Spring 1999

Vibration control on linear robots with digital servocompensator

Roger Kobla Kwadzogah

New Jersey Institute of Technology

Follow this and additional works at: https://digitalcommons.njit.edu/theses

Part of the Electrical and Electronics Commons

Recommended Citation

https://digitalcommons.njit.edu/theses/859

This Thesis is brought to you for free and open access by the Theses and Dissertations at Digital Commons @ NJIT. It has been accepted for inclusion in Theses by an authorized administrator of Digital Commons @ NJIT. For more information, please contact digitalcommons@njit.edu.
Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement.

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation.

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen.
The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.
ABSTRACT

VIBRATION CONTROL ON LINEAR ROBOTS WITH DIGITAL SERVOCOMPENSATOR

by
Roger Kobla Kwadzogah

Control application for active damping of structural vibrations and acoustic noise in mechanical systems is one of the engineering fields that can benefit from advances made in digital signal processors. This thesis project is one such application. It is about a vibration control at the loading point of a high speed linear robotic workcell. A lead zirconate titanate piezoelectric ceramic is used as the actuator and an accelerometer provides the sensing. From experimentally measured frequency response of this system, a shaping filter is designed and added on. The reshaped system is fitted with a third order transfer function design model. And based on this model, a discrete-time control scheme designated "servocompensator" is designed and implemented on a Digital Signal Processing board to control structural vibrations on the robotic workcell. Servocompensator is a control scheme based on the principle of Internal Model Design.

The results have demonstrated the servocompensator as a powerful scheme for controlling independently the individual modes within the spectrum of a given vibration signal. In a typical result, as much as 40 dB of attenuation is produced on the targeted mode, where 0 dB is equal to 1 g of acceleration in this application. Furthermore, with the multi-tasking capability of the digital hardware, multiple mode control is demonstrated by multiplexing a number of single-mode servocompensators.
VIBRATION CONTROL ON LINEAR ROBOTS WITH DIGITAL SERVOCOMPENSATOR

by
Roger Kobla Kwadzogah

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering

Department of Electrical and Computer Engineering

May 1999
VIBRATION CONTROL ON LINEAR ROBOTS WITH DIGITAL SERVOCOMPENSATOR

Roger Kobla Kwadzogah

Dr. Timothy Chang, Thesis Advisor
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Andrew Meyer, Committee Member
Professor of Electrical and Computer Engineering, NJIT

Dr. Edwin Hou, Committee Member
Associate Professor of Electrical and Computer Engineering, NJIT
BIOGRAPHICAL SKETCH

Author: Roger Kobla Kwadzogah

Degree: Master of Science in Electrical Engineering

Date: May 1999

Undergraduate and Graduate Education:

- Master of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 1999

- Bachelor of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 1997

Major: Electrical Engineering
To my family: Cathy, Tina and Toni
ACKNOWLEDGMENT

My deepest respect, love, admiration and gratitude go to Dr. Timothy N. Chang, my advisor, for his trust in me, his guidance, his support and his assurance. I also like to extend special appreciation to the committee members Dr. Andrew Meyer and Dr. Edwin Hou. I like to thank my wife Cathy and my two children, Tina and Toni for all the moral support. I will use this opportunity to extend my respect and love for my brother-in-law and my friend, Mr Daniel Yador, without whose confidence in my academic ability and without whose support I would not have been here today. Many thanks are also due to many fellow students, for the company. I must also extend my gratitude to Mr. Kedar Godbole and Mr. Vincenzo Pappano for their invaluable advice and their effort in writing the 'C' functions necessary to interface the Dalanco board with 'C'language. I really appreciate the financial support afforded me from the NIST ATP Grant number 70NANB5H1092 for Precision Optoelectronics Assembly.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Objective</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Background</td>
<td>2</td>
</tr>
<tr>
<td>1.2.1 A Brief Survey of Micromotion Control Methods</td>
<td>3</td>
</tr>
<tr>
<td>1.3 General System Description</td>
<td>4</td>
</tr>
<tr>
<td>2 SYSTEM HARDWARE DESCRIPTION</td>
<td>6</td>
</tr>
<tr>
<td>2.1 The Robot Hardware</td>
<td>6</td>
</tr>
<tr>
<td>2.1.1 Motion of the Robot Modules and Structural Vibration</td>
<td>7</td>
</tr>
<tr>
<td>2.2 The Control System Hardware</td>
<td>8</td>
</tr>
<tr>
<td>2.2.1 The Vibration Actuator</td>
<td>9</td>
</tr>
<tr>
<td>2.2.2 The Vibration Sensor</td>
<td>9</td>
</tr>
<tr>
<td>2.2.3 The Power Amplifier</td>
<td>10</td>
</tr>
<tr>
<td>2.3 The Digital Hardware</td>
<td>11</td>
</tr>
<tr>
<td>2.3.1 The TMS320C31 DSP</td>
<td>11</td>
</tr>
<tr>
<td>2.3.2 Programming the Dalanco Board</td>
<td>12</td>
</tr>
<tr>
<td>3 MODELLING THE SYSTEM DYNAMICS</td>
<td>15</td>
</tr>
<tr>
<td>3.1 Derivation of the Design Model</td>
<td>15</td>
</tr>
<tr>
<td>3.1.1 Experiment to Measure the Frequency Response of the System</td>
<td>15</td>
</tr>
<tr>
<td>3.1.2 Frequency Response of the Open-loop System</td>
<td>16</td>
</tr>
<tr>
<td>3.1.3 Model for the Open-loop System</td>
<td>18</td>
</tr>
<tr>
<td>3.1.4 Analog Shaping/Antialiasing Filter</td>
<td>18</td>
</tr>
<tr>
<td>3.1.5 Frequency Response of the Filtered System</td>
<td>19</td>
</tr>
<tr>
<td>4 THEORY OF THE SERVOCOMPENSATOR</td>
<td>22</td>
</tr>
<tr>
<td>4.1 Concept of the Servocompensator</td>
<td>22</td>
</tr>
<tr>
<td>Chapter</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.2 Model of Deterministic Disturbance Signals</td>
<td>23</td>
</tr>
<tr>
<td>4.3 Servocompensator Design for a General Disturbance</td>
<td>24</td>
</tr>
<tr>
<td>4.4 Single-mode Servocompensator Design for Vibration Control</td>
<td>28</td>
</tr>
<tr>
<td>4.4.1 Discrete-Time Design</td>
<td>28</td>
</tr>
<tr>
<td>4.4.2 Designing the Stabilizing Gain for the Augmented Plant-disturbance System</td>
<td>30</td>
</tr>
<tr>
<td>4.4.3 Observer Design</td>
<td>31</td>
</tr>
<tr>
<td>4.5 Double-mode Servocompensator Design for Vibration Control</td>
<td>33</td>
</tr>
<tr>
<td>4.6 Multiplexing a Number of Single-mode Servocompensitors</td>
<td>35</td>
</tr>
<tr>
<td>5 CONTROL IMPLEMENTATION SOFTWARE</td>
<td>38</td>
</tr>
<tr>
<td>5.1 The Adept VME Motion Interface Controller</td>
<td>38</td>
</tr>
<tr>
<td>5.2 The Control Implementation Algorithm</td>
<td>38</td>
</tr>
<tr>
<td>5.3 Control Loop for the Single-mode Servocompensator</td>
<td>39</td>
</tr>
<tr>
<td>5.4 Multiplexing Single-mode Servocompensators</td>
<td>40</td>
</tr>
<tr>
<td>6 TEST RESULTS</td>
<td>43</td>
</tr>
<tr>
<td>6.1 Results for a Single-mode Servocompensator</td>
<td>43</td>
</tr>
<tr>
<td>6.2 Results for Two-Multiplexed Single-mode Servocompensators</td>
<td>43</td>
</tr>
<tr>
<td>6.2.1 Results for a Double-mode Servocompensator</td>
<td>44</td>
</tr>
<tr>
<td>6.3 Results for Three-Multiplexed Single-mode Servocompensators</td>
<td>45</td>
</tr>
<tr>
<td>6.4 Results for Four-Multiplexed Single-mode Servocompensators</td>
<td>46</td>
</tr>
<tr>
<td>6.5 Results for Multiplexing More Than Four Single-mode Servocompensators</td>
<td>46</td>
</tr>
<tr>
<td>7 CONCLUSIONS AND RECOMMENDATION</td>
<td>48</td>
</tr>
<tr>
<td>7.1 Conclusions</td>
<td>48</td>
</tr>
<tr>
<td>7.2 Recommendations</td>
<td>48</td>
</tr>
<tr>
<td>APPENDIX A PARAMETER SPECIFICATIONS</td>
<td>49</td>
</tr>
<tr>
<td>APPENDIX B DESIGN MATLAB M-FILE</td>
<td>51</td>
</tr>
<tr>
<td>APPENDIX C SAMPLE 'C' PROGRAMS FOR THE CONTROL LOOP</td>
<td>54</td>
</tr>
<tr>
<td>Chapter</td>
<td>Page</td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>60</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Block Diagram of the Robotic Workcell</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Dominant Band of Modes of the Vibrations Observed on the Robot</td>
<td>8</td>
</tr>
<tr>
<td>2.3 Block Diagram of the Plant</td>
<td>8</td>
</tr>
<tr>
<td>2.4 Block Diagram of the Dalanco Board Inside the Computer</td>
<td>11</td>
</tr>
<tr>
<td>3.1 Experimental Setup for Frequency Response Measurement</td>
<td>16</td>
</tr>
<tr>
<td>3.2 Frequency Response of the Open-loop System: no Filter</td>
<td>17</td>
</tr>
<tr>
<td>3.3 Unfiltered Frequency Response: Experimental and Theoretical</td>
<td>19</td>
</tr>
<tr>
<td>3.4 The Shaping/Antialiasing Filter</td>
<td>20</td>
</tr>
<tr>
<td>3.5 Frequency Response of the Filtered System</td>
<td>20</td>
</tr>
<tr>
<td>3.6 Filtered Frequency Response: Experimental and Theoretical</td>
<td>21</td>
</tr>
<tr>
<td>4.1 Configuration of the Servocompensator</td>
<td>23</td>
</tr>
<tr>
<td>4.2 Servocompensator with an Observer-based State Feedback</td>
<td>27</td>
</tr>
<tr>
<td>4.3 Simulated Closed Loop Response of the Single-mode Servocompensator for $R = 0.00001$</td>
<td>30</td>
</tr>
<tr>
<td>4.4 Simulated Closed Loop Response of the Single-mode Servocompensator for $R = 0.00005$</td>
<td>31</td>
</tr>
<tr>
<td>4.5 Simulated Closed Loop Response of the Single-mode Servocompensator for $R = 0.00001$</td>
<td>31</td>
</tr>
<tr>
<td>4.6 Simulated Closed Loop Response of the Single-mode Servocompensator for $R = 0.00005$</td>
<td>32</td>
</tr>
<tr>
<td>4.7 Simulated Closed Loop Response of the Single-mode Servocompensator for $R = 0.001$</td>
<td>32</td>
</tr>
<tr>
<td>4.8 Simulated Frequency Response the Closed Loop System with the Double-mode Servocompensator</td>
<td>35</td>
</tr>
<tr>
<td>4.9 Block Diagram of $N$-Multiplexed Single-mode Servocompensators</td>
<td>36</td>
</tr>
<tr>
<td>4.10 Simulated Response for Seven-Multiplexed Single-modes Servocompensators</td>
<td>37</td>
</tr>
<tr>
<td>5.1 Flowchart of the Control Program</td>
<td>39</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>5.2</td>
<td>Block Diagram of Single-mode Control Loop</td>
</tr>
<tr>
<td>5.3</td>
<td>Block Diagram of the Multiplexing Control Loop</td>
</tr>
<tr>
<td>6.1</td>
<td>Response for a Single-mode Servocompensator</td>
</tr>
<tr>
<td>6.2</td>
<td>Response for Two-Multiplexed Single-mode Servocompensators</td>
</tr>
<tr>
<td>6.3</td>
<td>Response for a Double-mode Servocompensator</td>
</tr>
<tr>
<td>6.4</td>
<td>Response for Three-Multiplexed Single-mode Servocompensators</td>
</tr>
<tr>
<td>6.5</td>
<td>Response for Four-Multiplexed Single-mode Servocompensators</td>
</tr>
<tr>
<td>6.6</td>
<td>Response for Five-Multiplexed Single-mode Servocompensators</td>
</tr>
<tr>
<td>6.7</td>
<td>Response for Six-Multiplexed Single-mode Servocompensators</td>
</tr>
<tr>
<td>6.8</td>
<td>Response for Seven-Multiplexed Single-mode Servocompensators</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

