NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

ACTIVE VIBRATION CONTROL OF FLEXIBLE STRUCTURES USING THE MODULAR CONTROL PATCH (MCP)

by

Steven P. Schmidt

March, 1997

Thesis Advisor: Brij N. Agrawal

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE March 1997	3. REPORT TYPE AND DATES COVERED Engineer's Thesis

4. TITLE AND SUBTITLE ACTIVE VIBRATION CONTROL OF FLEXIBLE STRUCTURES USING THE MODULAR CONTROL PATCH (MCP)

5. FUNDING NUMBERS

6. AUTHOR(S) Steven P. Schmidt

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Active vibration control has been increasingly used as a solution for spacecraft structures to achieve the degree of vibration suppression required for precision pointing accuracy that is not easily achieved with passive damping. This thesis examines the effectiveness and suitability of the Modular Control Patch (MCP) to achieve active vibration control on flexible structures. The MCP was developed by TRW for the United States Air Force and uses a digital signal processor to implement control algorithms. The objective of the MCP program was to design a miniaturized multi-channel digital controller suitable for space-based vibration control. Three different control laws: Positive Position Feedback (PPF), Strain Rate Feedback (SRF), and Integral control were realized using the MCP. These control laws were used independently and in combination in order to discover the most effective damping for the first two modal frequencies on a cantilevered aluminum beam. Two PPF filters in parallel provided the most effective multi-mode damping. Further experiments tested the robustness of the PPF control law implemented by the MCP. Increasing the compensator damping greatly improved PPF robustness and expanded its capability as an effective controller.

14. SUBJECT TERMS
MCP, piezoceramic, active damping, robustness

15. NUMBER OF PAGES 120

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

Approved for public release; distribution is unlimited.

ACTIVE VIBRATION CONTROL OF FLEXIBLE STRUCTURES USING THE MODULAR CONTROL PATCH (MCP)

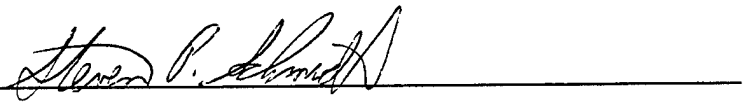
Steven P. Schmidt
Lieutenant Commander, United States Navy
B.S.E.E., University of Illinois, 1984

Submitted in partial fulfillment
of the requirements for the degree of

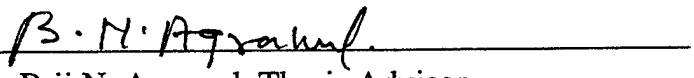
AERONAUTICAL AND ASTRONAUTICAL ENGINEER
MASTER OF SCIENCE IN ASTRONAUTICAL ENGINEERING

from the

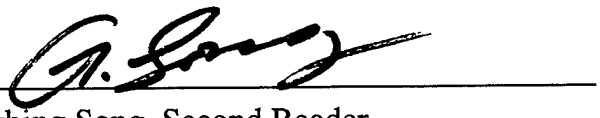
NAVAL POSTGRADUATE SCHOOL
March 1997

Author: 

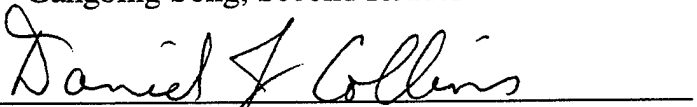
Steven P. Schmidt

Approved by: 

Brij N. Agrawal, Thesis Advisor



Gangbing Song, Second Reader



Daniel J. Collins, Chairman
Department of Aeronautics and Astronautics

ABSTRACT

Active vibration control has been increasingly used as a solution for spacecraft structures to achieve the degree of vibration suppression required for precision pointing accuracy that is not easily achieved with passive damping. This thesis examines the effectiveness and suitability of the Modular Control Patch (MCP) to achieve active vibration control on flexible structures. The MCP was developed by TRW for the United States Air Force and uses a digital signal processor to implement control algorithms. The objective of the MCP program was to design a miniaturized multi-channel digital controller suitable for space-based vibration control. Three different control laws: Positive Position Feedback(PPF), Strain Rate Feedback(SRF), and Integral control were realized using the MCP. These control laws were used independently and in combination in order to discover the most effective damping for the first two modal frequencies on a cantilevered aluminum beam. Two PPF filters in parallel provided the most effective multi-mode damping. Further experiments tested the robustness of the PPF control law implemented by the MCP. Increasing the compensator damping greatly improved PPF robustness and expanded its capability as an effective controller.

TABLE OF CONTENTS

ACKNOWLEDGEMENT

The author would like to acknowledge the individuals who have made this thesis possible. First and foremost, thanks to my wife, Cathy, and my children, Amelia, Ben, and Catrina. They patiently endured my long days and late nights at work and were a constant source of love and support. Their sacrafices were numerous and are graciously acknowledged and greatly appreciated. Thanks also to Dr. Gangbing Song whose enthusiasm, advice, and assistance in the laboratory were invaluable to the author in collecting, processing, and analyzing data. He truly was an inspiration and a tremendous asset. Finally, thanks to Dr. Brij Agrawal for his continual guidance in defining the focus of this thesis and for his encouragement during especially difficult times. His astute, reasoned advice kept the author on track through the entire process.

I. INTRODUCTION

A. BACKGROUND

The constant pressure to make spacecraft lighter has led to the increasing use of lighter and more flexible materials for spacecraft. This has resulted in very low natural frequencies for spacecraft. These frequencies are easily excited by a number of disturbances. Along with this, the evolution of payloads launched into space demand ever increasing pointing accuracy and minimal vibration noise. As a result of these opposing demands, extensive research is being conducted in actively controlling, suppressing, or even isolating structural vibration.

Structural vibrations are excited by numerous spacecraft functions. Examples include, but are not limited to, antenna slewing or pointing maneuvers, station repositioning, and thruster firing for momentum wheel desaturation. To dampen these structural excitations in an expeditious manner, smart structures are a promising technology. In general, these are system elements that can sense structural vibration and act to dampen it in order to meet the required performance. Smart structures can accomplish vibration damping to a degree that is not easily achieved with passive structures. Also, smart structures are known to have a number of advantages such as low weight, low noise, and easy implementation.

There are several different types of materials that can be used for actuators and sensors. Ones that can be used as sensors are piezoceramic wafers, strain gages, and fiber optics. For actuators, piezoceramic wafers, electrostrictive ceramic wafers, piezoelectric polymer film and shape memory metal wires are candidates. For this experiment, piezoceramic sensors and actuators are used. Piezoceramics have several desirable characteristics for this type of application. These include a high strain sensitivity, high stiffness, low noise, good linearity, temperature insensitivity, ease of implementation, and low power consumption.[Ref. 1]

When vibration suppression was first accomplished using smart structures, analog circuits were solely utilized as compensators. These worked reasonably well but are rather inflexible and can be hampered by circuit noise if not constructed properly.

Also, once built, even minor adjustments to the controller required the hardware to be changed. In the last several years, digital signal processors have been increasingly used as compensators as they have become faster and more powerful. Digital processors can now handle the high throughput and minimum time delay requirements necessary for real time control. Once implemented, any changes necessary can be made via software making it a very adaptable and versatile tool.

B. SCOPE

This thesis examines the effectiveness and suitability of the MCP to achieve active vibration control on a flexible aluminum beam. The MCP is a miniaturized digital controller with an ultimate objective of being an operational space-qualified component. It is used in this experiment to implement Positive Position Feedback (PPF), Strain Rate Feedback (SRF), and Integral control laws. These control laws as realized by the MCP are evaluated on their effectiveness in damping the vibration of an aluminum beam. Control laws are tested individually and in combination in order to achieve the highest damping ratios possible. Both single mode and multi-mode suppression are investigated.

The robustness of the PPF controller is also investigated. The main documented disadvantage of PPF has been its lack of robustness. Typically, PPF has been realized with a compensator damping value of around 0.1. For this thesis, compensator damping values greater than 0.1 are examined. It will be shown that this change greatly increases the robustness of PPF to variations in the targeted modal frequencies.

II. MODULAR CONTROL PATCH

A. BACKGROUND

The Module Control Patch(MCP) program was funded by the Air Force Phillips Laboratory and the Ballistic Missile Defense Office and was tasked with the development of a miniaturized multi-channel digital controller. The MCP was specifically designed to be a controller for space-based vibration control and pointing systems. It is especially well suited for systems requiring high throughput of analog data and computations. The program was comprised of two different phases.

Phase I included a survey of potential target applications and their requirements for active vibration control. This was followed by a survey of electronic components which could meet the needs of the target applications. From this, two digital microprocessors were chosen as possibilities. The first system was a fixed-point two-input, one-output system centered around the Motorola DSP56156 microprocessor. The second was a two-input, two-output system suitable for control of coupled systems using floating point arithmetic and utilizing the Texas Instruments TMS320C30 chip.[Ref. 2]

1. Phase I Summary

a) Fixed Point Controller

The Motorola DSP5616 is a 16-bit fixed point processor with on-chip analog to digital(A/D) and digital to analog(D/A) conversion. The on-chip sigma-delta CODEC(coder-decoder) was found to give excellent analog data processing capability with 16-bit resolution. The chip could process two analog inputs using an input multiplexer and a gain stage with three selectable levels. It possesses a single analog output. Software implementation for this chip, however, proved troublesome. A simple two-pole bandpass filter implemented in 16-bit code required 2.5 pages of software with resolution and dynamic range being very poor. Thirty two bit code was then attempted using twelve custom built macros to implement the higher resolution arithmetic. This

required 25 pages of code and caused the filter to run slowly with still very poor dynamic range. At this point, the fixed point controller was dropped from consideration.[Ref. 2]

2. Floating Point Controller

The Texas Instrument TMS320C30 (C30) incorporates a 32-bit floating point arithmetic, parallel instruction capability, and on-chip Random Access Memory (RAM). A simple input-output device provided by Texas Instruments was found to have a time delay of greater than ½ millisecond. Additionally, harmonic distortion was around -50 dB. This interface for the C30 chip was rejected and separate A/D and D/A converters were chosen for implementation.[Ref. 2]

The processing performance of the C30 was found to be outstanding. Testing showed that the TI processor achieved near-theoretical performance in implementing a bandpass filter at 1 kHz.[Ref. 2]

3. Phase II Summary

The objective of Phase II was to develop a small multi-channel digital controller capable of adapting to a variety of spacecraft. Two printed circuit board controllers were first developed. The first controller(MCP-I) was a 2-input and 2-output board using serial data I/O between the processor and analog converters. Following successful tests of MCP-I, it was decided that having more analog channels would greatly enhance its' capability. The second controller, MCP-II, had 16 analog input channels and 8 output channels connected to the C30 and also utilized a field programmable gate array. All parts except the C30 were available in a radiation hardened, die versions as well as regular commercial versions.[Ref. 2]

Testing of the MCP-II was generally successful except for noise issues. This resulted in an MCP-III which incorporated several parts changes and a revised circuit board layout. This achieved a controller with excellent signal to noise ratio and 6 MCP-IIIs were produced with one being obtained by the Naval Postgraduate School in May

1996. The MCP-III controller at the Naval Postgraduate School contains a capability of 8 analog inputs with 6 analog outputs. In order to handle this many analog input and output channels, a time division multiplexing approach was adopted. This design moves all the digitized data to and from the processor using the C30 expansion bus. Since the expansion bus moves the data in parallel from the different inputs and outputs, the data from all can be moved in a single processor cycle. Timing of the numerous devices is controlled by the ACTEL field programmable gate array (FPGA).

The MCP-III was built mainly for laboratory test purposes in order to be a building block for a space qualified model. A subsequent model, the MCP-IV, was produced for the Air Force which involved a miniaturized version of the MCP-III controller using radiation-hardened (rad-hard) dies and remotely programmable read only memory(ROM) on the multi-chip module.[Ref. 1]

B. MCP-III

1. Overview

As mentioned above, the MCP-III and analog interface delivered to the Naval Postgraduate School allows for 8 inputs and 6 outputs with the analog board being specifically designed for piezoceramic sensors and actuators. Figure 2.1 shows a functional overview of the MCP and its interface. The only component not shown is an ACTEL field programmable gate array (FPGA) which is used to control the timing of the numerous channels through the C30. The MCP digital board was intentionally designed to contain only the digital signal processing components and the A/D and D/A converter. All sensor and actuator interface electronics were omitted from the MCP. The intent was to then design analog interface boards for each individual application. This way the MCP could be kept a general purpose device and the analog interface would be made to utilize a number of different sensors and actuators. These include but are not limited to

accelerometers, piezoceramics, and inductive probes. The analog interface at the Naval Postgraduate School is designed to have all piezoceramic inputs and outputs.

The basic flow is to send an address through the FPGA to the input multiplexer (MUX), instructing it which analog input to receive. A field effect transistor (FET) is used to select the desired input and transfer it the MDAC. The MDAC's output buffer amplifier then applies to the signal a gain from 0 to 9 before it is sent to a sample and hold device. The sample and hold device ensures the input signal remains stable while it is being converted to a digital signal. The A/D device is the 12-bit Analog Devices 774, a CMOS device which operates at 80,000 samples per second (sps). Twelve bit resolution is currently the greatest resolution available in rad-hard devices. The analog signal range of all the input devices is ±10 volts (V). The A/D converter is operated with a 10 V reference so that 0 Volt signals get an output code in the middle of the range. The code corresponding to the digitized analog signal is then transmitted to the processor across the expansion bus. The offset is subtracted in software before calculations are performed on the signal.[Ref. 2]

Data from all of the input channels is acquired before digital signal processing commences. Following calculations for each time step, digital command data is fed to all of the output channels for conversion to analog. The offset is added back to the actuator command code before it is sent to the D/A converter, as the D/A converter also employs a 10 V offset.[Ref. 2]

The D/A converter signal is scaled by the output MDAC, which behaves identically to the input MDAC. The analog output is then held by one of eight sample and hold devices. These sample and holds were built from a combination FET switch and hold circuit. The FPGA commands the FET switch to steer the voltage from the output MDAC to one of the eight hold circuits.[Ref. 2]

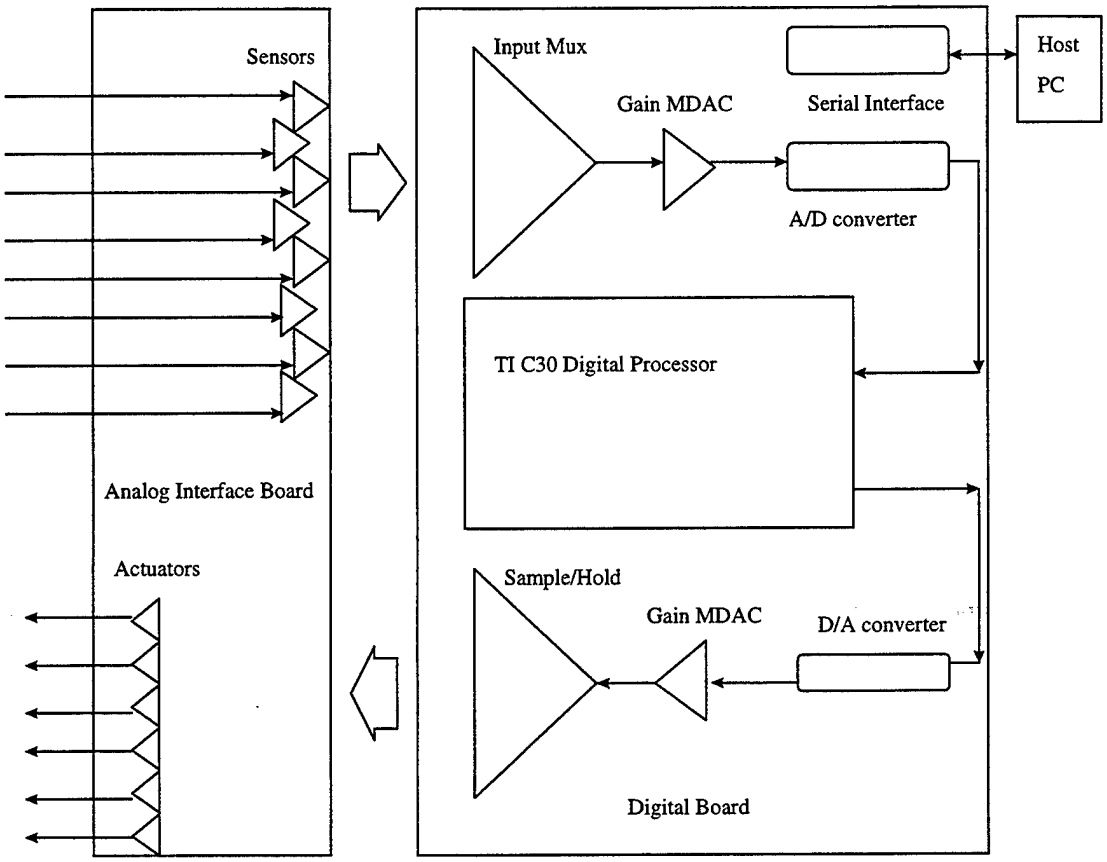


Figure 2.1 MCP Overview

2. MCP-III digital filter implementation

The MCP is designed to be used as a digital feedback controller. Analog sensor inputs are digitized, processed through an algorithm, converted back to analog and sent to the actuators. At the simplest levels, it is possible to realize 6 single input-single output (SISO) feedback controllers. Again, the analog interface for the MCP at the Naval Postgraduate School is capable of 8 inputs but only 6 outputs. The maximum number of channels feasible for the real time digital feedback by the MCP was not explored.

The MCP filters a signal by converting a digital input sequence, x_n, into an output sequence, y_n, by use of z-domain techniques to realize an equivalent s-domain transfer function. There are two types of digital filters, finite impulse response (FIR) and infinite

impulse response (IIR). Software written for C30 chip implements a selected transfer function primarily through the use of IIR digital filters.

IIR filters in general, involve fewer parameters, less memory requirement, and lower computational complexity than FIR filters. The transfer function of IIR filters has both poles and zeroes and all poles must be stable. Its output depends on both the input and past output. Although IIR filters involve less computation, they are sensitive to coefficient quantization. Most common implementation of IIR filters are as a cascade of second order filters call biquads.[Ref. 3] The recursion relation for the biquad form in terms of the input and output sequences appears as:

$$y_n = a_1 y_{n-1} + a_2 y_{n-2} + b_0 x_n + b_1 x_{n-1} + b_2 x_{n-2}. \tag{2.1}$$

The *a* coeffients times the previous outputs are the autoregressive terms, which are the poles of the transfer function. The b coefficients times current and previous inputs are the moving average terms, which define the zeros of the transfer function. It must be noted that this is Texas Instrument's definition of IIR as implemented on the C30 chip. In other textbooks, the autoregressive terms, $a_i y_{n-i}$, are subtracted from the moving average terms. TI adds the auto-regressive and moving average terms which simplifies the software.[Ref. 2] The resulting z-domain transfer function of the discrete biquad realized on the MCP is

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}} \tag{2.2}$$

Most transfer functions can be directly transformed from the s-domain to the z-domain using IIR filters.

Figure 2.2 shows a biquad recursion flow chart as implemented by software on the C30 chip. Each one of the biquads are implemented in the processor using an intermediate delay variable d_n. The first recursion relation defines the intermediate variable

$$d_n = x_n + a_1 d_{n-1} + a_2 d_{n-2}. \tag{2.3}$$

This relation is autoregressive, and supplies the poles. After computation of d_n for the current step, the second recursion relation provides the final output

$$y_n = b_0 d_n + b_1 d_{n-1} + b_2 d_{n-2}. \tag{2.4}$$

This relation is strictly a moving average and supplies the zeroes. The advantage of the intermediate delay is that only it and its delays need to be stored, rather than the delays of both the input and output variables.[Ref. 2]

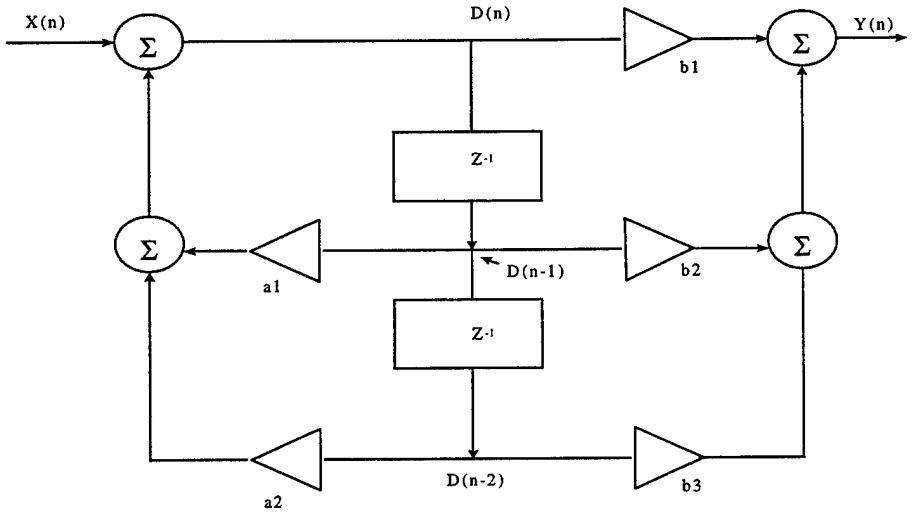


Figure 2.2 Biquad Filter Diagram

The MCP multi-channel interface currently implements three biquads in series. If less are needed the extra biquads are programmed to pass data with no gain or delay by setting b_0=1 and the other coefficients equal to 0. Overall gain is added to the filter by multiplying all of the b coefficients in the leading biquad by the desired gain magnitude.

The recursion shown in Figure 2.2 is implemented on the C30 chip in assembly language using hardware circular buffers. The software employs parallel multiply, add fetch, and increment operations. The delays are stored in a RAM circular buffer with 32

bit accuracy. The coefficients are stored in static RAM, also with 32 bits. The partial sums stay in the 40 bit accumulator registers. The input and output signals go through 12 bit converters.[Ref. 2]

C. SOFTWARE INTERFACE

The MCP-III employs two software programs to allow the user to realize a desired control program. The graphical user interface to allows implementation of several single input single output (SISO) transfer functions through the use of digital biquad filters. Six different transfer functions can be run simultaneously on six different channels. The second program uses state space methods to achieve true multiple input multiple output (MIMO) capability. It can utilize all 8 input channels to drive the 6 possible output channels. Both programs were developed by TRW and written in C software language. Both C programs are compiled into assembly level language by a TI C compiler. The compiler uses an optimization program that restructures the program to ensure maximum execution speed. A bootloader code is then used to download the C30 executable programs over a RS-232 port at a 19,200 baud rate. Appendix A contains step by step instructions on achieving a desired control law by either program.

1. Multi-channel Graphical User Interface

A multi-channel graphical user interface developed for the Windows 3.1/Windows 95 environment to ease implementation of SISO filters on the MCP is shown in Figure 2.3. After downloading an initialization program in a DOS environment, the user simply opens the interface, and sets up the intended filter. The first item to be selected should be the channel on which the control is to run. Then, the number of filters to implement on that particular channel is chosen. This is followed by selecting the actual filters to be implemented under the "first structure", "second structure", and "third structure" headings. The desired frequencies and Q also need to be entered. Up to three

filters in series can be achieved on a single channel. The overall gain for the channel, along with the sampling frequency, are the last items to be entered. Theoretically, there is no limit to the gain that can be selected, however output voltage on the analog board is limited to 150 V to avoid damaging the actuators. Individual channel gains can also be set by use of the MDAC. This is done by selecting the options button on the tool bar and then selecting gains. A gain range of up to 9 times the input can be realized this way. The MDAC gain varies linearly in 255 increments of 0.0353 each. Thus, to achieve a gain of 1, a code of 28 is entered for the appropriate channel. A code of zero will null the analog signal on that channel.

After all items are entered, the control program is started by simply depressing the run button and stopped by selecting the stop button.

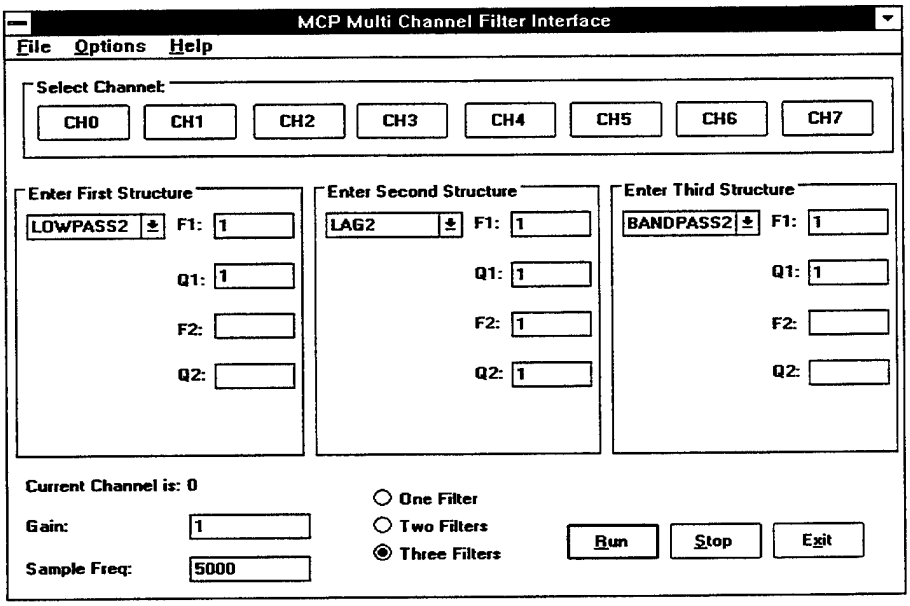


Figure 2.3 Multi-Channel Graphical User Interface

As mentioned above, the sampling frequency is selected by the user. The default is 5,000 sps. Maximum sampling frequency depends on the number of channels selected as depicted in the Table 2.1 below. The user must ensure that these limits are met otherwise the MCP will not function properly.

Number of Channels	Sample Rates
Channel 1	25.5 kHz
Channel 2	17.5 kHz
Channel 3	13.5 kHz
Channel 4	10.5 kHz
Channel 5	9.0 kHz
Channel 6	7.5 kHz
Channel 7	6.5 kHz
Channel 8	6.0 kHz

Table 2.1. Maximum Sampling Frequencies for the MCP

The sampling frequencies available are more than sufficient for MCP applications. The main focus is to dampen low frequencies usually under 50 Hz. Using the Nyquist criteria, the lowest potential sampling frequency would be 100 Hz. Even using all 8 input channels, the maximum sampling frequency is well above the Nyquist frequency.

There are several limitations to using the multi-channel interface. The biggest is the fact that only SISO can be realized thus limiting a designers options. Second is that filters can only be constructed in series and not parallel. This prevents the designer in trying to achieve more than one control scheme with one actuator/sensor pair.

The number of different types of filters that the multi-channel interface software is able to implement is a variety of one and two pole filters. The different types of first order filters that can be realized are listed in Table 2.2 and the second order filters in Tables 2.3 and 2.4.

Type	Lowpass	Allpass	Lead $\omega_p>\omega_z$ or Lag $\omega_p<\omega_z$	Highpass
poles	1	1	1	1
zeros	0	1	1	1
H(s)	1/(1+s/w)	(1-s/w)/(1+s/w)	$(1+s/w_z)/(1+s/w_p)$	

Table 2.2 MCP First Order Digital Filters

Type	Lowpass	Bandpass	Highpass
poles	2	2	2
zeros	0	1	2
H(s)	$\omega^2/(s^2+s\omega/Q+\omega^2)$	$s\omega/(s^2+s\omega/Q+\omega^2)$	$s^2/((s^2+s\omega/Q+\omega^2)$

Table 2.3 MCP Second Order Digital Filters

Type	Lead $\omega_p>\omega_z$ or Lag $\omega_p<\omega_z$	All Pass
poles	2	2
zeros	2	2
H(s)	$\omega_p^2(s^2+s\omega_z/Q_z+\omega_z^2)/\omega_z^2(s^2+s\omega_p/Q+\omega_p^2)$	$(s^2-s\omega/Q+\omega^2)/(s^2+s\omega/Q+\omega^2)$

Table 2.4 MCP Second Order Digital Filters (Phase Shifters)

The MCP second order filters listed in the above tables have poles and zeros defined in terms of frequency ω and amplification quality factor Q. Q is used frequently in filter engineering and is equal to $1/(2\varsigma)$. The lowpass, lead or lag, and allpass filters all have unity gain at DC(s=0). The highpass filter has unity gain at the Nyquist frequency except for the attenuation caused by the sample and hold. The bandpass filter has unity gain at the cutoff frequency.

