# A digital signal processor based optical position sensor and its application to flexible beam control 

Thomas J. Spirock<br>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

Spirock, Thomas J., "A digital signal processor based optical position sensor and its application to flexible beam control" (1996). Theses. 1378.
https://digitalcommons.njit.edu/theses/1378

This Thesis is brought to you for free and open access by the Electronic 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.
by
Thomas J. Spirock

[^0]
# A DIGITAL SIGNAL PROCESSOR BASED OPTICAL POSITION SENSOR AND ITS APPLICATION TO FLEXIBLE BEAM CONTROL 

Thomas J. Spirock

Dr. Timothy Chang, Thesis Advisor Date
Assistant Professor of Electrical Engineering, NJIT
Dr. Walter Kosonocky, Committee Member
Distinguished Professor of Electrical Engineering
and Holder of Foundation Chair in Optoelectronics and Solid-State Circuits, NJIT


#### Abstract

\title{ A DIGITAL SIGNAL PROCESSOR BASED OPTICAL POSITION SENSOR AND ITS APPLICATION TO FLEXIBLE BEAM CONTROL }


by Thomas J. Spirock

A Digital Signal Processor (DSP) based optical position sensor was developed. The sensor system consists of the following components: 1) analog electronics, 2) the DSP based synchronous demodulation software, 3) PC based interface software which samples and saves the data, and 4) PC based control codes for a flexible beam experiment.

The ability of the system to determine the distance from the optical sensor to the power modulated light source was assessed by the following tests: 1) a stationary drift test to evaluate the system's noise, 2) a short-range test to determine the resolution of the optical sensor over a 25 mm range and, 3) a long-range test to evaluate the ability of the system to predict the location of the optical sensor over a 600 mm range. It was found that the resolution of the system is approximately 0.5 mm for the short range test and 5 mm for the long range test.

Finally, the sensor was deployed for the position feedback of a flexible beam experiment. Performance indices used to evaluate the response of the system were: 1) the sum of the squared position error, 2) the final steady state position error of the end of the flexible beam, and 3 ) the $5 \%$ settling time of the flexible beam. A number of control laws were evaluated and it was determined that a variable PID controller produced the best overall performance. The system can consistently position the end of the flexible beam from a $+/$ 20 cm to within 5 mm of the command position in approximately 8 seconds with a properly tuned controller.


# BIOGRAPHICAL SKETCH 

Author: Thomas J. Spirock<br>Degree: Master of Science in Electrical Engineering<br>Date: May, 1996

## Undergraduate and Graduate Education:

- Master of Science in Electrical Engineering,

New Jersey Institute of Technology, Newark, NJ, 1996

- Bachelor of Science in Electrical Engineering,

New Jersey Institute of Technology, Newark, NJ, 1993

Major: Electrical Engineering

## TABLE OF CONTENTS

Chapter Page
1 INTRODUCTION. ..... I
1.1 Overviev. ..... 1
1.2 Signal Flow and Description of Variables. ..... 2
1.3 Thesis Organization ..... 4
2 HARDWARE DEVELOPMENT ..... 5
2.1 General Description ..... 5
2.2 Light Sensor Module. ..... 5
2.3 Laser Diode Module ..... 7
2.4 TMS320C25 Digital Signal Processor ..... 10
2.5 Data Sampling Schemes ..... 10
2.5.1 Time Sampling Scheme. ..... 11
2.5.2 Analog Sampling Scheme. ..... 20
2.5.3 Comparison of Sampling Techniques ..... 22
2.6 One-Dimensional Testing Setup Description. ..... 22
2.7 Flexible Beam Testing Setup Description ..... 23
3 SOFTWARE DEVELOPMENT. ..... 25
3.1 General Description. ..... 25
3.1.1 C25 Digital Signal Processor Software. ..... 25
3.1.2 PC Software.. ..... 29

## TABLE OF CONTENTS

 (Continued)Chapter Page
3.2 Synchronous Demodulation Implementation. ..... 30
3.2.1 Bandpass Filter. ..... 33
3.2.2 Multipliers ..... 34
3.2.3 Lowpass Filter ..... 38
3.2.4 Elliptic Filter. ..... 39
3.2.5 Squarer and Summer. ..... 41
3.2.6 Filter Development. ..... 42
3.3 Channel Three -- Laser Modulation. ..... 45
3.4 PC Interface. ..... 49
3.4.1 Data Sampling and Saving. ..... 50
3.4.2 Control Program. ..... 52
4 TESTING AND DATA. ..... 59
4.1 Signal Time Stability ..... 59
4.1.1 Time Stability of Pstat ..... 59
4.1.2 Time Stability of Dpin and Ppin. ..... 61
4.2 Calibration and Verification Procedures ..... 63
4.3 Short-Range Test using Pstat. ..... 66
4.4 Long-Range Test using Two Pstat. ..... 68
4.5 Long-Range Test using Ppin ..... 73

