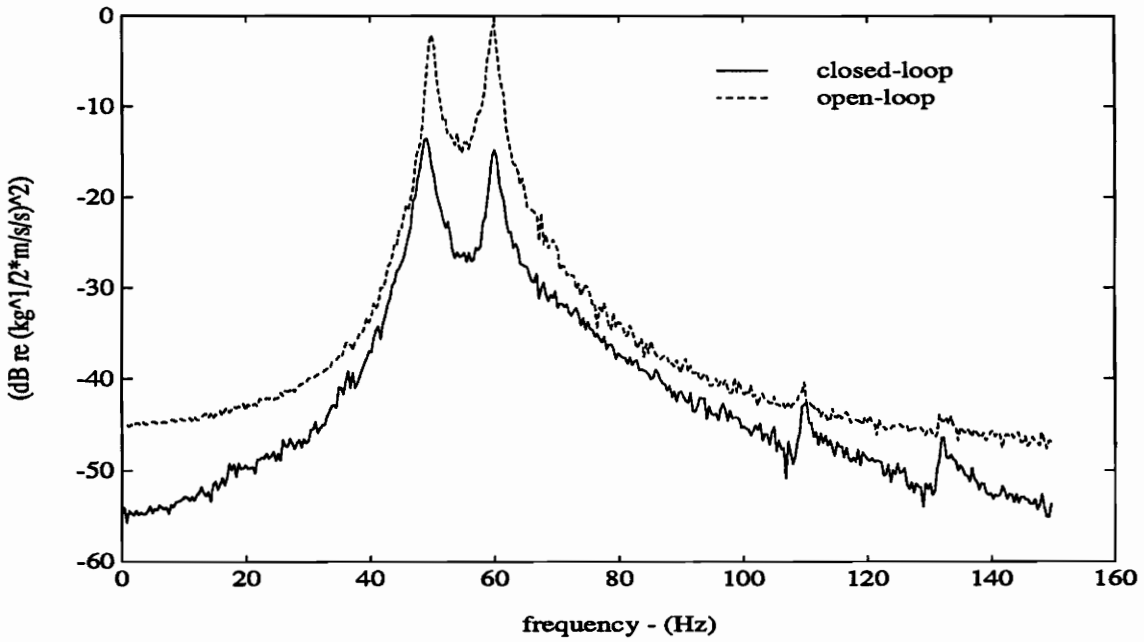


Figure 5.6: Experimental Narrowband Disturbance Rejection - Mode 1. The dotted line is the open-loop response and the solid line is the closed-loop response.

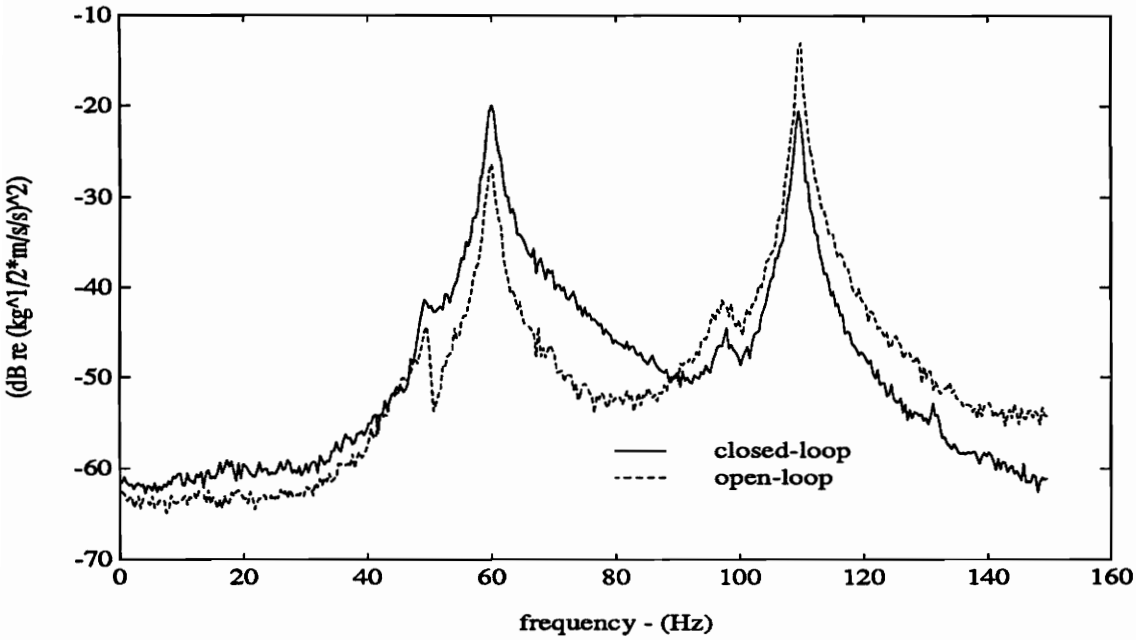


Figure 5.7: Experimental Narrowband Disturbance Rejection - Mode 2. The dotted line is the open-loop response and the solid line is the closed-loop response.

input also drives the plate at its natural frequencies since there is energy in those bins. We see from the autospectra that the response at the natural frequencies is suppressed by the controller which is indicitive of the damping added by a feedback controller. This leads us to our next section where we discuss the results from an impulse applied to the center of the plate.

5.4 Transient Suppression

Transient suppression is an inherent property of feedback controllers. An impulse was applied to the center of the plate during both the open- and closed-loop. The first modal acceleration is shown in figure 5.8. Notice that the closed-loop acceleration decays more quickly than the open-loop response. This is indicitive of the additional modal damping which increased by 100%. Since the plate was struck in the middle, along the nodal line of mode 2 we only see the first modal response.

5.5 Summary of Results

The compensator was shown to reject persistant harmonic and narrowband disturbances. Harmonic excitation of the first modal acceleration was reduced by an order of magnitude while narrowband excitation can be reduced by 15 dB power. Since there is only one control input, we chose to optimally reject the first mode. However, n modes can be rejected optimally if there are n control actuators. Finally, the transient suppression nature inherent to the feedback controller was demonstrated as the closed-loop damping of the fundamental mode is twice the open-loop damping. The additional damping introduced by the feedback controller is also seen in the