2. State Space Controller

A state space realization is well suited for implementation on the MCP due to its strong floating point capability. It is especially useful because the control system designer often works within the state space framework in constructing control systems. Although state space is particularly suited with MIMO systems, it is not limited to that. A controller can be achieved with the MCP that uses MIMO, SISO, or both. This allows the designer more flexibility and options to achieve effective control on a number of applications that is not possible with multi-channel filter interface. The continuous time version of a state space system or equivalent state space controller is based on the following equations.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ \dot{y} &= Cx + Du \end{aligned} \tag{2.5}$$

As with the traditional state space notation, x is the state vector of size *1 x n*. The variable y is the system output, and u is the set of input signals of size *1 x m*. The variables *A, B, C and, D* represent the state space system matrices. The continuous time solution to this system is expressed in the following equations.

$$\begin{aligned} x(t) &= e^{A(t-t0)} \cdot x(t0) + \int_{t0}^{t} e^{A(t-\tau)} \cdot Bu(\tau)d\tau \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{2.6}$$

It is possible to obtain a discrete state-space representation by use of the following.

$$\begin{aligned} x_{n+1} &= A_d x_n + B_d u_n \\ y_n &= Cx_n + Du_n \end{aligned} \tag{2.7}$$

This is not dependent on the type of hold used but does assume that the input is constant over the sample interval. With this assumption, the continuous time solution is integrated over one sample period. The discrete time initial condition for each iteration is the state variable x_n and the state transition matrix A_d is given by

$$A_d = e^{AT} \tag{2.8}$$

where T is the sample period. The input matrix B_d is given by

$$B_d = \int_0^T e^{A\tau} \cdot B \cdot d\tau \tag{2.9}$$

The C and D matrices which are the output and feed through matrices are equivalent to the continuous time case.[Ref 2]

Appendix A contains the C language program for implementing the discrete time controller and instructions for realizing a state space controller. The *A, B, C, and D* matrices are generated in Matlab and then converted into C language format. The matrices can be constructed to accomplish any number of filters in series or parallel. The maximum number of states due to C30 limitations is 126. A Matlab program that constructs the state space matrices is also included in Appendix A.

To implement filters in series the two transfer functions must be convolved together. This can be easily accomplished in Matlab by multiplying the numerators and denominators of the two polynomials together. It is easiest to do this before converting to state space format. Any number of filters can be constructed in series by using this method.

For filters in parallel, it is most convenient, unlike the series case, to accomplish this after converting the transfer functions to the state space realization. Once in the state space format, the two systems are combined by using the Matlab "append" command. This will produce an aggregate state space system that is in fact a parallel realization of the two transfer functions. Just as in the series case, any number of filters can be constructed in parallel this way.

To set the matrices, for the desired number of inputs and output, the B, C, and D matrices must be manipulated. Just as in traditional state space notation, the B matrix is the input matrix and the number of columns this matrix has corresponds to the number of inputs. Thus, the B matrix must be manipulated to reflect the correct number of inputs. All that is required to achieve this is to collapse the matrix to the appropriate number of columns as shown below.

$$\begin{bmatrix} 0.4996 & 0 & 0 \\ 0.005 & 0 & 0 \\ 0 & 0.236 & 0 \\ 0 & 0.002 & 0 \\ 0 & 0 & 0.344 \\ 0 & 0 & 0.003 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.4996 \\ 0.005 \\ 0.236 \\ 0.002 \\ 0.344 \\ 0.003 \end{bmatrix} \tag{2.10}$$

The matrix will always be able to be collapsed in this fashion. The number of rows will vary depending on the number of states or poles in the transfer function. It also should be noted that the MCP will always correlate the first column of the *B* matrix to the input to channel one and the second column to channel two and so forth.

The *C* matrix corresponds to the output of the system. The number of outputs must equal the number of rows in the *C* matrix. Just as with the input matrix, the output matrix must be collapsed to the required number of rows as shown below.

$$\begin{bmatrix} .01 & -0.192 & 0 & 0 & 0 & 0 \\ 0 & 0 & .02 & .321 & 0 & 0 \\ 0 & 0 & 0 & 0 & .03 & .123 \end{bmatrix} \Rightarrow \begin{bmatrix} .01 & -0.192 & .02 & .321 & .03 & .123 \end{bmatrix} \tag{2.11}$$

Again, the *C* matrix can always be collapsed in this fashion. The MCP will also always correspond the first row of the *C* matrix to the output of channel 1 and the second row to channel 2 and so forth.

The *D* matrix will be a square matrix with entries only on the diagonal and the size related to the number of outputs. For example, a two output system would have a *D* matrix with 2 rows and 2 columns. The *D* matrix must be collapsed to match the number of columns of the *B* matrix and the number of rows of the *C* matrix. If the *B* matrix contains only one column and the *C* matrix is collapsed to one row, then the *D* matrix will be a *1x1* matrix with the diagonal entries summed together as shown below.

$$\begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.02 & 0 \\ 0 & 0 & 0.03 \end{bmatrix} \Rightarrow [0.06] \tag{2.12}$$

Typically, the actual numbers in the *D* matrix are very small compared to the *C* matrix.

Once the matrices have been constructed to reflect the number of actuators and sensors, the *C* language program must be adjusted to reflect the number of states, number of inputs, number of outputs, and desired sample frequency. This is done at the very top of the program as can be seen in Appendix A. Finally, this program must be altered to sample the appropriate number of channels. At the very bottom of the program just above the "while loop" is the command line

$$Gain[16] = 0x5 \tag{2.13}$$

The *0x5* on the right side of the equation must be changed to reflect the channels the MCP should sample. The *0x5* is a C language command to interpret the number 5 as hexadecimal. The compiler will then convert this in assembly language to a binary number. Each place value of the binary number corresponds to a channel on the MCP. For example, since the five is converted to 101, the MCP will sample channels 1 and 3. To sample channels 1, 2, 3, the user would change the *0x5* to *0x7*. Seven converted to binary is 111.

Once this step is completed, the program is ready to be compiled and then downloaded via the bootloader program.

D. PHYSICAL CONFIGURATION

A diagram showing the external configuration of the MCP-III is shown in Figure 2.4. Connectors to the analog interface are on the front panel. The back panel has the power connector and a serial port connector for communication to a computer. On the right side is a reset pin.

Interface with analog board is through J10 and J11 in the diagram. Both are nine pin connectors with one being a ground. The sensor input and actuator output signals are assumed to come from and go to separate analog electronics. Analog electronics along with the sensors, actuators, and structure it connects to constitute the "plant" in controls terminology. If any anti-alias filtering or additional gain is desired it must be placed on the analog board.[Ref 2]

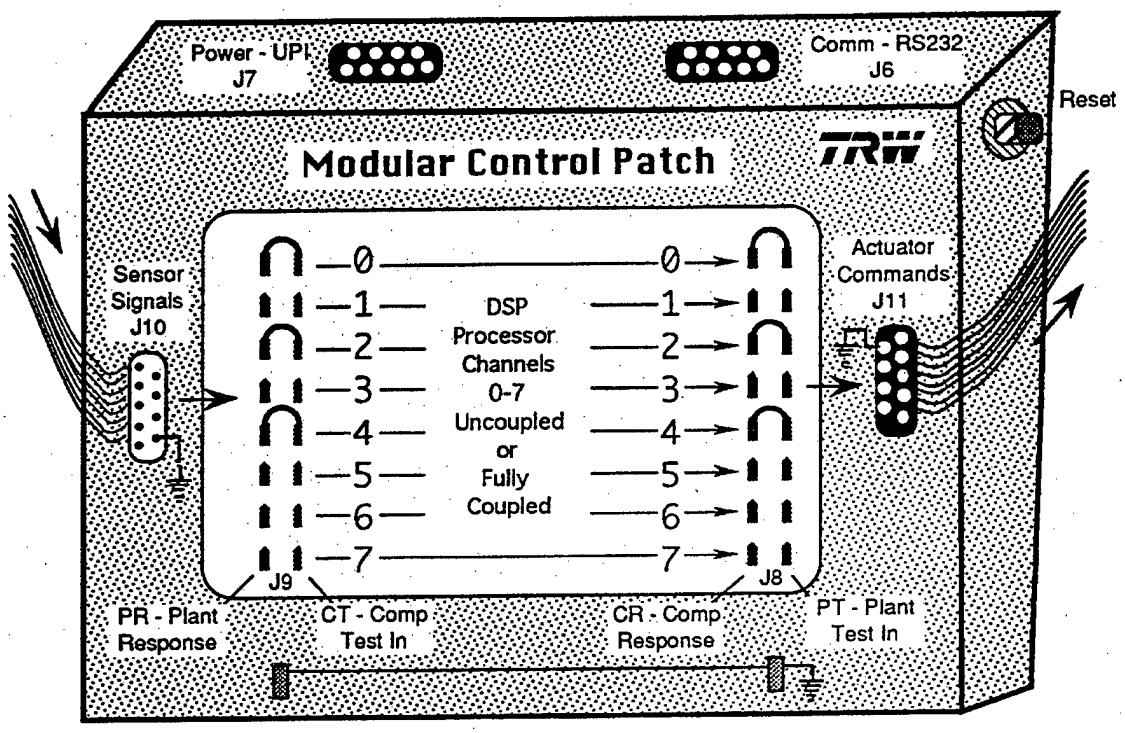


Figure 2.4 From Ref. [2], MCP External Configuration

The sensor signals from J10 are wired directly to the Plant Response(PR) pins on the front panel. To connect the sensor input to the MCP, a jumper must be placed to the corresponding compensator test in (CTin) pin. Similarly, the actuator command signals from J11 are wired to the Plant Test in(PTin) pins on the front panel. To input a test signal into an actuator simply connect to the PTin. To close the loop between a MCP

output channel and actuator input, connect the Compensator Response(CR) pin and its corresponding PTin pin. The compensator output signal can still be monitored with the jumper attached. Do not, however, attempt to drive PTin with the jumper connected to CR since this will back drive the MCP D/A sample and hold output.

The compensator is composed of the digital control hardware and implements the transfer function downloaded from the computer. The input mux is directly attached to the Ctin pins. The output sample and hold signals are wired directly to the CR pins. To close the loop for the entire system, both pin sets must be connected by jumper. This means PR is connected to CTin and CR is connected to PTin.

A summary of tests that can be performed is given below in Table 2.5.

Test type	Procedure
Open Loop Plant Transfer Function	1. Input signal to Ptin and measure CTin. 2. Ensure CR and Ptin are open.
Open Loop Compensator Transfer Function	1. Input signal to Ctin and measure CR. 2. Ensure PR and Ctin open.
Plant x Compensator Transfer Function	1. Input signal to Ptin and measure CR. 2. Ensure PR is connected to CTin and that CR and PTin are open.
Compensator x Plant Transfer Function	1. Input signal to Ctin and measure PR. 2. Ensure CR is connected to PTin and that PR and CTin are open.
Closed Loop Response	1. Measure closed loop sensor at PR and compensator response at CR 2. Ensure PR connected to CTin and that CR is connected to Ptin.

Table 2.5. Summary of System Tests

The reset button on the right panel of the MCP is used to clear any program still resident on the C30 before downloading the desired software. The software is delivered

from the computer through a standard nine pin RS232 serial port and is designated J6 on the MCP external diagram.

Power supply requirements for the MCP and the analog board are shown in Table 2.6. The MCP and the analog board are connected via the Universal Power Interface(UPI) format. The MCP power connector (J7) is a 9-pin female D-connector. A listing of the pin assignments for the MCP and analog board is in Appendix A.

The main purpose of the analog interface board is to condition the input and output signals for the MCP. The input signal needs to ±15 V. This puts the signal in an acceptable range for sample and hold device. The board uses charge amps to process incoming signals On the output side, the signal must be boosted to a maximum of 150 V in order to drive the actuators. This is accomplished through the use of op amps for each channel.

Voltage	MCP	Analog Board	Current
+5 V	Required	Not required	Nominal 0.5 A
+15 V	Required	Required	Nominal 150 mA (MCP) Maximum 100 mA/channel (Analog Board)
-15 V	Required	Required	same as +15 V
+150 V	Not Required	Required	Maximum 50 mA/channel
-150 V	Not Required	Required	Maximum 50 mA/channel

Table 2.6 MCP and Analog Interface Board Power Requirements

E. ELECTRONIC PERFORMANCE

1. Noise and Harmonic Distortion

Overall, the performance of the MCP-III is excellent, particularly considering that multi inputs and outputs are being processed through a single data path. As mentioned

earlier, the resolution of the system is limited to 12 bits due to the limit of rad-hard converters. A 12 bit converter has one sign bit and 11 amplitude bits, which gives a theoretically possible dynamic range of 2^{11}or 66dB. Measured signal to noise ratio and total harmonic distortion for the MCP are at this level. Figure 3.4 shows the measured power spectral density at the output of the MCP-III with MDAC gains set at nearly 1, and all analog inputs grounded. The sample rate used was 8,000 sps. The noise floor below 1,000 Hz , even at harmonics of 60 Hz, is far below a milli-Volt. Given the 10 Volt range, the in-band noise floor is over four orders of magnitude below the maximum signal level, or greater than -80dB. When viewed at frequencies near the Nyquist frequency there were a few milli-Volts of noise at the sample rate, which also aliased down to the first six sub-harmonics. Output smoothing filters will eliminate this sampling noise.[Ref 2]

Signal to noise ratio and total harmonic distortion are illustrated in Figure 2.5. The test was performed by providing a 5 Volt, 17.5 Hz sinusoid into an input channel and monitoring power spectral density at the output channel. MDAC gains were again approximately 1. As seen in the figure, the 5V fundamental is transmitted as well as 1.8 mV of harmonic distortion at the 35 Hz second harmonic. Total harmonic distortion is 68 dB below the 5 V signal. The minimum discernible voltage with the 11 bit resolution and 10 V range should be 10/2047=0.005V. Therefore, harmonic distortion and noise are well below the theoretical digital noise floor. It was found experimentally that using the MDAC at full gain allows 72 dB signal to noise and harmonic distortion ratio.[Ref. 2]

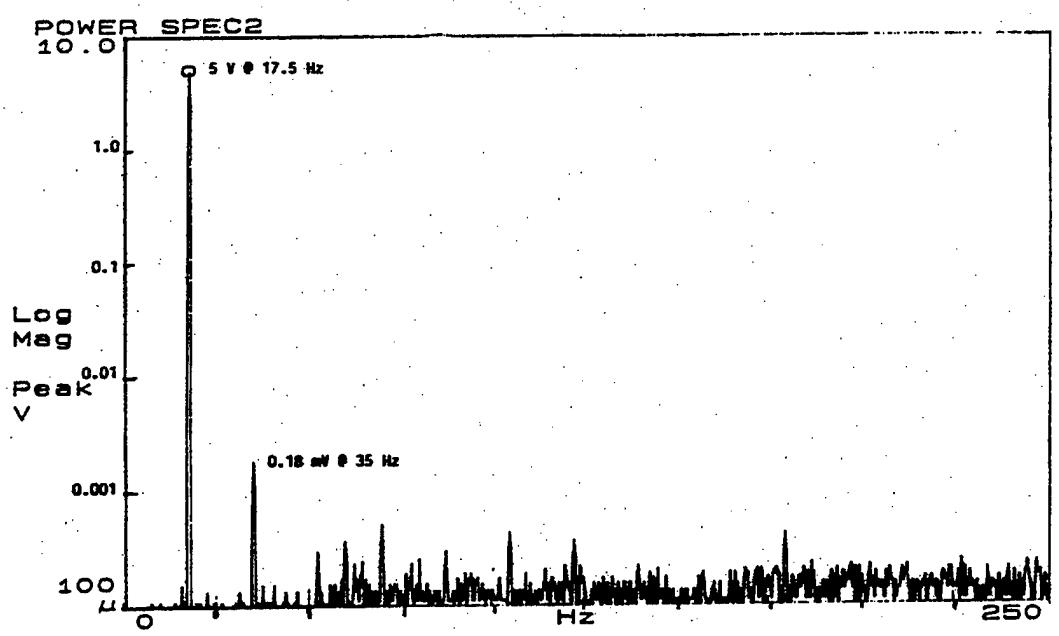


Figure 2.5 From Ref. [2], MCP Signal to Noise and Harmonic Distortion

2. Digital Filter Realization

The true test for the MCP as a digital controller is how accurately it can realize a desired transfer function. Of course, there are a number of factors that will affect how well the MCP can do this with sampling frequency being one of the more important ones. Figure 2.6 shows the Bode plot of a PPF transfer function calculated in Matlab as well as a bode plot of the PPF filter implemented by the MCP. The Bode plot of the MCP realization was generated by recording the input and output of the MCP while running a PPF control law. Comparison of the two plots shows that the MCP realizes the control law reasonably well. The MCP was set at 5,000 sps although the data recording equipment was only sampling at 100 sps. This gives the MCP bode plot a slightly jagged look and the higher frequency data is unreliable as the phase starts to roll off. There is a slight phase shift incurred due to the sample and processing time for the MCP. The delay is about 1.5 sample periods[Ref. 2]. The zero order hold device will add a half sample period and the rest of the MCP components add one sample period delay.

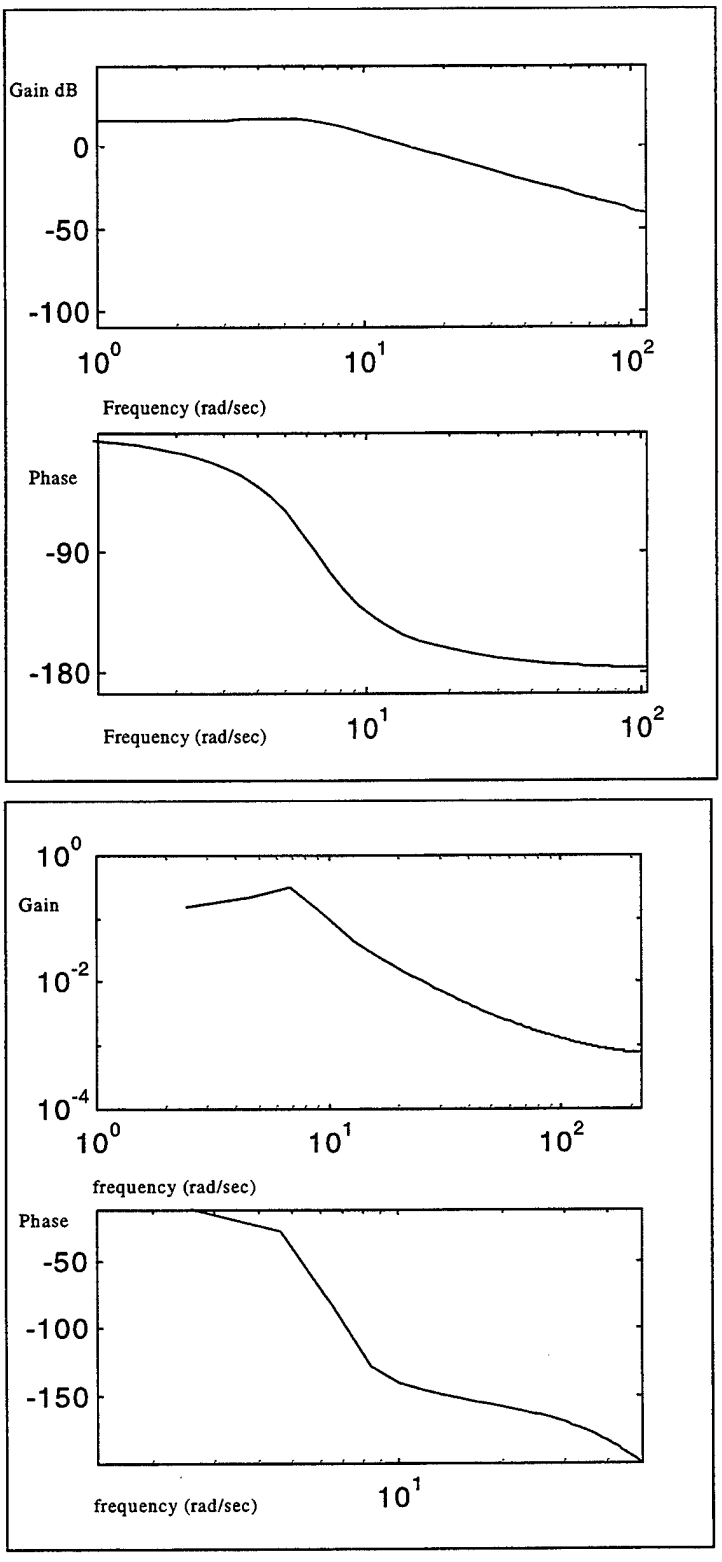


Figure 2.6 Theoretical Bode Plot (top) vs MCP Bode Plot (bottom) for PPF Control Law.

III. PIEZOCERAMICS

A. BACKGROUND

Piezoceramics are part of a class of materials which demonstrate piezoelectric properties. The main characteristic of which is the ability to develop an electrical charge when subject to mechanical strain. The charge is directly proportional to the amount of strain and the direction of the electric field or current flow depends on whether the material is under compression or tension. Conversely, an applied electric voltage will produce a mechanical deformation proportional to the amount of voltage applied. Thus, piezoceramics make excellent electromechanical transducers.

Piezoelectricity occurs naturally in some crystalline materials and can be induced other polycrystalline materials such as ceramics by a process known as "poling". The materials crystal lattice structure may be poled by applying a large electric field while heated to a temperature above its Curie point[Ref. 4]. This realigns the crystalline structure such that it becomes elongated in direction of the electric field as shown in Figure 3.1. This becomes the "poling axis" of the material. Subsequent application of an electric field in the poling direction will lengthen the material in that direction while it shrinks laterally. Conversely, an electric field applied opposite the poling direction will contract the material along the poling axis while expanding it laterally. This occurs roughly according to Poisson's ratio. Also, the poling axis is normally designated as the 3-axis with the 1-axis and 2-axes being orthogonal to the poling axis as illustrated in Figure 3.1.

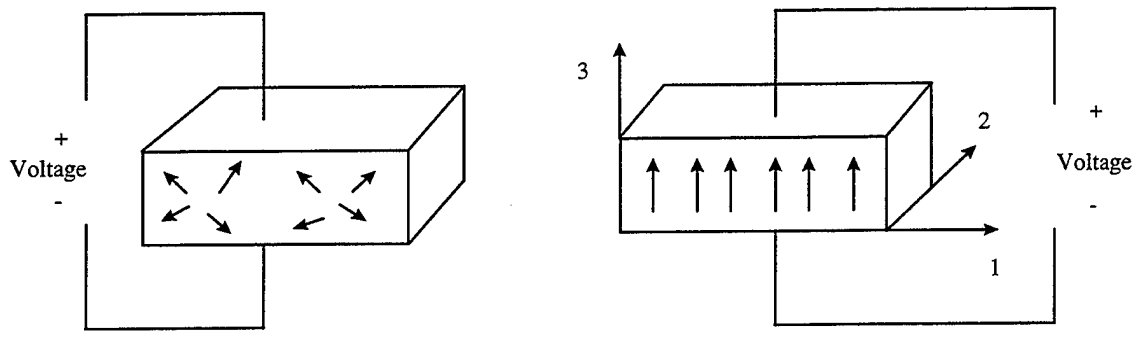


Figure 3.1 Piezoceramic Poling Direction

Barium titanate, discovered during the 1940's became the first widely used piezoceramic. This was followed by a solution of lead zirconate and lead zirconate titanate (PZT) which was discovered in 1954 and has become more widely used than its predecessor due to its stronger piezoelectric effects. The application of piezoceramics as a solution to low frequency structural control did not occur until the late 1980's. Piezoceramics are a hard dense material that can be manufactured in almost any given shape or size and thus tailored to many different applications. They are chemically inert and immune to moisture and other atmospheric conditions. Also, the near linear response, adequate stiffness, and low electrical power consumption make them very desirable for use on spacecraft structures.

B. PIEZOELECTRIC PROPERTIES

Piezoceramics are usually transversely isotropic materials meaning that the material properties are the same in all directions in the plane orthogonal to the poling axis. Also, the dielectric constant of piezoceramics are relatively high.

When used as a sensor, piezoceramics produce a charge when under lateral strain as given by

$$Q = AEd_{31}(\varepsilon_1 + \varepsilon_2) \tag{3.1}$$

where A is the lateral area of the piezoceramic, E is Young's modulus, d_{31} is the lateral strain coefficient, and ε_1, ε_2 represent the strain in the lateral strain directions. The capacitance generated is equal to

$$C = DA / t \tag{3.2}$$

where D is the dielectric constant or permittivity and t is the thickness of the piezoceramic sensor. The output voltage is then given by

$$V = \frac{Q}{C} = \frac{Ed_{31}}{D} t(\varepsilon_1 + \varepsilon_2) \tag{3.3}$$

Thus, a piezoceramic sensor can detect a mechanical strain and convert it to a voltage which can easily be measured and processed as desired.

When used as an actuator, an applied voltage is used to generate a moment to counteract a vibration. The applied voltage produces an electric field across the actuator given by

$$\phi = \frac{V}{t} \tag{3.4}$$

Lateral strain is produced from the applied voltage from the relation

$$\varepsilon = d_{31}\phi = d_{31}\frac{V}{t} \tag{3.5}$$

By using Young's modulus, the corresponding stress can be found and then the resultant force. The force is given by the following equation

$$F = wEd_{31}V \tag{3.6}$$

where w is the width of the actuator. The moment generated is then

$$M = wEd_{31}(\frac{t}{2} + \frac{t_b}{2})V \tag{3.7}$$

where t_b is the thickness of the beam to where the actuator is attached. Actuator strains on the order of 1,000 μstrain have been produced with piezoceramics. Strains are non-dimensional ratios of the change in the length to the original length. It must be noted that the poling direction of the piezoceramic can be damaged by the application of a strong electric field. Limits for most materials are between 500 and 1000 volts/mm.[Ref. 5]

IV. EXPERIMENTAL SETUP

The experiment was essentially conducted in two parts. The first part consisted of using the MCP implementation of a control law to provide active damping on a single modal frequency of the aluminum beam. The second part attempted to use the same control laws to achieve multi-mode damping. PPF, SRF, and Integral control were all tested independently and in conjunction with another in order to maximize damping effectiveness.

Additionally, in both parts of the experiment, a robustness analysis of PPF control was conducted by varying the compensator damping and by varying the compensator frequency. Damping effectiveness by these variations was studied with a single PPF filter targeting a single mode and then two PPF filters in parallel targeting different modes.

A. PHYSICAL CONFIGURATION

A schematic of the equipment setup for the aluminum beam is shown in Figure 4.1. It is a single degree of freedom system with the beam clamped such that its length was parallel to the granite table below it. This allowed the bending to be strictly in the horizontal plane. Although it can not be seen in the schematic, the piezoceramic patches were placed on both sides the beam. A picture of the aluminum beam with the MCP, analog interface, and low voltage power supply is shown in Figure 4.2 and the entire experiment setup is shown in Figure 4.3.