## TABLE OF CONTENTS

## (Continued)

Chapter Page
4.6 Summary and Test Results. ..... 77
5 FLEXIBLE BEAM EXPERIMENT ..... 79
5.1 Control Implementation. ..... 79
5.2 Test Results ..... 80
5.2.1 Position Control with Constant Gain. ..... 80
5.2.2 Position and Derivative Control with Constant Gain. ..... 86
5.2.3 Position, Derivative and Integral Control with Constant Gain ..... 93
5.2.4 Position, Derivative and Integral Control with Variable Gain. ..... 100
6 CONCLUSIONS. ..... 108
6.1 Sensor Module Characterization ..... 108
6.2 Flexible Beam Experiment ..... 109
6.3 Future Improvements ..... 110
APPENDIX 1: Specification Sheet of the TSL220 Light to Frequency Converter ..... 112
APPENDIX 2: Specification Sheet of the Panasonic LN9705 Laser Diode. ..... 117
APPENDIX 3: Time Sampling Assembly Code. ..... 122
APPENDIX 4A: Description of Inverstion Assembly Language Program Variables ..... 123
APPENDIX 4B: Inversion Assembly Program. ..... 125
APPENDIX 5: Specification Sheet of the DC Motor ..... 129
APPENDIX 6A: Description of Assembly Language Program Variables. ..... 130

## TABLE OF CONTENTS

 (Continued)Chapter Page
APPENDIX 6B: DSP Assembly Language Program ..... 139
APPENDIX 7: Program to Link the Assembly Program. ..... 176
APPENDIX 8A: Description of Data Saving C- Program Variables. ..... 177
APPENDIX 8B: Data Saving C-Program ..... 179
APPENDIX 9A: Description of Control C-Program Variables. ..... 182
APPENDIX 9B: Control C- Program. ..... 186
REFERENCES. ..... 194

## LIST OF FIGURES

Figure ..... Page
1.1 Block diagram of sensor system and DSP signal flow. ..... 2
1.2 Block diagram of software signal flow in the host PC. ..... 3
2.1 General hardware structure ..... 5
2.2 TSL220 circuit. ..... 6
2.3 Output frequency of the TSL220 Vs the external capacitor. ..... 6
2.4 Photodiode spectral response ..... 7
2.5 Block diagram of the laser diode power amplifier ..... 8
2.6 Schematic of the laser diode's power amplifier circuit. ..... 9
2.7 Schematic of the highpass filter with gain circuit. ..... 10
2.8 Block diagram of the Time Sampling scheme. ..... 11
2.9 Block diagram of the Analog Sampling scheme. ..... 11
2.10 Signal flow in the Time Sampling scheme ..... 12
2.11 Flow chart of the Time Sampling scheme. ..... 13
2.12 Flow chart of the Uniform Sampling scheme ..... 14
2.13 Timing sequence of the TSL220 interrupt trigger and the resultant sampling. ..... 14
2.14 The output waveform specifications of the TSL220. ..... 15
2.15 Output pulse duration Vs the external capacitor ..... 15
2.16 Pulse shortening One Shot circuit. ..... 16
2.17 Flow chart of the Inversion Loop. ..... 19

## LIST OF FIGURES (Continued)

Figure Page
2.18 Comparison of the non-uniform and uniform sampling signals. ..... 19
2.19 Pulse widening One-Shot circuit. ..... 20
2.20 Schematic of the State Variable bandpass filter circuit. ..... 21
2.21 Schematic of the Gain circuit. ..... 21
2.22 Block diagram of the Analog Sampling scheme ..... 22
2.23 Block diagram of the one-dimensional testing setup. ..... 23
2.24 Block diagram of the flexible beam setup. ..... 24
3.1 Block diagram of the general software structure. ..... 25
3.2 Simplified flow chart of the assembly language program. ..... 26
3.3 Flow chart of Channels Zero, One and Two. ..... 28
3.4 Block diagram of the $A / D^{\prime}$ s ..... 28
3.5 General block diagram of the PC interface. ..... 29
3.6 Flow chart of the synchronous demodulation scheme. ..... 31
3.7 Frequency and phase response of the bandpass filter. ..... 34
3.8 Demodulated signal with DC leakage through the bandpass filter. ..... 36
3.9 FFT of signal with DC leakage (Figure 3.8). ..... 37
3.10 FFT of Vc or Vs when the DC component has been properly attenuated. ..... 37
3.11 Frequency and phase response of the lowpass filter. ..... 39
3.12 Frequency and phase response of the elliptic filter ..... 41