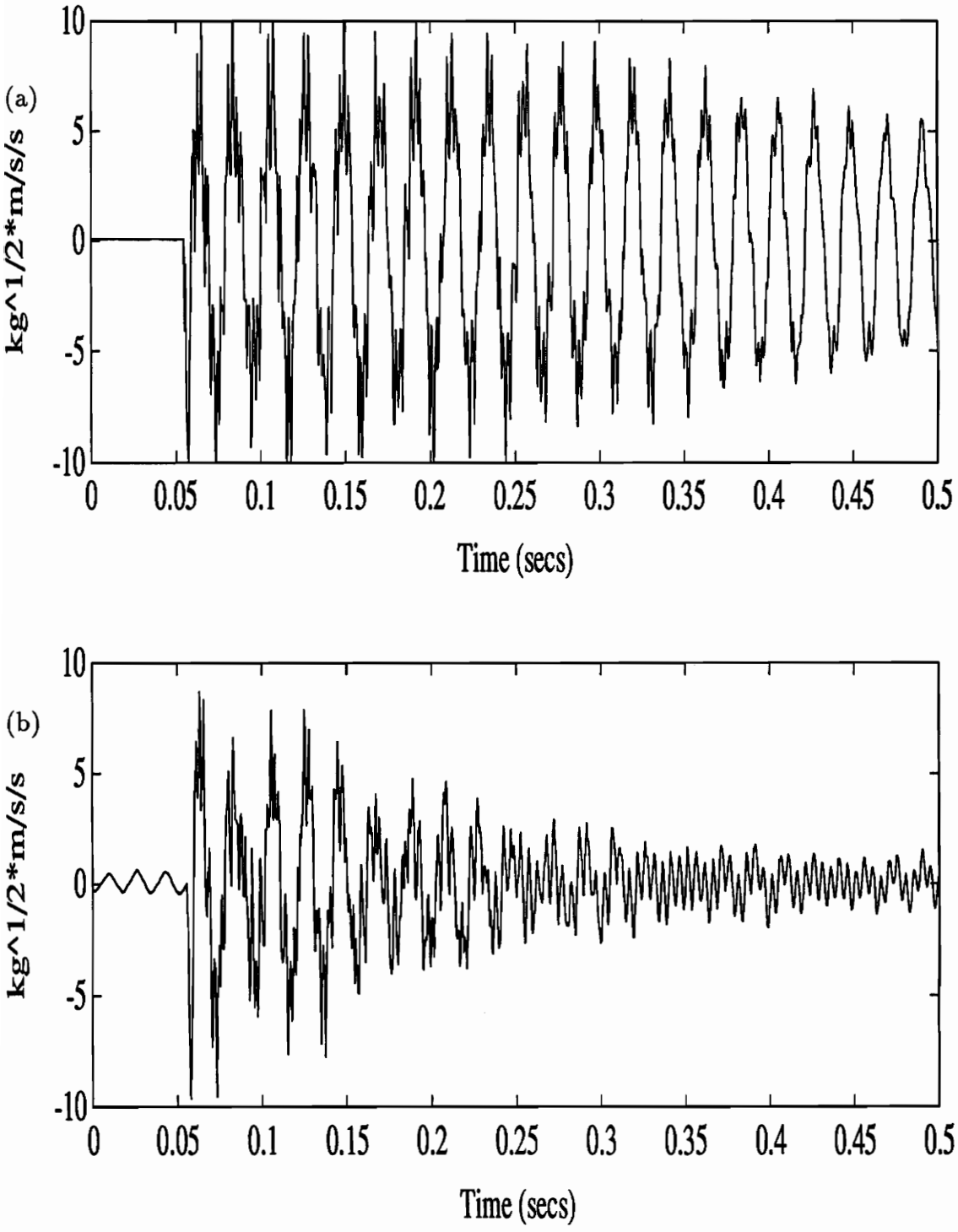


Figure 5.8: Transient Disturbance Suppression of the First Modal Acceleration: a) Mode 1, b) Mode 2. The dotted line is the open-loop response and the solid line is the closed-loop response.

autospectra of figure 5.6 and figure 5.7 where the response at the natural frequencies is attenuated in the closed-loop.

Chapter 6

Conclusions

The stated objectives of this research were to construct a testbed for state-space controller applications and implement a low-order LQ vibration controller to reject persistent disturbances. The testbed will be used as an evaluation platform for ASAC verification. We built a simply-supported plate and developed the instrumentation and controller hardware. We identified the need for both spatial and temporal oversampling to distinguish the modes of the plate. We modeled the plate analytically with finite-elements and experimentally verified that model. We implemented the LQG controller that is an identical design as the acoustic controller developed by Baumann [17, 18], with the only difference being the choice of a vibration cost function versus the acoustic cost function. It was shown that the two-mode controller rejected both harmonic and narrowband persistent disturbances of the simply-supported plate, and provided transient suppression. The framework used is easily extendable to MIMO controllers with both a vibration cost function and an acoustic cost function. Further, advanced modeling techniques and robust controllers, both tools available in the modern controls toolbox, can only enhance the controller performance. Finally, computational issues of implementing these higher-order compensators can be addressed in the modern control framework.

Bibliography

[1] Sievers, Lisa A. and Andreas H. von Flotow. Linear control design for active vibration isolation of narrowband disturbances. *Proceedings of the* 27^{th} *Conference on Decision and Control.* December, 1988.

[2] Sievers, Lisa A. and Andreas H. von Flotow. Comparison and extension of control methods for narrowband disturbance rejection. *Presented at the 1990 ASME Winter Annual Meetings.* November, 1990.

[3] Gupta, Narenda K. Frequency-shaped cost functionals: extension of linear-quadratic-Gaussian design methods. *Journal of Guidance and Control.* Volume 3, number 6. November-December, 1980.

[4] Stengel, Robert F. **Stochastic Optimal Control: Theory and Applications.** John Wiley & Sons. New York. 1986.

[5] Fuller, C.R., G.P. Gibbs, and R.J. Silcox. Simultaneous active control of flexural and extensional waves in beams. *Journal of Intelligent Material Systems and Structures.* 1 (2), pp. 235-247.

[6] Fuller, C.R. and L.O. Gonidou. Active vibration control of flexural power flow in beams. *Journal of the Acoustical Society of America.* 84, S47.

[7] Clark, R.L. and C.R. Fuller. Experiments on active control of structurally radiated sound using multiple piezoceramic actuators. Accepted for publication in the *Journal of the Acoustical Society of America.* 1990.

[8] Fuller, C.R., R.J. Wilcox, V.L. Metcalf, and D.E. Brown. Experiments on structural control of sound transmitted through an elastic plate. *Proceedings of the American Control Conference.* 1989. pp. 2079-2084

[9] Elliot, S.J., I.M. Strothers, and P.A. Nelson. A multiple error LMS algorithm and its appplication to the active control of sound and vibration. *IEEE Transactions on Acoustics, Speech, and Signal Processing.* Volume ASSP-35, number 10. pp.1423-1434.

[10] Meirovitch, L., H. Baruh, and H. Oz. A comparison of control techniques for large flexible structures. *Journal of Guidance and Control and Dynamics.* Volume 6, number 4. July-August, 1983. pp. 302-310.

[11] Mace, B.R. Active control of flexural vibrations. *Journal of Sound and Vibration.* Volume 114. pp. 253-270.

[12] Johnson, C.D. Accomodation of disturbances in optimal control problems. *International Journal of Control.* Volume 15, number 2. 1972. pp. 209-231.

[13] Sorrells, James E. Design of disturbance accomodating controllers for stochastic environments. *Proceedings of the American Control Conference.* 1989. pp. 2603-2608.

[14] Hall, Steven R. and Norman M. Wereley. Linear control issues in the higher harmonic control of helicopter vibrations. *Proceedings of the* 45^{th} *Annual Forum of the American Helicopter Society.* 1989. pp. 955-971.

[15] Parlos, Alexander G. and John W. Sunkel. Active disturbance rejection in large flexible space structures. *Proceedings of the American Control Conference.* 1990. pp. 2971-2978.

[16] Rao, S. Vittal, Michael S. Mattice, and Norman P. Coleman. Design of reduced order LQG/LTR controllers for turret-gun systems. *Proceedings of the American Control Conference.* 1989. pp. 312-315.

[17] Baumann, W.T., W.R. Saunders, and H.H. Robertshaw. Active suppression of acoustic radiation from impulsively excited structures. Accepted for publication in *Journal of the Acoustical Society of America.* 1990.

[18] Baumann, W.T., F.S. Ho, and H.H. Robertshaw. Active acoustical control of broadband structural disturbances. Presented at the 120^{th} *Meeting of the Acoustical Society of America.* November, 1990.

[19] Baz, A. and S. Poh. Experimental implementation of the modified independent modal space control method. *Journal of Sound and Vibration.* Volume 139. pp. 133-149.

[20] Wynn, Robert H., Jr. The control of flexible structure vibrations using a cantilevered adaptive truss. Doctor of Philosophy Dissertation. Virginia Polytechnic Institute and State University. 1990.

[21] Rubenstein, S.P., W.R. Saunders, G.K. Ellis, W.T. Baumann, and H.H. Robertshaw. Demonstration of a LQG vibration controller for a simply-supported plate. *Proceedings of the Recent Advances in Active Control of Sound and Vibration.* 1991. pp.618-630.

[22] Wallace, C.E. Radiation resistance of a rectangular panel. *Journal of the Acoustical Society of America.* Volume 51, number 3. 1972.

[23] Fahy, Frank. **Sound and Structural Vibration: Radiation, Transmission and Response.** Harcourt Brace Jovanovich. New York. 1985.

[24] Mitchell, L.D. and K.B. Nelson. How to design stingers for vibration testing of structures. *Sound and Vibration.* Volume 14, number 4. April, 1984. pp. 14-18.

[25] Anderson, I.A. Avoiding stinger rod resonance effects on small structures. *Proceedings of IMAC 1990.* pp. 673-678.

[26] Ellis, G.K. Design and programming of a medium performance data acquisition system based on transputers. Submitted for publication as a Center for Intelligent Material Systems and Structures internal report. 1990.

[27] Leissa, Arthur W. **Vibration of Plates.** NASA SP; 160. Scientific and Technical Division. Washington D. C. 1969.

[28] Ewins, D.J. **Modal Testing: Theory and Practice.** Research Studies Press Ltd. Letchworth, Hertfordshire, England. 1986.

[29] Meirovitch, L. and H. Baruh. On the problem of observation spillover in self-adjoint distributed-parameter systems. *Journal of Optimization Theory and Applications.* Volume 39, number 2. February, 1983.