1.1 Objective

The objective of this thesis is to present a control strategy that can be used to digitally drive piezoelectric actuators for control of structural vibrations on mechanical systems. This work is motivated by applications such as the NIST Precision Optoelectronics Assembly Project where an optoelectronic component is carried or positioned by a high speed linear robot whose acceleration may be as high as 3 g. This rapid slewing motion inevitably excites the structural resonance of the mechanical members of the robot and consequently degrades the performance of the system. Conventional vibration control methodologies, based on feedback stabilization of the platform, are not applicable due to bandwidth limitation of the robot's actuators. Now since the assembly process requires vibrational suppression at the load (component) only, it is significantly more efficient to insert a microactuator, such as piezoelectric ceramic stack, between the robotic platform and the load so that the vibration suppression can be carried out by the microactuator. In this approach, the microactuator is considered as the plant while the robotic platform is considered as the disturbance or the vibration generator which produces a spectrum of harmonics. The amplitude and phase of the disturbance signals are generally unknown but the frequencies are well defined. Therefore, the robust servomechanism method [1] is introduced to attenuate these disturbance signals. This control methodology which based on the principle of Internal Model Design will be referred to as servocompensator control.

In this method, an internal model of the disturbance is incorporated into the compensator to produce an augmented plant-disturbance system which is then stabilized with a feedback. For the case of vibration control, as undertaken in this project, the exogenous disturbances (the vibrations) are sinusoidal, signal modes
generated by the x-y linear robot during high speed slewing motion. The plant, in
this application, consists of a pzt stack actuator with an accelerometer serving as the
sensor. The designed servocompensator is implemented inside a Data Acquisition
and Signal Processing Board hosted inside an IBM-compatible personal computer.
The performance of the servocompensator is measured by comparing the frequency
response of the plant in open loop to that with the servocompensator running. In
a typical result, where the servocompensator is designed as a single-mode signal
controller, as much as 40 dB of attenuation is achieved in the amplitude of the
targeted signal mode. The success of this result — one servocompensator per one
signal mode— motivates a multi-loop implementation concept where a number of
signal modes within the vibration spectrum are concurrently controlled using as
many number of multiplexed single-mode servocompensators with each servocom-
pensator designed to control one and only one signal mode.

1.2 Background

High precision motion control is an important technology with scientific and
industrial applications. At the root of the success of most high technologies is the
improved methods of high resolution motion control. Some of these precision motion-
based technologies include integrated circuit manufacturing, optical mounting, fine-
pointing for spacecraft altitude control, assembling and manufacturing in optoelec-
tronics and operation of high-density magnetic data storage, to name just a few.
The ever increasing demand on the performance of these technologies, obviously,
has led to increased research for more and more precise motion control schemes.
Futami et al using a single-axis stage mechanism actuated by an AC linear motor
was able to achieve position in 1 nm with maximum range of 0.25 m by designing
controller around several microdynamic regions [13]. The most critical element
in micromotion control is the actuator and also the most difficult to design. The
physical characteristics of the motion control problem determine which actuator technology is best. In addition to the mechanical stiffness of the load, a primary factor in selecting the best actuator technology is the load spectrum, which is the frequency spectrum of the force that the actuator must deliver to the load when the control is operating [12]. If the application requires large displacements and low forces, then the actuator technology must be able to exert comparable displacement and force over the frequencies in which the load force is largest. On the other hand, if the application requires small displacements but low forces, a different actuator technology would be more suitable. Some the available actuator technologies include the low-stiffness type such as the moving coil actuator, as in a loudspeaker, and the variable reluctance actuator also known as the solenoid. Then is the high-stiffness solid-state actuators made from piezoelectric ceramics. Piezoelectric actuators have gained widespread use in the aerospace industry. They possess the desirable qualities for high force, high bandwidth and extremely precise positioning capabilities, making them ideal for precision motion control for both dynamic and static applications [12].

1.2.1 A Brief Survey of Micromotion Control Methods

The goal of a actuator controller is to obtain a precise positioning, independent of disturbances such as power supply voltage, friction, backlash or reactive force from the manipulated mechanism or medium. The performance of the conventional fixed-gain linear feedback controllers are usually limited in controlling actuator motion. These controllers, in general, lack good command tracking. Alternatives to the fixed-gain methods are nonlinear control methods. One of these methods is the famous computed torque method, which uses a dynamic model of the actuator to calculate the input torques for the specified trajectory. Another approach to the actuator controller design problem uses adaptive. These adaptive controllers can accomodate
varying environments and not sensitive to modelling error. Several studies have also been reported on the application of learning control techniques to actuator controller for improving the performance in trajectory following tasks over successive attempts at following the same trajectory. Also recently, increasing attention has been paid to the use of artificial neural networks in actuator control [17]. The vibration control project presented in this report is based on a piezoelectric actuator, with a controller design based on the principle of the Internal Model Design.

1.3 General System Description

All the hardware involved this thesis project can be classified into three groups:

(i) the platform hardware
(ii) the vibration control hardware
(iii) the digital implementation hardware

The platform hardware is the Adept robotic workcell. This is the system that produces structural vibrations to which the control scheme is to be applied. It is a two axis linear commercial robotic workcell equipped with an integrated VME controller.

The vibration control hardware, part of which is the robotic loading platform over which it is mounted, is an accelerometer (sensor) bonded on top of a stack of pzt actuator. Connected to the pzt actuator is a high voltage power amplifier. The vibrations over the loading platform are sensed by the accelerometer. Counter-vibrations to cancel the sensed vibrations are to be generated by the pzt actuator at the command of a designed controller. This control system consisting of the robotic loading platform, the power amplifier, the pzt actuator, and the accelerometer from here on will simply be referred to as the plant, the system or the open-loop system. The digital control hardware is a DSP-based Data Acquisition and Signal Processing Board hosted inside an IBM-compatible personal computer. A more detailed
description of these three groups of hardware is presented in Chapter 2. Chapter 3 will present modelling of the system dynamics. A complete theory on servocompensator, its design process and its response simulation are presented in Chapter 4. Chapter 5 will cover all the digital control software. The results of the actual implementation will be presented and discussed in Chapter 6. Finally, in Chapter 7, conclusions are drawn and suggestions on future direction of this project are given.
CHAPTER 2

SYSTEM HARDWARE DESCRIPTION

This chapter describes, into a little more detail, all the hardware used in this project. The descriptions, however, will be restricted to only those details relevant to this project.

2.1 The Robot Hardware

Figure 2.1 is a block diagram of the linear robotic workcell. The unit is a two-axis linear robot manufactured by Adept Inc., San Jose CA. There are a y-axis robot module and an x-axis robot module. Refer to Appendix A-6 for specifications on these modules. The y-axis module is the smaller of the two and is designed to be mounted on top and across the larger x-axis module. The design configuration allows the smaller y-module to slide linearly along the length of the x-module to generate the x-component of the system motion. The sliding y-module itself carries the loading platform which can slide linearly along the length of the y-module to generate the y-component of the system motion. In this manner, the system motion which is the resultant motion experienced by the loading platform is cartesian two-dimensional in x-y plane. However, due to mechanical flexure, the z-direction vibration can be quite significant, especially during rapid acceleration and deceleration in the x or y directions. As much as 3 \textit{g} of acceleration is observed along the z-direction under some of these conditions. In this project, the entire unit is mounted horizontally on a vibration-proof table top, as such the system motions are in the horizontal plane of the table top. Each robot module consists of a precision ground ball-screw drive mechanism with linear guides to enable sliding. The screw is driven by an AC servo-motor. The controlled position of the loading platform is measured by rotary type of incremental encoders mounted on the drive shaft of the modules. The platform
motions are controlled by an integrated VME controller. The VME controller is equipped with a keyboard, a hard-disk storage, a monitor etc., all of which makes it easy to program and drive the robotic loading platform in software. In this project, the VME control is responsible for the robot motion and the vibration control resides in the PC/DSP.

2.1.1 Motion of the Robot Modules and Structural Vibration
Most of the various types of motion, executed by the robotic platform in the horizontal plane, are observed to be characterized by disturbing structural vibrations in the vertical direction. With the robots excited along both axis with sinusoidal motion of amplitude 1.5 mm and at frequency of about 10 Hz, a band of modes of structural resonance of the robotic loading platform are excited. Figure 2.2 shows a typical spectrum of the band of the dominating signal modes within vibrations observed on the loading platform of the robotic workcell. The most dominant of these signal modes is at about 130Hz with a relative amplitude of −30 dB. It is this band of signals within the vibrations that is the subject of this research project — to actively control these disturbance signal modes digitally with the servocompensator.

Figure 2.1 Block Diagram of the Robotic Workcell
2.2 The Control System Hardware

The control system hardware for this vibration control project, as already outlined in Chapter One— the plant or the open-loop system— , consists of the robotic loading platform, a PZT vibration actuator, a high voltage power amplifier and an accelerometer. The PZT actuator and the accelerometer are physically mounted over the surface the robotic loading platform. Figure 2.1 is a block diagram of the whole setup and Figure 2.3 is a block diagram of the open-loop control system hardware.

Figure 2.2 Dominant Band of Modes of the Vibrations Observed on the Robot

Figure 2.3 Block Diagram of the Plant
2.2.1 The Vibration Actuator

The actuator used in this project is a PZT actuator model LTZ-2H manufactured by PT Transducer Products, Goshen, CT. PZT is a piezoelectric material made with lead zirconate titanate materials. Whenever a piece of such a material is excited with an electric field, its shape is distorted in a direction perpendicular to that of the applied field. Provided the applied voltage is not excessive as to exceed some characteristic threshold, the amplitude of distortion is proportional to the excitation voltage [13]. Thus, where the excitation voltage is AC, the distortion alternates at the excitation voltage frequency, which explains the vibration-actuating property of the piezoelectric material. The LTZ-2H model used in this project is designed for applications requiring high efficiency and sensitivity. The LTZ-2H actuator is mounted on the horizontal loading platform carried by the y-robot module. Along a pair of opposite vertical faces on the actuator are provided ohmic contacts for connection to a high voltage power amplifier. The PZT actuator specifications are listed in the Appendix A-1.