## LIST OF Figures <br> (Continued)

Figure Page
3.13 Example of the division error in the DSP. ..... 42
3.14 Example of the division with the DSP incorporating the correction. ..... 43
3.15 Example of the lowpass filter error. ..... 44
3.16 Block diagram of Channel Three. ..... 46
3.17 Block diagram of the laser diode voltage determination scheme. ..... 47
3.18 Block diagram of the data sampling process. ..... 50
3.19 Flow chart of the data saving C-program ..... 51
3.20 Block diagram of the control program scheme ..... 52
3.21 Flow chart of the control C-program. ..... 54
3.22 Flow chart of the Kp control. ..... 55
3.23 Frequency and phase response of the control program's lowpass filter ..... 56
3.24 Flow chart of Kp , Kd control ..... 56
3.25 Block diagram of the motor control system incorporating the integrator ..... 57
3.26 Flow chart of the $\mathrm{Kp}, \mathrm{Kd}, \mathrm{Ki}$ control with constant gains. ..... 57
3.27 Flow chart of the $\mathrm{Kp}, \mathrm{Kd}, \mathrm{Ki}$ control with variable gains. ..... 58
4.1 Time stability of Pstat at 12.7 mm . ..... 60
4.2 Time stability of Pstat at 0 mm ..... 61
4.3 Time stability of Dpin ..... 62
4.4 Time stability of Ppin. ..... 63

## LIST OF FIGURES

## (Continued)

Figure
Page
4.5 Step one of testing --. calibration ..... 65
4.6 Step three of testing -- position prediction. ..... 65
4.7 Calibration curve for the short range test. ..... 67
4.8 Location prediction error for the short-range test. ..... 67
4.9 Calibration curve \#1 for the long-range test using Pstat ..... 69
4.10 Location prediction error from the first calibration polynomial. ..... 70
4.11 Calibration curve \#2 for the long-range test using Pstat ..... 71
4.12 Location prediction error from the second calibration polynomial. ..... 71
4.13 Calibration curve \#3 for the long-range test using Pstat. ..... 72
4.14 Location prediction error from the third calibration polynomial. ..... 73
4.15 Calibration curve \#1 for the long-range test using Ppin. ..... 74
4.16 Location prediction error from the first calibration polynomial. ..... 75
4.17 Calibration curve \#2 for the long range test using Ppin. ..... 76
4.18 Location prediction error from the second calibration polynomial. ..... 76
5.1 Typical $Y$ and $U$ time plots for $K_{P}=5$ ..... 81
5.2 Typical Y and U time plots for $\mathrm{K}_{\mathrm{P}}=10$. ..... 82
5.3 Typical Y and U time plots for $\mathrm{K}_{\mathrm{P}}=20$. ..... 82
5.4 Typical $Y$ and $U$ time plots for $K_{P}=30$ ..... 83
5.5 Typical $Y$ and $U$ time plots for $K_{P}=5$. ..... 84

## LIST OF FIGURES

## (Continued)

Figure Page
5.6 Typical Y and U time plots for $\mathrm{K}_{\mathrm{P}}=10$ ..... 85
5.7 Typical $Y$ and $U$ time plots for $K_{P}=20$. ..... 85
5.8 Typical $Y$ and $U$ time plots for $K_{P}=30$ ..... 86
5.9 Typical $Y$ and $U$ time plots for $K_{D}=5$. ..... 88
5.10 Typical $Y$ and $U$ time plots for $K_{D}=10$ ..... 88
5.11 Typical Y and U time plots for $\mathrm{K}_{\mathrm{D}}=20$ ..... 89
5.12 Typical $Y$ and $U$ time plots for $K_{D}=30$. ..... 89
5.13 Typical $Y$ and $U$ time plots for $K_{D}=5$ ..... 91
5.14 Typical $Y$ and $U$ time plots for $K_{D}=10$. ..... 91
5.15 Typical Y and U time plots for $\mathrm{K}_{\mathrm{D}}=20$. ..... 92
5.16 Typical Y and U time plots for $\mathrm{K}_{\mathrm{D}}=30$. ..... 92
5.17 Typical Y and U time plots for $\mathrm{K}_{\mathrm{I}}=0.005$ constant gain ..... 95
5.18 Typical $Y$ and $U$ time plots for $K_{I}=0.01$ constant gain. ..... 95
5.19 Typical $Y$ and $U$ time plots for $K_{1}=0.02$ constant gain. ..... 96
5.20 Typical $Y$ and $U$ time plots for $K_{I}=0.04$ constant gain. ..... 96
5.21 Typical $Y$ and $U$ time plots for $\mathrm{K}_{1}=0.005$ constant gain. ..... 98
5.22 Typical $Y$ and $U$ time plots for $K_{1}=0.01$ constant gain. ..... 98
5.23 Typical $Y$ and $U$ time plots for $K_{l}=0.02$ constant gain. ..... 99
5.24 Typical $Y$ and $U$ time plots for $K_{1}=0.04$ constant gain. ..... 99

## LIST OF FIGURES <br> (Continued)

Figure Page
5.25 Typical $Y$ and $U$ time plots for $K_{1}=0.005$ variable gain. ..... 102
5.26 Typical $Y$ and $U$ time plots for $K_{I}=0.01$ variable gain. ..... 102
5.27 Typical $Y$ and $U$ time plots for $K_{I}=0.02$ variable gain. ..... 103
5.28 Typical $Y$ and $U$ time plots for $K_{I}=0.04$ variable gain. ..... 103
5.29 Typical $Y$ and $U$ time plots for $\mathrm{K}_{\mathrm{I}}=0.005$ variable gain. ..... 105
5.30 Typical $Y$ and $U$ time plots for $K_{1}=0.01$ variable gain. ..... 105
5.31 Typical $Y$ and $U$ time plots for $K_{I}=0.02$ variable gain. ..... 106
5.32 Typical $Y$ and $U$ time plots for $K_{1}=0.04$ variable gain. ..... 106