[30] Zhang, Qiang, R.J. Allemang, and D.L. Brown. Modal filter: concept and applications. Structural Dynamics Research Laboratory, University of Cincinnati. 1990.

[31] Reddy, J.N. **An Introduction to the Finite-Element Method.** McGraw-Hill Book Company. New York. 1984.

[32] Zhang, Q., C.Y. Shih, and R.J. Allemang. Orthogonality criterion for experimental modal vectors. *ASME Publication DE.* Volume 18-4. pp. 251-258.

[33] Kirk, Donald E. **Optimal Control Theory, an Introduction.** Prentice-Hall Inc. Englewood Cliffs, New Jersey. 1970.

[34] Ackerman, Jurgen. **Sampled-Data Control Systems: Analysis and Synthesis, Robust System Design.** Springer-Verlag. New York. 1985.

Appendix A

Pictures of the Experimental Rig

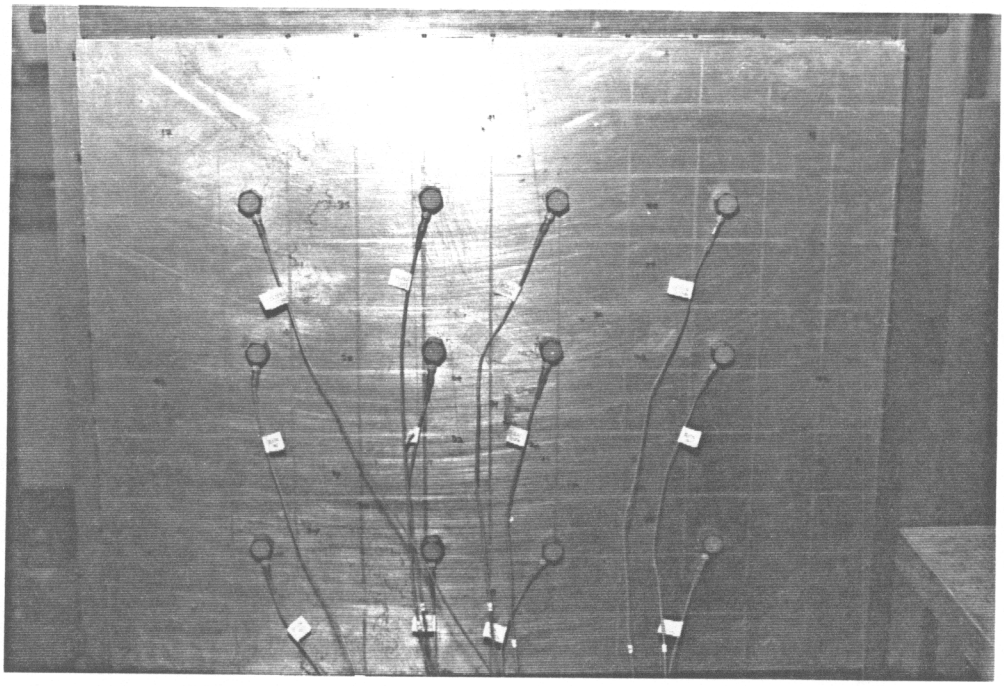


Photo by Michele Leshan

Figure A.1: Picture of the Plate Front with the Accelerometer Array

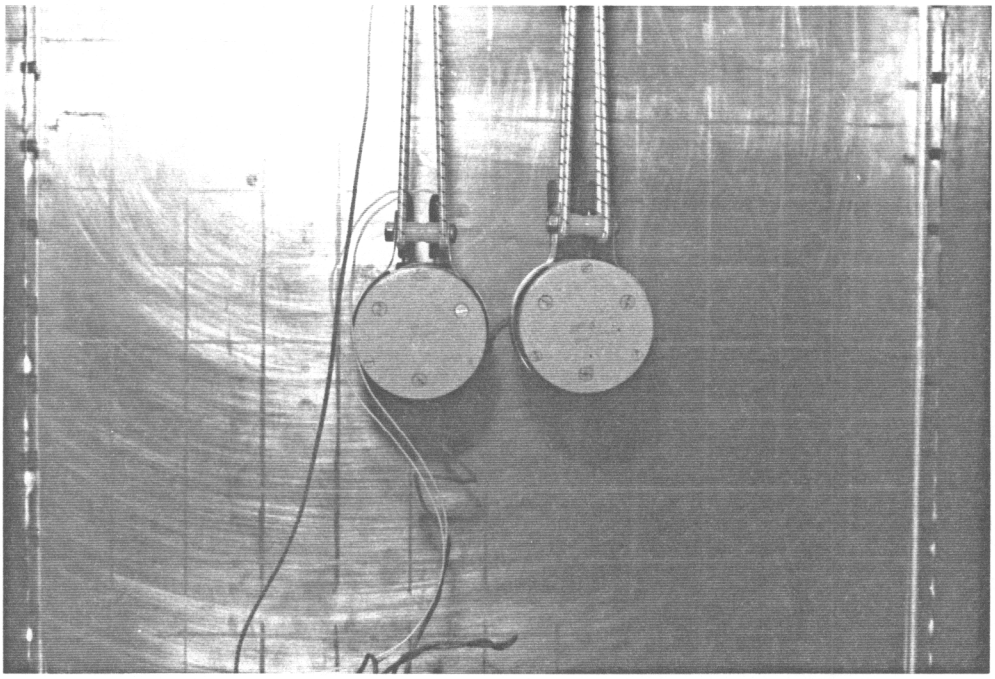


Photo by Michele Leshan

Figure A.2: Picture of the Plate Back with the Disturbance Shaker, on right, and the Control Shaker