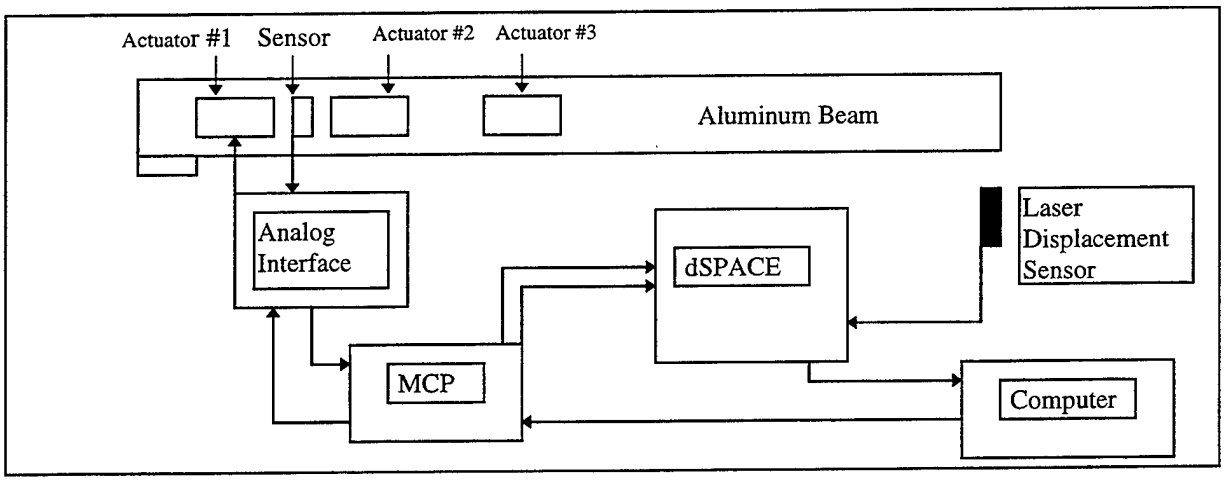


Figure 4.1. Experimental Setup Schematic

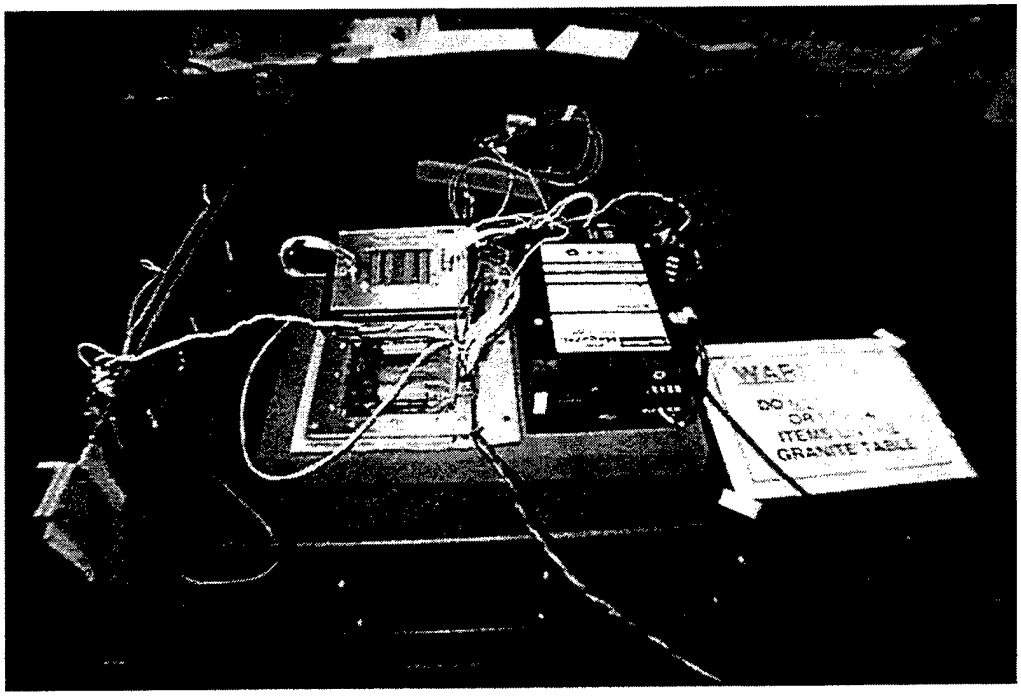


Figure 4.2 Aluminum Beam, MCP, Analog Interface, and MCP Power Supply

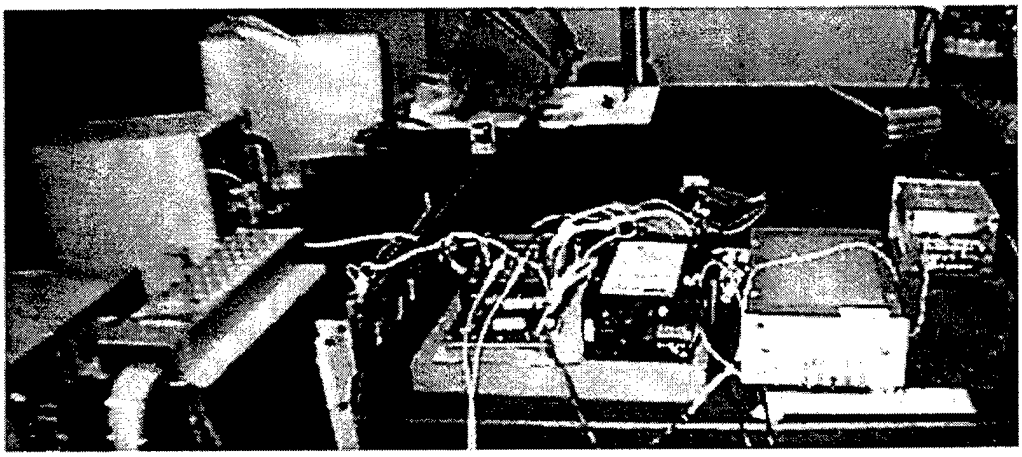


Figure 4.3 Complete Experimental Setup(From left to right, dSpace Computer and DS2003 board, Aluminum Beam, MCP, Analog Interface, MCP Power Supply, and Analog Interface Low and High Voltage Power Supplies)

A dSPACE digital signal processor was used to record the experimental data. The processor incorporates a TMS320C40 digital signal processing chip very similar to the one utilized in the MCP. It is designed to work with multiple types of interface boards allowing it to be extremely versatile. It has the potential to perform the same function as the MCP although its size precludes it from being considered for use in a space platform.

Utilizing a DS2003 MUX/AD board, dSPACE can convert up to 32 analog inputs and outputs to digital signals for processing. The dSPACE interfaces with Matlab and Simulink which allows for convenient post-experiment data processing. The Real-Time Trace Module for Matlab is a windows based dSPACE graphical user interface and was used for all testing. This permitted the saving all data in a Matlab .MAT format for processing and plotting. Matlab programs were written to identify the modal frequencies, calculate damping on each mode, and then plot the results along with the numerical data. A copy of the Matlab programs used for data reduction is included in Appendix B.

Three inputs were provided to dSPACE to be recorded. The sensor input to the MCP as well as the MCP output were the first two signals provided. These were essentially the input and output of the compensator. The third input was the beam's tip displacement which was provided by an NAiS ANL1651AC infrared laser analog displacement sensor. The laser provides an output of 0.1 volts per millimeter and has a dynamic range that is adjustable of up to 1 kHZ. It was typically set at 100 Hz for the

beam experiments. This was more than sufficient as the first two modes were of primary interest and both of those are below 10 Hz.

B. EXPERIMENTAL PROCEDURE

The objective of the experiment was to investigate the effectiveness and suitability of the MCP digital controller in achieving active damping on the beam by implementing several control algorithms. A second objective was test the robustness of the PPF control law. Both open and closed loop tests were performed. Each test conducted was started by manually exciting the beam. This was a simple but very effective method to vibrate the beam. The first mode could be excited exclusively or a multiple mode excitation could be accomplished. Both types of excitations were used.

Initially, tests were run with all three actuators operational. This worked well when trying to dampen the first mode only. However, for higher modes, the actuator placement was found to adversely impact damping effectiveness. In some cases, it excited the higher modes. Because of this, the majority of the tests were conducted with only one actuator and one sensor. The other two being disconnected.

Data was obtained for a time interval of 15 seconds after beam excitation. This allowed ample time to measure any damping effects. The experimental results were processed to provide data in several different formats. Most importantly, a Fast Fourier Transform (FFT) was performed in Matlab to provide a power spectral density (PSD) of the beam response. PSD gives a measure of signal energy at different frequencies. A comparison of the ratios of the final PSD to the initial PSD provides an excellent indication of the damping performed on each particular mode. Also, when referenced to an open loop excitation of the beam, a direct comparison can be made to the effectiveness of the each control algorithm. Figure 4.4 shows the PSD for a multi-mode open loop vibration. The solid line is the first second of the test with the bottom line representing the last second. A Matlab program was written to select the modes excited and compute the difference between the initial and final PSD in dB at the desired frequencies. Table 4.1 shows the natural damping of the first four modes in dB after 15 seconds.

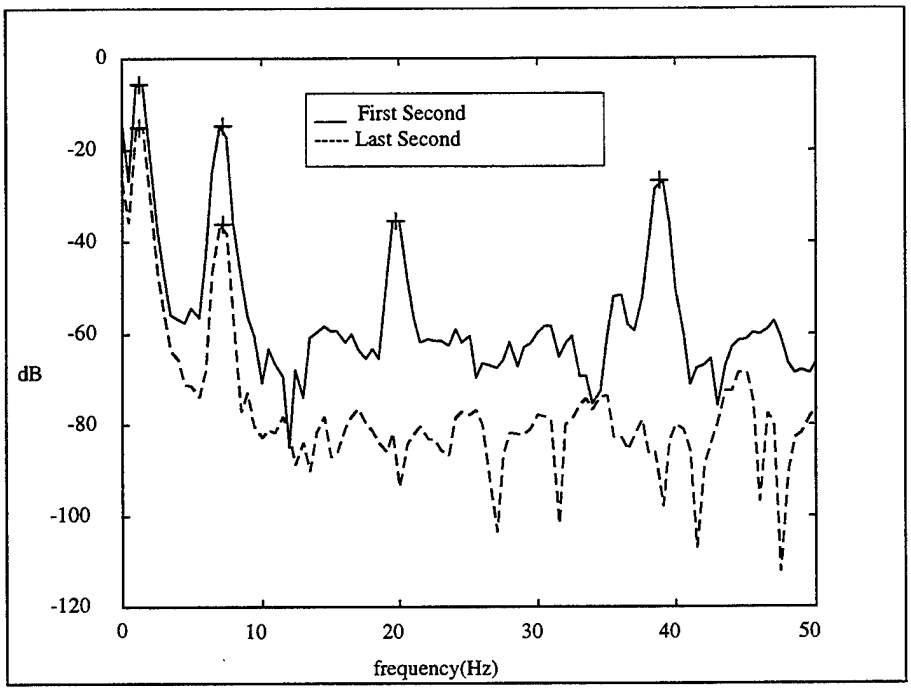


Figure 4.4. Open Loop Vibration of Aluminum Beam

Mode #	First Mode 1.33 Hz	Second Mode 7.1 Hz	Third Mode 19.0 Hz	Fourth Mode 38.2 Hz
dB drop in 15 seconds	9.52 dB	22.38 dB	48.98dB	61.94 dB

Table 4.1 Natural Damping of Aluminum Beam

Another data processing technique is to measure the damping ratio from amplitude measurements of the PZT sensor or laser signal. The amplitude at a given time was divided by the initial amplitude at $t = 0$. The limitation to this procedure is that for multi-mode excitation it is impossible to determine the damping effectiveness on any particular mode. Only when the first mode was predominantly excited could a single mode be analyzed.

Lastly, the data was used to generate the actual transfer function being implemented as was seen in Chapter II. This helped verify that the actual control law was the same as the intended one and gave a performance measure as to how well the MCP could realize a given control algorithm.

V. FINITE ELEMENT MODEL

A. BACKGROUND

Modern aerospace structures consist of many structural elements and shapes that contain a variety of materials including metals and composites. These structures tend to be analytically indeterminate and traditional structural analysis methods tend not to be practical for these types of problems. The finite element stiffness method was developed in the late 1950's as a result of the need for a systematic analysis procedure for complex structures that could be implemented on a digital computer. Conceptually, the finite element method breaks down the structure into a finite number of discrete elements. With the entire structure broken down into elements, each element is analyzed separately for equilibrium. The structure is then tied back together by imposing compatibility or equilibrium requirements at the boundaries or nodes where the elements are connected.

B. ELEMENT STIFFNESS MATRIX

The direct assembly approach is used to construct the mass and stiffness matrices.[Ref. 5] The stiffness matrix relates a displacement vector to a force vector. The differential equation for the displacement $w(x)$ of a finite element is

$$EI\frac{d^4w(x)}{dx^4}=0 \qquad 0<x<h \tag{5.1}$$

where E is Young's modulus for the material and I is the area moment of inertia for the beam cross-section.

The equation for element displacement at any point is achieved by integrating equation 5.1 four times. This results in the following equation

$$w(x)=\frac{1}{6}c_1x^3+\frac{1}{2}c_2x^2+c_3x+c_4 \tag{5.2}$$

where c_1 through c_4 are constants of integration and can be determined by applying conditions as follows

$$w(0) = w1 \qquad \left.\frac{dw(x)}{dx}\right|_{x=0} = \theta_1 \qquad w(h) = w_2 \qquad \left.\frac{dw(x)}{dx}\right|_{x=h} = \theta_2 \tag{5.3}$$

From equation 5.2 and 5.3 the constants of integration can be found and are given by

$$c_1 = \frac{6}{h^3}(2w_1 + h\theta_1 - 2w_2 + h\theta_2)$$

$$c_2 = \frac{2}{h^2}(-3w_1 - 2h\theta_1 + 3w_2 - h\theta_2) \tag{5.4}$$

$$c_3 = \theta_1$$

$$c_4 = w_1$$

Substituting equation 5.4 into 5.2, the expression for the element bending displacement is

$$w(x) = \left[1 - 3\left(\frac{x}{h}\right)^2 + 2\left(\frac{x}{h}\right)^3\right]w_1 + \left[\left(\frac{x}{h}\right) - 2\left(\frac{x}{h}\right)^2 + \left(\frac{x}{h}\right)^3\right]h\theta_1 \tag{5.5}$$

$$+\left[3\left(\frac{x}{h}\right)^2 - 2\left(\frac{x}{h}\right)^3\right]w_2 + \left[-\left(\frac{x}{h}\right)^2 + \left(\frac{x}{h}\right)^3\right]h\theta_2$$

The bending displacement can now be related to the nodal forces for adjacent elements

$$EI\left.\frac{d^3w(x)}{dx^3}\right|_{x=0} = Q_1 \qquad EI\left.\frac{d^2w(x)}{dx^2}\right|_{x=0} = -M_1 \tag{5.6}$$

$$EI\left.\frac{d^3w(x)}{dx^3}\right|_{x=h} = -Q_2 \qquad EI\left.\frac{d^2w(x)}{dx^2}\right|_{x=h} = M_2$$

where Q_1, Q_2 are shear forces and M_1, M_2 are moment forces. Combining equations 5.4 and 5.5 yields

$$Q_1 = \frac{EI}{h^3}\left(12w_1 + 6h\theta_1 - 12w_2 + 6h\theta_2\right)$$

$$M_1 = \frac{EI}{h^3}\left(6w_1 + 4h\theta_1 - 6w_2 + 2h\theta_2\right) \tag{5.7}$$

$$Q_2 = \frac{EI}{h^3}\left(-12w_1 + 6h\theta_1 + 12w_2 - 6h\theta_2\right)$$

$$M_2 = \frac{EI}{h^3}\left(6w_1 + 2h\theta_1 - 6w_2 + 4h\theta_2\right)$$

Equation 5.7 can be written in matrix form as

$$[k]\{q\} = \{F\} \text{ where } \{q\} = \begin{Bmatrix} w_1 \\ \theta_1 \\ w_2 \\ \theta_2 \end{Bmatrix} \text{ and } \{f\} = \begin{Bmatrix} Q_1 \\ M_1 \\ Q_2 \\ M_2 \end{Bmatrix} \tag{5.8}$$

and the element stiffness matrix is given by

$$[k] = \frac{EI}{h^3}\begin{bmatrix} 12 & 6h & -12 & 6h \\ 6h & 4h^2 & -6h & 2h^2 \\ -12 & -6h^2 & 12 & -6h \\ 6h & 2h^2 & -6h & -4h^2 \end{bmatrix} \tag{5.9}$$

C. ELEMENT MASS MATRIX

The equation for bending displacement can also be expressed as a function of nodal displacements and interpolation function *L(x)*

$$w(x) = L_1(x)w_1 + L_2(x)\theta_1 + L_3 w_2 + L_4(x)\theta_2 \tag{5.10}$$

where

$$L_1(x) = 1 - 3\left(\frac{x}{h}\right)^2 + 2\left(\frac{x}{h}\right)^3 \qquad L_2(x) = x - 2h\left(\frac{x}{h}\right)^2 + h\left(\frac{x}{h}\right)^3 \tag{5.11}$$

$$L_3(x) = 3\left(\frac{x}{h}\right)^2 - 2\left(\frac{x}{h}\right)^3 \qquad L_4(x) = -h\left(\frac{x}{h}\right)^2 + h\left(\frac{x}{h}\right)^3 \tag{5.12}$$

In matrix form

$$w = \{L(x)\}^T\{q\} \tag{5.13}$$

L(x) is a four dimensional vector of the interpolation functions and *x(t)* a *4x1* vector of nodal displacements. The element kinetic energy has the form

$$T(t) = \frac{1}{2}\int_0^h m(x)\left[\frac{\partial^2 w(x,t)}{\partial t^2}\right]^2 dx = \frac{1}{2}\left\{\dot{q}\right\}^T [m]\left\{\dot{q}\right\} \tag{5.14}$$

where [m] is the mass matrix and is equal to

$$[m] = \int_0^h m(x)\{L(x)\}\{L(x)\}^T dx \tag{5.15}$$

Substituting in the interpolation function yields the element mass matrix

$$[m] = \frac{mh}{420}\begin{bmatrix} 156 & 22h & 54 & -13h \\ 22h & 4h^2 & 13h & -3h^2 \\ 54 & 13h & 156 & -22h \\ -13h & -3h^2 & -22h & 4h^2 \end{bmatrix} \tag{5.16}$$

D. GLOBAL MATRICES BY THE DIRECT METHOD

With each elements mass and stiffness matrices obtained, the next step is to construct the global mass and stiffness matrices for the entire structure. The matrices are simply appended along the diagonal. Common node points between two elements are simply added together. An example for a *2x2* element case is shown below

$$k^1 = \begin{bmatrix} k_{11}^1 & k_{12}^1 \\ k_{21}^1 & k_{22}^1 \end{bmatrix} \qquad k^2 = \begin{bmatrix} k_{11}^2 & k_{12}^2 \\ k_{21}^2 & k_{22}^2 \end{bmatrix} \qquad k^{total} = \begin{bmatrix} k_{11}^1 & k_{12}^1 & 0 \\ k_{21}^1 & k_{22}^1 + k_{11}^2 & k_{12}^2 \\ 0 & k_{21}^2 & k_{22}^2 \end{bmatrix} \tag{5.17}$$

This method is the same for both the stiffness and mass matrices. If boundary conditions require a displacement or rotation to be zero, that row and column corresponding to the displacement or rotation is eliminated. For example, a cantilevered beam has boundary conditions at the root that make w_1 and θ_1 zero. Thus, the first and second rows and columns are eliminated from the global mass and stiffness matrices.

E. FINITE ELEMENT MODEL FOR PIEZOCERAMICS

The piezoceramic elements used in the structure must be incorporated into the finite element model. The basic equations for both piezoceramic actuators and sensors are the same as for ordinary structural elements discussed previously. The distance the piezoceramic is placed from the center of the beam must be taken into account.

To model the constitutive behavior of piezoceramics, the electromechanical coupling effect is used. Piezoceramic material satisfy the following linear equations:

$$\begin{Bmatrix} D_3 \\ S_1 \end{Bmatrix} = \begin{bmatrix} \varepsilon_3^T & d_{31} \\ d_{31} & s_{11}^E \end{bmatrix} \begin{Bmatrix} E_3 \\ T_1 \end{Bmatrix} \tag{5.18}$$

where D represents the electric displacement with units of charge per unit area, E_3 represents the applied field density, s_i represents the strain, T_1 is the stress, ε^T_3 is the permitivity of the piezoelectric material, d_{31} is the piezoelectric charge coefficient, and $S^e_{ii} = 1/E_p$ is the elastic constant for the piezoelectric material. The equations can also be written as

$$\begin{Bmatrix} D_3 \\ S_1 \end{Bmatrix} = \begin{bmatrix} \varepsilon_3^T - d_{31}^2 E_p & d_{31} E_p \\ -d_{31} E_p & E_p \end{bmatrix} \begin{Bmatrix} E_3 \\ T_1 \end{Bmatrix} \tag{5.19}$$

Now the potential energy of the element is expressed as

$$-U = \frac{1}{2}\int_V \left(-T_1 S_1 + D_3 E_3\right) dV \tag{5.20}$$

where $-T_1S_1$ is the mechanical potential energy and D_3E_3 the electrical potential energy. This expression can be rewritten if the width (w_p) is held constant and pulled out of the integration as shown below

$$\begin{aligned} -U &= \frac{1}{2} w_p \int_0^h \int_\zeta^{\zeta+t_p} (-T_1 S_1 + D_3 E_3) dx dz \\ &= \frac{1}{2} w_p \int_0^h \int_\zeta^{\zeta+t_p} \begin{Bmatrix} D_3 \\ T_1 \end{Bmatrix}^T \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{Bmatrix} E_3 \\ S_1 \end{Bmatrix} dx dz \end{aligned} \tag{5.21}$$

The strain, S_1, can be written using small angle displacement theory as

$$S_1 = \varepsilon_x = -z\frac{\partial^2 w}{\partial x^2} \tag{5.22}$$

substituting equation (5.19) into equation (5.21)

$$\begin{aligned} -U &= \frac{1}{2} w_p \int_0^h \int_\zeta^{\zeta+t_p} \begin{Bmatrix} E_3 \\ \varepsilon_x \end{Bmatrix}^T \begin{bmatrix} \varepsilon_3^T - d_{31}{}^2 E_p & d_{31} E_p \\ d_{31} E_p & -E_p \end{bmatrix} \begin{Bmatrix} E_3 \\ \varepsilon_x \end{Bmatrix} dx dz \\ &= \frac{1}{2} w_p \int_0^h \int_\zeta^{\zeta+t_p} \left[(\varepsilon_3^T - d_{31}{}^2 E_p) E_3{}^2 + 2 d_{31} E_p E_3 \varepsilon_x - E_p \varepsilon_x{}^2 \right] dx dz \end{aligned} \tag{5.23}$$

now using equation (5.22)

$$-U=\frac{1}{2}w_p\int_0^h\int_\zeta^{\zeta+t_p}\left[(\varepsilon_3^T-d_{31}{}^2E_p)E_3{}^2+2d_{31}E_pE_3z\frac{\partial^2 w}{\partial x^2}-E_pz^2\left(\frac{\partial^2 w}{\partial x^2}\right)^2\right]dxdz \tag{5.24}$$

Rewriting equation (5.13) in terms of a summation

$$w(x,t)=\sum_{i=1}^{4}\Phi_i(x)q_i(t) \tag{5.25}$$

where Φ is the vector of interpolation functions or "mode shapes" and q is the nodal vector from equation. Inserting this equation along with the interpolation functions, the general form of the energy equation is

$$-U=\frac{1}{2}ye^2-q^Tbe-\frac{1}{2}q^Tk_pq \tag{5.26}$$

where

$$\begin{aligned}
y&=\frac{w_ph}{t_p}\left(\varepsilon_3^T-d_{31}{}^2E_p\right),\quad e=t_pE_3\\
b_i&=d_{31}E_pw_p\left(\zeta+\frac{t_p}{2}\right)\int_0^h\frac{d^2\Phi_i(x)}{dx^2}dx\\
\left[k_p\right]_{ij}&=w_pE_pt_p\left[\zeta^2+\varsigma\cdot t_p+\frac{t_p{}^2}{3}\right]\int\frac{d^2\Phi_i(x)}{dx^2}\frac{d^2\Phi_j(x)}{dx^2}dx
\end{aligned} \tag{5.27}$$

Substituting the interpolation functions from equations (5.11 and 5.12) into the b vector, yields

$$\begin{aligned}
b_1&=0\\
b_2&=-d_{31}E_pw_p\left(\zeta+\frac{t_p}{2}\right)\\
b_3&=0\\
b_4&=d_{31}E_pw_p\left(\zeta+\frac{t_p}{2}\right)
\end{aligned} \tag{5.28}$$

and into the piezoceramic elemental stiffness matrix, $[k_p]$

$$[k_p] = \frac{\kappa}{h}\begin{bmatrix} \frac{12}{h^2} & \frac{6}{h} & -\frac{12}{h^2} & \frac{6}{h} \\ \frac{6}{h} & 4 & -\frac{6}{h} & 2 \\ -\frac{12}{h^2} & -\frac{6}{h} & \frac{12}{h^2} & -\frac{6}{h} \\ \frac{6}{h} & 2 & -\frac{6}{h} & 4 \end{bmatrix}$$

$$\kappa = w_p t_p E_p (\zeta^2 + \zeta t_p + \frac{t_p^{\,2}}{3}) \tag{5.29}$$

The piezoceramic elemental mass matrix is derived in the same fashion as an ordinary structural element with the exception that the piezoceramic material linear mass density be used instead of the ordinary material density of the structure to which it is bonded.

For structural elements that have piezoceramic material bonded to them, their respective mass and stiffness matrices are given by the simple addition of the beam elemental matrices and the piezoceramic elemental matrices which are then assembled into global mass and stiffness matrices as in the previous section.

$$[M]_{element} = [m]_{beam} + [m]_{piezo}$$

$$[K]_{element} = [k]_{beam} + [k]_{piezo} \tag{5.30}$$

F. FINITE ELEMENT MODEL OF FLEXIBLE BEAM

The flexible aluminum beam used in the experiment is modeled using the finite element method (FEM). This model was necessary initially as it was the only method available to determine the natural frequencies of the beam. Later, a dSPACE controller was available to experimentally confirm the natural frequencies. The dSPACE was used to collect data from the sensor on the beam and then processed in Matlab as mentioned in Chapter IV. The beam was excited without the compensator being connected. The flexible aluminum beam used is illustrated in Figure 5.1 with the placement of the piezoceramic patches shown. Due to geometry involving the piezoceramics, a 10 element

model was chosen. Piezoceramic actuators are on elements 2, 4, and 6 and the single sensor element is on element 3. There were actuators and a sensor on each side of the beam

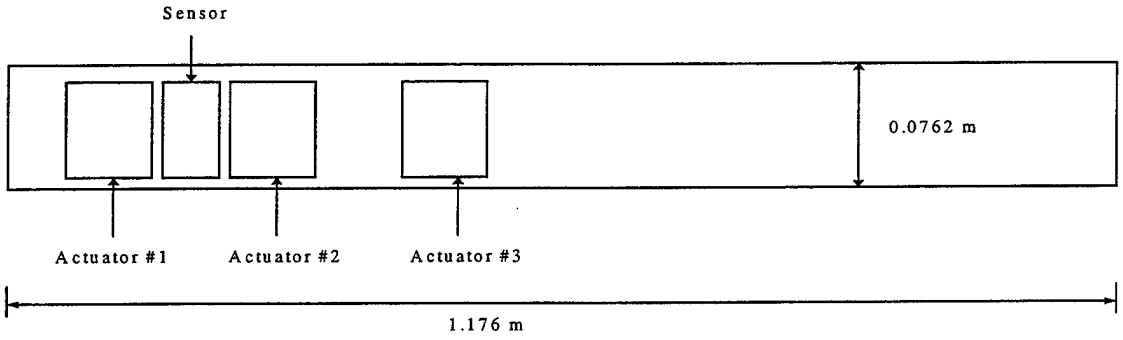


Figure 5.1 Aluminum Beam Dimensions

Appendix A contains the Matlab code used to model the beam. Tables 5.1 and 5.2 show the properties of the beam and piezoceramics respectively.