## LIST OF TABLES

Table Page
1.1 Description of signals. ..... 3
2.1 Description of signals in the Time Sampling scheme. ..... 12
2.2 Description of the variables involved in the inversion loop. ..... 18
3.1 Description of symbols used in the synchronous demodulation scheme ..... 32
3.2 Effectiveness of gaining the input of the lowpass filters. ..... 45
3.3 Description of variables involved in laser diode voltage determination. ..... 48
3.4 Data memory location summary. ..... 49
3.5 Description of control variables. ..... 53
4.1 Description of symbols used in the calibration and verification procedures ..... 64
4.2 Summary of test results ..... 77
5.1 Listing of typical plots of the Y and U signals for $\mathrm{K}_{\mathrm{P}}$ tests. ..... 80
5.2 Summary of results: Initial position $=+20 \mathrm{~cm}, \mathrm{~K}_{\mathrm{P}}$ control. ..... 81
5.3 Summary of results: Initial position $=-20 \mathrm{~cm}, \mathrm{~K}_{\mathrm{P}}$ control. . ..... 84
5.4 Listing of typical plots of the Y and U signals for $\mathrm{K}_{\mathrm{p}} \mathrm{K}_{\mathrm{D}}$ tests. ..... 87
5.5 Summary of results: Initial position $=+20 \mathrm{~cm}, \mathrm{~K}_{\mathrm{P}} \mathrm{K}_{\mathrm{D}}$ control, $\mathrm{K}_{\mathrm{P}}=10$ ..... 87
5.6 Summary of results: Initial position $=-20 \mathrm{~cm}, \mathrm{~K}_{\mathrm{P}} \mathrm{K}_{\mathrm{D}}$ control, $\mathrm{K}_{\mathrm{P}}=10$. ..... 90
5.7 Listing of typical time plots of the $Y$ and $U$ signals for constant gain tests. ..... 94
5.8 Summary of results: Initial position $=+20 \mathrm{~cm}$, Constant $K_{p} K_{D} K_{1}$ control $-\mathrm{K}_{\mathrm{p}}=10, \mathrm{~K}_{\mathrm{D}}=10$ ..... 94

## LIST OF TABLES <br> (Continued)

Table Page
5.9 Summary of results: Initial position $=-20 \mathrm{~cm}$, Constant $K_{P} K_{D} K_{I}$ control -- $K_{P}=10, K_{D}=10$. ..... 97
5.10 Listing of typical time plots of the Y and U signals for constant gain tests ..... 101
5.11 Summary of results: Initial position $=+20 \mathrm{~cm}$, Variable $K_{P} K_{D} K_{I}$ control $-K_{P}=10, K_{D}=10$ ..... 101
5.12 Summary of results: Initial position $=-20 \mathrm{~cm}$, Variable $\mathrm{K}_{\mathrm{p}} \mathrm{K}_{\mathrm{D}} \mathrm{K}_{\mathrm{I}}$ control -- $K_{P}=10, K_{D}=10$. ..... 104
6.1 Summary of test results from Chapter 4. ..... 108
A3.1 Description of memory locations in the Time Sampling code. ..... 122
A4. 1 Symbol name cross reference. ..... 123
A4.2 Description of assembly program variables. ..... 124
A6.1 Symbol name cross reference ..... 130
A6.2 Description of assembly program variables ..... 130
A8.1 Symbol cross reference in data saving C-program. ..... 177
A8.2 Description of data saving C-program variables ..... 178
A9.1 Symbol cross reference in control C-program. ..... 182
A9.2 Description of control C-program variables. ..... 182

## CHAPTER 1

## INTRODUCTION

In this thesis work, an optical position sensor has been designed and implemented. A number of tests were performed to characterize the drift, short-range, and long-range properties. The system was then deployed in a flexible beam control experiment.

### 1.1 Overview

The purpose of the project is to design and build an optical position sensor and to apply this sensor in motion control systems. The sensor's operation is based on the inverse square law property of light propagation which is used to derive the distance between a power-modulated laser diode and a light intensity sensor.

A Digital Signal Processor (DSP) is used to 1) power-modulate the laser diode, 2) sample the light intensity signal from the optical sensor, 3) perform synchronous demodulation on the intensity signal, and 4) pass the data to the host PC for further calculations. To perform the stated functions of the system both the analog electronics; which power the laser diode and pre-process the signals from the light intensity sensors, and the software; which performs the synchronous demodulation and power-modulate the laser diode, have been developed.