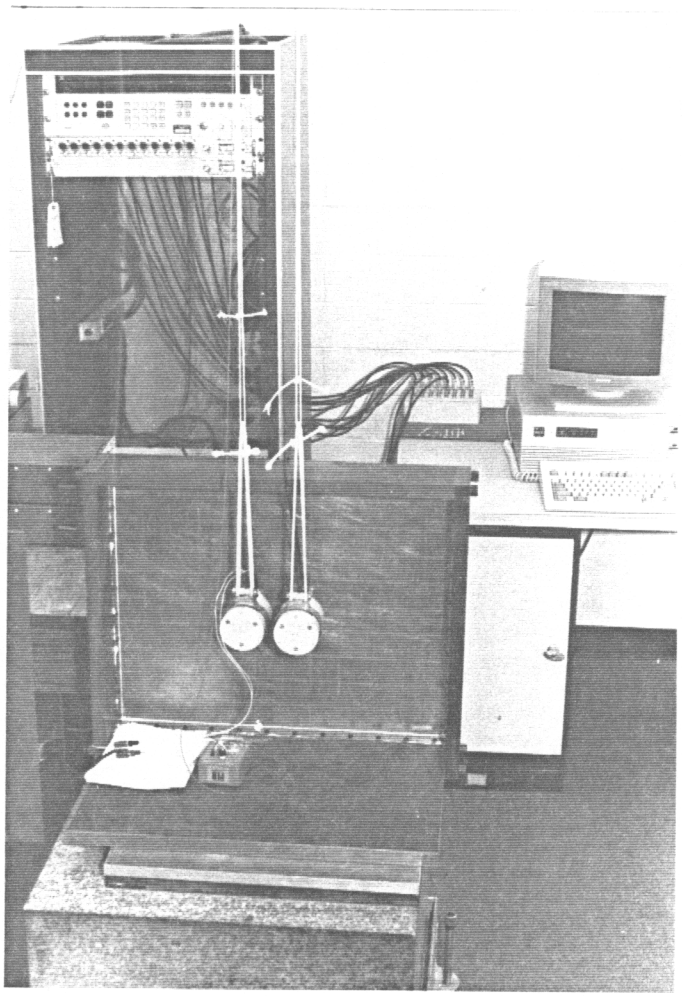


Photo by Michele Leshan

Figure A.3: Picture of the Experimental Set-up

Appendix B

Simulation Code

The following MATLAB files simulate the response of the simply-supported plate. The first file, filt_2m.m, is the real-time simulation of the closed-loop system. The second file, int_qd.m, calculates the covariance of the discrete-time disturbance sequence which represents the continuous, mean-zero, Gaussian disturbance input. The simulation code was written by William R. Saunders and Stephen P. Rubenstein.

```
%   filt_2m.m   This file is the real time simulation of the DARPA
%               plate.  The simulation is set up for the two-mode
%               controller which will be implemented in the lab.
%               Only the modal variables are considered in this
%               code, i.e., no modal filtering is performed here.
%
%               The narrowband disturbance is centered at 60 Hz
%               and acts continuously.  The Kalman Filter gains
%               are calculated based on this disturbance model.
%               Both a sinusoidal and narrowband disturbance
```

```
%           input can be simulated.  Since the disturbance
%           acts continuously, the disturbance does not meet
%           the zero-order hold assumption.  It is assumed
%           to act through a first-order hold.  The first-order
%           hold is based upon the one in LSIM.M.
%
%           This code will be used to generate the LQR controller
%           gains and the Kalman filter gains used in the
%           experiments.  The discrete state matrices and vectors
%           are also calculated here for entry into ***.h, which
%           stores most of the defines to be used by the digital
%           controller for the plate.
%
%           This file is written to emulate the closed-loop
%           experiment code, filthost.c, in which the plate
%           response is fully described by the first four modes
%           of the plate.  The estimator is designed to estimate
%           the states associated first two modes, the smoothing
%           filter states, and the disturbance states.  Only these
%           estimated states are used in the control law.
%
%           The control force, u, is fed through a smoothing
%           filter before it reaches the plate.  This is used to
%           smooth the discrete control force and remove the high
%           frequency content of the control force.
```

```
%
%               A vibration cost function is used here.  No radiation
%               dynamics are included in this code.
%
%               Note:  The size of the compensator can each be adjusted
%                      independently between one and four modes.
%
%
%                          SPR, WRS
%                          9-25-90   modified 10-26-90
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format short e
clear
clg
T=1/2850;          % sampling time
%
% generalized stiffness and generalized damping matrices from Nastran:
%
% use only two modes (n = 2)
n = 2;                  %no. of modes in the controller and estimator
np = 4;                 %no. of modes in the plate simulation
a = zeros(np,np);
b = zeros(np,np);
w1 = 2*pi*49.447;       %rad/s - MODHAN quantities
w2 = 2*pi*108.96;
```

```
w3 = 2*pi*130.25;
w4 = 2*pi*188.53;
a(1,1) = w1^2;
a(2,2) = w2^2;
a(3,3) = w3^2;
a(4,4) = w4^2;
K = a;
zeta1 = 0.7722826E-02;
zeta2 = 0.1171460E-01;
zeta3 = 0.8318498E-02;
zeta4 = 0.2731109E-02;
b(1,1) = 2 * zeta1 * w1;
b(2,2) = 2 * zeta2 * w2;
b(3,3) = 2 * zeta3 * w3;
b(4,4) = 2 * zeta4 * w4;
Damp = b;
%
% Define the F matrix for the system and the estimator:
F_p = [zeros(np,np) eye(np,np);-K(1:np,1:np) -Damp(1:np,1:np)];
F = [zeros(n,n) eye(n,n);-K(1:n,1:n) -Damp(1:n,1:n)];
%
% Define the G and L matrices for the following case:
%
%       One control force (m = 1)  applied at accelerometer
%          location 8, grid point 208 - 360 node NASTRAN model
```

```
%       One disturbance input (s = 1) applied at accelerometer
%          location 5, grid point 154
%
%   Define Phi (control force) and Gamma (disturbance):
m = 1;
s = 1;
load phi_exp;   %the experimentally extracted eigenvectors
%
p=[phi_exp(:,1) phi_exp(:,2) phi_exp(:,3) phi_exp(:,4)];
p(8,1)=p(8,1)/1.2;   %exp. correction of the eigenvector
% plate matrices
Phi_p = p(8,1:np)';
D_p = Phi_p;
Gamma_p = p(5,1:np)';
D_d_p = Gamma_p;
G_p = [zeros(np,m);Phi_p];
L_p = [zeros(np,s);Gamma_p];
% estimator matrices
Phi = p(8,1:n)';
D = Phi;
Gamma = p(5,1:n)';
D_d = Gamma;
G = [zeros(n,m);Phi]
L = [zeros(n,s);Gamma];
% The output equation coef. matrices:
```

```
C_p = [-K(1:np,1:np) -Damp(1:np,1:np)];    %plate system
C = [-K(1:n,1:n) -Damp(1:n,1:n)];          % n x 2n
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Define the disturbance spectrum
%
w_d = 60.0*2*pi;    %narrowband disturbance centered at 60 Hz
z_d = .01;          %damping model for sinusoidal excitation
%z_d = .1;          %damping model for narrowband excitation
F_w = [0 1;-w_d^2 -2*z_d*w_d];
G_w = [0;1];
C_w = [0 1];        %see notes on formulation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%    Define the unity-gain Sallen-Key low pass filter on
%    the control force.
%
t = 2;                  %smoothing filter order
w_lp = 120*2*pi;        %cut-off freq of 120 Hz.
z_lp = 0.707;           %damping ratio that has best flatness
F_lp = [0 1;-w_lp^2 -2*z_lp*w_lp];
G_lp = [0;w_lp^2];
C_lp = [1 0];
D_lp = 0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%    Set up the augmented matrix equation
%    Use these for continuous-time augmented system of the
%    estimator.
%
Fa = [F G*C_lp L*C_w;zeros(t,2*n) F_lp zeros(t,2*s);..
     zeros(2*s,2*n) zeros(2*s,t) F_w];
Ga = [zeros(2*n,m);G_lp;zeros(2*s,m)];
La = [zeros(2*n,s);zeros(t,s);G_w];
Ca = [C D*C_lp D_d*C_w];
D = zeros(n,m);
H = eye(2*n);
Ha = [H zeros(2*n,t) zeros(2*n,2*s)];
% Discretize the augmented system - plate & disturbance
[dFa,dGa]=c2d(Fa,Ga,T);
[dFa,dLa]=c2d(Fa,La,T);
seig=eig(Fa);
ob=obsv(dFa,Ca);
rob=rank(ob);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   Augment the plate equations with the smoothing filter.
%
F_p = [F_p G_p*C_lp;zeros(t,2*np) F_lp];
G_p = [zeros(2*np,m);G_lp];
L_p = [L_p;zeros(t,s)];
```

```
C_p = [C_p D_p*C_1p];
D_p = zeros(np,m);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%       Calculate the discrete feedback and Kalman filter gains
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Q = Ha'*Ha;                         %controller cost matrices
rho = 5e-4;
R = rho * eye(m);
dC_1q = DLQR(dFa,dGa,Q,R)           %discrete feedback gains
dC_1q(7)=-350;                      %optimized to compensate mode 1
dC_1q(8)=0.8;
clear Q;
clear R;
Q_kf = 5e2;
[QQ_kf,S] = int_Qd(Fa,T,La,Q_kf);   %assume that the disturbance acts
                                    %continuously
qf = 25;
R_kf =  qf * eye(n);                %noise covariance
dKf = DLQE(dFa,eye(2*n+2*m+2*s),Ca,QQ_kf,R_kf);
                                    %discrete estimator gains
%
%  The following lines save the coefficients used in the
%  digital controller.
```

```
%This is for chan_count = 12
chan_count = 12;
MM = p;
MF = inv(MM'*MM)*MM';
freq = 1/T;
cont_n = 2*n + 2*s + t;
p_c = n;
p_r = np - n;
save test.dat freq cont_n m p_c p_r chan_count dFa dGa
                Ca D dC_lq dKf MF /ascii
%
% closed loop poles
z_clpoles=eig(dFa-dGa*dC_lq)
z_obspls = eig((eye(2*n+2*m+2*s)-dKf*Ca)*dFa);
zeig=eig(dFa);
zeig=[zeig z_clpoles z_obspls];
seig = 1/T*log(zeig);
zz=abs(zeig)
ss = abs (seig);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%       Real time open loop and closed loop simulation of the
%       augmented system.  The disturbance is the only input
%       until the controller is implemented.
%
```

```
%          First-order hold approximation is used on the
%          disturbance input in the plate simulation.  The actual
%          system, as opposed to the augmented system simulates
%          the plate response.
%
clear Q;
N_limit = 3000;
%
%  disturbance input sequence
%tm = [0:T:2.0];        %time axis for the sinusoidal disturbance
%f = 60;                %excitation frequency - Hz
%omega = 2 * pi * f;
%w = 2 * sin(omega*tm);     %excitation force
load nbsim             %simulated narrowband disturbance sequence

% First Order Hold Approximation
[q,s] = size(L_p);
% For first order hold approximation first add m
%                  integrators in series
[F_p,L_p,C_p,D_d_p] =
         series(zeros(s),eye(s),eye(s),zeros(s),F_p,L_p,C_p,D_d_p);
G_p = [zeros(s,1);G_p];
% Get equivalent zero order hold discrete system
[dF_p,dL_p] = c2d(F_p,L_p,T);
[dF_p,dG_p] = c2d(F_p,G_p,T);
```

```
%  Initial conditions
x0 = zeros(2*np+t,1);              %plant states
u(1) = 0.0;

% For first order hold add (z-1)/T in series
% This is equivalent to differentiating u.
% Transfer first sample to initial conditions.
x0=[zeros(s,1);x0(:)]+(G_p*u(1)+L_p*w(1));
x(1,:) = x0';

Qminus(1,:) = zeros(1,2*n+2*m+2*s);       %predicted estimate
yhat(1,:) = (Ca*Qminus(1,:)')';

y(1,:) = (C_p*x(1,:)' + D_p*u(1) + D_d_p*w(1))';
k=1;

Qplus(1,:) = (Qminus(1,:)' + dKf*(y(1,1:n) - yhat(1,:))')';
                                  %meas. updated state estimates

u(2) = 0.0;                %plant and estimated states at t=T
x(2,:) = (dF_p*x(1,:)' + dG_p*u(1) + dL_p*((w(2)-w(1))/T))';
y(2,:) = (C_p*x(2,:)' + D_p*u(2) + D_d_p*w(2))';  % testf_2m only

k=2;
```

```
Qminus(2,:) = (dFa*Qplus(1,:)' + dGa*u(1))';
yhat(2,:) = (Ca*Qminus(2,:)' + D*u(2))';
Qplus(2,:)  = (Qminus(2,:)' + dKf * (y(2,1:n) - yhat(2,:))')';

for k=3:N_limit;
   Qminus(k,:) = (dFa*Qplus(k-1,:)' + dGa*u(k-1))';
                                %K. F. predictor equation
%controller input
   if k<2000,
      u(k) = 0;
   else
      u(k) = -dC_lq*Qminus(k,:)';
   end

% clip the controller input

    if u(k)>5.3878,
       u(k) = 5.3878;
    elseif u(k)<-5.3878,
       u(k) = -5.8378;
    end

   ww(k-1) = (w(k)-w(k-1))/T;
   x(k,:) = (dF_p*x(k-1,:)' + dG_p*u(k-1) + dL_p*ww(k-1))';
                                %plant output equation
```

```
    y(k,:) = (C_p*x(k,:)' + D_p*u(k) + D_d_p*w(k))';
                                    %measured modal accelerations


    yhat(k,:) = (Ca*Qminus(k,:)' + D*u(k))';

    Qplus(k,:)  = (Qminus(k,:)' + dKf * (y(k,1:n) - yhat(k,:))')';
                                    %Kalman update filter equation

end
end
```

```
function [Q_d,S] = int_Qd(F,T,L,Q_c)
%
%   int_Qd.m  This file will calculate the discrete covariance
%             matrix, Q_d, of the continuous time disturbance
%             with a spectral density matrix, Q_c.  The other
%             necessary inputs are the state matrices, F and L,
%             and the sampling time, T.
%
%             The rectangular rule is implemented to perform the
%             numerical integration of eq. 4.2-52 (Stengel, p.328).
%             The Cholesky decomposition of Q_d forms S which is
%             to simulate the cross-corelated white noise, w_k,
%             with the Gaussian white noise, v_k, such that
%                          w_k = S*v_k .
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
N = 100;        %number of intervals in the integration
delt = T/N;     %time interval

sum = 0;
dum=0;

for k=1:N;

   t = (k-1)*delt;
```

```
    dum = expm(F*t)*L*Q_c*L'*expm(F'*t)*delt;  %rectangular rule
    sum = sum + dum;

end

Q_d = sum;
R = chol(Q_d);
S = R';
end;
```