2.2.2 The Vibration Sensor

The vibration sensor used in this project is the triaxial charged capacitor accelerometer model 2223D manufactured by Endevco Inc., Pasadena, CA. This model is designed for applications requiring measurements of shock or vibration simultaneously in three mutually perpendicular axes. It is a self-generating piezoelectric transducer that requires no external power for operation. It is, however, used in this project with an Endevco model 104 charge signal conditioner. The accelerometer specifications are listed in the Appendix A-2.

2.2.2.1 The Endevco Charge Signal Conditioner: The Endevco model 104 charge signal conditioner is a two-channel signal conditioner designed for use with
piezoelectric transducers. The output voltage of each amplifier is proportional to the charge at the input of each channel. There are front panel controls to provide adjustments for input sensitivity, gain and filter enable-disable. This model is designed to be used with the Endevco model 109 power supply. The specifications on the signal conditioner can be found in Appendix A-3.

2.2.2.2 The Endevco Power Supply: The Endevco model 109 power supply is designed for use with Endevco model 100 series signal conditioners. It provides 15 volts to power the conditioner. The specifications on this power supply are also listed in Appendix A-4.

2.2.2.3 The Integrated Accelerometer Circuit: The accelerometer, the signal conditioner together with the power supply unit represents the vibration sensor circuit. In this project, only the vertical channel on the accelerometer is used because, the control of vibrations along only this direction is of interest. In order to minimize any phase shift between the pzt actuator and the accelerometer signals, the accelerometer is mounted directly on top of the actuator. This configuration is known as sensor and actuator collocation and helps to avoid a nonminimum phase plant [13].

2.2.3 The Power Amplifier
The power amplifier used in this project is a high voltage amplifier model 601B-PCB from Trek Inc., Medina NY. This amplifier provides the necessary voltage amplification to drive the pzt actuator. Refer to Appendix A-5 for the specifications on this high voltage amplifier.
2.3 The Digital Hardware

The digital implementation hardware is a Data Acquisition and Signal Processing Board Model-310B manufactured by Dalanco Spry Inc., Rochester, NY. The board, which is controlled by the Texas Instruments' TMS320C31 DSP, is designed to be hosted inside an IBM-compactible personal computer. In addition to the DSP, other circuits on the Dalanco board include a 12bit two channel DAC, a four channel multiplexed 14bit ADC and 128k word memory. Figure 2.4 is a block diagram of the Dalanco Board inside a personal computer.

2.3.1 The TMS320C31 DSP

The TMS320C31 DSP is a 32bit floating point DSP. It runs on a clock at speed of 50MHz and has instruction execution time of 40ns and a floating point processing speed at 50MFLOPS. This translates to instruction execution rate of 25MIPS. Inside the DSP, is a CPU, a DMA controller, an instruction cache, a RAM, a ROM, a serial port and a number of timers etc. The CPU, contains an ALU, a 32bit barrel shifter, a 32bit multiplier, a number of ARAU's and about 28 registers. The multiplier is capable of performing a full 32bit multiplication in one cycle independent of the ALU. The ALU itself can perform a single cycle operations on 32bit integers and
40-point floating point numbers. Two of the ARAU's, ARAU-0 and ARAU-1 are used for generating memory addresses. The RAM is divided into block-0 and block-1 each of which is 1k by 32bits. The ROM is about 4k by 32bits while the instruction cache is 64 by 32bits and uses the strategy of Least Recently Used (LRU) for instruction caching. The TMS320C31 DSP, itself is equipped with a full duplex bi-directional serial port which can transfer in 1, 2, 3 or 4 bytes per word. This serial port can be programmed in asynchronous mode to allow continuous data transfer without any new synchronization pulses. The timer inside the DSP is a 32bit counter, however, synchronization clock for the ADC conversions is provided on the Dalanco board external to the DSP. With the Texas Instruments' Optimizing C-compiler the DSP can be programmed in 'C' as well as assembly language. However, because the rest of the circuits on the Dalanco Board could not be programmed in 'C', the Dalanco Board as supplied is to be programmed in only assembly language. However, using assembly language-embedded 'C' codes, a number of 'C' functions have been written by Kedar Godbole and Vincenzo Pappano. With these functions stored in a library file to be included in any 'C' program, it becomes possible to program the entire Dalanco Board in 'C'. The control algorithms in this project are therefore completely written in 'C'.

2.3.2 Programming the Dalanco Board

The main job of the host PC is to perform supervisory functions such as starting and loading control programs into the DSP on the Dalanco board. The program loading is performed by running the loader program, load300, as an external command on the PC. The Dalanco board has four input analog channels and two analog output channels. To run control algorithms inside this board requires some initialization. The three 'C' functions that make it possible to write control codes for the Dalanco
board in 'C' are contained in the library header file *d3lbio.h* which must be included in any control program written in 'C'.

2.3.2.1 **Initialization of the Dalanco Board:** Part of the initialization routine necessary to use the Dalanco board involves setting up the timer and the sampling period. The 'C' function responsible for doing this is the *InitDSP( )* function. The prototype for this function is simply *void InitDSP(void);*

2.3.2.2 **Analog Input:** In order to sample an analog signal on one of the four ADC channels requires calling the *ReadAdc( )* function. The prototype for this function call is *int ReadAdc(int channel).* This function will cause an external analog voltage connected to its argument channel to be sampled in. The value returned by this function call is an integer ranging in value from -2047 to 2047, corresponding to voltages ranging from -5volts to 5volts. The necessary sign extension on the sampled data is done automatically. To be able to read from more than one channel, will require multiple calls of the *ReadAdc( )* function. Since the ADC has four channels, the range of possible channel argument is from 0 to 3. If the voltage sampled is to be calculated, simply multiply the sampled value by the scale factor 5/2047. The clock that triggers this sampling also provides the synchronization for software control loop, and is also the source of all timing in all the control programs.

2.3.2.3 **Analog Output:** The output of the analog channels is written via DAC. The function call for this is provided by the *WriteDAC( ).* The prototype for this is *int WriteDAC(int value, int channel).* Since the DAC on the Dalanco board is two-channeled, the possible channel arguments for the *WriteDAC( )* function are 0 and 1. The value written out on these channels range between -2047 to 2047. Whenever
the value is greater than 2047, it is automatically clamped off to 2047. Also, should the value be less than -2047, it is also clamped to -2047.

2.3.2.4 Setting the Sampling Rate for the Dalanco Board: The sampling rate chosen for the Dalanco board operation can be programmed in through another constant \( \text{TIMPER0} \). The chosen sampling rate must first be used to calculate \( \text{TIMPER0} \) according to the formula

\[
\text{TIMPER0} = \frac{\text{PcClock}}{8 \times \text{Numcal} \times \text{Sampfreq}}
\]

where

\( \text{PcClock} \) = the clock rate of the PC.

\( \text{Numcal} \) = the total number of function calls made to both the \text{WriteDAC( )} and \text{ReadAdc( )} in each control loop. In this application

\( \text{PcClock} = 50\,\text{MHz} \)

\( \text{Numcal} = 2 \), one \text{ReadAdc( )} plus one \text{WriteDAC( )} \)

\( \text{Sampfreq} = 10\,\text{kHz} \)

this leads to \( \text{TIMPER0} = 312.5 \).

Since \( \text{TIMPER0} \) is to be used in the header file \text{d3lbio.h} to set the sampling period for the board it is necessary to initialize \( \text{TIMPER0} \) prior to the \text{include d3lbio.h} statement.
CHAPTER 3
MODELLING THE SYSTEM DYNAMICS

Conventional approach towards active vibration control on flexible structures consists of two steps: 1) modelling and 2) controller synthesis where Kalman filter based stabilizers are designed. The design objectives are usually 1) closed loop stability and 2) asymptotic regulation of the amplitude of each resonant mode of the flexible structure. Furthermore, it is desired to conserve control bandwidth so as to minimize costs [11]. The modelling problem is addressed in this chapter. In this application, a model of the piezoelectric stack is required, so the presentation in this chapter is about how this model is obtained. Presented in the chapter is an experiment to measure the frequency response of the open loop system. There will also be discussion on the criterion used in selecting an analog shaping/antialiasing filter to reduce the effect of high frequency resonant associated with the pzt actuator dynamics.

3.1 Derivation of the Design Model

A transfer function design model of the open loop system is derived from its experimentally measured frequency response. A preliminary frequency response is first measured from which the characteristic of a shaping filter are determined. The shaping filter is necessary to reshape the open-loop system characteristic by increasing the overall system attenuation in the high frequency range to prevent spillover effects [3]. Now since digital control is used, there is the need for an analog antialiasing filter to be added on. In this project, a single analog low pass filter is selected to double as both the shaping and the antialiasing filters.

3.1.1 Experiment to Measure the Frequency Response of the System

Frequency response of the open-loop system is measured by the Hewlet Packard digital spectrum analyzer model HP3582A. The spectrum analyzer is connected to
the open loop plant as shown in Figure 3.1. The high voltage amplifier input is connected to an internal random signal generator on the analyzer. This input signal is also tapped to the channel A of the analyzer, as required in transfer function measuring mode of the analyzer. To complete the connections, the accelerometer output, is connected to channel B of the analyzer. The random noise generator on the analyzer is then set to bandwidth of 10kHz and the magnitude and phase response of the system are automatically displayed on the screen of the spectrum analyzer.

![Vibration Control System Diagram](image)

**Figure 3.1** Experimental Setup for Frequency Response Measurement

### 3.1.2 Frequency Response of the Open-loop System

The measured frequency response of the open-loop system is shown in Figure 3.2. It is observed that the unity gain crossover frequency; the frequency at which the gain plot first crosses downward the zero-dB line, is 320Hz. The response again shows a second crossover frequency at approximately 3100Hz, where this time the gain plot crosses the zero-dB line upward. This rising of the gain plot in the high frequency region is indicative of the high frequency resonances characteristic the open-loop system. The second gain crossover frequency marks, approximately, the beginning
of the region of these high frequency resonant modes [4]. The open-loop system over
the region of frequencies, shown in Figure 3.2, has no high frequency attenuation,
therefore needs an analog low pass or lag filter to produce gain roll-off over the high
frequency region.

Figure 3.2 Frequency Response of the Open-loop System: no Filter

The required low pass filter's characteristic, and the resulting reshaped system, can
be determined analytically if a model can be fitted for the open-loop system from
the measured frequency response. Thus, in order to use this analytical approach,
an approximate seventh order transfer function model is fitted to the experimentally
measured frequency response. The model fitting is done with Matlab. By observing
the slopes, the locations of break points and the low frequency asymptote of the
measured frequency response and with some experience with Bode plot techniques,
a dc gain-pole-zero model of the system can be obtained iteratively. Each trial is
plotted on same graph as the experimentally obtained response until a good match
is observed [6,7]. This is the approach used in this application to fit the models. A
better approach to obtaining the design model consists of using the bandwidth and
resonant modes distribution information, obtainable from the measured frequency
response, to select the characteristic of the required low pass filter. The selected
filter can then be designed, built and then added on to the system. The resulting filtered (reshaped) system can then have its frequency response measured to which a design model can be fitted. This is the approach used in this project and the next two sections explain how this is done.