### 1.2 Signal Flow and Description of Variables

In this section, a brief overview of the system configuration and pertinent signals is provided. All subsequent discussions will be referenced to the diagrams and table in this section. Block diagrams of the signal flow are shown in Figure 1.1 and Figure 1.2. A description of the signals appears in Table 1.1

Figure 1.1 is a block diagram of sensor system and DSP signal flow. The signals from the analog electronics are sampled at 10 KHz by the AD (analog to digital converter). Synchronous demodulation is then performed on the sampled signals to extract their amplitudes which correspond to the non-normalized intensity signals from the sensor modules. The average values of the amplitudes are stored in the external memory where it can be accessed by the host PC for further analysis.


Figure 1.1 Block diagram of sensor system and DSP signal flow.

Figure 1.2 describes the signal flow in the host PC. The output data from the DSP are stored in dual ported external memory. Post processing of the sensor data, such as normalization and interpolation is performed by a C-program executed in the host PC.


Figure 1.2 Block diagram of software signal flow in the host PC.

Table 1.1 Descriptions of Signals

| Signal | Description |
| :---: | :---: |
| Vmove | The analog sine wave output from the mobile sensor module. |
| Vstat | The analog sine wave output from the stationary sensor module. |
| Vpin | The analog signal from the laser diode feedback circuit. |
| Dmove | The 10KHz sampled Vmove signal. |
| Dstat | The loKHz sampled Vpin signal. |
| Dpin | Average of the Non-normalized mobile sensor module's position. |
| Amove | Average of the Non-normalized stationary sensor module's position. |
| Astat | Average of the laser diode's output signal. |
| Apin | Position signal normalized by the laser diode's feedback signal. |
| Nstat | Time average of the Nstat signal. |
| Npin | Time average of the Npin signal. |
| Pstat | Thormalized by the stationary sensor module. |

### 1.3 Thesis Organization

The organization of this thesis is as follows: Chapter Two describes the hardware development such as the optical sensor modules, the laser-diode module, data sampling schemes, the one-dimensional testing setup and the application of the sensor module as the position sensor for a flexible beam experiment. Chapter Three describes the software development such as the DSP assembly language program which performs synchronous demodulation on the optical sensor output signals and the C-programs, which are executed on the host PC, which save the position data from the DSP and execute the control program for the flexible beam experiment. Chapter Four characterizes the performance of the sensor system. Finally, Chapter Five describes the application of the system to the flexible beam experiment.

## CHAPTER 2

## HARDWARE DEVELOPMENT

### 2.1 General Description

A general block diagram of the hardware is shown in Figure 2.1. A stationary, red laser diode, power-modulated by a 500 Hz sine wave, illuminates a mobile light sensor module. The light sensor module consists of a TSL220 light sensor and analog electronics. The TMS320C25 Digital Signal Processor (DSP) performs the following tasks: 1) modulate the laser diode, 2) sample the signal from the sensor module, 3) demodulate the sensor signal, and 4) pass the results to the host PC for future signal processing.


Figure 2.1 General hardware structure.
A detailed description of the hardware components are given in the following sections.

### 2.2 Light Sensor Module

The heart of the sensor module is the TSL220 light intensity to frequency converter chip from Texas Instruments. The TSL220 consists of a photo-diode and a current-tofrequency converter. The output voltage is a pulse train whose frequency is directly proportional to the incident light intensity on the photodiode. The output frequency range is determined by an external capacitor; so that the desired output frequency is adjustable
for a given intensity of light. The TSL220 circuit is shown in Figure 2.2. Figure 2.3 shows the output frequency of the TSL220 Vs it's external capacitor and Figure 2.4 shows the spectral response of the photodiode. Complete specifications of this chip are included in Appendix 1


Figure 2.2 TSL220 circuit.


Figure 2.3 Output frequency of the TSL220 Vs the external capacitor.


Figure 2.4 Photodiode spectral response.

### 2.3 Laser Diode Module

The light source which illuminates the light sensor modules is a Panasonic LN9705 laser diode. The LN9705 is a visible red GaAlAs laser diode with a nominal illumination wavelength of 670 nm . Automatic power control is possible by utilizing a built-in pin photodiode to monitor light power.

The primary source of noise on the light sensor is the 120 Hz overhead room lights. The frequency of oscillation of the laser diode was chosen to be 500 Hz so that it is sufficiently far enough away from the frequency of the room lights for demodulating purposes.

The TMS320C25 Digital Signal Processor, which will be described in detail in the following sections, generates the signal to modulate the laser diode. The signal consists of
a 500 Hz sine wave with an amplitude of 0.2 volts and a 2.6 volt DC bias. It should be noted that the drive signal should not exceed 3.0 volts, as indicated in the specification sheet, because the laser diode is very susceptible to over-voltage damage. Refer to Appendix 2 for the specifications of the laser diode.

The laser diode's power amplifier circuit consists of a set of unity gain inverters, implemented with LF353 op-amps, which buffer the DSP's D/A (Digital to Analog Converter). The current gain is provided by a unity gain amplifier implemented with an LM675 op-amp. A block diagram of the Laser Diode Module is shown in Figure 2.5 Figure 2.6 shows the schematic of the laser diode's power amplifier circuit. The calculation of the laser diode modulation voltage (V laser) is described in Section 3.3.