Aluminum Beam Type: 7075 T-6			
Quantity	**Description**	**Units**	**Value**
t_b	Beam thickness	meters	1.5875×10^{-3}
w_b	Beam width	meters	0.0254
ρ_b	Beam density	kg/m^3	2800
E_b	Young's Modulus	N/m^2	1.029×10^7

Table 5.1. Beam material properties

Piezoceramic Type: PZT-5A (Navy Type II)			
Quantity	**Description**	**Units**	**Value**
d_{31}	Lateral strain coefficient	*m/V* or *Coul/N*	1.8×10^{-10}
E_p	Young's Modulus	N/m^2	6.3×10^{10}
ν	Poisson's ratio	N/A	0.35
D	Absolute permittivity	*Farad/m* or N/V^2	1.5×10^{-8}
t_p	Piezoceramic thickness	meters	1.905×10^{-4}

Table 5.2. Piezoceramic properties

Using these properties and a ten element model, twenty natural modes of vibration were produced. Table 5.3 shows the first 10 natural frequencies as computed by the FEM.

Mode	Frequency (Hz)	Frequency (radians)
1	1.3823	8.6855
2	7.6745	48.220
3	18.796	118.10
4	30.917	194.26
5	53.196	334.24
6	86.349	542.55
7	155.24	975.43
8	174.50	1096.4
9	200.70	1,261.0
10	322.57	2,273.7

Table 5.3 Natural Frequencies Determined from Finite Element Model

The number of frequencies generally considered accurate from a finite element model is equal to half the number of elements used. In this case, 10 elements are used and so the first 5 modes are considered sufficiently accurate. The first two modes are the ones that are being targeted to be damped in this experiment so they are of primary interest.

The experimental results showed the first 4 modes to be 1.33 Hz, 7.1 Hz, 19.0 Hz, and 38.2 Hz. The gives a percentage error of 3.7%, 8%, 1.0%, and 19%. The modal shapes for the first three modes of the aluminum beam are shown below in Figure 5.2.

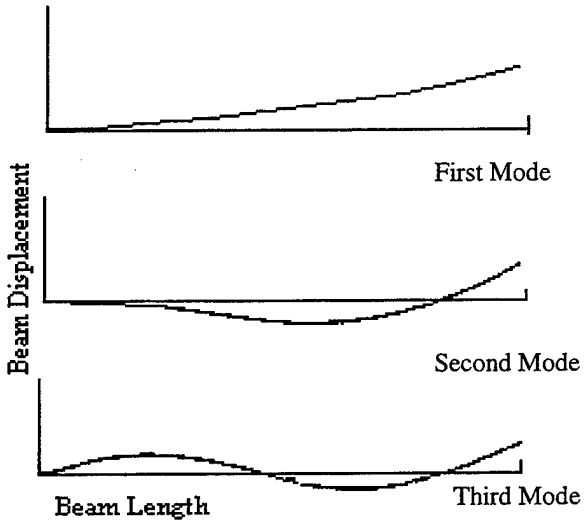


Figure 5.2 First Three Modal Shapes of Aluminum Beam

VI. SINGLE MODE VIBRATION SUPPRESSION

A. POSITIVE POSITION FEEDBACK

1. Theory

Positive Position Feedback (PPF) control is achieved by feeding the structural position coordinate directly to the compensator. The output of the compensator is then fed directly to the actuator times a desired gain. This is realized by applying the voltage from the sensor, which is proportional to the strain, directly to the compensator and then feeding the compensator output voltage to the actuator. Thus, unlike most feedback, loop gain is positive. The equations of motion describing PPF are given by

Structure $$\ddot{\xi} + 2\varsigma w\dot{\xi} + w^2\xi = gw^2\eta \tag{6.1}$$

Compensator $$\ddot{\eta} + 2\varsigma_c w_c{}^2\dot{\eta} + w_c{}^2\eta = w_c{}^2\xi \tag{6.2}$$

where ξ is the structure modal coordinate, η is the compensator modal coordinate, ζ and ζ_c are the structure and compensator damping ratios, ω and ω_c are the structure and compensator natural frequencies and G is the gain factor. A block diagram illustrating PPF control is shown in Figure 6.1.

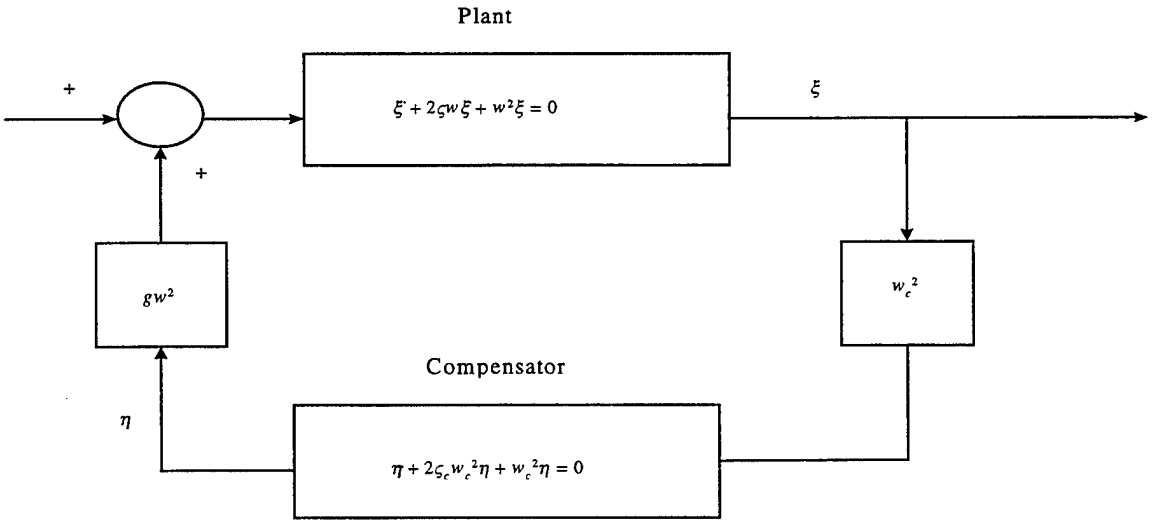


Figure 6.1. Block diagram for PPF control

Assuming a single degree of freedom and that the solution is of the form

$$\xi(t) = \alpha \cdot e^{iwt} \tag{6.3}$$

the output of the compensator will be

$$\eta(t) = \beta \cdot e^{i(wt-\phi)} = \frac{A\frac{w}{w_c} \cdot e^{i(wt-\phi)}}{\sqrt{(1-\frac{w^2}{w_c^2})^2 + (2\varsigma_c \frac{w}{w_c})^2}} \tag{6.4}$$

the phase angle ϕ is given by

$$\phi = \tan^{-1}\left(\frac{2\varsigma_c \frac{w}{w_c}}{1-\frac{w^2}{w_c^2}}\right) \tag{6.5}$$

When the structure vibrates at a frequency much lower than the compensator natural frequency, the phase angle approaches zero. Substituting equation 6.4 with ϕ=0 back into equation 6.1 results in

$$\ddot{\xi} + 2\varsigma w\dot{\xi} + (w^2 - G\beta w^2)\xi = 0 \tag{6.6}$$

This results in the stiffness term being decreased. When the compensator and the structure have the same natural frequency, the phase is $\pi/2$. In this case, the structural equation is

$$\ddot{\xi} + (2\varsigma w + G\beta w)\dot{\xi} + w^2\xi = 0 \tag{6.7}$$

This case shows an increase in the damping term. When the compensator frequency is greater than the structure, the phase angle approaches π. This results in a structure equation of

$$\ddot{\xi} + 2\varsigma w\dot{\xi} + (w^2 + G\beta w^2)\xi = 0 \tag{6.8}$$

This results in an increase in the stiffness term. A plot of the phase angle versus frequency is shown in Figure 6.2. As can be seen from the figure and the above discussion, to achieve maximum damping, w_c should be closely matched to w. Also any structural natural mode below w_c will experience increased flexibility. How much effect will depend on the natural damping of the structural frequency and the gain added to the compensator. Additionally, the exact area of active damping is not known. Larger values of ζ_c will result in a less steep slope thereby increasing the region of active damping. Figure 6.3 shows the bode plot for ζ_c=0.5 and for ζ_c=0.1. The difference in the slopes of the phase angle can easily be seen. This will have a positive effect on the robustness of the compensator. However, it will also result in slightly less effective damping and could result in increased flexibility any lower modes.

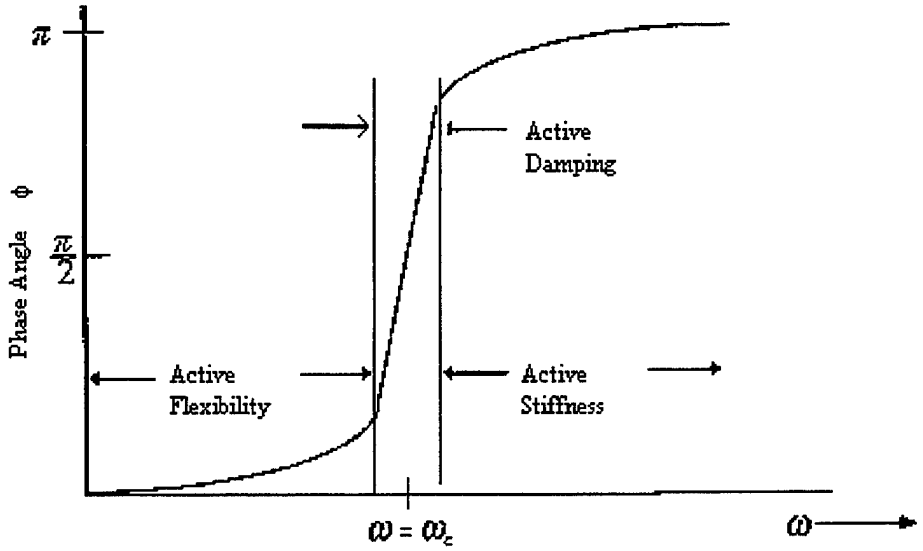


Figure 6.2 PPF Phase Angle Diagram

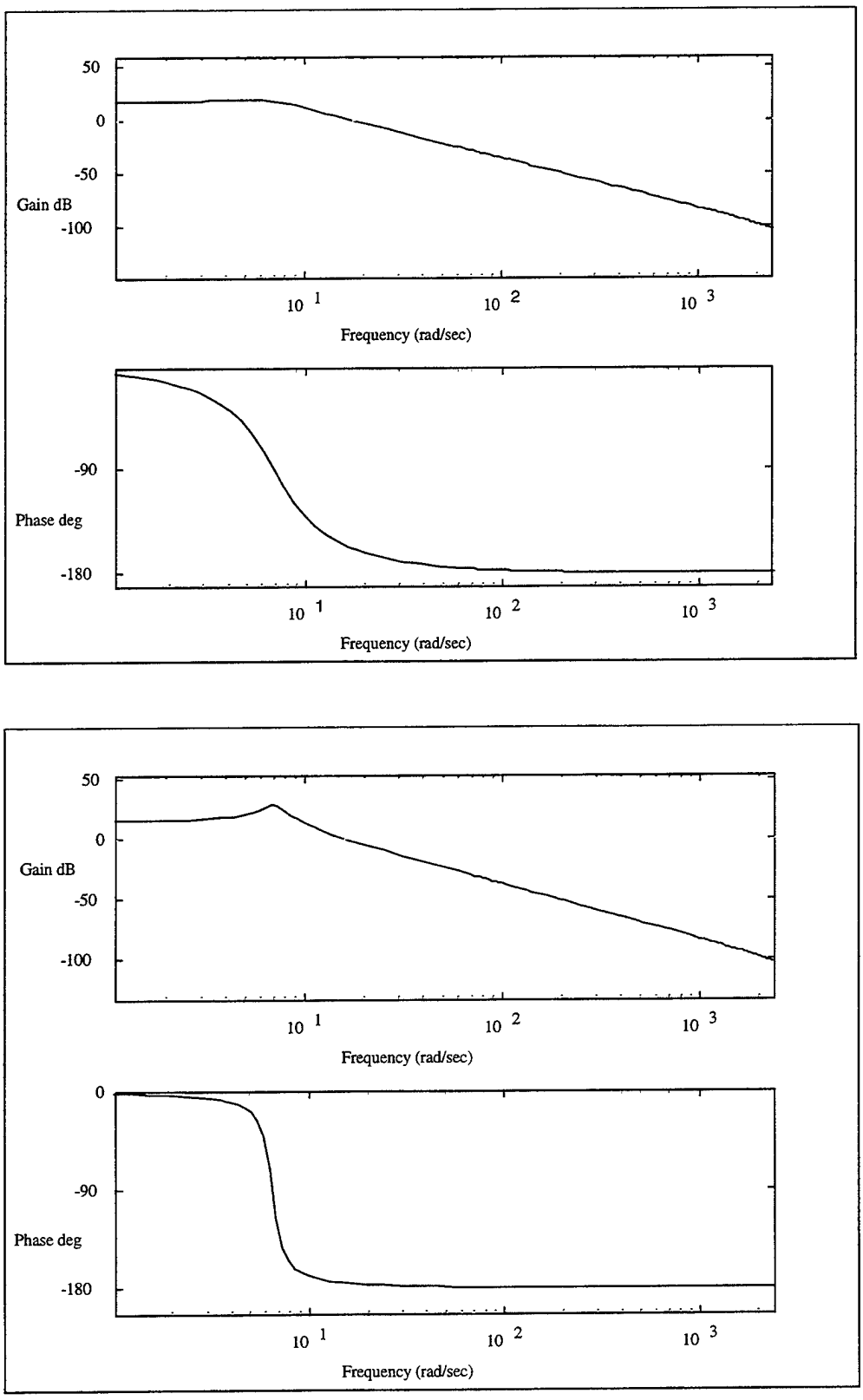


Figure 6.3 Bode Plot for Positive Position Feedback with ζ_c=0.5(top) and ζ_c=0.1(bottom)

2. Experimental Results

Damping effectiveness of a PPF implementation is shown in Table 6.1. In the last column, the achieved damping is compared to the natural damping of the beam. All three actuators were connected for this series of tests. Gain was increased until the actuators caused instability in the beam. Gain applied by the compensator was also the negative of the desired gain as the analog interface board inverted the signal. The compensator, ζ_c, damping was set 0.5. This was chosen as a compromise between damping effectiveness and robustness. The compensator frequency was set at the first natural frequency of the beam where it would give maximum PPF efficiency.

Using three actuators, PPF was very effective in damping the first mode but was ineffective when targeting the second mode. Figures 6.4 and 6.5 are PSD plots which graphically illustrate these results.

The location of actuator #3 was determined to be the cause of poor second mode damping. Originally, the actuators were placed on the beam in locations of maximum strain energy to optimize their effectiveness. Use of three actuators was intended to increase the total moment available to dampen the vibration. However, three actuators worked well only when trying to dampen the first mode. For higher modes, the actuator placement was found to adversely affect the damping. In some cases, it excited the higher modes. The explanation for this is that the distance between the sensor and actuator is too great for this beam. Actuator #3 should apply a moment that will counteract the strain in the beam. It generates a moment, though, that is determined by the strain at the sensor's location. Comparing the second and third mode shapes with the placement of actuator #3, it can be seen that for the higher modes the strain changes direction in-between the sensor and actuator #3. Thus, for higher modes, actuator #3 applies a moment that actually augments the vibration vice dampening it.

Because of this, the actuators were rewired so that each one could be connected individually. The majority of the tests were then conducted with only actuator #1 and the sensor connected. Actuator #1 was left connected since it was physically located right before the sensor. This configuration matched the actuator and sensor positions on the

flexible spacecraft simulator (FSS) at the Naval Postgraduate School which has successfully demonstrated second mode damping.

Test results implementing PPF control with a single actuator and sensor is shown in Table 6.2. Also Figure 6.6 is a PSD plot showing effective second mode damping using one actuator and a gain of -6.

Target Mode	Parameters	Damping Observed	% Change
1	f_c=1.3 Hz, ζ_c=0.5, gain = -2	39.20 dB	312%
1	f_c=1.3 Hz, ζ_c=0.5, gain = -4	52.31 dB	449%
1	f_c=1.3 Hz, ζ_c=0.5, gain = -6	71.71 dB	653%
2	f_c=7.1 Hz, ζ_c=0.5, gain = -6	18.86 dB	-37%

Table 6.1 PPF Results Using Three Actuators

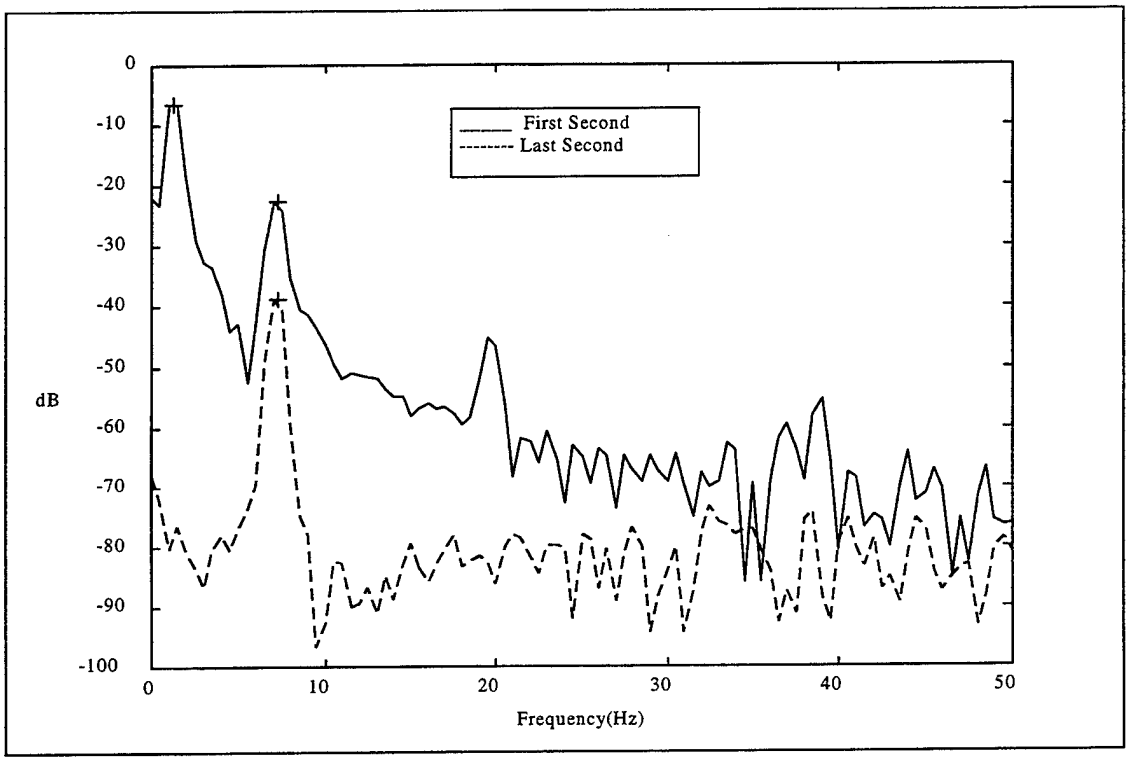


Figure 6.4 PSD Plot for PPF Using Three Actuators and Targeting the First Mode

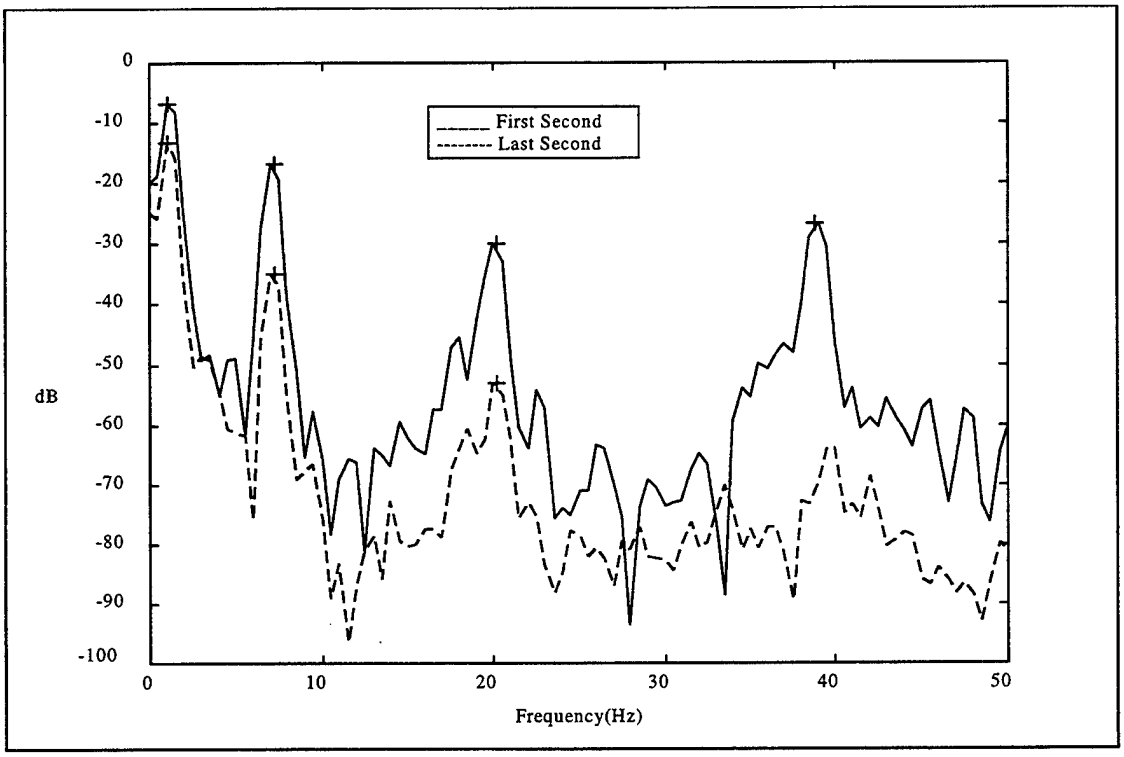


Figure 6.5 PSD Plot for PPF with 3 Actuators and Targeting Second Mode

Target Mode	Parameters	Damping Observed	% Change
1	f_c=1.3 Hz, ζ_c=0.5, gain = -4	44.20 dB	364%
1	f_c=1.3 Hz, ζ_c=0.5, gain = -6	61.89 dB	550%
2	f_c=7.1 Hz, ζ_c=0.5, gain = -2	34.81 dB	266%
2	f_c=7.1 Hz, ζ_c=0.5, gain = -6	44.00 dB	362%

Table 6.2 PPF Results Using One Actuator

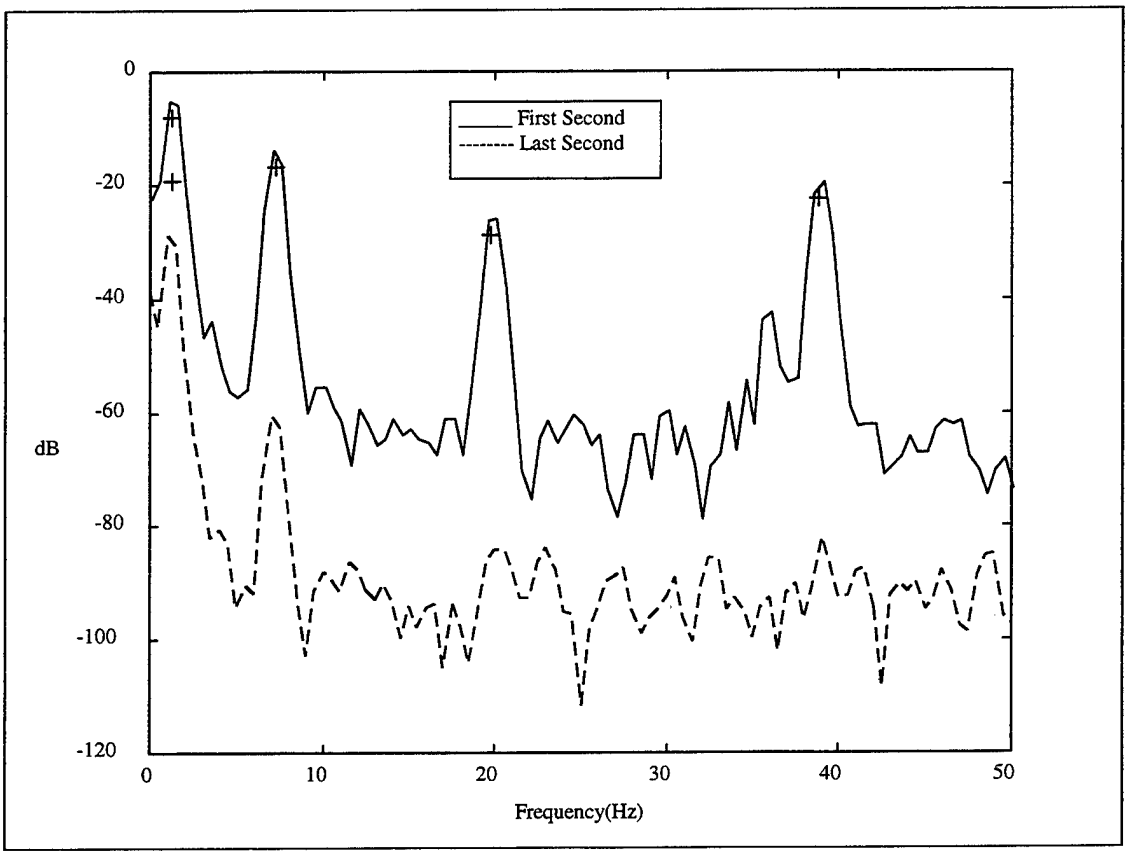


Figure 6.6 PSD Plot for PPF Using 1 Actuator and Targeting the Second Mode

Although damping is slightly reduced from the three actuator case, the one actuator case is still quite effective. Also, disconnecting the two actuators allowed the second mode to be damped using PPF.

Damping on the second mode using two actuators was briefly investigated. It was found that the maximum gain was -2 to maintain stability. This produced a damping of 42 dB or roughly equal to the one actuator case with a gain of -6. Thus, two actuators did produce any significant benefits.

The biggest documented drawback of PPF has been its lack of robustness. If something occurred to alter the targeted frequency or its was simply miscalculated, PPF damping would be severely affected. Also the increased flexibility it adds to frequencies below the compensator frequency has usually limited PPF to targeting the fundamental mode only.

3. PPF Robustness Analysis

Most PPF filters have been designed with ζ_c being 0.1 or less. This experiment tested the robustness with a higher ζ_c of 1, 0.5, and 0.25. As mentioned earlier, increasing ζ_c should increase the robustness but with a decrease in effectiveness.

The gain was increased from -6 to -10 for ζ_c=1. This was due to the fact that increasing ζ_c caused the gain of the controller to decrease around f_c. Thus gain could be increased without driving the beam unstable. It should also be noted that tests for ζ_c of 1 and a gain of -6 were conducted with a noticeable decrease in effectiveness. The results of these tests are shown below in Table 6.3.

Compensator Frequency (f_c)	Parameters Gain: $\zeta_c(.5)$ = -6; $\zeta_c(.25)$ = -6; $\zeta_c(1)$ = -10	Damping-dB (% increase) ζ_c=0.25	ζ_c =0.5	ζ_c =1
0.65 Hz	f_c = 1.3 x (0.5)	15.19(60%)	17.75(86%)	23.65(148%)
0.84 Hz	f_c = 1.3 x (0.65)	22.10(132%)	-	-
0.975 Hz	f_c = 1.3 x (0.75)	37.35(292%)	40.04(320%)	41.93(340%)
1.105 Hz	f_c = 1.3 x (.85)	52.02(446%)	52.78(454%)	-
1.3 Hz	f_c = 1.3 x (1.0)	60.41(535%)	61.64(547%)	56.50(493%)
1.495 Hz	f_c = 1.3 x (1.15)	-	56.44(493%)	51.22(438%)
1.625 Hz	f_c = 1.3 x (1.25)	41.58(337%)	54.94(447%)	-
1.885 Hz	f_c = 1.3 x (1.45)	-	45.62(379%)	-
1.95 Hz	f_c = 1.3 x (1.5)	26.06(174%)	-	62.98(562%)
2.275 Hz	f_c = 1.3 x (1.75)	-	53.45(461%)	-
2.60 Hz	f_c = 1.3 x (2)	-	28.62(201%)	56.39(492%)
3.90 Hz	f_c = 1.3 x (3)	-	17.17(80.3%)	-

Table 6.3 PPF Robustness Results

It can be seen from the table that the robustness increases as ζ_c is increased. This is as expected. Also, the damping ratio at f_c decreases as ζ_c increases. It can also be seen that the change in damping ratio from ζ_c=0.5 to ζ_c=1, is not as dramatic as from 0.25 to 0.5. PPF showed positive damping on all frequencies tried but dropped off rapidly when going below 75% of the fundamental frequency or 200% above it. Overall, these results are much better than expected. In addition, increased flexibility was not measurable even at 300% above the fundamental frequency. Optimal ζ_c for a given structure would depend on how accurate the modes are known and how much they would be expected to change, but a ζ_c of at least 0.5 gives strong damping when f_c is tuned to the first mode and good robustness. Figure 6.7 graphically illustrates the strong effectiveness and improved robustness achieved by using higher ζ_c values.