### 3.1.3 Model for the Open-loop System

Using Matlab the experimentally measured frequency response of the open-loop system is approximated with a 7th order transfer function model given by:

\[
G_o(s) = \frac{183950}{s + 3769.9} \times \frac{s^2 + 1390.4s + 2860000}{s^2 + 2990.5s + 2731200} \times \frac{s^2 + 1390.4s + 2860000}{s^2 + 1885s + 1421200} \times \frac{s^2 + 1390.4s + 2860000}{s^2 + 11551s + 334150000}
\]

With poles at: \([-3.7699e+003, -5.7805e+003, \pm 5.7516e+004j, -1.0452e+003, ±5.1205e+003j, -9.4248e+002, ±3.6502e+003j]\)

and zeros at: \([-6.9524e+002, ±5.3025e+003j, -6.9524e+002, ±5.3025e+003j, -6.9524e+002, ±5.3025e+003j]\)

Frequency response of the fitted model and the experimentally measured frequency response are shown together in figure 3.3. This fitted 7th order open-loop system model is used in this project to compute the characteristics of the required lag filter and the design model as suggested in the previous section.

### 3.1.4 Analog Shaping/Antialiasing Filter

Because the closed loop system will be implemented in discrete time, it is necessary to add an analog antialiasing filter to the open loop plant. The antialiasing filter is required at the output of the accelerometer. As already mentioned, the frequency response of the open-loop system, starting from approximately 3.1kHz, is characterized by a number of resonant modes. Thus, in order to avoid exciting these high frequency resonances and the resulting instability of spill-over, it is necessary to reshape the open-loop system characteristic with a lag filter. The cut-off frequency of the filter, in this application, can be set to lower bound of 320Hz — the bandwidth
of the open loop system. A single analog low pass filter is chosen, in this application, to double as both the needed shaping filter and the necessary antialiasing filter. The upper bound on cut-off frequency of the filter can be the Nyquist frequency, provided this frequency is well below the first resonant mode of the open-loop system [2]. This is not the case in this project because the sampling frequency chosen is 10kHz, which makes a Nyquist frequency of 5kHz. This Nyquist frequency is rather well above 3.1kHz, the approximate beginning of the region of the resonances. Thus, in order to avoid exciting any resonant mode, the filter’s cut-off frequency must be chosen far below 3.1kHz. For this project, an effective choice of cut-off frequency that satisfies all these constraints is 500Hz. The filter is designed as a second order butterworth type [8]. This filter is shown in figure 3.4. The filter component values are:

\[ C1 = 0.3 \mu F, \quad C2 = 0.06 \mu F, \quad R1 = 4.7k, \quad R2 = 4.7k, \quad R3 = 1.0k \]

3.1.5 Frequency Response of the Filtered System

The chosen filter is connected to the output of the accelerometer and the resulting reshaped open-loop system, — the desired open-loop system— has its frequency
response measured. Figure 3.5 shows the measured frequency response of the reshaped open loop system.

![Figure 3.4 The Shaping/Antialiasing Filter](image)

**Figure 3.4** The Shaping/Antialiasing Filter

Using Matlab, a third order transfer function model:

$$G(s) = \frac{-602.35(s^3 + 298.58s + 30572000)}{(s + 4555.3)(s^2 + 1407.4s + 4042600)}$$

is fitted to the measured frequency response.

With poles at: \([-4.5553e+003\ -7.0372e+002\ \pm 1.8834e+003j]\)

and zeros at: \([-1.4929e+002\ \pm 5.5272e+003j]\)

This transfer function is the design model and its derivation completes the modelling process. The experimentally measured frequency response of the filtered system and that of the fitted third order design model are shown together in Figure 3.6. It is
observed that both the amplitude and phase responses of $G(s)$ are sufficiently close to the experimental data.

\textbf{Figure 3.6} Filtered Frequency Response: Experimental and Theoretical
CHAPTER 4

THEORY OF THE SERVOCOMPENDATOR

Presented in this chapter is the general theory of the servocompensator. Also presented here is the specific servocompensator design pertaining to the vibration control objective of this project. One design will be for a single-mode servocompensator and a second one will be for the multi-mode, specifically, double-mode servocompensator. The chapter will continue with presentation of the simulated responses of the designed single- and double-mode servocompensators. The concluding section of the chapter will introduce and explain the concept of multiple mode control which is a time-multiplexed servocompensator.

4.1 Concept of the Servocompensator

Servocompensator is a feedback control strategy. It is a scheme that allows any exogenous deterministic disturbance signal into a plant to be modelled into \[1\]. Since the transfer function of a linear system is the Laplace transform of its response to a unit impulse excitation, any deterministic disturbance signal can therefore be modelled by its Laplace transform, its transfer function model; effectively by its poles (eigenvalues) \[5\],\[7\]. When such a deterministic disturbance signal model is used to compensate the plant,— in series with the plant, along the forward path as shown in Figure 4.1— the result is an augmented plant-disturbance system. With a feedback gain designed to stabilize such a compensated system, the disturbance signal modes represented by the poles of the compensator, become the transmission zeros of the resulting closed loop system therefore blocking transmission through the closed loop system of the disturbance signal \[5\].
4.2 Model of Deterministic Disturbance Signals

Most deterministic disturbance signals encountered in control systems are either steps, ramps, sinusoids or signals obtainable from some combination these basic signals. As explained earlier, any such disturbance signal may be regarded as the pulse or impulse response of an appropriate dynamic system. Transfer function model of that dynamic system representing the disturbance being given by the Laplace transform of the disturbance signal itself [5].

Where the disturbance signal is a unit step, the dynamic system model \( G(s) \) is given by the integrator

\[
G(s) = \frac{1}{s} \quad \text{(4.1)}
\]

which using state space formulation is equivalent to a dynamic matrix \( A \) which in this case is a scalar of value zero therefore a first order plant with a single eigenvalue of 0 given as

\[
A = 0
\]

For a unit ramp disturbance signal, the system model \( G(s) \) is the double integrator

\[
G(s) = \frac{1}{s^2} \quad \text{(4.2)}
\]
which in state space is given by
\[ A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \]
a second order dynamic matrix with its two eigenvalues equal 0, 0

Finally, where the disturbance signal is sinusoidal at frequency of \( \omega \), the dynamic system model is the harmonic oscillator at this frequency given as
\[ G(s) = \frac{\omega^2}{s^2 + \omega^2} \]
which in state space is given by
\[ A = \begin{bmatrix} 0.00 & 1.00 \\ -\omega^2 & 0.00 \end{bmatrix} \]
a second order matrix with is two eigenvalues at \( \pm j\omega \)

Evidently, these deterministic disturbance signal models are characterized by dynamic system matrices with eigenvalues either at the origin or along the imaginary axis in the s-plane.

4.3 Servocompensator Design for a General Disturbance

\[ \dot{x} = A_x x + B_x u + v \]  
(4.4)
\[ y = C_x x \]  
(4.5)

The system represented by the dynamic equation above is \(( A_x, B_x, C_x )\) linear with a number of exogenous deterministic disturbance inputs represented by the state vector \( v \). Let \( v \) be generated by the following linear process:

\[ \dot{w} = \hat{A} w \]  
(4.6)
\[ v = \hat{C} w \]  
(4.7)

where \( w(0) \neq 0 \)
and \((\hat{C}, \hat{A})\) is observable but not necessarily known except for the eigenvalues of \(\hat{A}\).

Let \(A^*\) be the controllable canonical form of \(\hat{A}\) and define the following controllable pair \((A^*, B^*)\) as

\[
A^* = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\
-a_1 & -a_2 & -a_3 & \cdots & \cdots & \cdots & -a_n
\end{bmatrix} \quad \quad \quad B^* = \begin{bmatrix}
0 \\
0 \\
\vdots \\
0 \\
1
\end{bmatrix}
\]

If the plant \((A_x, B_x, C_x)\) is MIMO with a total of \(r\) input-output channels then with each channel compensated with the augmented disturbance \((A^*, B^*)\), the controller for all the \(r\) input-output represented of the plant will include a larger augmented disturbance model \((A_z, B_z)\) where \(A_z\) and \(B_z\) are block diagonal matrices having on their main diagonals a total of \(r\) blocks of \(A^*\) and \(r\) blocks of \(B^*\) respectively as

\[
A_z = \begin{bmatrix}
A^* & 0 & 0 & \cdots & 0 \\
0 & A^* & 0 & \cdots & 0 \\
0 & 0 & A^* & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & A^*
\end{bmatrix}
\]

\[
B_z = \begin{bmatrix}
B^* & 0 & 0 & 0 & \cdots & 0 \\
0 & B^* & 0 & 0 & \cdots & 0 \\
0 & 0 & B^* & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & \cdots & B^*
\end{bmatrix}
\]

Let \(z\) represent the state vector of the augmented disturbance model \((A_z, B_z)\) then since this model is in series with the plant, its control input is the plant’s output error vector \(e\) given as

\[
e = (y^{ref} - y)
\]  \hspace{1cm} (4.8)

Therefore the dynamics of the this augmented disturbance model are given as
In the special case of regulator control we have

\[ y^{ref} = 0 \]

so that

\[ e = -y \]

leading to simplified dynamics given as

\[ \dot{z} = A_z z + B_z e \] (4.9)

\[ \dot{z} = A_z z + B_z(y^{ref} - y) \] (4.10)

This leads to an augmented (coupled) plant-disturbance dynamics given as

\[ \dot{z} = A_z z - B_z y \] (4.11)

\[ y = C_z x \]

this dynamics simplifies further to

\[ \dot{z} = A_z z - B_z C_z x \] (4.12)

\[ y = C_z x \]

This leads to an augmented (coupled) plant-disturbance dynamics given as

\[ \dot{x} = A_z x + B_z u \] (4.13)

\[ \dot{z} = -B_z C_z x + A_z z \] (4.14)

\[ y = C_z x \]

\[
\begin{bmatrix}
\dot{x} \\
\dot{z}
\end{bmatrix} = 
\begin{bmatrix}
A_x & 0 \\
-B_z C_z & A_z
\end{bmatrix}
\begin{bmatrix}
x \\
z
\end{bmatrix} + 
\begin{bmatrix}
0 \\
B_z
\end{bmatrix} u
\]

(4.15)

\[ y = \begin{bmatrix} C_z & 0 \end{bmatrix} \]

As long as the plant \( (A_z, B_z, C_z) \) is controllable,

\[
\begin{bmatrix}
A_x & 0 \\
-B_z C_z & A_z
\end{bmatrix},\begin{bmatrix} 0 \\ B_z \end{bmatrix}
\]
will be controllable so that the stabilizing feedback gain given as

$$K = [K_x, K_z]$$  \hspace{1cm} (4.16)

can be designed using either the pole placement or the linear quadratic control algorithm, where

$K_x$ = the plant’s state feedback component of the stabilizing gain,

$K_z$ = the controller’s stabilizing gain. This means the control applied to the plant consists of two components given as

$$u = -K_x x - K_z z$$  \hspace{1cm} (4.17)

$(-K_x x)$ = the plant’s state feedback component,

$(-K_z z)$ = the controller’s state feedback component.