Figure 2.5 Block diagram of the laser diode power amplifier.


Figure 2.6 Schematic of the laser diode's power amplifier circuit.

The power monitoring feedback signal from the laser diode's built-in pin photodiode (Vpin) can be used to normalize the signal from the mobile sensor module, which will be discussed in detail in Section 3.4. The feedback signal from the laser diode is a sine wave with a DC bias whose voltage is directly proportional to the illumination power of the laser diode. A highpass filter is needed to remove the DC bias and a variable gain amplifier is synthesized to adjust the amplitude to match the $+/-5$ volts range of the A/D. The result is a sine wave whose amplitude is directly proportional to the illumination power of the laser diode which is sampled at 10 KHz by the DSP. Figure 2.7 shows the schematic of the first order highpass filter with a cutoff frequency of 50 Hz and variable gain which filters and amplifies the laser diode's feedback signal so it can be sampled by the DSP's A/D (Analog to Digital converter).


Figure 2.7 Schematic of the highpass filter with gain circuit.

### 2.4 TMS320C25 Digital Signal Processor

The TMS320C25 Digital Signal Processor (DSP) from Texas Instruments is used for sensor signal demodulation and laser modulation. It is located on a development plug-in card in the host PC. The DSP development card has the following peripherals: An internal timer with 0.1 us resolution, eight analog input channels multiplexed to an $\mathrm{A} / \mathrm{D}$, two independent analog output channels, 128 K words of dual-ported memory which is simultaneously accessible to both the DSP and the host PC. The DSP program is executed concurrently with the host PC. Therefore, a C-program can be executed in the host PC, in parallel with the DSP program, to manipulate the data in the DSP's memory. In the application stage, the DSP can be used in an embedded mode independent of a host PC to maximize efficiency.

### 2.5 Data Sampling Schemes

In order to acquire the sensor signal accurately and efficiently, two sampling approaches were investigated: Time Sampling and Analog Sampling. The Time Sampling approach uses the DSP's internal timer to clock the pulse stream generated by the TSL220. In the

Analog Sampling method the pulse stream from the TSL220 is first passed through a 500 Hz analog bandpass filter (BPF) to extract the carrier's fundamental harmonic which is then sampled by the A/D. Figures 2.8 and 2.9 show block diagrams of the two sampling schemes investigated. These two approaches are now described.


Figure 2.8 Block diagram of the Time Sampling scheme.


Figure 2.9 Block diagram of the Analog Sampling scheme.

### 2.5.1 Time Sampling Scheme

The first attempt to sample the TSL220 pulse stream was to use the internal timer of the DSP as a counter. A block diagram of the signal flow for the Time Sampling scheme is shown in Figure 2.10 and a description of the variables appears in Table 2.1


Figure 2.10 Signal flow in the Time Sampling scheme.
Table 2.1: Description of signals in the Time Sampling scheme.

| Signal Name | Description |
| :---: | :---: |
| F | Output of Inversion Routine = the intensity of the incident light. |
| $\mathrm{T}_{\mathrm{P}}$ | Non-uniform sampled time signal from TSL220 |
| $\mathrm{T}_{\mathrm{U}}$ | Uniformly sampled time signal from TSL220 |
| Tmove | Output signal from the mobile TSL220 |
| Tstat | Output signal from the stationary TSL220 |

The signal from the TSL220 sensor is fed into the DSP as an interrupt trigger. The interrupt routine determines the period of the incoming interrupts by performing the following operations: 1) Save the previous timer value, 2) Save the current timer value, 3) Reset the timer to its maximum value, and 4) Take the difference from the current and pervious timer values and save in memory as the time between the two previous pulses $\left(T_{p}\right)$. The resultant value ( $T_{p}$ ) would represent the number of clock ticks between two successive TSL220 pulses. Since the frequency of the DSP's internal timer is 10 MHz , a timing resolution of 0.1 uS can be achieved. A flow chart of this scheme is shown in Figure 2.11.


Figure 2.11 Flow chart of the Time Sampling scheme.

There are several problems associated with this intuitively appealing scheme. The first problem is non-uniform time sampling: the frequency of the TSL220 pulse train is continually changing due to the modulation of the laser diode and background fluctuations. This problem can be solved by uniformly sampling the timer data ( $T_{p}$ ) at a constant sampling rate. To accomplish this, another lower priority interrupt service routine, operating at 10 KHz , is used for the uniform sampling routine. Whenever the lower priority interrupt routine is triggered, the program copies the values of $T_{p}$ saved in memory and saves it as the uniform sampled time value $\left(\mathrm{T}_{\mathrm{U}}\right)$. A flow chart of the uniform sampling routine is shown in Figure 2.12


Figure 2.12 Flow chart of the Uniform Sampling scheme.
The second problem with using the TSL220 signal as the sampling trigger is that the DSP interrupts are both level and edge sensitive. The typical duration of a TSL220 pulse is approximately 3 us whereas the cycle time of the interrupt service routine which saves the time value $\left(T_{p}\right)$, is $2 u s$. Now since the interrupt is level sensitive, it will immediately re-trigger upon returning from the interrupt service routine if the TSL220 signal is still low, resulting in the incorrect timer value being saved. Figure 2.13 shows the timing sequence of the TSL220 interrupt trigger signal and the resulting sampling.