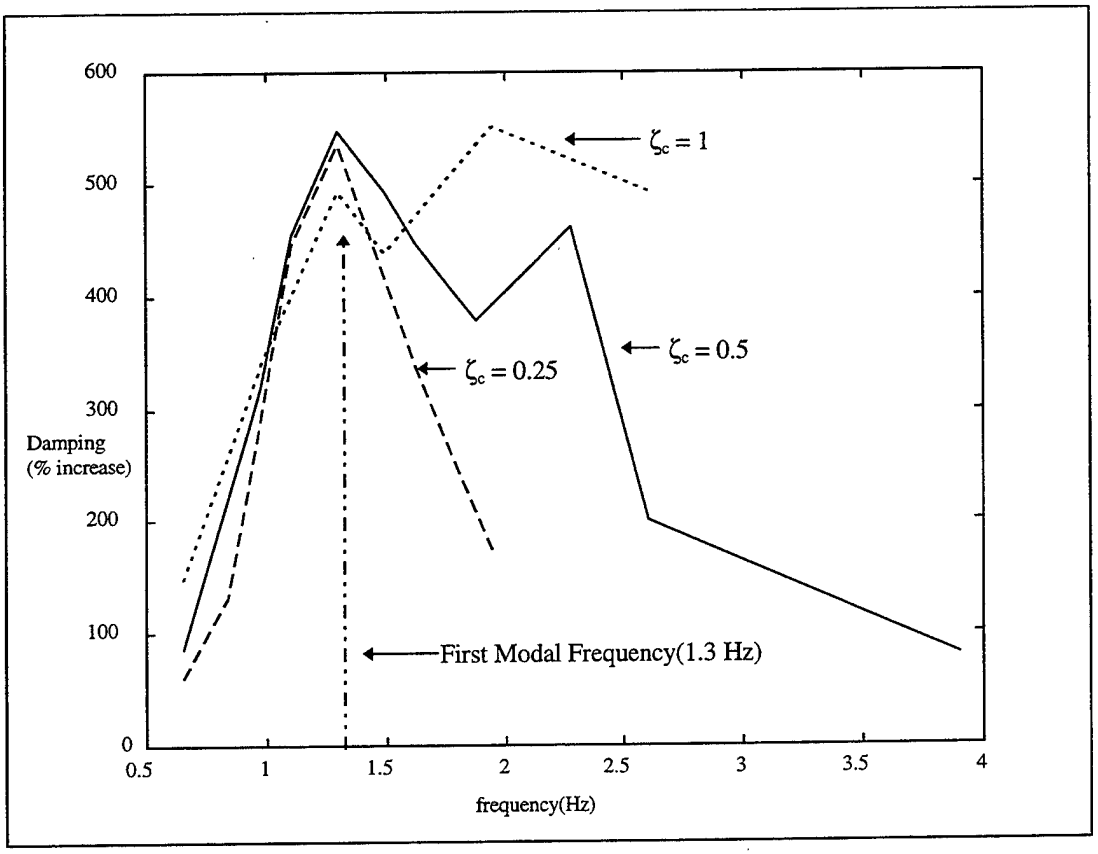


Figure 6.7 Robustness Results for different ζ_c

B. STRAIN RATE FEEDBACK

1. Theory

Strain rate feedback is achieved by feeding the structural velocity coordinate to the compensator input. The compensator position coordinate is then fed back to the structure after a negative gain is applied. This is realized by feeding the derivative of the voltage from the sensor, which is proportional to the strain rate, to the input of the compensator and applying the compensator output voltage to the actuator. The equations of motion in modal coordinates are

$$\ddot{\xi} + 2\varsigma w\dot{\xi} + w^2\xi = -gw^2\eta \tag{6.9}$$

$$\ddot{\eta} + 2\varsigma_c w_c{}^2\dot{\eta} + w_c{}^2\eta = w_c{}^2\dot{\xi} \tag{6.10}$$

where the variables are the same as those defined for PPF in the previous section. A block diagram illustrating this control scheme is shown in Figure 6.8

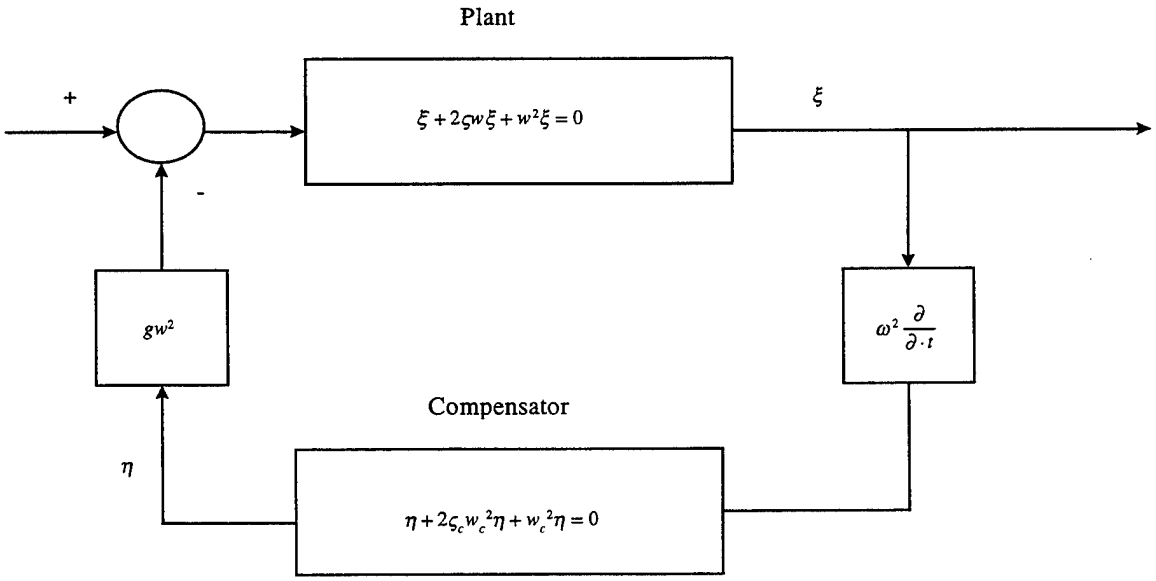


Figure 6.8 Block diagram for SRF control

The compensator bode plot for strain rate feedback is shown in figure 6.9. Again, assuming a single degree of freedom and that the solution is of the form

$$\xi(t) = \alpha \cdot e^{iwt} \tag{6.11}$$

the output of the compensator will be

$$\eta(t) = \beta \cdot e^{i(wt-\phi)} = \frac{A\frac{w}{w_c} \cdot e^{i(wt+\frac{\pi}{2}-\phi)}}{\sqrt{(1-\frac{w^2}{w_c^{\,2}})^2 + (2\varsigma_c \frac{w}{w_c})^2}} \tag{6.12}$$

the phase angle ϕ is given by

$$\phi = \tan^{-1}\left(\frac{2\varsigma_c \frac{w}{w_c}}{1-\left(\frac{w^2}{w^2{}_c}\right)}\right) \tag{6.13}$$

When the structure vibrates at a frequency much lower than the compensator natural frequency, the phase angle approaches $\pi/2$. Substituting equation 6.48 with $\phi=0$ back into equation 6.44 results in

$$\ddot{\xi} + (2\varsigma w + G\beta w^2)\dot{\xi} + w^2\xi = 0 \tag{6.14}$$

This results in an increase in the damping. When the compensator and the structure have the same natural frequency, the phase is 0°. In this case, the structural equation is

$$\ddot{\xi} + 2\varsigma w\dot{\xi} + (w^2 + G\beta w^2)\xi = 0 \tag{6.15}$$

This case shows an increase in the stiffness term. When the compensator frequency is greater than the structure, the phase angle approaches $(-\pi/2)$. This results in a structure equation of

$$\ddot{\xi} + (2\varsigma w - G\beta w)\dot{\xi} + w^2\xi = 0 \tag{6.16}$$

This results in a decrease in the damping term. Thus, in implementing SRF, the compensator should be designed so the targeted frequencies are below the compensator frequencies.

Selecting a precise compensator frequency is not as clearly defined as with PPF. However, SRF clearly has a much wider active damping region which gives the designer some flexibility. As long as the compensator frequency is greater than the structural frequency, a certain amount of damping will be provided. A big limitation to SRF is the fact that the magnitude of the transfer function in the active damping region becomes extremely small very quickly. Therefore, the amount of damping proving over a certain range is uncertain. Compensator damping ratio for this experiment was 0.02. This was done to limit the active stiffness area and maximize the active damping region with as much gain as possible.

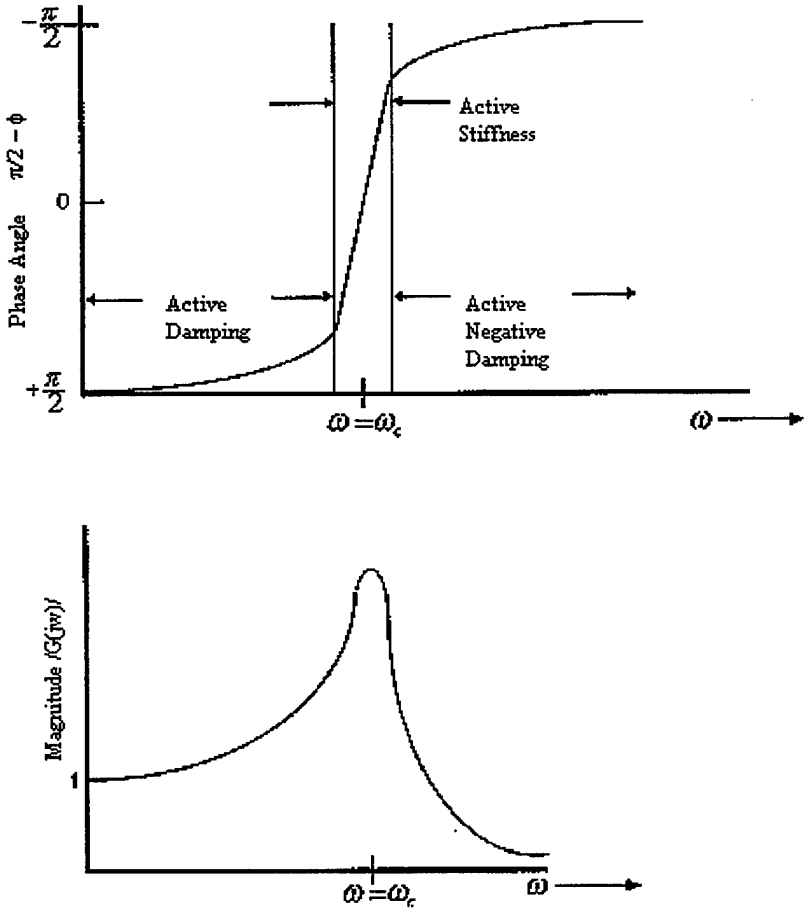


Figure 6.9 Bode Plot for SRF

2. Experimental Results

Damping effectiveness of a SRF implementation is shown in Table 6.4. In the last column, the achieved damping is compared to the natural damping of the beam. SRF was tested with one and three actuators. Gain was increased until the negative damping region began to excite the higher modes. Gain applied by the compensator was also the negative of the desired gain as the analog interface board inverted the signal. The compensator, ζ_c, damping was set 0.02 as mentioned above. The compensator frequency was set to put the targeted frequency in the active damping range with as high of magnitude as possible. Just as with PPF, SRF was ineffective in damping the second mode with three actuators but did work when one actuator was used. Figure 6.7 is a PSD plot of a SRF filter using 3 actuators and targeting the first mode.

As can be seen from the table, SRF was not as effective in damping the targeted mode as PPF. It only achieved 50% of the PPF damping ratio on the first mode. It performed better when targeted on the second mode. It was difficult to fine tune the SRF controller to optimize damping. The compensator frequency needed to be tweaked to put the targeted frequency as close as possible to it without placing it into the active stiffness range. SRF could be made more effective if a low pass filter is employed to attenuate the gain at higher frequencies. This might decrease the gain in the active negative damping region enough where it would not excite the higher modes but yet allow a higher gain to be achieved in the active damping region. It is not clear, though, how much higher the gain could be raised by implementing this technique.

Target Mode	Parameters	Damping Observed	% Change
1	f_c=2.5 Hz, ζ_c=0.02, gain = 1, 3 actuators	22.79 dB	139%
1	f_c=2.5 Hz, ζ_c=0.02, gain = 2 3 actuator	32.34 dB	240%
1	f_c=2 Hz, ζ_c=0.02, gain = 2 1 actuator	32.60 dB	242%
2	f_c=10 Hz, ζ_c=0.02, gain = 6 3 actuators	18.37 dB	-17.9%
2	f_c=10 Hz, ζ_c=0.02, gain = 2 1 actuator	44.10 dB	97%

Table 6.4 SRF Results on Single Mode Damping

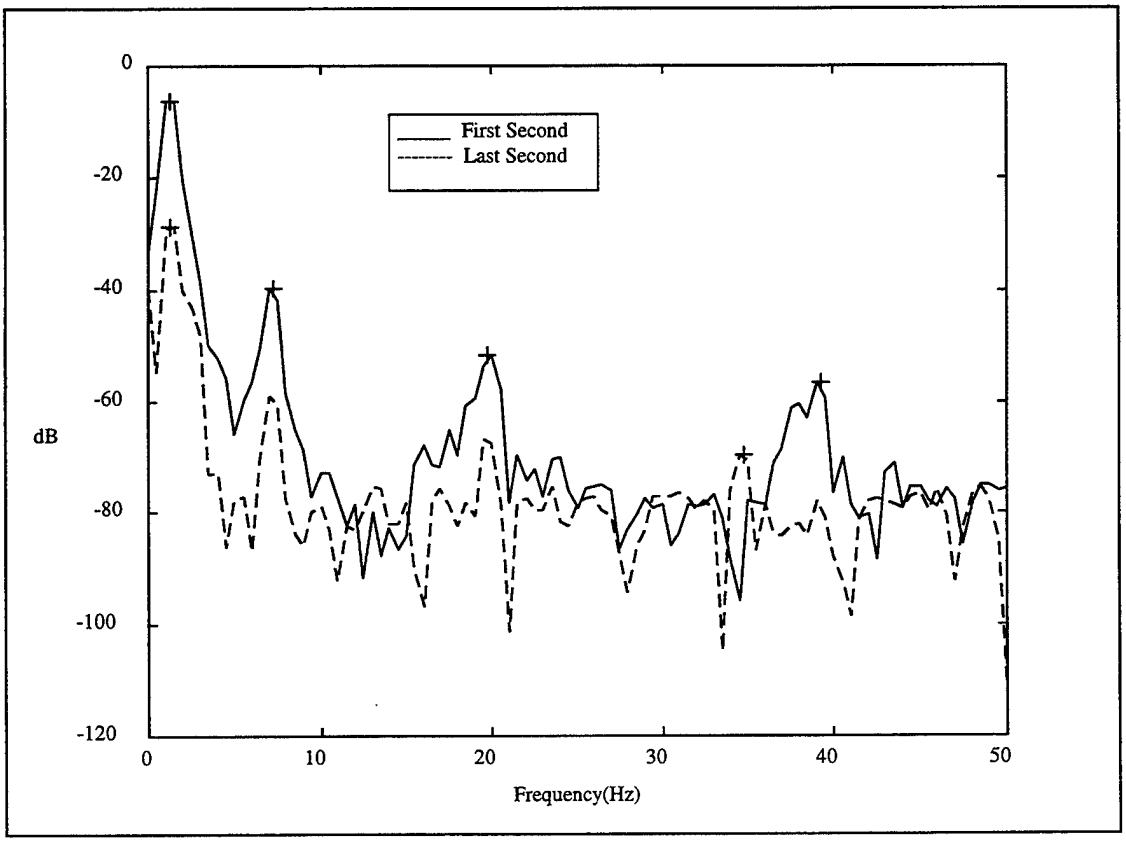


Figure 6.10 PSD Plot for SRF Using 3 Actuators and Targeting the First Mode

C. INTEGRAL FEEDBACK

1. Theory

Integral feedback is another popular control method. The primary reason for using integral control is to mitigate steady state errors, or to increase low frequency attenuation capability. Integral feedback transfer function has the form

$$G(s) = \frac{K}{T_I s} \tag{6.17}$$

where T_I is called the integral or reset time, and $1/T_I$ is a measure of the speed of the response and is referred to as the reset rate. T_I is the time for the integrator output to reach $1 \times K$ with an input of unity. Integral control has the primary virtue that it can provide a finite value of control signal with no error-signal input. This is because the output of the feedback is a function of all past values of the feedback input rather than just the current value. Therefore, past errors 'charge up' the integrator to some value that remains even if the error becomes zero and stays there. This means that disturbances can be canceled with zero error because the controller input no longer has to be finite to produce a control that will counteract a constant disturbance. A block diagram of an integral controller is shown in Figure 6.11 below.

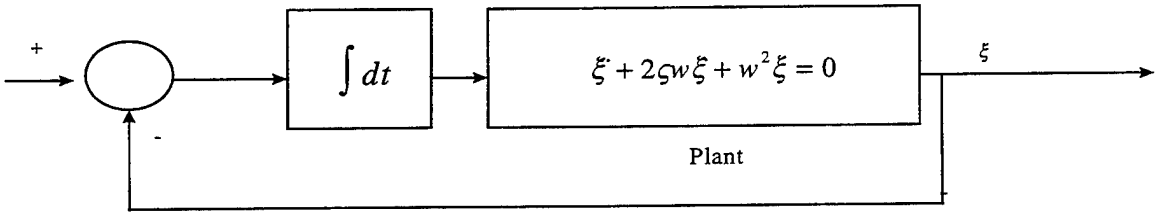


Figure 6.11 Block diagram for integral control

The main drawback with integral control tends to be actuator saturation because of the limited dynamic range of real actuators. The integrator builds up large values to constant or non-zero means inputs such as system noise. At low frequency, a pure

integral controller has extremely high gain as can be seen from the bode plot in Figure 6.12. This has to be counteracted or the actuator will become saturated almost instantly.

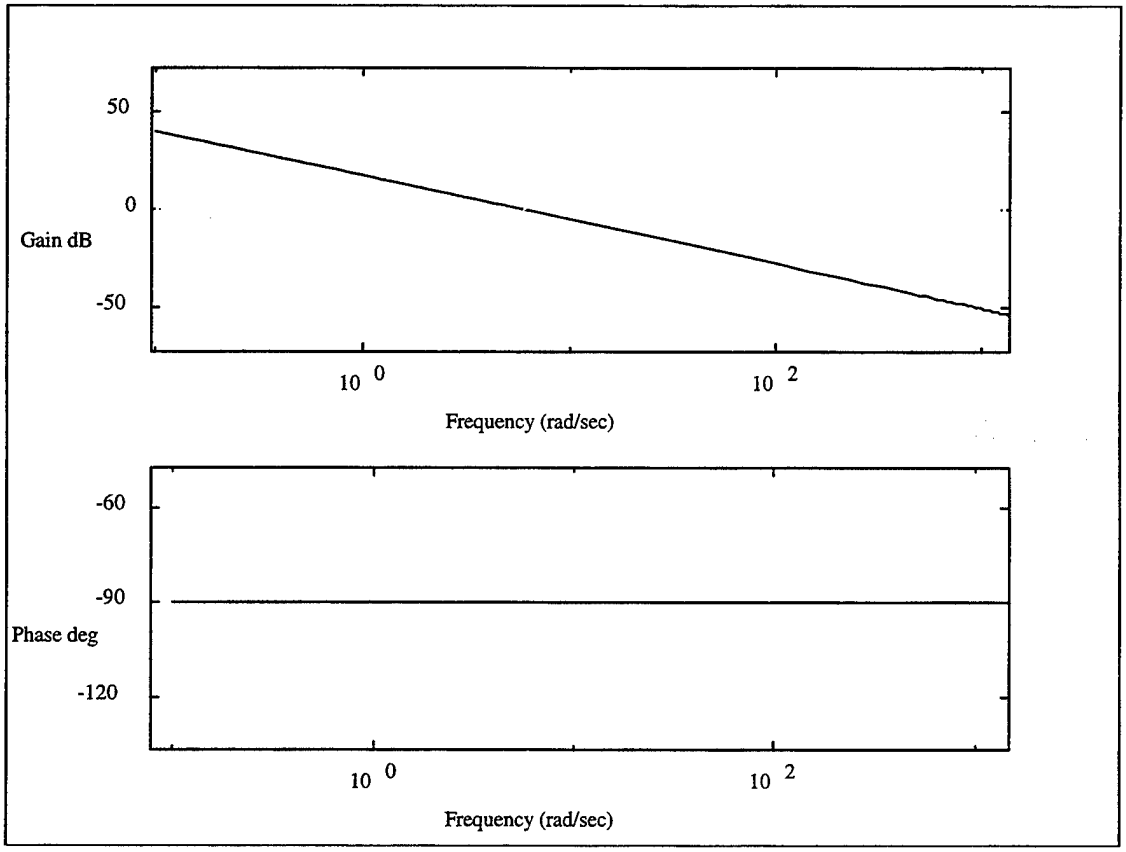


Figure 6.12 Bode Plot for Pure Integral Controller

One method of countering this effect is to activate the integral controller only when needed. Another way is to shut the controller down whenever actuator saturation is reached. This method is termed "integrator antiwindup"[Ref. 7]. A third alternative is to employ filters to reduce the low frequency gain but yet still achieve the 90° phase lag characteristic of integral controllers in the frequency range desired. Integral control was accomplished two ways during this experiment.

One implementation used several filters connected in series. This controller was developed by TRW. It begins with a one pole low pass filter at a very low frequency (0.01 Hz). This keeps the gain from infinite at zero frequency. A zero is then added to the transfer function at 0.4 Hz which essentially forms lag-lead filter. This is followed by

a lead-lag filter at 1.5 to 4 Hz which adds lead phasing at the crossover frequency. Then a 2 pole lowpass filter is added at 25 Hz for attenuation. A bode plot of this realization is shown in Figure 6.13. A slightly modified version of this controller was also implemented. The filters remained the same but the frequencies were shifted in an attempt to maximize the damping on the beam.

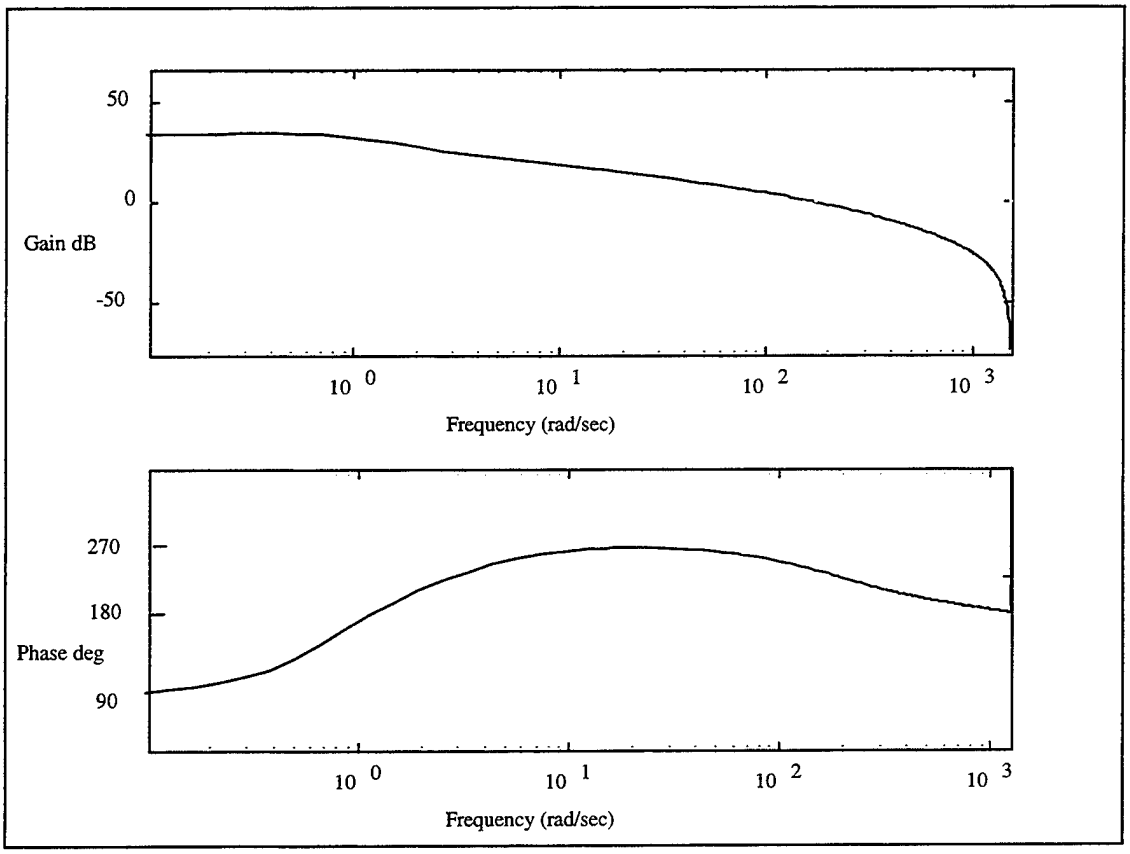


Figure 6.13 Bode Plot for TRW Integral Controller

The second way an integral feedback can be realized is by a simple one-pole lowpass filter. This attenuates the gain at low frequencies and also shifts the phase angle to the desired 90° lag. The bode plot of this filter is shown in Figure 6.14. This filter while effective is still susceptible to actuator saturation if left running for a period of time. However, when employed in conjunction with PPF, the gain is further reduced at low frequency and no actuator saturation occurs. This combination of filters worked reasonably well as is shown in Chapter VII.