Appendix C

Discrete Controller Code

The software used to drive the transputer-based, parallel-processing control system is included below. The first program, filthost.c, is the discrete controller implemented on the T800 tranputer. The coefficients of the equations are defined in c_t5e4.h which is included next. The second program, filt_t2.c, is the data-acquisition software implemented on the T222 transputer. The programs were written by Graham K. Ellis based on the control code that drives the DT2818 data acquisition board which was written by William R. Saunders.

```
/*****************************************************************
*
* filthost.c  This code is derived from dtcontrol.c, the four mode
*             controller for the simply supported plate vibration
*             control experiment.  Closed-loop control is generated
*             here.  The code writes the following
*             binary data files:
*
*               y.dat        measured modal acceleration amplitudes.
```

```
*                yhat.dat      estimated modal acceleration amplitudes.
*                qe_disp.dat   estimated modal displacement amplitudes.
*                qe_vel.dat    estimated modal velocity amplitudes.
*                qe_dist.dat   estimated disturbance states.
*                yacc.dat      measured physical acclerations (optional)
*
*
* Modifed for transputer control system
* 12/27/90
* GKE
*
**********************************************************************
*/
/*   To Communicate with the T2 data acquisition node, use LINK3
*
* NOTE: T8 integers are 32 bits
* T2 integers are 16 bits
*/

#include <stdio.h>
#include <math.h>
#include <conc.h>
#include "c_t5e4.h"

#define   ADCGAIN  3.005        /* board amplifier gain   */
```

```
#define   AD_SCALE 204.7/ADCGAIN
#define   NLOOPS   3000    /* number of control loops to perform */

#define UTOINT16(x)  ((x)*204.7 + 2047)
#define INT16TOACC(x, cal)  (((x)-2047) * (cal))

int savemat(FILE *fp, int type, char *pname, int mrows, int ncols,
              int imagf, double *preal, double *pimag);

double dummy[12*NLOOPS];
float uu[NLOOPS];
float y1[NLOOPS],       y2[NLOOPS],
      y3[NLOOPS],       y4[NLOOPS],
      yacc1[NLOOPS], yacc2[NLOOPS],
      yacc3[NLOOPS], yacc4[NLOOPS],
      yacc5[NLOOPS], yacc6[NLOOPS],
      yacc7[NLOOPS], yacc8[NLOOPS],
      yacc9[NLOOPS], yacc10[NLOOPS],
      yacc11[NLOOPS], yacc12[NLOOPS];
float ldcell[NLOOPS];
float yhat1[NLOOPS], yhat2[NLOOPS];
float qe5[NLOOPS], qe6[NLOOPS],
  qe7[NLOOPS], qe8[NLOOPS];

void main(void)
```

```
{
FILE *dataout1;
FILE *dataout2;
FILE *dataout3;
FILE *dataout4;
FILE *dataout5;
FILE *dataout6;
FILE *dataout7;

char *cpnt; /* character pointers used so we can output */
char *upnt; /* two bytes of a 4 byte integer */
int   i, u16; /* control voltage in correct DAC format  */
int acc[13]; /* integer acceleromter data from T2 */
float acc0, /* Real, calibrated acceleration values */
          acc1, /* in m/s^2/volt */
          acc2,
          acc3,
          acc4,
          acc5,
          acc6,
          acc7,
          acc8,
          acc9,
          acc10,
          acc11,
```

```
          lc;
float y1del, y2del;
float final_cal0, final_cal1,
       final_cal2, final_cal3,
  final_cal4, final_cal5,
  final_cal6, final_cal7,
         final_cal8, final_cal9,
      final_cal10, final_cal11, lc_cal;
float qe1, qe2, qe3, qe4;
float s1, s2, s3, s4, s5, s6, s7, s8;
float freq, u, uvolt;

int nloops, k;
int start_time, stop_time;

dataout1 = fopen("y.mat","wb");
dataout2 = fopen("yhat.mat","wb");
dataout3 = fopen("qe_filt.mat","wb");
dataout4 = fopen("qe_dist.mat","wb");
dataout5 = fopen("u_cont.mat","wb");
dataout6 = fopen("yacc.mat","wb");
dataout7 = fopen("dist.mat","wb");

/* set the character control pointer to point at u16 */
upnt = (char *) &u16;
```

```
/* The ADC data is converted to the correct units for use in
the control law using the appropriate gains and sensitivities
*/

final_cal0 = (SENS0 / CHAN0_GAIN)/AD_SCALE;     /* m/s^2/volt */
final_cal1 = (SENS1 / CHAN1_GAIN)/AD_SCALE;
final_cal2 = (SENS2 / CHAN2_GAIN)/AD_SCALE;
final_cal3 = (SENS3 / CHAN3_GAIN)/AD_SCALE;
final_cal4 = (SENS4 / CHAN4_GAIN)/AD_SCALE;
final_cal5 = (SENS5 / CHAN5_GAIN)/AD_SCALE;
final_cal6 = (SENS6 / CHAN6_GAIN)/AD_SCALE;
final_cal7 = (SENS7 / CHAN7_GAIN)/AD_SCALE;
final_cal8 = (SENS8 / CHAN8_GAIN)/AD_SCALE;
final_cal9 = (SENS9 / CHAN9_GAIN)/AD_SCALE;
final_cal10 = (SENS10 / CHAN10_GAIN)/AD_SCALE;
final_cal11 = (SENS11 / CHAN11_GAIN)/AD_SCALE;
            lc_cal = (SENSLC / LC_GAIN  )/AD_SCALE;

/*  Initial Values: Use a "guess" to estimate the initial
value of the x(k/k) so the Kalman Filter will converge faster.
*/

  u = 0.0;
```

```
  qe1    = 0.0; /* position  */
  qe2    = 0.0; /* position  */
  qe3    = 0.0; /* velocity */
  qe4    = 0.0;  /* velocity */
  qe5[0] = 0.0;  /* smoothing filter */
  qe6[0] = 0.0;  /* smoothing filter */
  qe7[0] = 0.0;  /* disturbance */
  qe8[0] = 0.0;  /* disturbance */

  uu[0] = 0.0;

  /* Send the number of control loops to the data acquisiton T2 */
  nloops = NLOOPS;
  cpnt = (char *) &nloops; /* char pointer to temp */
  ChanOut(LINK3OUT, cpnt, 2); /* send 2 bytes to T2 */

  printf("Press any key to start\n");
  getch();

/*-------------------------------------------------------------
  M A I N L O O P
 -------------------------------------------------------------
*/
  k = -1; /* set loop counter */
```

```
  start_time = Time();
  while (k<NLOOPS)
    {
     k++;

     /*  begin state estimator (Kalman Filter) */
     /*  state propagation equation - predictor (augmented system) */

    s1  = a11*qe1+a13*qe3+a15*qe5[k]+a16*qe6[k]
                          +a17*qe7[k]+a18*qe8[k]+b11*u;
    s2  = a22*qe2+a24*qe4+a25*qe5[k]+a26*qe6[k]
                          +a27*qe7[k]+a28*qe8[k]+b21*u;
    s3  = a31*qe1+a33*qe3+a35*qe5[k]+a36*qe6[k]
                          +a37*qe7[k]+a38*qe8[k]+b31*u;
    s4  = a42*qe2+a44*qe4+a45*qe5[k]+a46*qe6[k]
                          +a47*qe7[k]+a48*qe8[k]+b41*u;
    s5  = a55*qe5[k]+a56*qe6[k]+b51*u;
    s6  = a65*qe5[k]+a66*qe6[k]+b61*u;
    s7  = a77*qe7[k]+a78*qe8[k];
    s8  = a87*qe7[k]+a88*qe8[k];

  /*  calculate the control signal  */

/* open loop */
/* u = 0.0; */
```

```
 if (k < 400)
     u = 0.0;
 else
   u =  -(C_LQ11 * s1 + C_LQ12 * s2 + C_LQ13 * s3 + C_LQ14 * s4 +
          C_LQ15 * s5 + C_LQ16 * s6 + C_LQ17 * s7 + C_LQ18 * s8);


  uvolt = -(u * 1.787);
                  /* converts Newtons to volts from cont_tf.dat */
                  /* accounts for inversion by current amp */

if (uvolt > 10.0){
   uvolt = 9.9;
   u = -5.54;
   }
if (uvolt < -10.0){
   uvolt = -9.9;
   u = 5.54;
   }

   uu[k+1] = u;

/* send control voltage to T2. Note, upnt points to u16. */
```

```
u16 = UTOINT16(uvolt);
ChanOut(LINK3OUT, upnt, 2);  /* send 2 bytes */

  /*  Kalman filter on output */