If the plant $(A_x, B_x, C_x)$ is observable then an observer, say a full order,

$$\dot{\hat{x}} = A_x \hat{x} + B_x u + L_z (y - C_x \hat{x})$$  \hspace{1cm} (4.18)

can be designed to estimate the state of the plant for use in the feedback. The closed loop servocompensator is shown in Figure 4.2

![Figure 4.2 Servocompensator with an Observer-based State Feedback](image-url)
4.4 Single-mode Servocompensator Design for Vibration Control

In this project the disturbances are sinusoidal signals at a known frequencies. The design of a servocompensator at the single most dominant mode of the robot’s vibration is presented in this section. As seen from the spectrum of the structural vibration of the robot, presented earlier, the most dominant mode is at the frequency of 130Hz. The disturbance model at this single mode is a harmonic oscillator at this frequency and is given as

\[ A_{zc} = \begin{bmatrix} 0.0000 & 1.0000 \\ -(260\pi)^2 & 0.0000 \end{bmatrix} \]

\[ B_{zc} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \]

With the open-loop system design model (derived in Chapter 3) given by the third order state model \((A_{zc}, B_{zc}, C_{zc})\), where

\[ A_{zc} = \begin{bmatrix} 0.0000 & 1.00000000 & 0.00000000 \\ 0.0000 & 0.00000000 & 1.00000000 \\ -5962.7 & -1.0454e+7 & -18415e+6 \end{bmatrix} \]

\[ B_{zc} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \]

\[ C_{zc} = \begin{bmatrix} -1.8415e+10 & -1.7985e+5 & -6.0235e+2 \end{bmatrix} \]

These dynamics are all in continuous-time. The discrete-time design starting with these continuous-time models is presented next.

4.4.1 Discrete-Time Design

A complete Matlab m-file of the discrete-time design process is listed in the appendix. With a sampling frequency chosen at

\[ T_s = 10kHz \]
and using a Matlab command $c2d$ the discrete-time open-loop plant model is obtained as $(A_x, B_x, C_x)$ where

$$A_x = \begin{bmatrix} 4.1026e-009 & 9.8429e-005 & 9.9735e-001 \\ 7.3966e-005 & 9.5446e-001 & -7.5550e+001 \\ 5.1342e-001 & -8.4878e+002 & -1.3621e+006 \end{bmatrix}$$

$$B_x = \begin{bmatrix} 1.4378e-013 \\ 4.1026e-009 \\ 7.3966e-005 \end{bmatrix}$$

$$C_x = \begin{bmatrix} -1.8415e+010 & -1.7985e+005 & -6.0235e+002 \end{bmatrix}$$

and the corresponding discrete-time disturbance model is $(A_z, B_z)$ where

$$A_z = \begin{bmatrix} 9.9667e-001 & 9.9889e-005 \\ -6.6644e+001 & 9.9667e-001 \end{bmatrix}$$

$$B_z = \begin{bmatrix} 4.9972e-009 \\ 9.9889e-005 \end{bmatrix}$$

Augmenting the plant and the disturbance models together leads to the 5th order discrete-time augmented plant-disturbance model given as

$$A = \begin{bmatrix} 4.1026e-009 & 9.8429e-005 & 9.9735e-001 & 0.000000000 & 0.0000000 \\ 7.3966e-005 & 9.5446e-001 & -7.5550e+001 & 0.000000000 & 0.0000000 \\ 5.1342e-001 & -8.4878e+002 & -1.3621e+006 & 0.000000000 & 0.0000000 \\ 9.2025e+001 & 8.9875e-004 & 3.0101e-006 & 9.9667e-001 & 9.9889e-005 \\ 1.8395e+006 & 1.7965e+001 & 6.0169e-002 & -6.6644e+001 & 9.9667e-001 \end{bmatrix}$$

$$B = \begin{bmatrix} 1.4378e-013 \\ 4.1026e-009 \\ 7.3966e-005 \\ 0.000000000 \\ 0.000000000 \end{bmatrix}$$

$$C = \begin{bmatrix} -1.8415e+010 & -1.7985e+005 & -6.0235e+002 & 0.00000 & 0.00000 \end{bmatrix}$$
4.4.2 Designing the Stabilizing Gain for the Augmented Plant-disturbance System

Using matlab, the stabilizing gain is designed with the linear quadratic regulator algorithm. A set of designs and simulations are carried out for a state weighting matrix \( Q \) given as

\[
Q = I, \text{ (a 5th order identity matrix)}
\]

and control weighting matrix \( R \) of value between

\[
R = 0.00001 \text{ to } R = 0.001
\]

The simulated closed loop response for:

- \( R = 0.00001 \)
- \( R = 0.00005 \)
- \( R = 0.0001 \)
- \( R = 0.0005 \)
- \( R = 0.001 \)

are shown in the Figures 4.3, 4.4, 4.5, 4.6, 4.7 and 4.8. Observe that in each of these simulated responses, the targeted mode (at 130Hz) is attenuated. The attenuation increases with the loop gain, that is with decreasing \( R \). In real-time test, however, large loop gain, that is smaller \( R \) must be avoided so as not to destabilize unmodelled high frequency modes of the plant.

![Figure 4.3 Simulated Closed Loop Response of the Single-mode Servocompensator for \( R = 0.00001 \)]
The design for $R = 0.0001$ has pole location given by $P$, where $P = \begin{bmatrix} 0.63082 & 0.91176 + 0.17828j & 0.91176 - 0.17828j \end{bmatrix}$.

This particular design is chosen for real-time implementation, the results of which is shown in Chapter 6, Figure 6.1.

### 4.4.3 Observer Design

To be able to physically implement the state-space-formulated design, requires the state observer to provide the estimate of the plant’s state for use in computing the state feedback control [2]. A full order observer is designed for this application. Again Matlab provides the design tool. Pole placement algorithm is used to design the observer gain. The observer poles choice criterion is that the slowest observer
pole, in continuous-time, be at least twice as fast as the fastest of the continuous-time closed-loop system poles. Thus, with the closed-loop system poles, in this case, given as

\[ P = \begin{bmatrix} 0.63082 & 0.91176 \pm 0.17828j \end{bmatrix} \]

one set of observer poles that satisfies this selection criterion is given by

\[ P_o = \begin{bmatrix} 0.001 & 0.002 \pm 0.02j \end{bmatrix} \]

The poles of the observer are therefore placed here to complete the whole design process. This pole location gives observer gain of

\[ L_x = \begin{bmatrix} 8.6425e-004 & -1.1844e-006 & -1.5030e-010 \end{bmatrix} \]
4.5 Double-mode Servocompensator Design for Vibration Control

The case of extending the servocompensator control to more than one signal mode within a vibration, in theory, can be approached in two different ways. One way is to design a multiple-mode servocompensator; one designed with more than one signal mode within the vibration spectrum. This simply requires an augmented, sinusoidal disturbance model whose order increases by two for each additional signal mode of vibration included [2]. An example of a multiple-mode servocompensator for the case of two signal modes is designed as follows:

Let \( \varpi_1 \) and \( \varpi_2 \) be two component frequencies of the vibration we desire to control, then their disturbance model is a fourth order harmonic oscillator given as

\[
A_{zc} = \begin{bmatrix}
0.00000000 & 1.0000 & 0.00000000 & 0.0000 \\
0.00000000 & 0.0000 & 1.00000000 & 0.0000 \\
0.00000000 & 0.0000 & 0.00000000 & 1.0000 \\
-(\varpi_1 \varpi_2)^2 & 0.0000 & -(\varpi_1^2 + \varpi_2^2) & 0.0000
\end{bmatrix} \quad B_{zc} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}
\]

For a specific example that is tested in this project we choose

\[
\varpi_1 = 2\pi \times 120 \text{rad} / s
\]

and

\[
\varpi_2 = 2\pi \times 140 \text{rad} / s
\]

\[
A_{zc} = \begin{bmatrix}
0.00000000 & 1.0000 & 0.00000000 & 0.0000 \\
0.00000000 & 0.0000 & 1.00000000 & 0.0000 \\
0.00000000 & 0.0000 & 0.00000000 & 1.0000 \\
-4398.8e+8 & 0.0000 & -1.3423e+6 & 0.0000
\end{bmatrix} \quad B_{zc} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}
\]

For the 10kHz sampling frequency, the discrete-time disturbance model is \((A_z, B_z)\) where

\[
A_z = \begin{bmatrix}
1.0000e+00 & 1.0000e-04 & 4.9944e-09 & 1.6655e-013 \\
-7.3265e-02 & 1.0000e+00 & 9.9776e-005 & 4.9944e-009 \\
4.3890e+007 & -2.1970e+003 & -1.3400e+002 & 9.9329e-001
\end{bmatrix}
\]
Augmenting this model with the third order open-loop design model leads to an augmented 7th order plant-disturbance model \((A, B, C)\) where

\[
A = 
\begin{bmatrix}
0.5134200 & -849.0000 & -1.36+6 & 0.000000 & 0.00000 & 0.00000 & 0.00000 \\
7.40e-05 & 0.9544600 & -75.5500 & 0.0000000 & 0.00000 & 0.00000 & 0.00000 \\
4.10e-09 & 9.84e-05 & 0.997350 & 0.0000000 & 0.00000 & 0.00000 & 0.00000 \\
2.51e-15 & 7.50e-13 & 7.67-08 & 1.0000000 & 0.00100 & 4.99e-9 & 1.67e-13 \\
1.00e-10 & 2.996e-8 & 0.0030700 & -0.07330 & 1.00000 & 9.98e-5 & 4.99e-09 \\
3.01e-06 & 8.98e-04 & 91.973000 & 2197.0000 & -0.0733 & 0.99329 & 9.98e-05 \\
0.0601000 & 17.945000 & 1.84+06 & -4.39e+7 & -2197.0 & -134.00 & 0.993290 \\
\end{bmatrix}
\]

\[
B = 
\begin{bmatrix}
7.3966e-005 \\
4.1026e-009 \\
1.4378e-013 \\
0.000000000 \\
0.000000000 \\
0.000000000 \\
0.000000000 \\
\end{bmatrix}
\]

\[
C = 
\begin{bmatrix}
-6.0235e+002 & -1.7985e+005 & -1.8415e+010 & 0.00 & 0.00 & 0.00 & 0.00 \\
\end{bmatrix}
\]

Using matlab, the stabilizing gain is designed using the linear quadratic regulator algorithm.

A set of designs and simulations are carried out for a state weighting matrix \(Q\) given as:

\[
Q = I, \text{ (a 7th order identity matrix)}
\]

and control weighting matrix \(R\) of value between
\[ R = 0.00001 \text{ to } R = 0.001 \]

The simulation result for \( R = 0.0001 \) is shown in the Figure 4.8

![Graph](image)

**Figure 4.8** Simulated Frequency Response the Closed Loop System with the Double-mode Servocompensator

The multiple-mode servocompensator appears to work in simulation but results of real-time test fail to follow the prediction. The test is even carried for other designed loop gains and in all cases, even though simulations for these gains show good results, the test results show an unstable response, unmodelled high frequency resonances become excited. The test result for the double-mode servocompensator is presented in Chapter 6. Actuator saturation appears to be main problem with the dual-mode servocompensatory. The problem so serious as to destabilize the closed-loop system.

### 4.6 Multiplexing a Number of Single-mode Servocompensators

In the control multiplexing scheme, a number of servocompensators are designed with each one dedicated to a single disturbance frequency. Assume it is desired to control vibrations at a total of \( N \) different frequencies, then this will require a total \( N \) (single-mode) servocompensators.