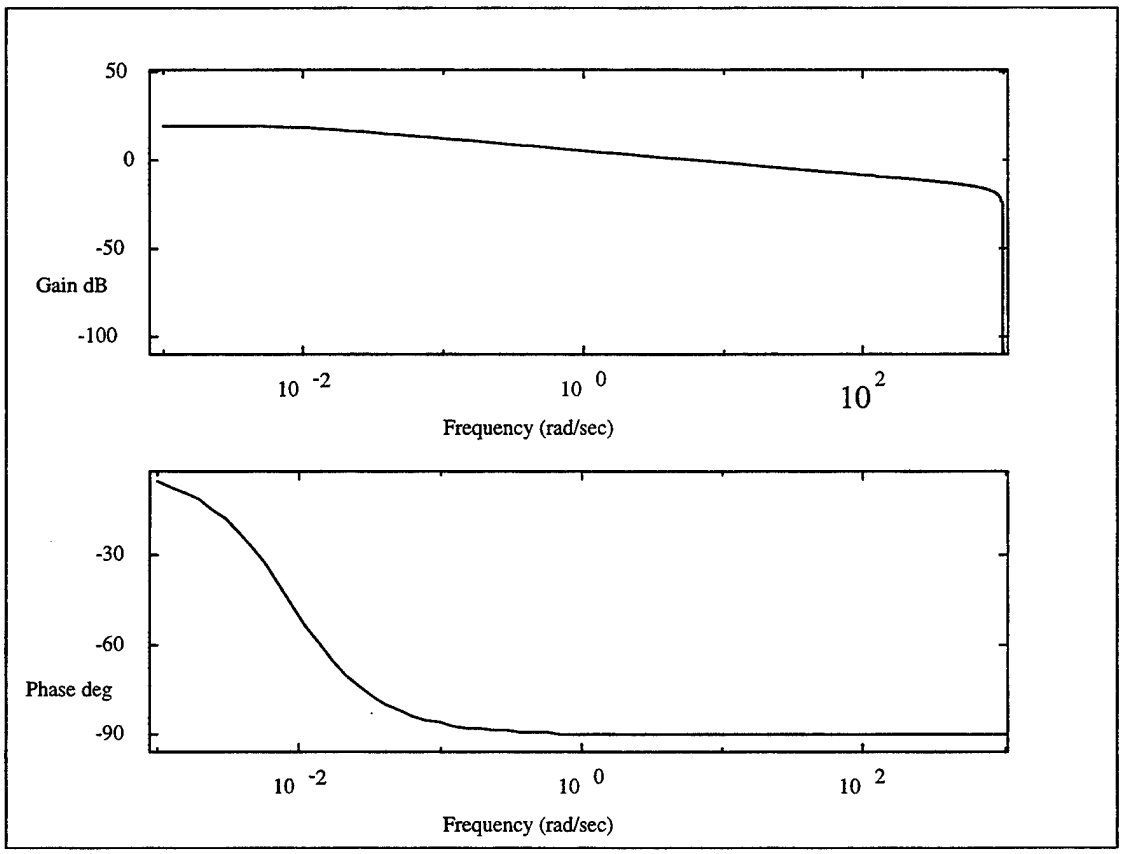


Figure 6.14 Bode Plot for Integral Control Realized by a One-Pole Lowpass Filter

2. Experimental Results

Integral feedback on single mode vibration is shown in Table 6.5. As mentioned above, the control realized was not pure integral feedback as this saturated the actuators very quickly. Instead the TRW filter was the actual design implemented. Data for the lowpass filter was only recorded when used in conjunction with PPF.

The TRW filter was tried on both the first and second modes. It actually became more effective when the filter frequencies were adjusted to optimize it for the natural frequencies of the aluminum beam. From the table, it can be seen that the filter was effective but not as capable as PPF. The real benefit for this controller is the potential for it to be effective for multi-mode vibration suppression. Figure 6.15 is a PSD plot for the TRW filter targeting the first modal frequency. The PSD plot illustrates the results for

the least effective controller. Changing the compensator's frequencies slightly significantly increased the damping effectiveness on the first mode.

Figure 6.16 is a PSD plot for the TRW filter with frequencies modified to target the second mode. As can be seen, second mode damping attempts were unsuccessful. For targeting a single mode, PPF and SRF are much easier to implement and more effective.

Target Mode	Parameters	Damping Observed	% Change
1	f_1=0.01 Hz, f_2=0.4 Hz, f_3=1.5 Hz, f_4=4 Hz, f_5=25 Hz, gain = -400, 1 actuator	28.42 dB	198.5%
1	f_1=0.01 Hz, f_2=0.4 Hz, f_3=1.5 Hz, f_4=4 Hz, f_5=25 Hz, gain = -800, 1 actuator	34.04dB	257%
1	f_1=0.01 Hz, f_2=1 Hz, f_3=1.15 Hz, f_4=3 Hz, f_5=35 Hz, gain = -400, 1 actuator	41.75 dB	339%
2	f_1=0.01 Hz, f_2=4 Hz, f_3=10 Hz, f_4=13 Hz, f_5=35 Hz, gain = -400, 1 actuator	21.60 dB	-3.5%

Table 6.5 Integral Feedback Effectiveness on Single Mode Damping

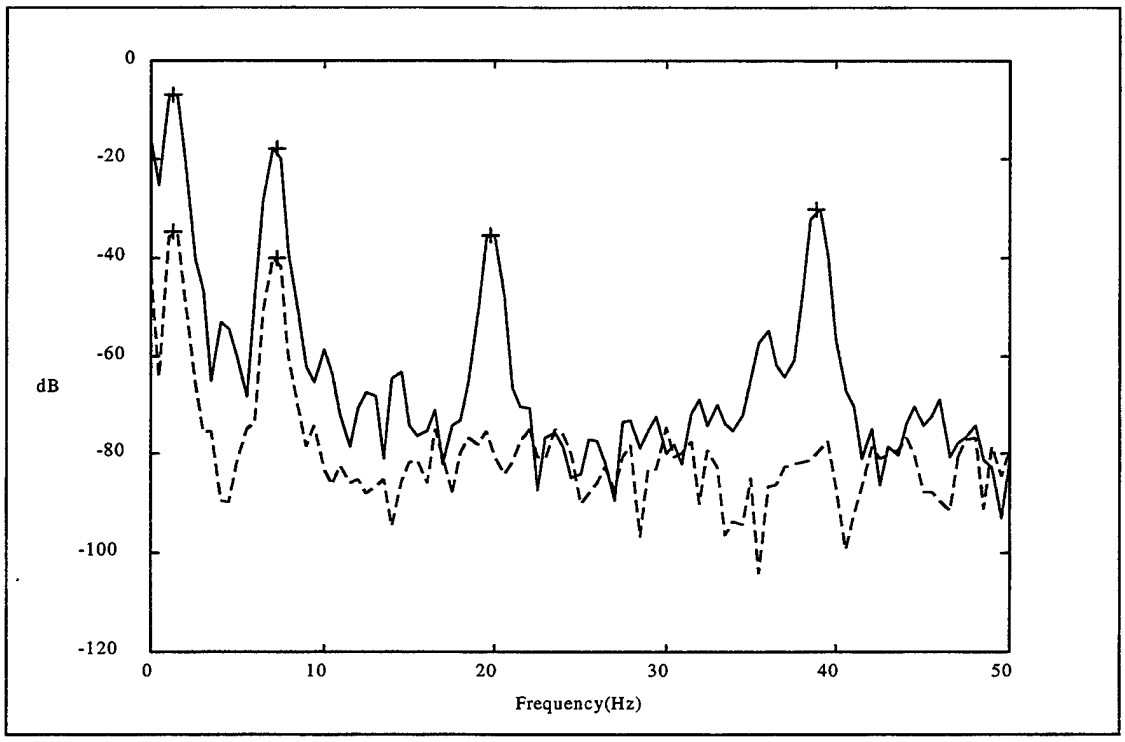


Figure 6.15 PSD Plot for TRW Filter Using 1 Actuator and Targeting the First Mode

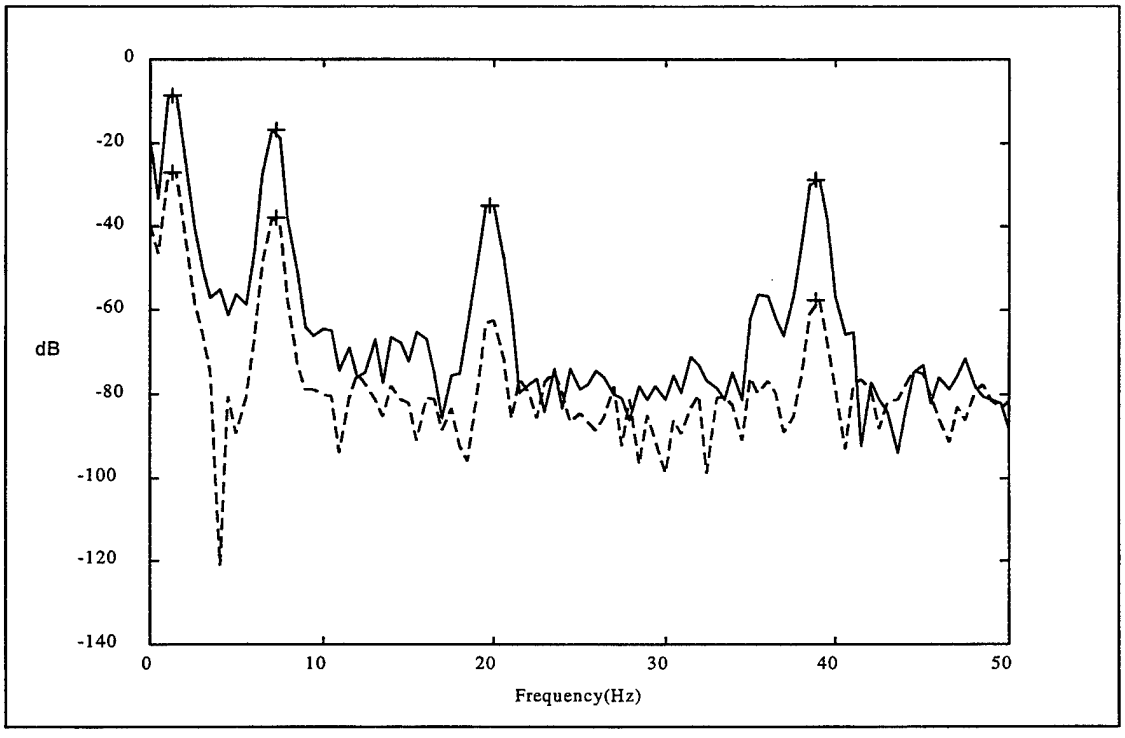


Figure 6.16 PSD Plot for TRW filter Using 1 Actuator and Targeting the Second Mode

VII. MULTI-MODE VIBRATION SUPPRESSION

The previous chapter discussed the effectiveness of several different types of control laws as implemented by the MCP on damping a single natural frequency of the flexible aluminum beam. This chapter will focus on the effectiveness of single control laws to dampen several modes. Effectiveness of multi-mode damping by combining different control laws in parallel is also examined. Combining control laws in parallel proves to be the most effective damping technique. In reality, active damping across a broad band of frequencies by a single control law or a simple combination of them is a highly desirable goal. A controller's robustness is also significant as the exact natural frequencies may not be known or may change with time.

The control law combinations include PPF & SRF, PPF & Integral, PPF & PPF, SRF & SRF. Results will show that two PPF filters together form the most effective damping pair and no single control law was very effective on multi-mode suppression. The biggest reason for this is due to the fact that the modes are far apart and the gain falls off fairly rapidly for all control laws. This makes it difficult for any single control law to be effective on both modes.

A. SINGLE CONTROL LAWS

1. Positive Position Feedback

Table 7.1 shows the results of single PPF filter with multi-mode excitations of the beam. A range of frequencies between the first and second mode were chosen to investigate if a single PPF controller could provide effective damping to two modes if f_c is properly chosen. Tests started with a compensator frequency set at the first mode and finished with the compensator frequency set at twice the second mode. All tests were conducted with a ζ_c set at 0.5.

It is fairly evident from the table that although PPF exhibits good robustness with ζ_c of 0.5, it is not effective at damping both modes. The reason is because the modes are spaced relatively far apart. If they were closer, PPF would be much more effective on damping both modes. The filter did achieve some positive damping on both modes with f_c = 3.90 being the most effective. However, the damping ratio was far smaller than accomplished when just a single mode was targeted and more effective multi-mode damping controllers will be shown.

Compensator Frequency (f_c)	Parameters Gain = -6 ζ_c = 1	Damping-dB (% increase) First Mode	Second Mode
1.3 Hz	f_c = 1.3 x (1.0)	56.30(491%)	21.90(-2.1%)
1.495 Hz	f_c = 1.3 x (1.15)	55.98(488%)	21.90(-2.1%)
1.625 Hz	f_c = 1.3 x (1.25)	46.85(392%)	23.02(2.8%)
1.885 Hz	f_c = 1.3 x (1.45)	48.28(407%)	22.70(1.4%)
2.275 Hz	f_c = 1.3 x (1.75)	40.97(330%)	22.71(1.4%)
2.60 Hz	f_c = 1.3 x (2)	35.50(273%)	23.63(5.6%)
3.90 Hz	f_c = 1.3 x (3)	19.44(104%)	25.89(15.7%)
5.235 Hz	f_c = 7.1 x (0.75)	13.52(42%)	29.81(33.2%)
7.1 Hz	f_c = 7.1 x (1)	12.09(27%)	34.93(132%)

Table 7.1 Single PPF Filter Results on Multi-Mode Damping

2. Strain Rate feedback

Table 7.2 shows the results of a single SRF filter on damping the first two modes of the aluminum beam. When the compensator frequency was placed just above the first mode, SRF was only effective on damping the that mode. Placing the compensator frequency just above the first mode, put the second mode in the active negative damping region of the SRF filter. This produced a slight but measurable reduction of the damping

ratio on the second mode. Conversely, when placing f_c above the second mode, both natural frequencies are in the active positive damping region of the SRF. However, the modes are far enough apart that the gain at the first mode is very low, and thus, damping accomplished on it is very small.

Target Mode	Parameters $\zeta_c = 0.02$ 1 actuator	Damping-dB (% increase) First Mode	Second Mode
1	$f_c = 2.5$ Hz, gain = 1	22.90(141%)	19.91(-11%)
1	$f_c = 2.0$ Hz, gain = 1	24.10(153%)	19.14(-14%)
1	$f_c = 2.0$ Hz, gain = 2	32.89(245%)	18.55(-17%)
2	$f_c = 10$ Hz, gain = 2	14.00(47%)	44.10(97%)

Table 7.2 Single SRF Filter Results for Multi-Mode Damping

3. Integral control

Table 7.3 shows the results of TRW's integral controller on the first two modes of the aluminum beam. The controller accomplished positive damping on the first mode but had virtually no effect on the second mode. The compensator's frequencies were modified to increase the effectiveness on both the first and second mode. First mode damping experienced a reasonable increase but no effect was measured on the second mode. Adjusting the filter parameters to dampen both modes was a tedious process and not successful. Again, the distance between the two modes made damping both of them by a single control law very difficult.

Target Mode	Parameters	Damping-dB (% Change)	
		First Mode	Second Mode
1	f_1=0.01 Hz, f_2=0.4 Hz, f_3=1.5 Hz, f_4=4 Hz, f_5=25 Hz, gain = -400, 1 actuator	28.50(199%)	22.90(2.3%)
1	f_1=0.01 Hz, f_2=0.4 Hz, f_3=1.5 Hz, f_4=4 Hz, f_5=25 Hz, gain = -800, 1 actuator	34.20(259%)	21.65(-3.2%)
1	f_1=0.01 Hz, f_2=1 Hz, f_3=1.15 Hz, f_4=3 Hz, f_5=35 Hz, gain = -400, 1 actuator	42.12(342%)	23.10(3.2%)
2	f_1=0.01 Hz, f_2=4 Hz, f_3=10 Hz, f_4=13 Hz, f_5=35 Hz, gain = -400, 1 actuator	17.20(81%)	22.05(-1.5%)

Table 7.3 Integral Control Results for Multi-Mode Damping

B. CONTROL LAW COMBINATIONS

The lack of success in damping the first two modes with a single control law led to several tests in which two control laws were combined together in parallel in an attempt to increase multi-modal damping. Table 7.4 shows the results for all combinations attempted.

The first combination tried was two PPF filters with each filter targeting a different mode. A Bode plot of this controller is shown in Figure 7.1. Note that the phase angle only reaches a value of approximately 45° at the first mode. Yet strong damping is still achieved. Figure 7.2 is a PSD plot showing the effectiveness of the controller with both PPF filters set at a gain of -6.

The first PPF combination implemented had the gain on the first filter set at -6 but the second filter gain was -2. This was done in order to reduce the increased flexibility impact that the second filter would have on the first mode. However, since the damping on the second mode was weak, it was decided to increase the second mode gain in order to improve the damping. Although this did lessen the damping on the first mode slightly,

it doubled the percentage damping on the second mode resulting in a more effective controller.

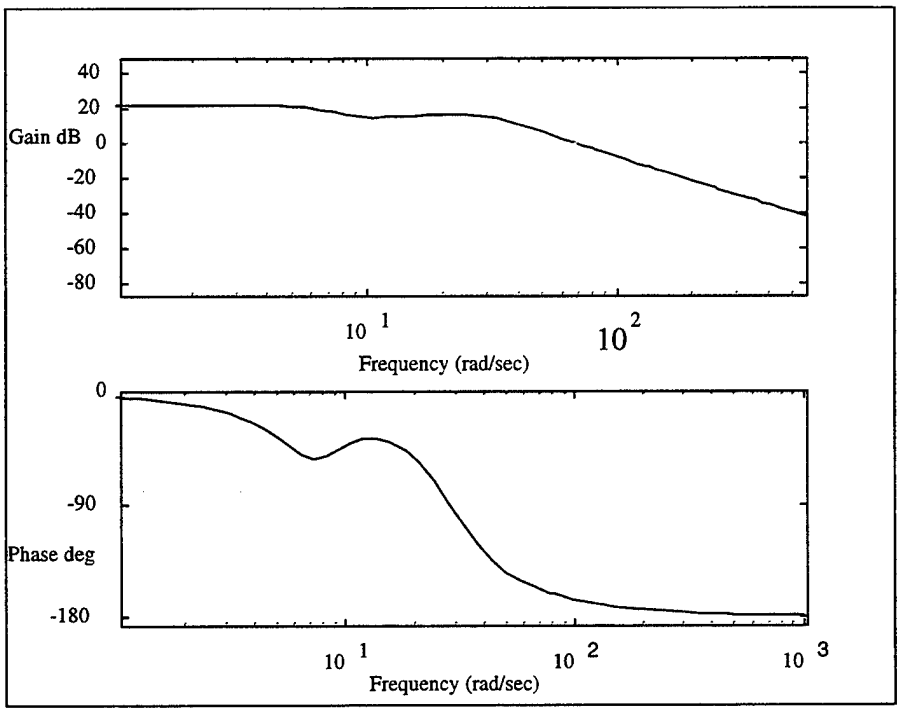


Figure 7.1 Bode Plot of Two PPF Filters in Parallel

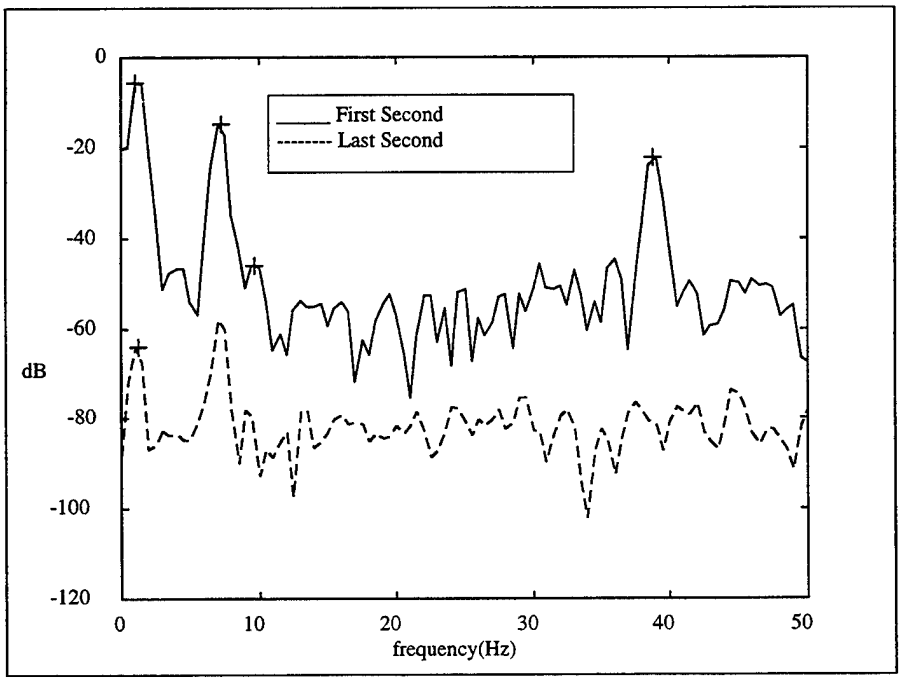


Figure 7.2 PSD Plot for Two PPF Filters in Parallel

The next pair attempted was a PPF with an SRF filter. The PPF targeted the first mode and the SRF the second. A Bode plot for this controller is shown in Figure 7.3. This was a very effective controller for both modes but the negative damping the SRF added to the higher frequencies exited the fourth mode. This can be seen on the graph of the PSD in Figure 7.4. The gain on the SRF was then lowered until the fourth mode was no longer excited. This reduced the damping on the second mode by almost 50% but the controller was still effective.

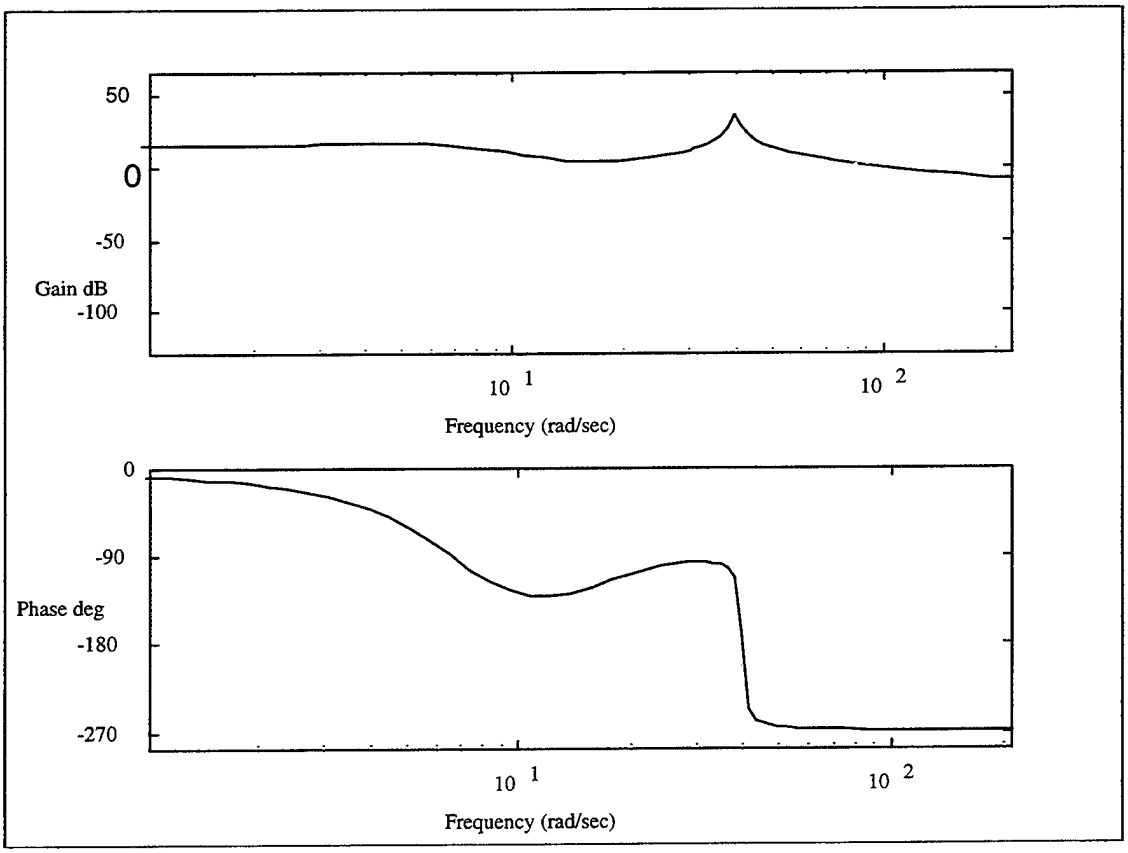


Figure 7.3 Bode Plot for PPF Combined with SRF

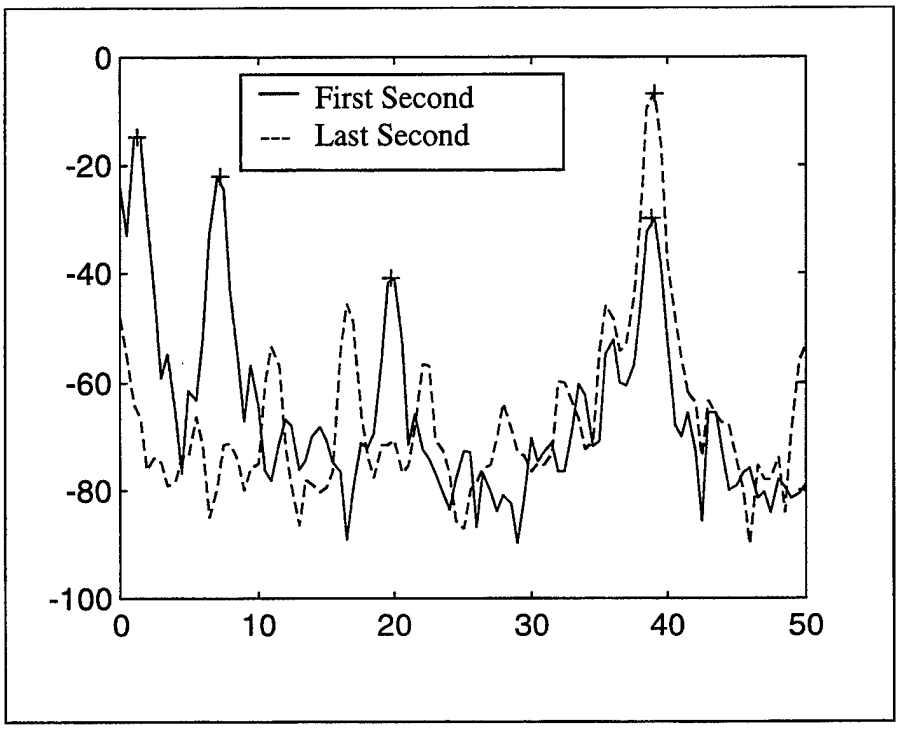


Figure 7.4 PPF and SRF Combination

The next combination tested was a controller using two SRF filters. A Bode plot for this combination is shown in Figure 7.5. The gain was initially set at 0.04 for both filters. This produced very effective damping on the second mode but very little on the first. Thus, the gain was increased as high as possible on the first filter to improve the results. First mode damping was increased dramatically but the increased gain also increased the active negative damping from the first filter and thus, the second mode damping decreased 39%. This can be seen on the PSD plot in Figure 7.6.