    yhat1[k+1] = ca11 * s1 + ca13 * s3 + ca15 * s5 + ca18 * s8;
    yhat2[k+1] = ca22 * s2 + ca24 * s4 + ca25 * s5 + ca28 * s8;

    /* Read in the acceleration values from the T2  */
    /* We are reading this in as two non-zero bytes, and  */
    /* then 2 zero bytes to simplify the code on this */
    /* transputer.  We expect a LSB -> MSB transfer  */

    ChanIn(LINK3IN, (char *) acc, sizeof(acc));

    /* now convert the integer value to voltage */
acc0  = INT16TOACC(acc[0], final_cal0);
acc1  = INT16TOACC(acc[1], final_cal1);
acc2  = INT16TOACC(acc[2], final_cal2);
acc3  = INT16TOACC(acc[3], final_cal3);
acc4  = INT16TOACC(acc[4], final_cal4);
acc5  = INT16TOACC(acc[5], final_cal5);
acc6  = INT16TOACC(acc[6], final_cal6);
acc7  = INT16TOACC(acc[7], final_cal7);
acc8  = INT16TOACC(acc[8], final_cal8);
```

```
acc9  = INT16TOACC(acc[9], final_cal9);
acc10 = INT16TOACC(acc[10], final_cal10);
acc11 = INT16TOACC(acc[11], final_cal11);
          lc = INT16TOACC(acc[12], lc_cal);


   /* acc 1 through 12 as output */

    y1[k+1] = mf11 * acc0 + mf12 * acc1 + mf13 * acc2 + mf14 * acc3
            + mf15 * acc4 + mf16 * acc5 + mf17 * acc6 + mf18 * acc7
            + mf19 * acc8 + mf110* acc9 + mf111* acc10+ mf112* acc11;

    y2[k+1] = mf21 * acc0 + mf22 * acc1 + mf23 * acc2 + mf24 * acc3
            + mf25 * acc4 + mf26 * acc5 + mf27 * acc6 + mf28 * acc7
            + mf29 * acc8 + mf210* acc9 + mf211* acc10+ mf212* acc11;

    y3[k+1] = mf31 * acc0 + mf32 * acc1 + mf33 * acc2 + mf34 * acc3
            + mf35 * acc4 + mf36 * acc5 + mf37 * acc6 + mf38 * acc7
            + mf39 * acc8 + mf310* acc9 + mf311* acc10+ mf312* acc11;

    y4[k+1] = mf41 * acc0 + mf42 * acc1 + mf43 * acc2 + mf44 * acc3
            + mf45 * acc4 + mf46 * acc5 + mf47 * acc6 + mf48 * acc7
            + mf49 * acc8 + mf410* acc9 + mf411* acc10+ mf412* acc11;

    yacc1[k+1] = acc0;
```

```
  yacc2[k+1] = acc1;
  yacc3[k+1] = acc2;
  yacc4[k+1] = acc3;
  yacc5[k+1] = acc4;
  yacc6[k+1] = acc5;
  yacc7[k+1] = acc6;
  yacc8[k+1] = acc7;
  yacc9[k+1] = acc8;
  yacc10[k+1] = acc9;
  yacc11[k+1] = acc10;
  yacc12[k+1] = acc11;
  ldcell[k+1] = lc;


  y1del = y1[k+1] - yhat1[k+1];
  y2del = y2[k+1] - yhat2[k+1];

  /*  update equation - corrector */

qe1      = s1 + K11*y1del + K12*y2del;
qe2      = s2 + K21*y1del + K22*y2del;
qe3      = s3 + K31*y1del + K32*y2del;
qe4      = s4 + K41*y1del + K42*y2del;
qe5[k+1] = s5 + K51*y1del + K52*y2del;
qe6[k+1] = s6 + K61*y1del + K62*y2del;
```

```
qe7[k+1] = s7 + K71*y1del + K72*y2del;
qe8[k+1] = s8 + K81*y1del + K82*y2del;

}  /* end control loop */

stop_time = Time();
printf("Number of Low-Pri ticks = %d\n", stop_time - start_time);
freq = k/((stop_time - start_time)*64e-6);
printf("Frequency %4.1f Hz\n", freq);

printf("Writing Data File....\n");


for (i=0;i<NLOOPS;i++) {
  dummy[i] = (double) y1[i];
  dummy[i+NLOOPS] = (double) y2[i];
  dummy[i+2*NLOOPS] = (double) y3[i];
  dummy[i+3*NLOOPS] = (double) y4[i];
}
savemat(dataout1, 0, "y", NLOOPS, 4, 0,
                          (double *) dummy, (double *)0);

for (i=0;i<NLOOPS;i++){
    dummy[i] = (double) yhat1[i];
    dummy[i+NLOOPS] = (double) yhat2[i];
```

```
}
savemat(dataout2, 0, "yhat", NLOOPS, 2, 0,
                              (double *) dummy, (double *)0);

for (i=0;i<NLOOPS;i++){
    dummy[i] = (double) qe5[i];
    dummy[i+NLOOPS] = (double) qe6[i];
}
savemat(dataout3, 0, "qe_filt", NLOOPS, 2, 0,
                                 (double *) dummy, (double *)0);

for (i=0;i<NLOOPS;i++){
    dummy[i] = (double) qe7[i];
    dummy[i+NLOOPS] = (double) qe8[i];
}

savemat(dataout4, 0, "qe_dist", NLOOPS, 2, 0,
                                 (double *) dummy, (double *)0);

for (i=0;i<NLOOPS;i++)
    dummy[i] = (double) uu[i];

savemat(dataout5, 0, "u_cont", NLOOPS, 1, 0,
                                (double *) dummy, (double *)0);
```

```
for (i=0;i<NLOOPS;i++){
    dummy[i] = (double) yacc1[i];
    dummy[i+NLOOPS] = (double) yacc2[i];
    dummy[i+2*NLOOPS] = (double) yacc3[i];
    dummy[i+3*NLOOPS] = (double) yacc4[i];
    dummy[i+4*NLOOPS] = (double) yacc5[i];
    dummy[i+5*NLOOPS] = (double) yacc6[i];
    dummy[i+6*NLOOPS] = (double) yacc7[i];
    dummy[i+7*NLOOPS] = (double) yacc8[i];
    dummy[i+8*NLOOPS] = (double) yacc9[i];
    dummy[i+9*NLOOPS] = (double) yacc10[i];
    dummy[i+10*NLOOPS] = (double) yacc11[i];
    dummy[i+11*NLOOPS] = (double) yacc12[i];
}
savemat(dataout6, 0, "yacc", NLOOPS, 12, 0,
                              (double *) dummy, (double *)0);

for (i=0;i<NLOOPS;i++)
    dummy[i] = (double) ldcell[i];

savemat(dataout7, 0, "dist", NLOOPS, 1, 0,
                              (double *) dummy, (double *)0);

fcloseall();
}
```

```
/*
 * savemat - C language routine to save a matrix in a MAT-file.
 *
 * Here is an example that uses 'savemat' to save two
 * matrices to disk, the second of which is complex:
 *
 * FILE *fp;
 * double xyz[1000], ar[1000], ai[1000];
 * fp = fopen("foo.mat","wb");
 * savemat(fp, 0, "xyz", 2, 3, 0, xyz, (double *)0);
 * savemat(fp, 0, "a", 5, 5, 1, ar, ai);
 *      fclose(fp);
 *
 * Author J.N. Little 11-3-86
 */


typedef struct {
    long type;   /* type */
    long mrows;  /* row dimension */
    long ncols;  /* column dimension */
    long imagf;  /* flag indicating imag part */
    long namlen; /* name length (including NULL) */
```

```
} Fmatrix;

savemat(FILE *fp, int type, char *pname, int mrows, int ncols,
                int imagf, double *preal, double *pimag)

/*  FILE *fp;          File pointer */
/*  int type;          Type flag: Normally 0 for PC, 1000 for Sun, Mac,
                  and Apollo, 2000 for VAX D-float, 3000 for VAX G-float
    Add 1 for text variables.
    See LOAD in reference section of guide for more info. */
/*  int mrows;         row dimension */
/*  int ncols;         column dimension */
/*  int imagf;        imaginary flag */
/*  char *pname;       pointer to matrix name */
/*  double *preal;     pointer to real data */
/*  double *pimag;     pointer to imag data */
  {
   Fmatrix x;
   int mn;
   int n;

   x.type = type;
   x.mrows = mrows;
   x.ncols = ncols;
   x.imagf = imagf;
```

```
 x.namlen = strlen(pname) + 1;
 mn = x.mrows * x.ncols;

 fwrite(&x, sizeof(Fmatrix), 1, fp);
 fwrite(pname, sizeof(char), (int)x.namlen, fp);
 n=fwrite(preal, sizeof(double), mn, fp);
      if (imagf) {
      fwrite(pimag, sizeof(double), mn, fp);
 }
}
```

```
/* Include file for plate control code */
/* c_t5e4.h */

/* sample frequency 2087.0 */
/* controller order 8 */
/* number of inputs 1 */
/* number of outputs (controlled)  2 */
/* number of outputs (observed)    2 */
/* number of analog input channels 12 */

#define CHAN0_GAIN 60   /* acc 1  */
#define CHAN1_GAIN 50   /* acc 2  */
#define CHAN2_GAIN 60   /* acc 3  */
#define CHAN3_GAIN 50   /* acc 4 */
#define CHAN4_GAIN 40   /* acc 5 */
#define CHAN5_GAIN 50   /* acc 6 */