Figure 2.13 Timing sequence of the TSL220 interrupt trigger and the resultant sampling.

The output waveform specifications are shown in Figure 2.14 and the output pulse duration Vs the external capacitor is shown in Figure 2.15. Refer to Appendices 3 and 4 for the assembly codes for the Time Sampling and Uniform Sampling programs.


OUTPUT WAVEFORM
Figure 2.14 The output waveform specifications of the TSL220.


Figure 2.15 Output pulse duration Vs the external capacitor.

The solution to this problem is to shorten the pulse width of the TSL220 signal with a Mono-Stable-Multi-Vibrator (One-Shot). The circuit diagram for the One-Shot is shown in Figure 2.16. The One-Shot shortens the pulse-width of the TSL220 from 3uS to approximately 35 nS . Since this is much shorter than the time required for the interrupt service routine to sample and save the DSP timer value, the interrupt correctly triggers only once for every pulse from the TSL220.


Figure 2.16 Pulse shortening One Shot circuit.

A third problem associated with the Time Sampling scheme is that the sampled data is inversely proportional to the intensity of the incident light on the sensor module. The inverse of the intensity cannot be directly used in the synchronous demodulation process due to the presence of a residue. Specifically, if the inverse of the intensity is used, which is given by $\operatorname{Vin}=1 /[A \sin (\omega t)]$, then an output residue in the form of $\cot (\omega t)$ is created by the demodulation process shown in the equations below (2.5-1) to (2.5-3). The resulting output of the demodulation process will be incorrect because the lowpass filter in
the demodulation process will not properly filter the $\cot (\omega t)$ term shown in equation (2.53). The entire correct demodulation process is described in detail in Chapter 3.

$$
\begin{align*}
V i n & =1 /[A \sin (\omega t)]  \tag{2.5-1}\\
V c & =\cos (\omega t) /[A \sin (\omega t)]  \tag{2.5-2}\\
& =\cot (\omega t) / A \tag{2.5-3}
\end{align*}
$$

The solution here is to invert the time signal to get an intensity signal which can be fed into the synchronous demodulation scheme.

This solution method is quite time consuming as the DSP is a fixed point procussor and an inversion program is needed in the interrupt service routine to invert the uniformly sampled signal $\left(\mathrm{T}_{\mathrm{U}}\right)$. The inversion process iteratively calculates an integer frequency value (F) which, when multiplied with the integer time value ( $\mathrm{T}_{\mathrm{U}}$ ) approximately equals a preset target bound $\left(2^{\mathrm{N}}\right)(2.5-4)$. The iterative process continues until the error ( $\mathrm{I}_{\mathrm{E}}$ ) given by $(2.5-5)$ is less than a prespecified tolerance $\left(2^{\mathrm{n}}\right)$.

$$
\begin{gather*}
\mathrm{T}_{U} * \mathrm{~F}=2^{\mathrm{N}}  \tag{2.5-4}\\
\mathrm{I}_{\mathrm{E}}=2^{\mathrm{N}}-\mathrm{T}_{\mathrm{U}}^{*} \mathrm{~F} \tag{2.5-5}
\end{gather*}
$$

The main constraint on the inversion scheme is that it must be completed in less that 12 us because the demodulation routine consumes about 13 us out of the 25 us total allocated loop time. It was determined by experiment that approximately 20 iterations of the inversion loop are required to properly calculate the frequency value ( F ). This would require 60 us , which is several times longer than the available quantum. Table 2.2 gives a description of the variables involved in the inversion loop. The flow chart for the inversion
loop is shown in Figure 2.17 and a comparison of the non-uniform and uniform sampled signals is shown in Figure 2.18.

Table 2.2 Description of the variables involved in the inversion loop.

| Variable | Description |
| :---: | :---: |
| $\mathrm{I}_{\mathrm{E}}$ | Error at the $\mathrm{i}^{\text {th }}$ step |
| F | Frequency value at the $\mathrm{i}^{\text {th }}$ step |
| K | Error gain |
| $\mathrm{T}_{U}$ | Uniformly sampled time value at the $\mathrm{i}^{\text {th }}$ step |
| $\mathrm{N}_{\mathrm{m}}$ | Maximum number of iterations |
| $2^{\mathrm{N}}$ | Preset target bound |
| $2^{\mathrm{n}}$ | Maximum error tolerance |

The Time Sampling Scheme was not pursued further due to the excessive time required to resample and invert the signal. In order to increase the available time to properly implement the inversion routine, the sampling frequency would have to be decreased form the desired 10 KHz to 3 KHz which is too low to properly process the 500 Hz laser signal (Sampling at 3 KHz will cause aliasing of the higher order harmonics during demodulation).