The plant(observer) \((A_z, B_z, C_z)\) and the observer gain \( L_z \) are the same for each single-mode servocompensator but the state feedback gain \( K_z \) and the mode feedback gain \( K_z \) are different. That is, each signal mode model \((A_z, B_z)_i\) is augmented
with the same plant \((A_z, B_z, C_z)\) to design its characteristic gain \([K_x, K_z]_i\) where \(i = 1, 2, 3, \ldots N\). The requirement then is that each single-mode servocompensator be designed at the same sampling frequency say \(T_s\). This will allow the digital implementation hardware to be programmed to sample faster by a factor of \(N\), that is with sampling period of \(\frac{T_s}{N}\). If all of these designs are arranged in a cyclical sequence inside the control loop software, the result is similar to a multi-loop control scheme where each single-mode servocompensator is guaranteed to get its control updated every \(T_s\) seconds as its design requires. During each cycle of the control loop, the same observer is used to update the plant's state, but the specific signal mode state that is updated and that to receive the control update at the time will depend on the position along the cyclical sequence of the mode model in the software at that instant. The block diagram of the design configuration is shown in Figure 4.9.

For each signal mode at frequency \(f_i\), a servocompensator \((A_z, B_z, K_z, K_x)_i\) is designed where \(i = 1, 2, 3, \ldots N\). Then with a software implemented (\(N\) by 1) multiplexer, the servocompensators are sequenced for control and state updates over a period of \(T_s\). With this control command multiplexing scheme a multiple modes of signals

![Figure 4.9 Block Diagram of N-Multiplexed Single-mode Servocompensators](image-url)
are concurrently controlled. This is equivalent to multiplexing the \( N \) single-mode servocompensators in such a way as to avoid actuator saturation. Figure 4.10 is the simulated response for the case of seven modes. The modes are at 90Hz, 100Hz, 110Hz, 120Hz, 130Hz, 140Hz and 180Hz.

![Figure 4.10 Simulated Response for Seven-Multiplexed Single-modes Servocompensators](image)

The simulation results for the control multiplexing scheme shows that it can be extended to as many modes as the digital hardware speed could allow. The gains:

\( (Kz)i \) and \( (Kx)i \) for \( R = 0.0001 \) for the modes:

110Hz 120Hz 130Hz 140Hz 150Hz 180Hz are:

\[
Kx110 = \begin{bmatrix} 2.3898e+02 & 1.3555e+06 & 2.2149e+09 \end{bmatrix};
Kz110 = \begin{bmatrix} -6.3113e+04 & 3.7761e+01 \end{bmatrix};
Kx120 = \begin{bmatrix} 2.3998e+02 & 1.3806e+06 & 2.1297e+09 \end{bmatrix};
Kz120 = \begin{bmatrix} -7.2539e+04 & 2.2504e+01 \end{bmatrix};
Kx130 = \begin{bmatrix} 1.1199e+02 & 5.9788e+05 & 1.1481e+09 \end{bmatrix};
Kz130 = \begin{bmatrix} -4.5248e+04 & 8.2596e+01 \end{bmatrix};
Kx140 = \begin{bmatrix} 1.1322e+02 & 6.0695e+05 & 1.1485e+09 \end{bmatrix};
Kz140 = \begin{bmatrix} -5.2549e+04 & 7.9503e+01 \end{bmatrix};
Kx150 = \begin{bmatrix} 1.1453e+02 & 6.1667e+05 & 1.1485e+09 \end{bmatrix};
Kz150 = \begin{bmatrix} -6.0384e+04 & 7.6047e+01 \end{bmatrix};
CHAPTER 5
CONTROL IMPLEMENTATION SOFTWARE

This chapter will present all the control software aspects of this project. The implementation software, as already explained in the chapter on the digital hardware, can be written in 'C' therefore this is what is done. A brief description of the robot control software, the VME, is also included in this chapter.

5.1 The Adept VME Motion Interface Controller

The Adept VME Motion Interface module is an integrated motion module. Each module can drive up to four servo amplifiers and can keep track of up to four incremental encoders. Most of the robot module motion control programs are provided by Adept in a file known as dlogger file. In this project, the robot modules are excited with sinusoidal signal of amplitude 1.5 \text{mm} and frequency of about 10 \text{Hz} in both x and y directions. This excitation, because of structural flexure causes, the robotic loading platform to vibrate in the z (vertical) direction generating a spectrum of harmonics typical of the one shown in Chapter 2, Figure 2.2. Thus in order to test the designed controllers, the robots are put on these sinusoidal trajectories again, this time with the control running.

5.2 The Control Implementation Algorithm

The servocompensator control program starts with, as is the rule for 'C' programming, include directives and variable declarations, followed by the body of the actual program. In this application, the include directive is used to include the d310bio.h the library file containing definitions of InitDSP( ), the ReadAdc( ) and the WriteDAC( ) functions required for to run the Dalanco board on the 'C' language. The variables, in this case, are the matrices and the vectors to define the parameters of the observer
and the signal mode models, vectors to represent the feedback gains, the present and next states of the observer and signal mode models and finally scalars to represent the sampling period, the control and other variables. The body of program in this case, is simply an infinite loop that essentially consists of reading the accelerometer output, updating states of the signal modes and the observer, computing and writing out control to the high voltage amplifier. The Figure 5.1 is the flowchart of this control loop.

![Flowchart of the Control Program](image)

**Figure 5.1** Flowchart of the Control Program

### 5.3 Control Loop for the Single-mode Servocompensator

The control loop for running one single-mode servocompensator starts with sampling in the accelerometer output. Next, control is computed from the feedback gains and the present states of the signal mode and the observer. Next, the computed control is written out via the DAC to the high voltage amplifier. Next, the accelerometer output, the computed control, the present states of the observer and that of the signal mode model, are used to compute the corresponding next state of the observer and that of the signal mode model. And prior exiting the present loop to enter the next,
the computed next state of the observer and that of the signal mode are copied into their corresponding present states, the accelerometer output—the system response—is also copied into a storage buffer for processing later. Figure 5.2 is a block diagram of the digital control loop for the single-mode servocompensator control. A complete list of a sample program is included in the Appendix C.

Figure 5.2 Block Diagram of Single-mode Control Loop

5.4 Multiplexing Single-mode Servocompensators

The control algorithm for controlling more than one servocompensator is basically same as for the case of running one servocompensator. The only differences are as follows:
(i) where there are $N$ servocompensators to control, with each designed at sampling frequency of $T_s$ then the sampling frequency programmed into the Dalanco board must be $\frac{T_s}{N}$

(ii) each servocompensator has its own independent state model for the signal mode it represents. These signal mode models are arranged in the control software, in a cyclical sequence. The software sequencing logic used to emulate the multiplexing among the servocompensators is the 'C' branch construct switch. Thus during each cycle of control loop, the same observer is used to update the state of the plant, but the specific signal model whose state is updated and that to receive control update will depend on the position along the cyclical sequence of the models at the instant. This position is decided by the logic argument expression of the switch construct. This logical expression is defined using the 'C' modullus operator. A sample control program for multiplexing four single-mode servocompensators is listed in Appendix C.
Figure 5.3 Block Diagram of the Multiplexing Control Loop
CHAPTER 6
TEST RESULTS

Test results of the control system designs are presented and discussed in this chapter. The format of the results presentation is simply a concurrent graphical plot of the response with no control and one with the servocompensator running. The real-time tests simply consists of driving the robotic workcell with its integrated VME controller while running the the designed servocompensator so as to control the vibrations experienced over part of the robotic loading platform covered by the collocated actuator and the accelerometer.

6.1 Results for a Single-mode Servocompensator
This is the fundamental result of the entire project; a single-mode servocompensator designed to regulate the vibration at a single frequency. The single-mode result is very impressive. The scale factor of the accelerometer for this application is set at 1 volts per 1 g of acceleration hence in these test results 0 dB is equivalent to 1 volt or 1 g. Figure 6.1 is this result for the signal mode at 130Hz, the most dominant signal mode within the observed vibrations on loading platform of the robotic workcell. Observe that the amount of attenuation produced by control on this single 130Hz signal mode is about 40 dB. Observe also the spread of the attenuation into the neighboring signal modes.

6.2 Results for Two-Multiplexed Single-mode Servocompensators
The result presented in this section is for the concurrent control of two single-mode servocompensators. The first servocompensator is designed for the mode at 120Hz and the second is targeted to the mode at 130Hz. These are the two most dominant
signal modes of vibration observable on the robotic workcell. The result is shown in Figure 6.2. The level of attenuation produced in the band of frequencies centered around two targeted signal modes averages to about 25dB.

Figure 6.1 Response for a Single-mode Servocompensator

6.2.1 Results for a Double-mode Servocompensator

It is not possible to run the designed double-mode servocompensator without exciting unmodelled resonances. Even though a number different loop gains have been designed and tested, the results show virtually no attenuation and worst of all, in all cases the system continues to be unstable as unmodelled high frequency modes remained excited, even though the corresponding simulation results show otherwise. The test result shown in Figure 6.3 is for loop gain at $R = 0.0001$; the same loop
gain for the simulated response shown in Figure 4.4 in Chapter 4. The design signal modes are set at 120Hz and 140Hz. Therefore the multiple-mode in general, and the double-mode servocompensator specifically does not work in real test as predicted.

Figure 6.3 Response for a Double-mode Servocompensator

6.3 Results for Three-Multiplexed Single-mode Servocompensators
The result, in Figure 6.4, is for the concurrent control of three single-mode servocompensators. The first servocompensator is designed for the signal mode at 110Hz. The second is designed at the 120Hz mode, while the third is targeted to the mode at 130Hz. In this case, observe the attenuation produced across the band of frequencies centered around the three target frequencies averages to about 15dB.

Figure 6.4 Response for Three-Multiplexed Single-mode Servocompensators
6.4 Results for Four-Multiplexed Single-mode Servocompensators

The result for the four-multiplexed single-mode Servocompensators is shown in Figure 6.5. The first servocompensator is designed to reduce the mode at 110Hz. The second servocompensator is designed to reduce the signal at 120Hz. The third servocompensator is dedicated to 130Hz signal, while the fourth is set to attenuate the 140Hz mode. The result shows an average attenuation of about 15dB for the band of frequencies across the four targeted modes.

![Figure 6.5 Response for Four-Multiplexed Single-mode Servocompensators](image-url)

6.5 Results for Multiplexing More Than Four Single-mode Servocompensators

The results in this section are for multiplexing more than four servocompensators. Figure 6.6: Result for five-multiplexed single-mode servocompensators at 110Hz, 120Hz, 130Hz, 140Hz and 150Hz. Figure 6.7: Result for six-multiplexed single-mode servocompensators at 110Hz, 120Hz, 130Hz, 140Hz, 150Hz and 180Hz. Figure 6.8: Result for seven-multiplexed single-mode servocompensators at 110Hz, 120Hz, 130Hz, 140Hz, 150Hz, 180Hz, and 190Hz.

Apart from saturation problem with the PZT actuator, the limit to the number of multiplexed signal modes tested, in this project, is the size of memory needed to hold all the variables in the control program. In each of the results shown below, there is
an average attenuation of about 15dB across the band of frequencies centered around the targeted signal modes.