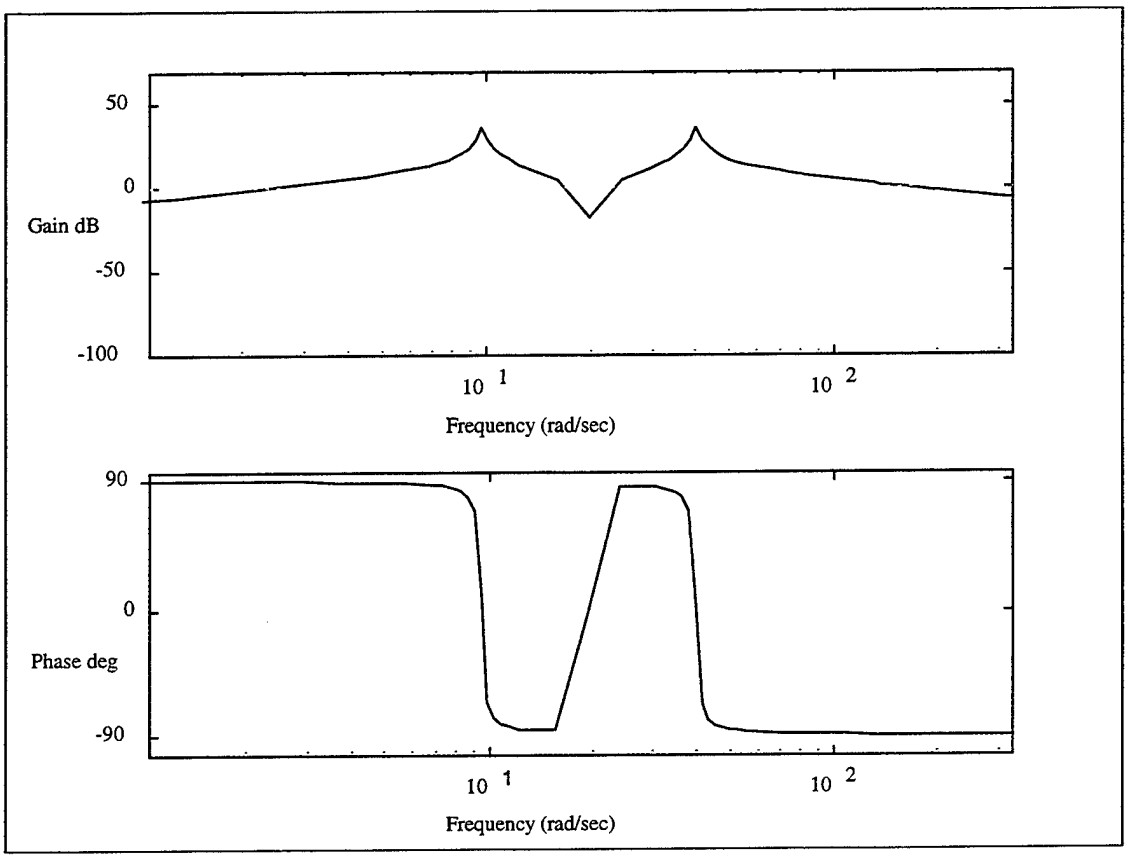


Figure 7.5 Bode Plot for Two SRF Filters Combined

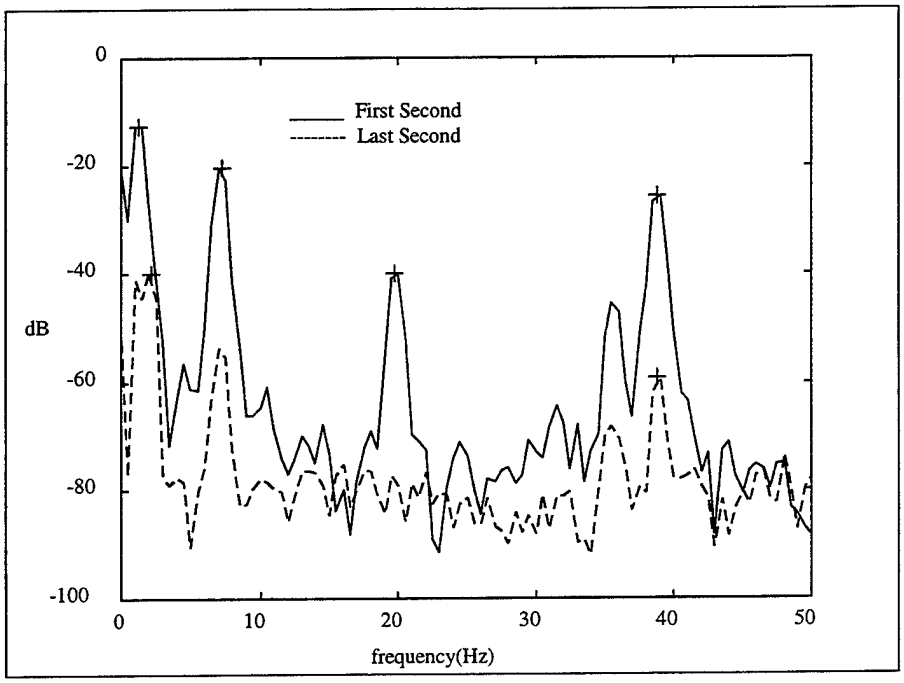


Figure 7.6 PSD Plot for Two SRF Filters Combined

Following the two SRF combination, a controller was attempted with a PPF and the TRW Integral filter. The Bode plot for this combination is shown in Figure 7.7. This controller proved to be ineffective on both modes. As can be seen from the Bode plot, the two filters interfere with one another and the proper phase is never achieved at the targeted modes. The ineffectiveness of this controller is shown in the PSD plot in Figure 7.8.

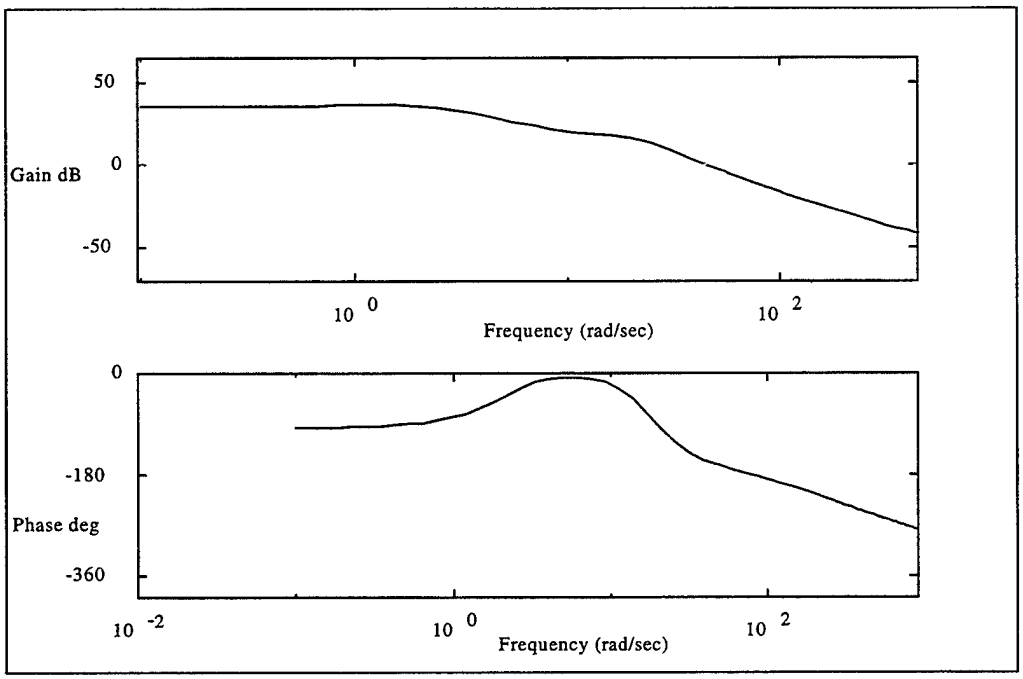


Figure 7.7 Bode Plot for PPF Combined with TRW Integral Filter

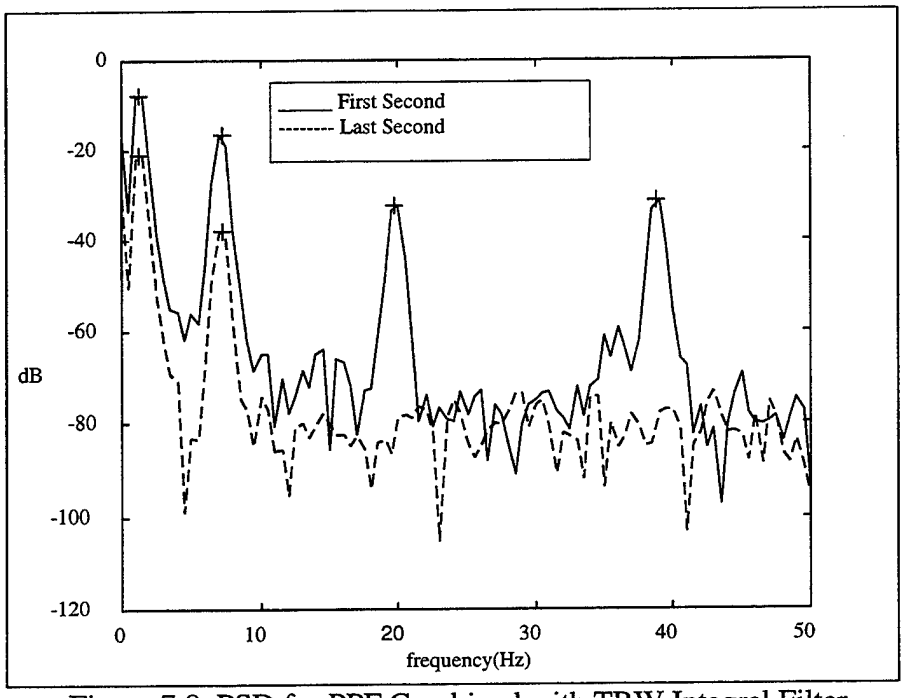


Figure 7.8 PSD for PPF Combined with TRW Integral Filter

The final combination attempted was a one-pole-lowpass filter implemented with a PPF filter. The lowpass filter, as discussed earlier, effectively achieves the 90° phase lag characteristic of integral feedback and when combined with PPF, the actuator saturation does not occur. The Bode plot for this controller is shown in Figure 7.9. Again, note that the PPF filter never reaches a phase angle of 90° at the first mode but yet is still very effective in damping the first mode.

The PPF and one-pole low pass filter controller had reasonable success as damping was achieved on both modes. However, damping of the second mode was not as high as achieved with previous controllers. A PSD plot for this combination is illustrated in Figure 7.10.

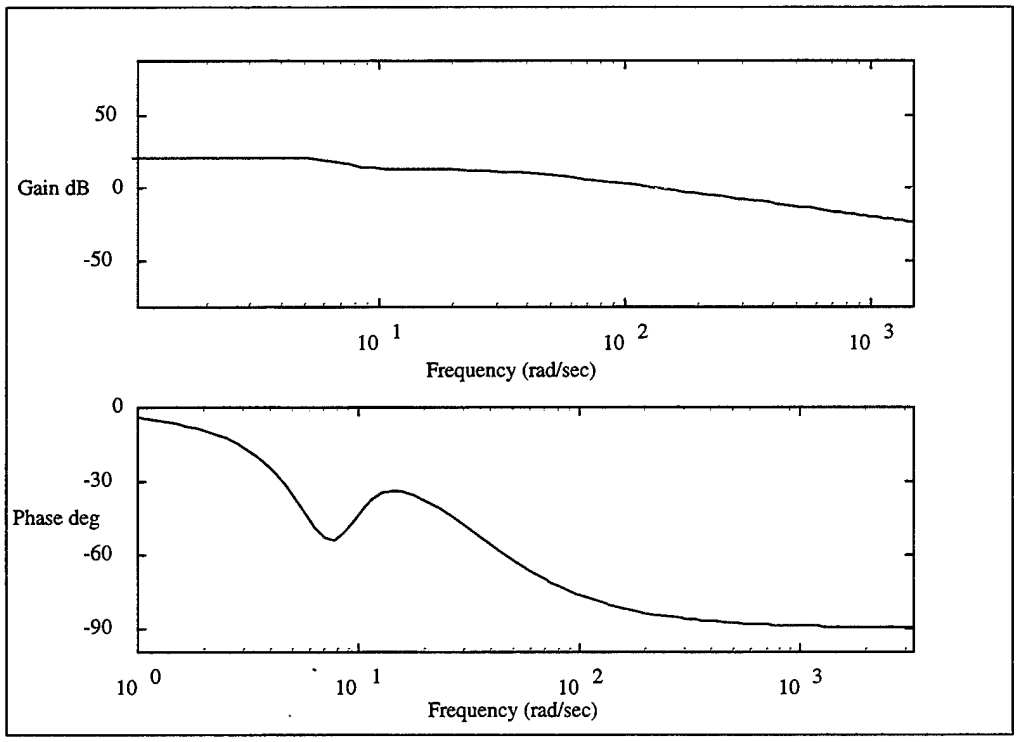


Figure 7.9 Bode Plot for PPF Combined with Lowpass Filter

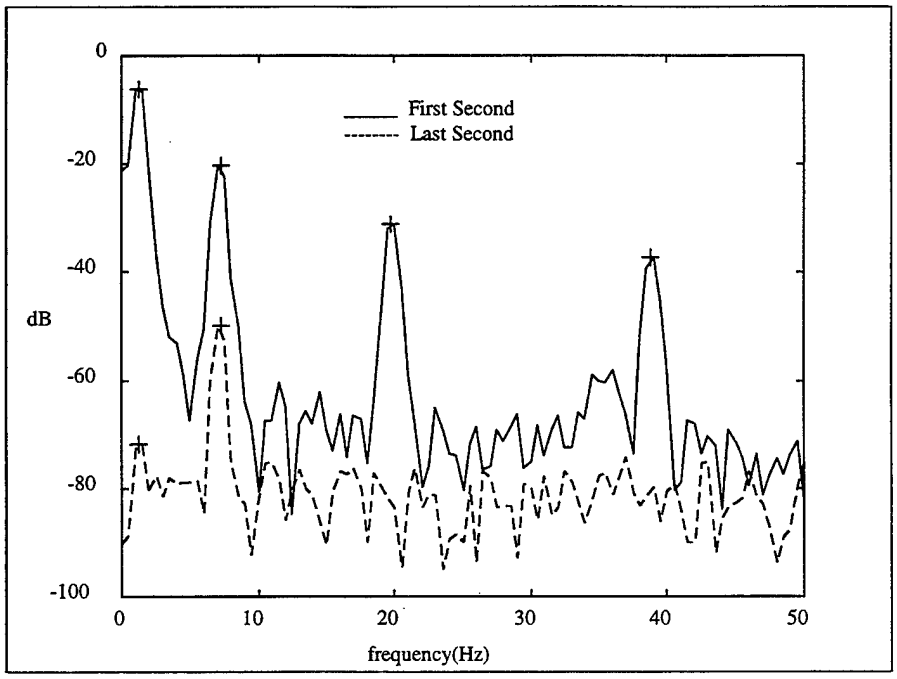


Figure 7.10 PSD for PPF Combined with a One-Pole-Lowpass Filter

From the Table 7.4, it can be seen that the PPF combination was the most effective in damping both modes. This was a little surprising since targeting the second mode with PPF puts the first mode in the increased flexibility region which will reduce the damping on that mode. However, the first PPF filter was not adversely affected by this even when the gain was increased.

Compensator Frequency	Parameters 1 actuator	Damping-dB (% Change)	
		First Mode	Second Mode
PPF(1st Mode) PPF(2nd Mode)	f_1=1.3, ζ_c=0.5, gain = -6 f_2=7.1, ζ_c=0.5, gain = -2	68.02(614%)	31.68(42%)
PPF(1st Mode) PPF(2nd Mode)	f_1=1.3, ζ_c=0.5, gain = -6 f_2=7.1, ζ_c=0.5, gain = -6	58.33(513%)	44.00(97%)
PPF(1st Mode) SRF(2nd Mode)	f_1=1.3, ζ_c=0.5, gain = -6 f_2=10, ζ_c=0.02, gain = 0.9	50.85(434%) 4th mode	52.11(133%) excited
PPF(1st Mode) SRF(2nd Mode)	f_1=1.3, ζ_c=0.5, gain = -6 f_2=10, ζ_c=0.02, gain = 0.04	72.84(665%)	38.98(74%)
SRF(1st Mode) SRF(2nd Mode)	f_1=1.3, ζ_c=0.02, gain = 0.04 f_2=10, ζ_c=0.02, gain = 0.04	16.09(69%)	42.70(91%)
SRF(1st Mode) SRF(2nd Mode)	f_1=1.3, ζ_c=0.02, gain = 0.1 f_2=10, ζ_c=0.02, gain = 0.04	30.04(216%)	34.04(52%)
PPF(1st Mode) TRW Integral (2ndMode)	f_1=1.3, ζ_c=0.5, gain = -6 f_1=0.01 Hz, f_2=1 Hz, f_3=1.15 Hz, f_4=30 Hz, f_5=35 Hz, gain = -100	13.65(43%)	22.10(-1.2%)
PPF(1st Mode) Integral (2nd Mode)	f_1=1.3, ζ_c=0.5, gain = -6 f_2=7, gain = -100	66.23(596%)	31.00(39%)

Table 7.4 Control Combination Results on Multi-Mode Damping

C. ROBUSTNESS ANALYSIS FOR TWO PPF FILTERS

Given the effectiveness of combining two PPF filters, a test was conducted to investigate the robustness of this controller. Table 7.5 shows the results of this test. A ζ_c of 0.5 and a gain of -6 is used for both filters. Both compensator frequencies are moved progressively higher than the targeted natural frequency. As is seen from the table, the first mode damping falls off much quicker than the second mode. Part of the reason for this is the increased stiffness region of the first filter is moving closer to the second mode thereby helping to increase the damping effect. Just as in the single PPF case, the PPF combination shows good robustness for both modes. These results suggest that along with being robust, the two PPF filter combination is effective in damping multiple modes over a range of frequencies. A Bode plot of the controller with both compensator frequencies set at 1.5 times the targeted modal frequencies is shown in Figure 7.11. A PSD plot for the same controller graphically illustrating damping effectiveness is shown in Figure 7.12.

Parameters ζ_c=0.5, gain = -6 1 actuator	Damping-dB (% Change) First Mode	Second Mode
f_1=1.3 x 1.0 = 1.3 f_2=7.1 x 1.0 = 7.1	58.33(513%)	44.00(97%)
f_1=1.3 x 1.25 = 1.625 f_2=7.1 x 1.25 = 8.875	42.16(343%)	38.10(70%)
f_1=1.3 x 1.5 = 1.95 f_2=7.1 x 1.5 = 10.65	36.14(280%)	36.35(62%)
f_1=1.3 x 1.75 = 2.275 f_2=7.1 x 1.75 = 12.45	30.11(216%)	32.99(47%)
f_1=1.3 x 2 = 2.6 f_2=7.1 x 2 = 14.2	17.88(88%)	26.57(19%)

Table 7.5 Robustness Results for Two PPF Filter Combination

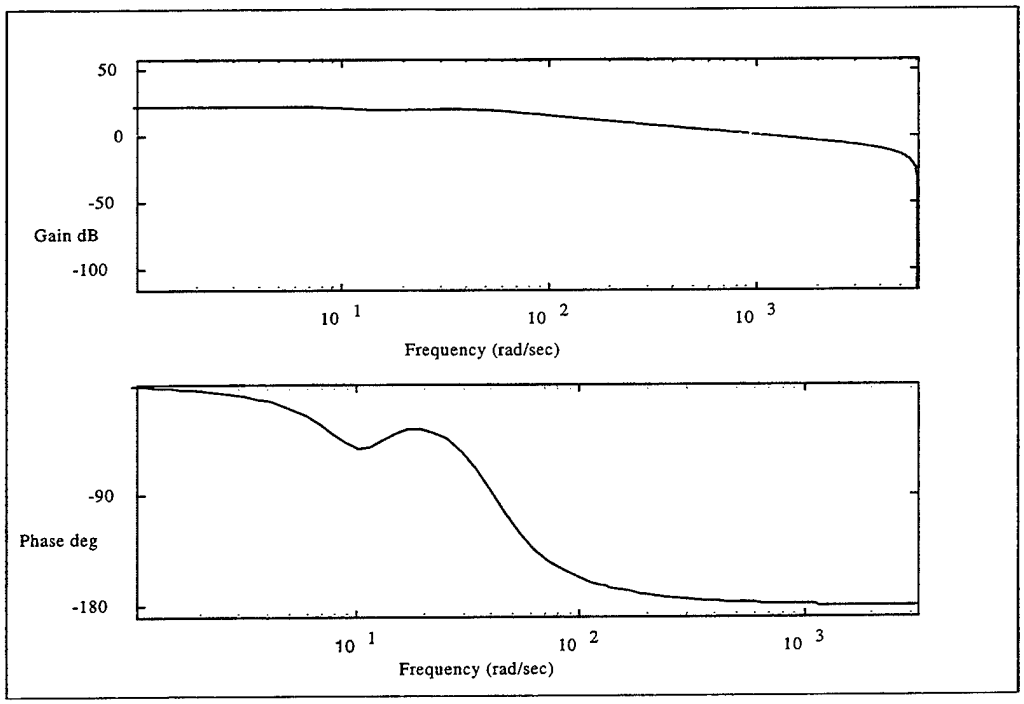


Figure 7.11 Bode Plot for Two PPF Filters. Compensator frequencies are 1.5 times the targeted modal frequencies

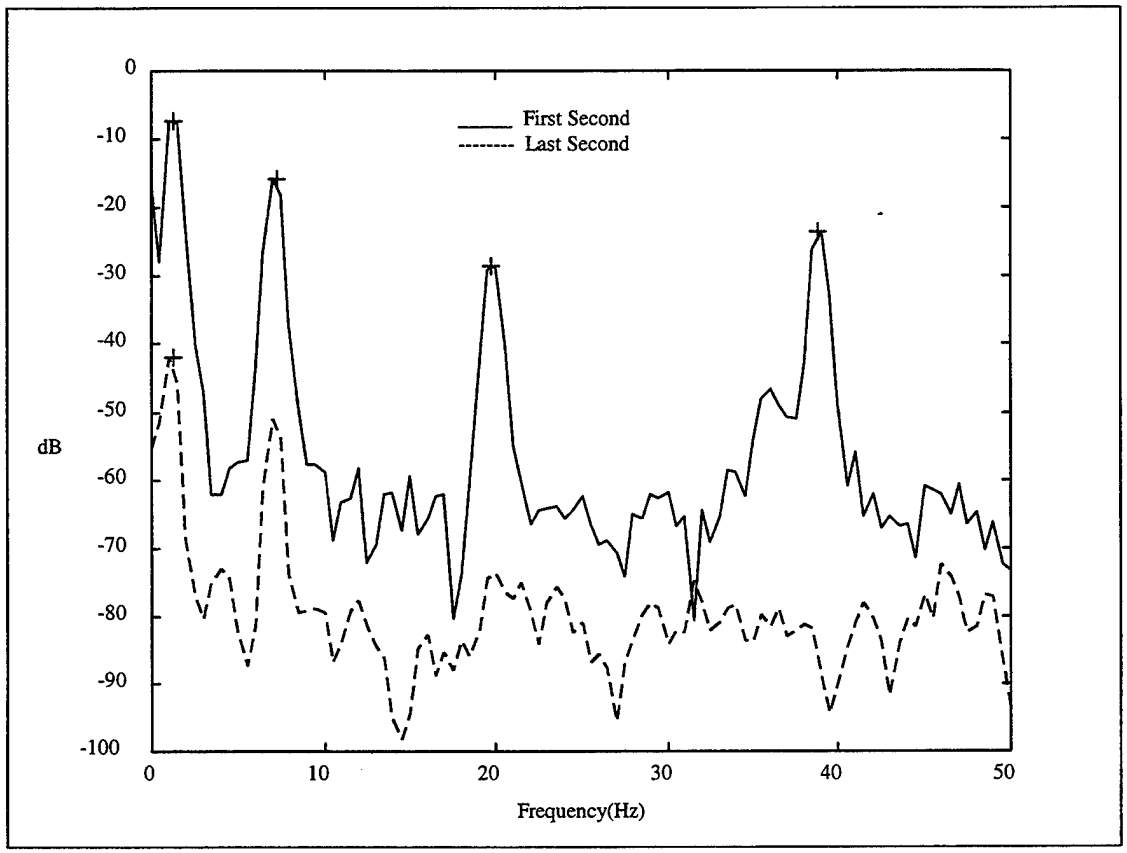


Figure 7.12 PSD Plot for Two PPF Filters. Compensator frequencies are 1.5 times the targeted modal frequencies

VIII. CONCLUSION

The MCP demonstrated the capability to effectively implement real time control laws. It proved to be a very capable and versatile instrument. The MCP has the potential to be a functional space-based vibration controller. The main drawbacks observed from this experiment are the user interface and its reliability.

The MCP malfunctioned twice during the course of this four month experiment. It was operated in a very controlled environment under near ideal conditions. To be suitable for space-based operations requires the equipment to be reliable and rugged enough to withstand the harsh space environment. The MCP did not demonstrate that high level of dependability in this experiment.

The user interface was also less than desirable. The multi-channel filter windows interface was easy to use but it was severely limited in the type of controllers it could implement. The state space program, on the other hand, was extremely versatile and was used for the majority of this experiment. Realization of any desired controller or combination of them was accomplished. However, the lack of a simple software interface required the user to perform extra steps to construct a controller. This proved cumbersome. A windows or Matlab interface for the state space program would be a big improvement.

This experiment also showed the capability of PPF, SRF, and Integral control laws. PPF performance has typically been very dependent on matching the compensator frequency to the structural frequency. However, increasing the compensator damping to 0.5 greatly increased the robustness of the controller. PPF also demonstrated the best damping ratios, and by combining two PPF filters in parallel, the most effective multi-mode damping was achieved. While SRF and Integral were successful in modal damping, neither one could achieve the level of damping that PPF provided. Although SRF has a wider active damping frequency region, the gain roll off requires the modes to be close together or the gain simply becomes too low for the controller to be effective. Integral control while functional did not provide the broad band damping that was hoped for or desired from it. Integral control proved difficult to implement and was the least effective of the three control laws.

Recommendations for further research include implementing the MCP on the Flexible Spacecraft Simulator (FSS) at the Naval Postgraduate School. The FSS will allow the multiple channel capability of the MCP to be explored. The failure of the MCP prevented this from being accomplished for inclusion in this thesis. Additionally, continued experiments on the effectiveness and robustness of PPF should be conducted. Specifically, tests should be conducted on a structure with more densely pack modes. The FSS would also be appropriate for this experiment.