#define CHAN6_GAIN 60   /* acc 7 */
#define CHAN7_GAIN 40   /* acc 8 */
#define CHAN8_GAIN 50   /* acc 9 */
#define CHAN9_GAIN 60   /* acc 10 */
#define CHAN10_GAIN 50   /* acc 11 */
#define CHAN11_GAIN 60   /* acc 12 */
#define LC_GAIN   1  /*  LC    */

#define SENS0 (10.1497*9.81) /* m/s^2/volt */
```

```
#define SENS1 (10.085*9.81)
#define SENS2 (9.845*9.81)
#define SENS3 (9.919*9.81)
#define SENS4 (9.827*9.81)
#define SENS5 (10.004*9.81)
#define SENS6 (9.764*9.81)
#define SENS7 (9.994*9.81)
#define SENS8 (10.059*9.81)
#define SENS9 (9.886*9.81)
#define SENS10 (9.715*9.81)
#define SENS11 (9.799*9.81)
#define SENSLC (4.44*.9542)

#define a11 9.88948e-01
#define a12 0.00000e+00
#define a13 4.76840e-04
#define a14 0.00000e+00
#define a15 6.64438e-08
#define a16 9.42675e-12
#define a17 -1.82507e-06
#define a18 8.02142e-08

#define a21 0.00000e+00
#define a22 9.46812e-01
#define a23 0.00000e+00
```

```
#define a24 4.68805e-04
#define a25 4.11721e-08
#define a26 5.85959e-12
#define a27 7.90165e-07
#define a28 -3.46139e-08

#define a31 -4.60270e+01
#define a32 0.00000e+00
#define a33 9.86660e-01
#define a34 0.00000e+00
#define a35 2.74105e-04
#define a36 5.63937e-08
#define a37 -1.14002e-02
#define a38 3.32952e-04

#define a41 0.00000e+00
#define a42 -2.19729e+02
#define a43 0.00000e+00
#define a44 9.39293e-01
#define a45 1.68455e-04
#define a46 3.49250e-08
#define a47 4.91941e-03
#define a48 -1.42511e-04

#define a51 0.00000e+00
```

```
#define a52 0.00000e+00
#define a53 0.00000e+00
#define a54 0.00000e+00
#define a55 9.45146e-01
#define a56 3.67125e-04
#define a57 0.00000e+00
#define a58 0.00000e+00

#define a61 0.00000e+00
#define a62 0.00000e+00
#define a63 0.00000e+00
#define a64 0.00000e+00
#define a65 -2.08707e+02
#define a66 5.53743e-01
#define a67 0.00000e+00
#define a68 0.00000e+00

#define a71 0.00000e+00
#define a72 0.00000e+00
#define a73 0.00000e+00
#define a74 0.00000e+00
#define a75 0.00000e+00
#define a76 0.00000e+00
#define a77 9.83749e-01
#define a78 4.75695e-04
```

```
#define a81 0.00000e+00
#define a82 0.00000e+00
#define a83 0.00000e+00
#define a84 0.00000e+00
#define a85 0.00000e+00
#define a86 0.00000e+00
#define a87 -6.76069e+01
#define a88 9.80162e-01


#define b11 6.59370e-10

#define b21 4.10604e-10

#define b31 5.35900e-06

#define b41 3.33112e-06

#define b51 5.48540e-02

#define b61 2.08707e+02

#define b71 0.00000e+00
```

```
#define b81 0.00000e+00



#define ca11 -9.65250e+04
#define ca12 0.00000e+00
#define ca13 -4.79873e+00
#define ca14 0.00000e+00
#define ca15 7.03290e-01
#define ca16 0.00000e+00
#define ca17 0.00000e+00
#define ca18 7.03343e-01

#define ca21 0.00000e+00
#define ca22 -4.68699e+05
#define ca23 0.00000e+00
#define ca24 -1.60400e+01
#define ca25 3.66434e-01
#define ca26 0.00000e+00
#define ca27 0.00000e+00
#define ca28 -3.06230e-01



#define dca11 0.00000e+00


#define dca21 0.00000e+00
```

```
#define C_LQ11 -7.24525e+03
#define C_LQ12 -1.03127e+04
#define C_LQ13 2.82101e+01
#define C_LQ14 1.69241e-01
#define C_LQ15 4.60050e-02
#define C_LQ16 4.42722e-05
#define C_LQ17 -3.50000e+02
#define C_LQ18 8.00000e-01


#define K11 -1.06512e-06
#define K12 7.39606e-08

#define K21 -2.67493e-08
#define K22 -7.57691e-08

#define K31 2.53506e-06
#define K32 4.05447e-05

#define K41 -3.82900e-05
#define K42 1.71724e-06

#define K51 -2.15949e-22
```

```
#define K52 -1.16386e-22

#define K61 -4.60618e-20
#define K62 -2.19688e-20

#define K71 -1.82844e-04
#define K72 3.68598e-05

#define K81 5.19002e-02
#define K82 6.52848e-03


#define mf11 1.26841e-01
#define mf12 1.94040e-01
#define mf13 9.56130e-02
#define mf14 1.84904e-01
#define mf15 3.04824e-01
#define mf16 1.78077e-01
#define mf17 1.59920e-01
#define mf18 2.87672e-01
#define mf19 2.11887e-01
#define mf110 6.18393e-02
#define mf111 1.72559e-01
#define mf112 1.35064e-01
```

```
#define mf21 -1.86425e-01
#define mf22 -2.98902e-01
#define mf23 -2.50592e-01
#define mf24 -6.73414e-02
#define mf25 -1.26491e-01
#define mf26 -1.10323e-01
#define mf27 8.92666e-02
#define mf28 1.48819e-01
#define mf29 1.14976e-01
#define mf210 1.61056e-01
#define mf211 2.92332e-01
#define mf212 2.34215e-01

#define mf31 1.51168e-01
#define mf32 1.87860e-02
#define mf33 -1.94451e-01
#define mf34 2.28256e-01
#define mf35 3.42570e-02
#define mf36 -2.75825e-01
#define mf37 1.72012e-01
#define mf38 3.15452e-02
#define mf39 -1.95192e-01
#define mf310 8.63878e-02
#define mf311 3.18677e-02
#define mf312 -6.11377e-02
```

```
#define mf41 -1.88112e-01
#define mf42 -3.93751e-02
#define mf43 2.06468e-01
#define mf44 -1.07694e-01
#define mf45 -4.85205e-02
#define mf46 1.53280e-01
#define mf47 8.89856e-02
#define mf48 -2.46614e-03
#define mf49 -1.13855e-01
#define mf410 2.27128e-01
#define mf411 -3.60397e-04
#define mf412 -2.91816e-01
```

```
/*****************************************************************
*
* filt_t2.c
*
* Data Acquitsion code for 2-mode plate controller
*
* Programmed By:
* GKE
* 12/27/90
*
*****************************************************************
*/

#include <conc.h>
#include <inline.h>

#define HEAP 1024

#define ADC 0x6000
#define SH 0x6100
#define MUX 0x6200
#define DAC 0x6800

#define ADCMASK 0x0fff /* Since we have a 12-bit converter */
/* mask off the high nibble.    */
```

```
inp(Process *);
outp(Process *, int);

void main(void)
{
int nloops;
Process *ain, *dout;

/* read in the number of control loops from the
                        root transputer */
nloops = ChanInInt(_boot_chan_in);

/* allocate the processes */
ain = ProcAlloc(inp, HEAP, NULL);
dout = ProcAlloc(outp, 256, 1, nloops+1);

ProcPriPar(ain, dout);
}

inp(Process *p)
{
int *dac1;
int temp;
int i;
```

```
int *adc, *sh, *mux; /* Hardware addresses */
int outdat[26];  /* we need 8 bytes to pad each word with */
 /* 2 zero bytes to make it look like   */
 /* 32-bit ints we are sending back to  */
 /* control node.  */
dac1 = (int *) 0x6808;
/* zero out the outdat[2i] */
for(i=0; i<sizeof(outdat)/sizeof(int); i++) {
outdat[i] = 0;
}

*sh = 0; /* track mode */
adc = (int *) ADC; /* assign the hardware pointers */
sh  = (int *) SH;
mux = (int *) MUX;

while(1) {
*mux = 0; /* channel 0 */
*sh = 1; /* hold the data */
ProcWait(1);
*adc = 0; /* start the conversion, channel 0 */
ChanInInt(EVENT); /* are we done? */
outdat[0] = *adc & ADCMASK; /* save the data */
*dac1 = outdat[0];
*mux = 1; /* channel 1 */
```

```
ProcWait(1);
*adc = 0;
ChanInInt(EVENT);
outdat[2] = *adc & ADCMASK;
*dac1 = outdat[2];
*mux = 2;          /* channel 2 */
ProcWait(1);
*adc = 0;
ChanInInt(EVENT);
outdat[4] = *adc & ADCMASK;
*dac1 = outdat[4];
*mux = 3;          /* channel 3 */
ProcWait(1);
*adc = 0;
ChanInInt(EVENT);
outdat[6] = *adc & ADCMASK;
*dac1 = outdat[6];
*mux = 4;          /* channel 4 */
ProcWait(1);
*adc = 0;
ChanInInt(EVENT);
outdat[8] = *adc & ADCMASK;
*dac1 = outdat[8];
*mux = 5;          /* channel 5 */
ProcWait(1);
```