Figure 6.6 Response for Five-Multiplexed Single-mode Servocompensators

Figure 6.7 Response for Six-Multiplexed Single-mode Servocompensators

Figure 6.8 Response for Seven-Multiplexed Single-mode Servocompensators
CHAPTER 7
CONCLUSIONS AND RECOMMENDATION

7.1 Conclusions
Active vibration control via digital servocompensator has been presented in this thesis. The fundamental result presented here, is that of the single-mode control; a strategy that allows control to be focussed on a single mode within the spectrum of vibration. The single-mode application will be ideal for the narrow-banded vibration control cases, that is, where a single-dominant mode is identified as the problem to suppress. A typical result for this case in this project shows as much as 40 dB of amplitude reduction on the targeted tone. The success of the single-tone design motivates the time-multiplexing scheme where a number of single-mode servocompensators are designed and controlled independently and concurrently. A typical result of the multi-tone control, tested in this project, produces an of average 15 dB attenuation across the band centered around the target modes. The overall results of this project clearly demonstrates the feasibility of using the methods presented in precision motion control in microelectronic assembly and manufacturing.

7.2 Recommendations
This study successfully demonstrates the possibility of controlling multiple modes of signals within a vibration spectrum by multiplexing the control among designs targeted to the individual signal modes. Future direction of this project could extend the method presented in this thesis to vibration control in all the three, x, y and z directions using either three pzt stack actuators or using one with three DOF's. In addition other stabilization and disturbance rejection methods such as the H-infinity could be tried.
APPENDIX A

PARAMETER SPECIFICATIONS

A.1 Model LTZ-2H Pzt Actuator

Free Dielectric Constant 3400
Loss Tangent -0.15
Density (gm/cm³) 7.45
Curie Temperature (°C) 195
Coupling Coefficients $k_{33}$ 0.75
Coupling Coefficients $k_{31}$ 0.36
Coupling Coefficients $k_{p}$ 0.62
Piezoelectric Constant $(m/v) \times 10^{12} d_{33}$ 590
Piezoelectric Constant $(m/v) \times 10^{12} d_{31}$ -280
Piezoelectric Constant $(v/m)/(N/m) \times 10^{3} g_{33}$ 19.6
Piezoelectric Constant $(v/m)/(N/m) \times 10^{3} g_{31}$ -8.3
Young’s Moduli $(N/m^2) \times 10^{-10} E_{11}$ 7.27
Young’s Moduli $(N/m^2) \times 10^{-10} E_{33}$ 4.88
Mechanical-Q Q 70
Transverse Frequency Constant $(l.f_r) (kc.in) N_1$ 58
Radial Frequency Constant $(r.f_r) (kc.in) N_2$ 42
Thickness Frequency Constant $(t.f_r) (kc.in) N_3$ 76
Circumference Frequency Constant $(D_m.f_r) (kc.in) N_c$ 40.5

A.2 Endevco Model 2223D Accelerometer

Charge Sensitivity $(\rho C/g)$ 12.0
Frequency Response Z-axis (Hz) 3 - 6000
Frequency Response X-axis (Hz) 3 - 3000
Frequency Response Y-axis (Hz) 3 - 3000
Transducer Capacitance $(\rho F)$ $800 \pm 0.25$
Transducer Resistance $(M\Omega)$ 20000
Sinusoidal Acceleration Limits along any axis 1000g
Shock Acceleration Limits along any axis 2000g
A.3 Endevco Model 104 Charge Signal Amplifier

Maximum Input Charge 30000\(\mu\)C
Input Source Resistance 10\(M\Omega\)
Input Source Capacitance 30000\(\mu\)F
Output Voltage Swing 10v(peak)
Output Current Swing 3mA(peak)
DC Offset 15mv(peak)
Gain Range 0.1 - 100
Frequency Response (lower cut-off) 1Hz
Frequency Response (upper cut-off) 20kHz

A.4 Endevco Model 109 Power Supply

Input Voltage 108 - 32 VAC, 47 - 63Hz
Output Voltage/Current 24V DC, 400mA(peak)
Line Regulation ±0.002 for 0.1% line change
Load Regulation ±0.003 for 0.5% load change

A.5 The Tek Model 601B-PCB High Voltage Amplifier

Input Voltage Range 0v - ±10v
Input Impedance 10k\(\Omega\)
Output Voltage Range ±500v
Output Current Range 0 - ±10mA rms, 20mA(peak)
Slew Rate less than 35v/\(\mu\)s
Gain Bandwidth Product Greater than 3MHz
Noise less than 500mV (peak)
Power Requirements 115/230V AC, 50Hz - 100Hz, 30Watts

A.6 Specifications on the X and the Y Robot Modules

<table>
<thead>
<tr>
<th>Specification</th>
<th>X-module</th>
<th>Y-module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Speed((mm/sec))</td>
<td>1200</td>
<td>1200</td>
</tr>
<tr>
<td>Stroke((mm))</td>
<td>1000</td>
<td>550</td>
</tr>
<tr>
<td>Repeatability</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>Ball Screw Pitch((mm))</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Max. Payload((kg))</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>Motor mount</td>
<td>Direct</td>
<td>Direct</td>
</tr>
<tr>
<td>Rated Thrust Force((N))</td>
<td>300</td>
<td>300</td>
</tr>
</tbody>
</table>
APPENDIX B
DESIGN MATLAB M-FILE

% THIS IS A MATLAB M-FILE

% Servocompensator Design

format short e;

% DYNAMICS OF THE FITTED THIRD ORDER DESIGN MODEL
p1 = 2*pi*320;    % A quadratic pole pair at 320Hz
ep = 0.35;        % Damping factor at this quadratic pole pair
p2 = 2*pi*725;    % A single pole at 725Hz
z1 = 2*pi*880;    % A quadratic zero pair at 880Hz
ez = 0.027;       % Damping factor at this quadratic zero pair

% transfer function model is given by (num/den) where
num = -(p1^2*p2)/(z1^2)*[ 1 2*ez*z1 z1^2 ];
den = conv([ 1 p2 ], [ 1 2*ep*p1 p1^2 ]);;

% converting transfer function model (num/den) to
% state model (Axc, Bxc, Cxc, Dxc)
[Axc,Bxc,Cxc,Dxc] = tf2ss(num,den);

% Choose sampling frequency Fs = 10000Hz
Fs = 10000;  Ts = 1/Fs;

% Convert the continuous-time plant model (Axc, Bxc, Cxc, Dxc)
% to discrete-time model (Ax, Bx, Cx)
[Ax, Bx] = c2d( Axc, Bxc, Ts );
Cx = Cxc;
Dx = Dxc;
% Continuous-time model of the vibration mode at (fd)Hz is
% (Azc, Bzc) where
fd = 130; \quad wd = 2*\pi*fd;
Azc = [ 0 1 ; -wd^2 0 ];
Bzc = [ 0 ; 1 ];

% Convert the continuous-time disturbance model (Azc, Bzc)
% to discrete-time model (Az, Bz)
[Az, Bz] = c2d(Azc, Bzc, Ts);

% The discrete time augmented plant-disturbance model is
% of 5th order given as (A, B, C) where
A = [ Ax, zeros(3,2) ; -Bz*Cx, Az ];
B = [ Bx ; 0 ; 0 ];
C = [ Cx , 0 , 0 ];
D = Dx ;

% State feedback (stabilizer/regulator) design
Q = eye(5); \quad \% state vector weighting matrix
R = 0.0001; \quad \% control vector weighting matrix

% open loop pole of the augmented system
Po = [ 6.3411e-001,
9.1556e-001 + 1.7451e-001i,
9.1556e-001 - 1.7451e-001i,
9.9667e-001 + 8.1591e-002i,
9.9667e-001 - 8.1591e-002i ];

% design by discrete time Ricatti equation
K = dlqr(A, B, Q, R);
Kx = [K(1),K(2),K(3)]; \quad \% state feedback gain component
Kz = [K(4),K(5)]; \quad \% disturbance feedback gain component

% The unaugmented state estimator (plant state observer) design using
% pole placement algorithm:
Lx = place(Ax', Cx', [ 0.001, 0.002+0.02i, 0.002-0.02i ]);
% Average amplitude (Am) of the 130Hz mode vibration measured is -30db
% approximately 0.031623volts
% and since the scale factor of the endevco accelerometer is 1volt/g
% this is then the same as (0.031623*g)meters per second square.

Am = 0.03162;
APPENDIX C

SAMPLE 'C' PROGRAMS FOR THE CONTROL LOOP

C.1 Single Servocompensator Control Program

/* This program runs control and stores the output response. Targeted mode of vibration is at 130 Hz.*/

#define TIMPER0 625/2 // sampling frequency Fs = 10Khz;
#include "D310BIO.H"
#define SIZE 10000
#define MEMSTART 0x4e20

void main( )

{ float xhat1 = 0, xhat2 = 0, xhat3 = 0,
  xhat1n1 = 0, xhat2n1 = 0, xhat3n1 = 0,
  z1 = 0, z2 = 0, z1n1 = 0,
  z2n1 = 0, u = 0,

  Ax11 = 5.1342e-001, Ax12 = -8.4878e002, Ax13 = -1.3621e006,
  Ax21 = 7.3966e-005, Ax22 = 9.5446e001, Ax23 = -7.5550e001, // observer
  Ax31 = 4.1026e-009, Ax32 = 9.8429e-005, Ax33 = 9.9735e-001, // model

  Bx1 = 7.3966e-005, Bx2 = 4.1026e-009, Bx3 = 1.4378e-013,
  Cx1 = -6.0235e002, Cx2 = -1.7985e005, Cx3 = -1.8415e010,

  Kx1 = 1.1199e002, Kx2 = 5.9788e005, Kx3 = 1.1481e009, // R=0.0001

  Lx1 = 8.6425e-004, Lx2 = -1.1844e-006, Lx3 = -1.5030e-010,

  Az11 = 9.9667e-001, Az12 = 9.9889e-005,
  Az21 = -6.6644e001, Az22 = 9.9667e-001, // disturbance model

  Bz1 = 4.9972e-009, Bz2 = 9.9889e-005,
  Kz1 = -4.5248e004, Kz2 = 8.2596e001, // R=0.0001

  yin = 0.0,
C.2 Control Program for Four Servocompensators

/* This program runs control and stores the output response. Targeted disturbance frequencies (modes) are 110 120 130 140 Hz. */

#define TIMPERO 625/8 // this chooses the dalanco board sampling frequency at Fs = 40kHz (10kHz per mode)

#include "D310BI0.H"
#define SIZE 10000

```
yout = 0.0;
int *buff, temp = 0, counter = 1;
InitDisp();
buff = (int *)MEMSTART;
while(1)
{
    temp = ReadAdc(1);
yout = (float)(5.0*temp/2047.0);
    if ( counter >= 20*SIZE && counter <= 21*SIZE )
    {
        *buff = temp;
buff++;
    }
    counter++;
}

U = (- Kx1*xhat1n1 - Kx2*xhat2n1 - Kx3*xhat3n1 - Kz1*z1n1 - Kz2*z2n1);
    WriteDAC((int)(2047.0*U/5.0), 1);

xhat1 = Ax11*xhat1n1 + Ax12*xhat2n1 + Ax13*xhat3n1 + Bx1*U
        +Lx1*(yout - Cx1*xhat1n1 - Cx2*xhat2n1 - Cx3*xhat3n1);

xhat2 = Ax21*xhat1n1 + Ax22*xhat2n1 + Ax23*xhat3n1 + Bx2*U
        +Lx2*(yout - Cx1*xhat1n1 - Cx2*xhat2n1 - Cx3*xhat3n1);

xhat3 = Ax31*xhat1n1 + Ax32*xhat2n1 + Ax33*xhat3n1 + Bx3*U
        +Lx3*(yout - Cx1*xhat1n1 - Cx2*xhat2n1 - Cx3*xhat3n1);

z1 = Az11*z1n1 + Az12*z2n1 + Bz1*(yin - yout);
z2 = Az21*z1n1 + Az22*z2n1 + Bz2*(yin - yout);

xhat1n1 = xhat1; xhat2n1 = xhat2; xhat3n1 = xhat3;
z1n1 = z1; z2n1 = z2;
```
#define MEMSTART 0x4e20

void main()
{
    float xhat1 = 0,
    xhat2 = 0,
    xhat3 = 0,
    xhat1n1 = 0,
    xhat2n1 = 0,
    xhat3n1 = 0,
    z11 = 0,
    z12 = 0,
    z11n1 = 0,
    z12n1 = 0,
    z21 = 0,
    z22 = 0,
    z21n1 = 0,
    z22n1 = 0,
    z31 = 0,
    z32 = 0,
    z31n1 = 0,
    z32n1 = 0,
    z41 = 0,
    z42 = 0,
    z41n1 = 0,
    z42n1 = 0,
    U1 = 0,
    U2 = 0,
    U3 = 0,
    U4 = 0,