APPENDIX A. [MCP PROCDEURES AND PROGRAMS]

Checklist to realize state space matrices on MCP

1. Calculate A, B, C, D matrices in Matlab
2. Check bode plot with dbode command: **dbode(ad,bd,cd,dd,1/5000,1)**
3. Save matrices in ascii format: **save filename ad bd cd dd -ascii**
4. Open file in notepad and put matrices in C language format (see matrxdat.c)
5. Save file as **mtrxdat.c**
6. Change **mtrx.c** to reflect number of states, input, and outputs.
7. Compile program using TI C compiler. To accomplish this do the following:
   a. Switch to directory containing TI compiler: **cd c:\directory**
   b. Set path to TI compiler software: **set path=%path%; c:\directory**
   c. Include library files: **set c_dir=c:\directory\h; c:\cc\lib**
   d. Switch to directory containing matrix files
   e. Type: **wmake /f mtrxfltr.mak**
8. Reset MCP
9. Download to computer: **blload /p 1 /I mtrxfltr.lod**

Checklist for Multi-channel graphical user interface

1. In dos window, change to mcp directory
2. type **blload /p 1 /I mfltmcp.lod**

   You should see: Current Baud Rate is 19200 (immediately)

   Successful Data Transmission (after a few seconds)

   If you do not get a successful transmission message, press **Esc**

   and redo steps 7 and 8.
3. Start windows by typing **win**.
4. Start MFLTWIN in mcp program group
5. Load existing file or create new file

6. To create new file:
    a. Select Channel (0-7)
    b. Number of filters (1-3)
    c. Filter type and parameters for each filters
    d. Gain and sample frequency
    e. Bode plot (if desired)
        i) Select **Options, Bode Plot**
        ii) Start Matlab
        iii) In Matlab, enter: **cd c:\mcp**

            cntrl-V (paste)

            enter
7. Repeat a-e for each additional channel to be used
8. Save file if desired
9. To run MCP, select **Run** pushbutton.
10. To stop MCP, select **Stop** pushbutton

```
% Sample program to construct A, B, C, D matrices

% This program combines three PPF transfer functions
% in parallel to be used with three inputs and three
% outputs.

f1=0.285;   % frequency for first PPF filter
q1=1;       % Q for first PPF filter

f2=0.285;   % frequency for second PPF filter
q2=1;       % Q for second PPF filter

f3=0.285;   % frequency for third PPF filter
q3=2;       % Q for third PPF filter


                    % first filter
b1=[0 0 (2*pi*f1)^2];          % construct transfer function
a1=[1 2*pi*f1/q1 (2*pi*f1)^2]; % b1 is the numerator and
                    % a1 is the denominator
```

```
b1=b1*(-6);                % set gain

b2=[0 0 (2*pi*f1)^2];      % second filter
a2=[1 2*pi*f2/q2 (2*pi*f2)^2];
b2=b2*(-6);


b3=[0 0 (2*pi*f3)^2];      % third filter
a3=[1 2*pi*f3/q3 (2*pi*f3)^2];
b3=b3*(-3);


%bx=conv(b1,b2);   % example how to multiply two transfer
%ax=conv(a1,a2);   % functions together to realize filters
                   % in series


[a1,b1,c1,d1]=tf2ss(b1,a1);    % change transfer functions to
[a2,b2,c2,d2]=tf2ss(b2,a2);    % state space realization
[a3,b3,c3,d3]=tf2ss(b3,a3);


% Append transfer functions to realize the filters
% in parallel. Can only append two matrices at a
% time so it takes two steps for three transfer functions

[a4,b4,c4,d4]=append(a1,b1,c1,d1,a2,b2,c2,d2);
[a,b,c,d]=append(a4,b4,c4,d4,a3,b3,c3,d3);


% change matrices to discrete model
[ad,bd,cd,dd]=c2dm(a,b,c,d,1/1000,'tustin');


[t,a]=balance(ad); % balance scales diagonal to improve
            % eigenvalue accuracy and helps with
            % stability for complex filters. Not
            % necessary for simple filters but does
            % not hurt

ti=inv(t);

ad=ti*ad*t;
bd=ti*bd;
cd=cd*t;
```

```
%cd(1,3:4)=cd(2,3:4);  % example of how to sum two output
%cd=cd(1,:);          % channels into one output
%dd=sum(dd);
```

```
% SAMPLE MATRIX FORMAT
float A[]={  1.0000000e+000,  2.000000e-004,  2.4999980e-009,  1.9982156e-012,  0.0000000e+000,  0.0000000e+000,
  0.0000000e+000,  1.0000000e+000,  2.4999980e-005,  1.9982156e-008,  0.0000000e+000,  0.0000000e+000,
  0.0000000e+000,  0.0000000e+000,  9.9999841e-001,  1.5985725e-003,  0.0000000e+000,  0.0000000e+000,
  0.0000000e+000,  0.0000000e+000, -1.9883314e-003,  9.9821558e-001,  0.0000000e+000,  0.0000000e+000,
  0.0000000e+000,  0.0000000e+000,  0.0000000e+000,  0.0000000e+000,  1.0000000e+000,  2.0000000e-004,
  0.0000000e+000,  0.0000000e+000,  0.0000000e+000,  0.0000000e+000,  0.0000000e+000,  1.0000000e+000};
float B[]={  9.9910779e-013,
  9.9910779e-009,
  7.9928623e-004,
  9.9910779e-001,
  1.0000000e-004,
  1.0000000e+000};
float C[]={  0.0000000e+000,  0.000000e+000, -1.1940633e-002, -9.5439909e-006,  0.000000e+000,  0.0000000e+000};
float D[]={ -4.7719954e-006};
```

```
/* C language program for state space realization */
/* Written by Eric Rohleen, TRW Inc. */
#include <cmath.h>
#include <mcp3io.h>          /*board routines*/

#define NS          6          /*# of states*/
#define          NI                1                /*# of inputs*/
#define NO          1          /*# of outputs*/
#define MAXSZ  6          /*max(NS,NI,NO)*/

#define FS     5000          /*sample rate (hz)*/

/*MATLAB: [a,b,c,d]=butter(2,.1)*/
/*
float A[]={0.6012,-0.2536,0.2536,0.9598};
float B[]={0.3586,0.0568};
float C[]={0.0897,0.6929};
float D[]={0.0201};
*/
extern float A[],B[],C[],D[];
float X[NS];

long    Gain[17];
```

```
        long    In_Samp[16];

        float TV1[MAXSZ],TV2[MAXSZ];

        /*****************************************************************
*********
        c_int10
        1) gets all eight channels
        2) filters those channels which are set to on using the established coeffs
        3) sends out filtered samples
        *****************************************************************
*********/
        void c_int10(void)
        {
        long    i;
        float tmp[8];
        float tout[8];

        /*sample*/
        dataio(Gain,In_Samp);

        /*tmp vector*/
        for(i=0;i<8;i++)
              {
              tmp[i]=In_Samp[i];
              }

        /*C X*/
        #if NO < 2
        vectvec(C,X,TV1,NS);
        #else
        matvec(C,X,TV1,NO,NS);
        #endif

        /*D U*/
        #if NI < 2
        #if NO < 2      /*SISO*/
        TV2[0]=D[0]*tmp[0];
        #else           /*SIMO*/
        convec(D,*tmp,TV2,NO);
        #endif
        #else           /*MISO*/
        #if NO < 2
        vectvec(D,tmp,TV2,NI);
        #else           /*MIMO*/
```

```
matvec(D,tmp,TV2,NO,NI);
#endif
#endif

/*Y = C X + D U*/
#if NO < 2
tout[0]=TV1[0]+TV2[0];
#else
vecpvec(TV1,TV2,tout,NO);
#endif
for(i=0;i<NO;i++)
      In_Samp[i]=tout[i];

/*A X*/
matvec(A,X,TV1,NS,NS);

/*B U*/
#if NI < 2
convec(B,*tmp,TV2,NS);
#else
matvec(B,tmp,TV2,NS,NI);
#endif

/*Xp = A X + B U*/
vecpvec(TV1,TV2,X,NS);
}

void main(void)
{
long i;

initbus();
inittimer(8e6/FS);            /*FS=8e6/val*/
asm(" OR 0201H,IE");          /*200H IS TIMER 01 IS UART (enable timer/uart
ints)*/
asm(" OR 02800H,ST");         /*GIE=2000H  CE=800H (global int enable)*/

for(i=0;i<16;i++)
        Gain[i]=30;
Gain[16]=0x1;                 /*bitslice of on channels*/
while(1)                              /* wait for ISR*/
        {
        ;
        }
}
```

Pin Assignments for MCP III

Power Connector (J7, 9 pin)

1	2	3	4	5	6	7	8	9
+5V	GND	+15V	-15V	-5V	GND	GND	+HV	-HV

Sensor Input(J10, 9 pin)

1	2	3	4	5	6	7	8	9
GND	CH 0	CH 1	CH 2	CH 3	CH 4	CH 5	CH 6	CH 7

Sensor Output(J11, 9 pin)

1	2	3	4	5	6	7	8	9
GND	CH 0	CH 1	CH 2	CH 3	CH 4	CH 5	Pins 8 and 9 not used	

Pin Assignments for Analog Interface Board

50 Pin Connector

Input from sensor(pins 1-9)

1	sensor 1	26	Not used
2	sensor 2	27	Not used
3	sensor 3	28	Not used
4	sensor 4	29	Not used
5	sensor 5	30	Not used
6	sensor 6	31	Not used
7	sensor 7	32	Not used
8	sensor 8	33	Not used
9	GND	34	GND

Analog Board Input to MCP(pins 10-18)		Input from MCP(pins 35-43)	
10	CH 0	35	GND
11	CH 1	36	GND
12	CH 2	37	CH 0
13	CH 3	38	CH 1
14	CH 4	39	CH 2
15	CH 5	40	CH 3
16	CH 6	41	CH 4
17	CH 7	42	CH 5
18	GND	43	GND

		High Voltage Output to Actuator(pins 44-50)	
19	Not used	44	actuator 1
20	Not used	45	actuator 2
21	Not used	46	actuator 3
22	Not used	47	actuator 4
23	Not used	48	actuator 5
24	Not used	49	actuator 6
25	Not used	50	GND

Analog Interface Board Power Connector (9 pin)

1	2	3	4	5	6	7	8	9
+5V	GND	+15V	-15V	-5V	GND	GND	+150	-150

APPENDIX B. [MATLAB CODE FOR DATA REDUCTION]

```
% FFTLAZ

% Written by Dr Gangbing Song and LCDR Steve Schmidt, Jan 97

% fft of beam tip vibrations
% x is the datafile name, mat file

%function []=fftlaz(x)


laser1= trace_y(1,1:100);
laser1=laser1';
laser2= trace_y(2,401:500);
laser2=laser2';

%Y=fft(laser,512);
%Pyy=Y.*conj(Y)/512;
%f=1000*(0:255)/512;
%figure;
        %plot(f,Pyy(1:256))



% MRED

% Written by Dr Gangbing Song and LCDR Steve Schmidt, Jan 97

% mode vibration reduction rate

clear
load 1api  %load mat file
ampf= trace_y(2,1:200);  %"f" for first
ampf=ampf';
[pxxf,ff]=psd(ampf,[],100);

%identify mode for the first 2 second
[mode,amode]=mid(ampf);

amp_1st=sum(abs(ampf(1:100)))/100;

figure;
plot(ff, 10*log10(pxxf))
hold on
plot(mode,10*log10(amode),'c+')
```

```
zoom on

clear ans1
ans1=input('Are any modes unimportant? (y/n)','s');
if strcmp(ans1,'y')==1
   num_mode_unimp=input('How many unimportant modes are included,
num_mode_unimp=');
   for i=1:num_mode_unimp
     mode_unimp(i)=input(['Enter No.' num2str(i) ' unimportant mode is (1, 2, 3,...)']);
   end
end

%delete the unimportant modes
for k=1:length(mode_unimp)
  j=length(mode_unimp)-k+1;
 tempa=mode(1:(mode_unimp(j)-1));
 tempb=mode((mode_unimp(j)+1):length(mode));
 mode=[tempa,tempb];
  tempa1=amode(1:(mode_unimp(j)-1));
  tempb1=amode((mode_unimp(j)+1):length(mode));
  amode=[tempa1,tempb1];
end


clg
plot(ff, 10*log10(pxxf))
hold on
plot(mode,10*log10(amode),'c+')
zoom on

mode
amode

clear ans2
ans2=input('Are all important modes picked? (y/n)','s');
if strcmp(ans2,'n')==1
  num_mode_missing=input('How many important modes are missing,
num_mode_missing=');
    for i=1:num_mode_missing
      mode_new(i)=input(['Enter No.' num2str(i) ' new mode (Hz) is, mode_new('
num2str(i) ')= ']);
    end

  %find the corresponding amode_new
  for j=1:(length(mode_new))
```

```
    for i=1:(length(pxxf)-2)
      if mode_new(j)>ff(i) & mode_new(j)<ff(i+1)
        slopef(j)=(pxxf(i+1)-pxxf(i))/(ff(i+1)-ff(i));
        amode_new(j)=slopef(j)*(mode_new(j)-ff(i))+pxxf(i); %m for modified
         if amode_new(j)<=pxxf(i)
           amode_new(j)=pxxf(i);
           mode_new(j)=ff(i);
           mark=2;
         else
           amode_new(j)=pxxf(i+1);
           mode_new(j)=ff(i+1);
           mark=1;
         end
       end
       if mode_new(j) == ff(i)
         if pxxf(i)<pxxf(i+1)
           amode_new(j)=pxxf(i+1);
           mode_new(j)=ff(i+1);
         elseif i>1
           if pxxf(i-1)>pxxf(i)
                    amode_new(j)=pxxf(i-1);
             mode_new(j)=ff(i-1);
           end
         else
                  amode_new(j)=pxxf(i);
           mode_new(j)=ff(i);
         end
       end
     end
   end

   plot(mode_new,10*log10(amode_new),'m+');

   mode_old=mode;  %mode generated by mid function
   amode_old=amode;
   clear amode;
   mode_comb_unsorted=[mode_old,mode_new];
   [mode,ind_mode_comb]=sort(mode_comb_unsorted); %now mode including both old
and new
   for k=1:length(mode)
    if ind_mode_comb(k) <= length(mode_old);
      amode(k)=amode_old(ind_mode_comb(k));
    else
      amode(k)=amode_new(ind_mode_comb(k)-length(mode_old));
    end
```

```
  end

 mode
 amode

end  % math the if need for new modes

%for the 13-15th seconds
ampl= trace_y(2,1301:1500);  %"l" for last
ampl=ampl';
[pxxl,fl]=psd(ampl,[],100);

%indentify modes using function mid for last
[mode_last,amode_last]=mid(ampl);

amp_last=sum(abs(trace_y(2,1401:1500)))/100-sum(trace_y(2,1401:1500))/100;

for j=1:(length(mode))
 for i=1:(length(pxxl)-2)
   if mode(j)>fl(i) & mode(j)<fl(i+1)
     slope(j)=(pxxl(i+1)-pxxl(i))/(fl(i+1)-fl(i));
     amode_last_m(j)=slope(j)*(mode(j)-fl(i))+pxxl(i); %m for modified
   end
   if mode(j) == fl(i)
     amode_last_m(j)=pxxl(i);
   end
 end
end

%figure;
%plot(ff, 10*log10(pxxf))
title('LOG, red - last');
%hold on
%plot(mode,10*log10(amode),'c+')


%zoom on
plot(fl, 10*log10(pxxl),'r')
plot(mode_last,10*log10(amode_last),'b+')
%zoom on



%figure;
%plot(ff, pxxf, )
```

```
%hold on
%plot(mode,amode,'c+')
%zoom on



%ampred=(amp_1st-amp_last)/amp_1st;
for j=1:(length(mode))
   dbdrop_mode(j)=10*(log10(amode(j))-log10(amode_last_m(j)));
end
dbdrop_amp=10*(log10(amp_1st)-log10(amp_last));

%red1=(amode1-amode1_last)/amode1;
dbdrop_mode
dbdrop_amp



% MID

% Written by Dr Gangbing Song and LCDR Steve Schmidt, Jan 97


% A Function used to to identify modes


%for the first two seconds,

%ampf= trace_y(2,1:200);  %"f" for first
%ampf=ampf';


function [mode,amode]=mid(ampf);

if exist('tmode')==1,
  clear tip ftip bot fbot ampf pxxf tmode tamode fall ind
end



[pxxf,ff]=psd(ampf,[],100);     %power spectral density of the first two seconds
ff=round(ff*100)/100;%round off the frequency of .01 accuracy
lenampf=length(ff);              % length of vector ampf

%mark the tip and the bottom
k=0;
```

```
j=0;
for i=1:(lenampf-2)
  %deal with the first  point
  if i==1,
    if pxxf(i)>pxxf(i+1)  %first point is the tip
      k=k+1;
      fp_tip=1;
      tip(k)=pxxf(i);  %%%
      ftip(k)=ff(i);
      ind_tip(k)=i; %returns index in pxxf
    end
    if pxxf(i)<pxxf(i+1) %first point is the bottom
        j=j+1;
      fp_bot=1;
      bot(j)=pxxf(i);   %%%
      fbot(j)=ff(i);
    end
  end

  %deal with other points
  if i>1 &  pxxf(1)>pxxf(2)
    %find the tip
    if pxxf(i+1)>pxxf(i) & pxxf(i+2)<pxxf(i+1)
      k=k+1;
      tip(k)=pxxf(i+1);
        ftip(k)=ff(i+1);
      ind_tip(k)=i;
    end
    %find the bottom
    if pxxf(i)<pxxf(i-1) & pxxf(i+1)>pxxf(i)
        j=j+1;
      bot(j)=pxxf(i);
      fbot(j)=ff(i);
    end
  end

if i>1 & pxxf(1)<pxxf(2)
    %find the tip
    if pxxf(i)>pxxf(i-1) & pxxf(i+1)<pxxf(i)
      k=k+1;
      tip(k)=pxxf(i);
        ftip(k)=ff(i);
      ind_tip(k)=i;
    end
    %find the bottom
```

```
    if pxxf(i)<pxxf(i-1) & pxxf(i+1)>pxxf(i)
          j=j+1;
      bot(j)=pxxf(i);
      fbot(j)=ff(i+1);
    end
  end


  %deal with the last point
  if i==lenampf,
    if pxxf(i)>pxxf(i-1)  %last point is a tip
      k=k+1;
      lp_tip=1;
      tip(k)=pxxf(i);
      ftip(k)=ff(i);
      ind_tip(k)=i;
    end
    if pxxf(i)<pxxf(i-1) %last point is a bottom
          j=j+1;
      lp_bot=1;
      bot(j)=pxxf(i);
      fbot(j)=ff(i);
    end
  end

end

lentip=length(tip);
lenbot=length(bot);

%calculate the fall
if pxxf(1)>pxxf(2)
 for ii=1:lenbot-1
    fall(ii)=abs(tip(ii)-bot(ii));
 end
else
 for ii=1:lenbot-1
    fall(ii)=abs(tip(ii)-bot(ii+1));
 end
end

%for ii=1:k
%  if ii==1 & fp_tip==1
%     fall(ii)=abs(tip(ii)-bot(ii));
%  elseif ii==k & lp_tip==1
```

```
%     fall(ii)=abs(tip(ii)-bot(ii-1));
%   fall(ii)=max([abs(tip(ii)-bot(ii)),abs(tip(ii)-bot(ii+1))]);
%end

kk=lenbot-1;

%identify the modes
[amp,ind]=sort(fall); %ind: index of fall, tip
tmi=0;  %temp mode index
for jj=1:5
  if ind(kk+1-jj)==1;
     if ftip(ind(kk+1-jj))>0.5    %avoid to pick 0 freq as a mode
        tmi=tmi+1;
        tmode(tmi)=ftip(ind(kk+1-jj));
        tamode(tmi)=tip(ind(kk+1-jj));
     end
  end
  if ind(kk+1-jj)>1 & ind(kk+1-jj)<kk-1
   %if tip(ind(kk+1-jj))>tip(ind(kk+1-jj)-1) &
   if tip(ind(kk+1-jj))>tip(ind(kk+1-jj)+1)  %tip higher than next tip
      if ftip(ind(kk+1-jj))>0.5   %avoid to pick 0 freq as a mode
         if ftip(ind(kk+1-jj))<10
            if  fall(ind(kk+1-jj))>1e-5
                        tmi=tmi+1;
                        tmode(tmi)=ftip(ind(kk+1-jj));
                        tamode(tmi)=tip(ind(kk+1-jj));
            end
             elseif ftip(ind(kk+1-jj))<20
            if fall(ind(kk+1-jj))>5e-5  %ps density should be higher than 1e-5
                        tmi=tmi+1;
                            tmode(tmi)=ftip(ind(kk+1-jj));
                            tamode(tmi)=tip(ind(kk+1-jj));
            end
             elseif ftip(ind(kk+1-jj))<30
            if fall(ind(kk+1-jj))>5e-6  %ps density should be higher than 1e-6
                        tmi=tmi+1;
                            tmode(tmi)=ftip(ind(kk+1-jj));
                            tamode(tmi)=tip(ind(kk+1-jj));
            end
         else
                    if fall(ind(kk+1-jj))>1e-7  %ps density should be higher than 1e-7
                        tmi=tmi+1;
                            tmode(tmi)=ftip(ind(kk+1-jj));
                            tamode(tmi)=tip(ind(kk+1-jj));
                    end
```

```
        end

      end
    end
  end
end  %for



%sort the first four mode
[mode,mi]=sort(tmode);
mode;
amode=tamode(mi);
log_amode=10*log10(tamode(mi));

mark1=0;
mark2=0;
clear mark1
clear mark2
clear yfrise xfrise ybrise xbrise fslope bslope

% modify the the modes
for j=1:length(mode)
 for i=1:lenampf
  if mode(j)==ff(i) & i>2
    ybrise(j)=pxxf(i)-pxxf(i-1);
    xbrise(j)=ff(i)-ff(i-1);
    yfrise(j)=pxxf(i+1)-pxxf(i);
    xfrise(j)=ff(i+1)-ff(i);
    fslope(j)=abs(yfrise(j)/xfrise(j));
    bslope(j)=abs(ybrise(j)/xbrise(j));
         if abs(bslope(j))<abs(fslope(j))  %rising
       if abs(ybrise(j))<0.06*abs(xbrise(j))
         %if bslope<0.02
           mode(j)=ff(i-1)+(1+bslope(j))*xbrise(j)/2;
           mark1(j)=i;
         %end
       end
     end
     if abs(bslope(j))>abs(fslope(j))  %down
       if abs(yfrise(j))<0.04*abs(xfrise(j))
         %if fslope<0.02
           mode(j)=ff(i)+(1-fslope(j))*xfrise(j)/2;
```

```
            mark2(j)=i;
          %end
        end
      end
  end
 end
end

%mode
%amode
%log_amode

%figure;
%plot(ff, 10*log10(pxxf))
%title('LOG');
%hold on
%plot(mode,log_amode,'c+')
%zoom on

%figure;
%plot(ff, pxxf)
%hold on
%plot(mode,amode,'c+')
%zoom on

%rvd=10*log10(lmax);
%fm=ffmax;



% MEID
% Written by Dr Gangbing Song and LCDR Steve Schmidt, Jan 97

% plot the bodeplot using input and output data
% using model-establishing method

z2=[trace_y(2,1:500)' trace_y(3,1:500)'];
z2=dtrend(z2);

th=arx(z2,[2 2 3]);
th=sett(th,0.01);
gth=th2ff(th);
figure;
bodeplot(gth);
```

APPENDIX C. [FINITE ELEMENT MATLAB CODE]

```
% Matlab Finite Element Code
%
% This program models a simple beam with piezoceramics bonded to it.
%
% LCDR Steve Schmidt USN, Jan 97
%
% This is a 10 element model with piezo actuators located on elements
% 2, 6, 8, and a piezo sensor on element 4.
%

clear;clc;format short e


% Beam properties

numelem=10; % number of elements
length=[0.1016 0.0635 0.028575 0.03175 0.028575 0.0635 0.08255 0.0635
0.356 0.356];
                        % length of beam in meters

thick=0.00152;              % thickness in meters
height=0.0762;              % height in meters
density=2800;               % density in Kg/m^3
elem_mass=density*height*thick;    % Mass/length
elem_inertia=1/12*thick^3*height;  % Meter^4
E=72e9;                     % Mod of elasticity, N/m^2


% Compute element mass and stiffness matrices

Kelem=0*ones(4,4*numelem);
Melem=0*ones(4,4*numelem);

for i=1:numelem,...
    Kelem(:,4*i-3:4*i)=E*elem_inertia/length(i)^3*[12 6*length(i) -12
6*length(i);...
    6*length(i) 4*length(i)^2 -6*length(i) 2*length(i)^2;-12 -6*length(i) 12 -
6*length(i);...
    6*length(i) 2*length(i)^2 -6*length(i) 4*length(i)^2];
end
```

```
for i=1:numelem,...
   Melem(:,4*i-3:4*i)=elem_mass*length(i)/420*[156 22*length(i) 54 -
13*length(i);...
   22*length(i) 4*length(i)^2 13*length(i) -3*length(i)^2;54 13*length(i) 156 -
22*length(i);...
   -13*length(i) -3*length(i)^2 -22*length(i) 4*length(i)^2];
end

% Construct global mass and stiffness matrices

Kmatrix=0*ones(2*(numelem+1));
Mmatrix=0*ones(2*(numelem+1));

for i=1:numelem,...
      Mmatrix(2*i-1:2*i+2,2*i-1:2*i+2)=Mmatrix(2*i-1:2*i+2,2*i-1:2*i+2) +
Melem(:,4*i-3:4*i);
end

for i=1:numelem,...
      Kmatrix(2*i-1:2*i+2,2*i-1:2*i+2)=Kmatrix(2*i-1:2*i+2,2*i-1:2*i+2) +
Kelem(:,4*i-3:4*i);
end


%% For cantilevered beam (row 1,2 & column 1,2 = 0)

Mmatrix=Mmatrix(3:2*(numelem+1),3:2*(numelem+1));
Kmatrix=Kmatrix(3:2*(numelem+1),3:2*(numelem+1));

%Mmatrix(5,5:8)=Mmatrix(5,5:8)-Melem(1,13:16);% account for elbow
%Mmatrix(6:8,5)=Mmatrix(6:8,5)-Melem(2:4,13); % subtract row/col

%Kmatrix(5,5:8)=Kmatrix(5,5:8)-Kelem(1,13:16);% account for elbow
%Kmatrix(6:8,5)=Kmatrix(6:8,5)-Kelem(2:4,13); % subtract row/col


%%Define the piezo actuator and sensor elements

delta=thick/2;
tp=2*1.905e-4;
wp=0.038;
Ep=6.3e10;
d31=-1.8e-10;
eT3=1.5e-8;
pdensity=7700;
```

```
Kpiezo=wp*tp*Ep*(delta^2 + delta*tp + (tp^2)/3);

length(3)=0.3175;

Ka=Kpiezo/length(2)*[12/length(2)^2 6/length(2) -12/length(2)^2 6/length(2);...
 6/length(2) 4 -6/length(2) 2;-12/length(2)^2 -6/length(2) 12/length(2)^2 -
6/length(2);...
 6/length(2) 2 -6/length(2) 4];
Ks=Kpiezo/length(3)*[12/length(3)^2 6/length(3) -12/length(3)^2 6/length(3);...
 6/length(3) 4 -6/length(3) 2;-12/length(3)^2 -6/length(3) 12/length(3)^2 -
6/length(3);...
 6/length(3) 2 -6/length(3) 4];

Mp=pdensity*wp*tp;                          %%Mass/length piezo
%
Ma=Mp*length(2)/420*[156 22*length(2) 54 -13*length(2);22*length(2)
4*length(2)^2 13*length(2) -3*length(2)^2;...
 54 13*length(2) 156 -22*length(2);-13*length(2) -3*length(2)^2 -22*length(2)
4*length(2)^2];

Ms=Mp*length(3)/420*[156 22*length(3) 54 -13*length(3);22*length(3)
4*length(3)^2 13*length(3) -3*length(3)^2;...
 54 13*length(3) 156 -22*length(3);-13*length(3) -3*length(3)^2 -22*length(3)
4*length(3)^2];

%%Add the piezo eleMent Mass and stiffness Matrices to the structure

Mmatrix(1:4,1:4)=Mmatrix(1:4,1:4)+2*Ma(1:4,1:4);
Mmatrix(5:8,5:8)=Mmatrix(5:8,5:8)+2*Ms;
Mmatrix(9:12,9:12)=Mmatrix(9:12,9:12)+2*Ma(1:4,1:4);
Mmatrix(13:16,13:16)=Mmatrix(13:16,13:16)+2*Ma;

Kmatrix(1:4,1:4)=Kmatrix(1:4,1:4)+2*Ka(1:4,1:4);
Kmatrix(5:8,5:8)=Kmatrix(5:8,5:8)+2*Ks;
Kmatrix(9:12,9:12)=Kmatrix(9:12,9:12)+2*Ka(1:4,1:4);
Kmatrix(13:16,13:16)=Kmatrix(13:16,13:16)+2*Ka;

%clear Ks Ka Ms Ma Kpiezo;

%%Solve for natural frequencies and Mode shapes.

[omega2,Phi,Psi]=eign(Kmatrix,Mmatrix);
omega=sqrt(omega2);
Hertz=omega/(2*pi);
ttl=str2mat('   Omega','Hertz','        ');
```

[ttl(1,:) ttl(3,:) ttl(2,:)]
[omega Hertz]

LIST OF REFERENCES

1. Agrawal, B. N., and Bang, H., "Active Vibration Control of Flexible Space Structures by using Piezoelectric Sensors and Actuators, Proceedings for the 14th Biennial ASME Conference, Albuquerque, NM, September, 1993.

2. Bronowicki, A. J. and Rohleen E., *Modular Control Patch (MCP), Final Report*, TRW Space and Electronics Group, Redondo Beach, CA.

3. Texas Instruments, Inc., TMS320C3x, User's Guide, 1992.

4. Chaudhry Z. and Rogers, C., Actuators for Smart Structures, *Fiber Optic Smart Structures*, John Wiley & Sons, Inc., New York, 1995.

5. Agrawal, B. N., Bang, H., and Jones, E., "Application of Piezoelectric Actuators and Sensors in the Vibration Control of Flexible Spacecraft Structures," Forty Third Congress of the International Astronautical Federation, Washington, D.C., September, 1992

6. Allen, D. H. and Haisler, W. E., *Introduction to Aerospace Structural Analysis*, John Wiley & Sons, Inc., 1985.

7. Franklin, G. F., Powell, D. J., and Emami-Naeini, A., *Feedback Control of Dynamic Systems*, Addison-Wesley Publishing Company, Inc., 1994.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center 8725 John J. Kingman Road, Suite 0944 Ft. Belvoir, VA 22060-6218	2
2.	Dudley Knox Library Naval Postgraduate School 411 Dyer Road Monterey, CA 93943-5002	2
3.	Chairman, Code AA Department of Aeronautics and Astronautics Naval Postgraduate School Monterey, CA 93943	1
4.	Professor Brij N. Agrawal, Code AA/Ag Department of Aeronautics and Astronautics Naval Postgraduate School Monterey, CA 93943	2
5.	Dr. Gangbing Song, Code AA/Gs Department of Aeronautics and Astronautics Naval Postgraduate School Monterey, CA 93943	2
6.	LCDR Steve Schmidt 885 Buckeye Lane West Jacksonville, FL 32259	2