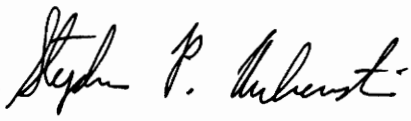
```
*adc = 0;
ChanInInt(EVENT);
temp=*adc;
outdat[10] = temp & ADCMASK;
*dac1 = outdat[10];
*mux = 6;       /* channel 6 */
ProcWait(1);
*adc = 0;
ChanInInt(EVENT);
outdat[12] = *adc & ADCMASK;
*dac1 = outdat[12];
*mux = 7;      /* channel 7 */
ProcWait(1);
*adc = 0;
ChanInInt(EVENT);
outdat[14] = *adc & ADCMASK;
*dac1 = outdat[14];
*mux = 8;     /* channel 8 */
ProcWait(1);
*adc = 0;
ChanInInt(EVENT);
outdat[16] = *adc & ADCMASK;
*dac1 = outdat[16];
*mux = 9;     /* channel 9 */
ProcWait(1);
```

```
*adc = 0;
ChanInInt(EVENT);
outdat[18] = *adc & ADCMASK;
*dac1 = outdat[18];
*mux = 10;    /* channel 10 */
ProcWait(1);
*adc = 0;
ChanInInt(EVENT);
outdat[20] = *adc & ADCMASK;
*dac1 = outdat[20];
*mux = 11;    /* channel 11 */
ProcWait(1);
*adc = 0;
ChanInInt(EVENT);
outdat[22] = *adc & ADCMASK;
*dac1 = outdat[22];
*mux = 12;    /* channel 12 */
ProcWait(1);
*adc = 0;
ChanInInt(EVENT);
outdat[24] = *adc & ADCMASK;
*dac1 = outdat[24];
ChanOut(_boot_chan_out, (char *) outdat, sizeof(outdat));
*sh = 0; /* track */
ProcWait(1);
```

```
}
}


outp(Process *p, int nloops)
{
int data;
int *dac;

dac = (int *) DAC;
*dac = 2047; /* zero the output */

/* control the number of loops in this routine */

while(nloops--) {
data = ChanInInt(_boot_chan_in);
*dac = data;
}
*dac = 2047;
}
```

Vita

Stephen P. Rubenstein was born in Flushing, New York on March 22, 1967. He grew-up in Rockville Centre, New York where he graduated from South Side High School in 1985. He attended Virginia Polytechnic Institute and State University, and he received a Bachelors of Science in Mechanical Engineering in 1989 and stayed to pursue a Masters of Science degree. Upon graduation he will accept a research engineering position with Nelson Industries in Stoughton, Wisconsin.



Stephen P. Rubenstein