    Ax11 = 5.1342e-001, Ax12 = -8.4878e+002, Ax13 = -1.3621e+006, // Observer
    Ax21 = 7.3966e-005, Ax22 = 9.5446e-001, Ax23 = -7.5550e+001, // model
    Ax31 = 4.1026e-009, Ax32 = 9.8429e-005, Ax33 = 9.9735e-001,

    Bx1 = 7.3966e-005, Bx2 = 4.1026e-009, Bx3 = 1.4378e-013,

    Cx1 = -6.0235e+002, Cx2 = -1.7985e+005, Cx3 = -1.8415e+010,

    Lx1 = 8.6425e-004, Lx2 = -1.1844e-006, Lx3 = -1.5030e-010,

    K1x1 = 6.9356e+001, K1x2 = 3.6707e+005, K1x3 = 7.2153e+008,
    K2x1 = 7.0058e+001, K2x2 = 3.7207e+005, K2x3 = 7.2234e+008,
    K3x1 = 7.0821e+001, K3x2 = 3.7754e+005, K3x3 = 7.2303e+008,
    K4x1 = 7.1643e+001, K4x2 = 3.8348e+005, K4x3 = 7.2350e+008,

    A1z21 = -4.7731e+001, A1z22 = 9.9761e-001,

    A2z11 = 9.9716e-001, A2z12 = 9.9905e-005, // mode at 120Hz
    A2z21 = -5.6795e+001, A2z22 = 9.9716e-001,

    A3z11 = 9.9667e-001, A3z12 = 9.9889e-005, // mode at 130Hz
    A3z21 = -6.6644e+001, A3z22 = 9.9667e-001,

    A4z11 = 9.9613e-001, A4z12 = 9.9871e-005, // mode at 140Hz
    A4z21 = -7.7278e+001, A4z22 = 9.9613e-001,

    B1z1 = 4.9980e-009, B1z2 = 9.9920e-005,
    B2z1 = 4.9976e-009, B2z2 = 9.9905e-005,
    B3z1 = 4.9972e-009, B3z2 = 9.9889e-005,
B4z1 = 4.9968e-009,    B4z2 = 9.9871e-005,
K1z1 = -2.0332e+004,  K1z2 = 5.5746e+001,
K2z1 = -2.4243e+004,  K2z2 = 5.4211e+001,
K3z1 = -2.8504e+004,  K3z2 = 5.2484e+001,
K4z1 = -3.3112e+004,  K4z2 = 5.0548e+001,

    yin = 0.0,
yout = 0.0;
    int *buff, which, temp = 0, counter = 1;
    InitDsp(  );
    buff = ( int *)MEMSTART;
    while(1)
        temp = ReadAdc(1);
        yout = (float)(5.0*temp/2047.0);
        if ( counter >= 20*SIZE && counter <= 21*SIZE )
        {
            *buff = temp;
            buff++;
            counter++;
            which = (counter) % 4;
            switch (which)
            {
case 0: { z11 = A1z11*z11n1 + A1z12*z12n1 + B1z1*(yin - yout);
                    z12 = A1z21*z11n1 + A1z22*z12n1 + B1z2*(yin - yout);
                    
                    xhat1 = Ax11*xhat1n1 + Ax12*xhat2n1 + Ax13*xhat3n1 + Bx1*U1);
                    +Lx1*(yout - Cx1*xhat1n1 - Cx2*xhat2n1 - Cx3*xhat3n1 );
                    
                    xhat2 = Ax21*xhat1n1 + Ax22*xhat2n1 + Ax23*xhat3n1 + Bx2*U1
                    +Lx2*(yout - Cx1*xhat1n1 - Cx2*xhat2n1 - Cx3*xhat3n1 );
                    
                    xhat3 = Ax31*xhat1n1 + Ax32*xhat2n1 + Ax33*xhat3n1 + Bx3*U1
                    +Lx3*(yout - Cx1*xhat1n1 - Cx2*xhat2n1 - Cx3*xhat3n1 );
                    
                    U1 = (-K1z1*xhat1n1-K1z2*xhat2n1-K1z3*xhat3n1-K1z1*z11n1-K1z2*z12n1 );
                    
                    WriteDAC( (int)(2047.0*U1/5.0), 1 );
                    
                    z11n1 = z11;  z12n1 = z12;
                    break;
            }
        }

    case 1: { z21 = A2z11*z21n1 + A2z12*z22n1 + B2z1*(yin - yout);
                z22 = A2z21*z21n1 + A2z22*z22n1 + B2z2*(yin - yout);
\[ x_{\text{hat}1} = A_{x_{11}}x_{\text{hat1}n1} + A_{x_{12}}x_{\text{hat}2n1} + A_{x_{13}}x_{\text{hat}3n1} + B_{x_1}U_2 + L_{x_1}(y_{\text{out}} - C_{x_1}x_{\text{hat}1n1} - C_{x_2}x_{\text{hat}2n1} - C_{x_3}x_{\text{hat}3n1}) \]

\[ x_{\text{hat}2} = A_{x_{21}}x_{\text{hat1}n1} + A_{x_{22}}x_{\text{hat}2n1} + A_{x_{23}}x_{\text{hat}3n1} + B_{x_2}U_2 + L_{x_2}(y_{\text{out}} - C_{x_1}x_{\text{hat}1n1} - C_{x_2}x_{\text{hat}2n1} - C_{x_3}x_{\text{hat}3n1}) \]

\[ x_{\text{hat}3} = A_{x_{31}}x_{\text{hat1}n1} + A_{x_{32}}x_{\text{hat}2n1} + A_{x_{33}}x_{\text{hat}3n1} + B_{x_3}U_2 + L_{x_3}(y_{\text{out}} - C_{x_1}x_{\text{hat}1n1} - C_{x_2}x_{\text{hat}2n1} - C_{x_3}x_{\text{hat}3n1}) \]

\[ U_2 = (-K_{2x_1}x_{\text{hat1}n1} - K_{2x_2}x_{\text{hat}2n1} - K_{2x_3}x_{\text{hat}3n1} - K_{2z_1}z_{21n1} - K_{2z_2}z_{22n1}) \]

WriteDAC( (int)(2047.0*U_2/5.0), 1 );

\[ z_{21n1} = z_{21}; \quad z_{22n1} = z_{22}; \]

break;

\}

case 2: { \[ z_{31} = A_{3z_{11}}z_{31n1} + A_{3z_{12}}z_{32n1} + B_{3z_1}(y_{\text{in}} - y_{\text{out}}) \]

\[ z_{32} = A_{3z_{21}}z_{31n1} + A_{3z_{22}}z_{32n1} + B_{3z_2}(y_{\text{in}} - y_{\text{out}}) \]

\[ x_{\text{hat}1} = A_{x_{11}}x_{\text{hat1}n1} + A_{x_{12}}x_{\text{hat}2n1} + A_{x_{13}}x_{\text{hat}3n1} + B_{x_1}U_3 + L_{x_1}(y_{\text{out}} - C_{x_1}x_{\text{hat}1n1} - C_{x_2}x_{\text{hat}2n1} - C_{x_3}x_{\text{hat}3n1}) \]

\[ x_{\text{hat}2} = A_{x_{21}}x_{\text{hat1}n1} + A_{x_{22}}x_{\text{hat}2n1} + A_{x_{23}}x_{\text{hat}3n1} + B_{x_2}U_3 + L_{x_2}(y_{\text{out}} - C_{x_1}x_{\text{hat}1n1} - C_{x_2}x_{\text{hat}2n1} - C_{x_3}x_{\text{hat}3n1}) \]

\[ x_{\text{hat}3} = A_{x_{31}}x_{\text{hat1}n1} + A_{x_{32}}x_{\text{hat}2n1} + A_{x_{33}}x_{\text{hat}3n1} + B_{x_3}U_3 + L_{x_3}(y_{\text{out}} - C_{x_1}x_{\text{hat}1n1} - C_{x_2}x_{\text{hat}2n1} - C_{x_3}x_{\text{hat}3n1}) \]

\[ U_3 = (-K_{3x_1}x_{\text{hat1}n1} - K_{3x_2}x_{\text{hat}2n1} - K_{3x_3}x_{\text{hat}3n1} - K_{3z_1}z_{31n1} - K_{3z_2}z_{32n1}) \]

WriteDAC( (int)(2047.0*U_3/5.0), 1 );

\[ z_{31n1} = z_{31}; \quad z_{32n1} = z_{32}; \]

break;

\}

case 3: { \[ z_{41} = A_{4z_{11}}z_{41n1} + A_{4z_{12}}z_{42n1} + B_{4z_1}(y_{\text{in}} - y_{\text{out}}) \]

\[ z_{42} = A_{4z_{21}}z_{41n1} + A_{4z_{22}}z_{42n1} + B_{4z_2}(y_{\text{in}} - y_{\text{out}}) \]

\[ x_{\text{hat}1} = A_{x_{11}}x_{\text{hat1}n1} + A_{x_{12}}x_{\text{hat}2n1} + A_{x_{13}}x_{\text{hat}3n1} + B_{x_1}U_4 \]
+Lx1*(yout - Cx1*xhat1n1 - Cx2*xhat2n1 - Cx3*xhat3n1 );

xhat2 = Ax21*xhat1n1 + Ax22*xhat2n1 + Ax23*xhat3n1 + Bx2*U4
+Lx2*(yout - Cx1*xhat1n1 - Cx2*xhat2n1 - Cx3*xhat3n1 );

xhat3 = Ax31*xhat1n1 + Ax32*xhat2n1 + Ax33*xhat3n1 + Bx3*U4
+Lx3*(yout - Cx1*xhat1n1 - Cx2*xhat2n1 - Cx3*xhat3n1 );

U4 = (-K4x1*xhat1n1-K4x2*xhat2n1-K4x3*xhat3n1-K4z1*z41n1-K4z2*z42n1);

WriteDAC( (int)(2047.0*U4/5.0), 1 );

z41n1 = z41;  z42n1 = z42;

break;

default: ;

} }

xhat1n1 = xhat1;  xhat2n1 = xhat2;  xhat3n1 = xhat3;

} }
REFERENCES


22. Dalanco Spry, "Model 310 Data Acquisition and Signal Processing Board for the IBM PC and Compatibles", 1993