Figure 2.17 Flow chart of the Inversion Loop.


Figure 2.18 Comparison of the non-uniform and uniform sampled signals.

### 2.5.2 Analog Sampling Scheme

The sampling scheme chosen to replace the Time Sampling scheme was the Analog Sampling scheme. In the Analog Sampling scheme, the TSL220 pulse stream is first converted into an analog sine wave by passing the pulse stream through a one-shot, followed by a 500 Hz bandpass filter. Since the amplitude of the resulting sine wave is directly proportional to the intensity of the light illumination on the TSL220's photo diode, no inversion is necessary.

In the Analog Sampling scheme, a one-shot is used to lengthen the pulse width from 3us to approximately 30 us so there is sufficient energy in the signal to excite the bandpass filter circuit. The output pulse width of the one-shot is chosen to be approximately $50 \%$ of the duty cycle when the laser diode is at maximum brightness. The circuit diagram of the pulse-widening circuit is shown in Figure 2.19. The 10K trim-pot controls the width of the output pulse to prevent over running of adjacent pulses.


Figure 2.19 Pulse widening One-Shot circuit.

The output of the One Shot (Tstat or Tmove) is then fed into the analog statevariable bandpass filter circuit, shown in Figure 2.20. The bandpass filter has a center frequency of 500 Hz and a Q of 50 which produces the 500 Hz sine wave by filtering out all but the fundamental frequency of 500 Hz . The output of the bandpass filter is then passed through a variable gain circuit so the sensor signal output range is matched to the $+1-5$ volt range of the $A / D$. The output of the gain circuit is represented by Vmove or Vstat. The schematic of the gain circuit is shown in Figure 2.21 and a block diagram of the final sampling scheme is shown in Figure 2.22.


Figure 2.20 Schematic of the State Variable bandpass filter circuit.
3.3 K


Figure 2.21 Schematic of the Gain circuit.


Figure 2.22 Block diagram of the Analog Sampling scheme.

### 2.5.3 Comparison of Sampling Techniques

Both sampling schemes investigated have relative merits over the other. The main advantage of the Time Sampling scheme is that it samples the time between successive pulses from the TSL220 very accurately with no external analog circuits. However, this method has two disadvantages, non-uniform sampling and signal inversion, which can only be solved by implementing time consuming assembly codes.

The Analog Sampling scheme avoids the above difficulties by using analog circuits to process the data from the TSL220. The analog signal is then sampled by the DSP with the $A / D$. The main disadvantages of this method are the instability of the analog electronics and the 12 -bit resolution of the $A / D$ which limits the resolution of the sensor.

### 2.6 One-Dimensional Testing Setup Description

A block diagram of the setup that was used to test the sensor system is shown in Figure 2.23.


Figure 2.23 Block diagram of the one-dimensional testing setup.

The laser diode, modulated at 500 Hz by the DSP, illuminates two TSL220 sensor modules. One module is setup on a stationary mount which is used to monitor the intensity of the laser diode. The second module is mounted on a platform that can be moved towards and away from the laser diode for either short range or long range tests. For the short range test, the platform can be moved over a range of 25 mm with a repeatability of approximately 0.02 mm . For the long range test, the platform can be moved over a range of approximately 600 mm with a repeatability of approximately 2 mm . The purpose of the stationary sensor module is to monitor the laser diode intensity so the signal from the mobile sensor module can be compensated for the drifts in laser.

### 2.7 Flexible Beam Testing Setup Description

Besides the one-dimension test, the sensor system is also applied to a flexible beam experiment. A sensor module is mounted at the end of a flexible beam which is a type 304
stainless steel beam, approximately 1500 mm long, 25 mm wide, and 6 mm thick. The position of the flexible beam is controlled by a brushless DC motor which is mounted in a vertical position on a platform. The sensor module is illuminated by the laser diode as in the previous test setup.

Refer to Figure 1.1, Figure 1.2 and Table 1.1 for a description of the signals. The DSP program samples the electrical output of the sensor module: V move from the mobile sensor module and Vpin from the laser diode. It then calculates Amove and Apin, the amplitudes of the sampled sine waves which represent the non-normalized intensity of the incident light on the mobile sensor module and the output power of the laser diode, respectively. The C-program fetches Amove and Apin from the DSP and calculates the normalized intensity Npin, as $\frac{\text { Apin }}{\text { Amove }}$, which is used as the position signal in the control law. The control signal is finally sent to the power amplifier and the $D C$ motor to generate the proper corrective action. Refer to Appendix 5 for the specifications of the DC motor. Chapter 3 describes the assembly language and C-programs in detail. A block diagram of the flexible beam test setup is shown in Figure 2.24.


Figure 2.24 Block diagram of the flexible beam setup.

## CHAPTER 3

## SOFTWARE DEVELOPMENT

### 3.1 General Description

A block diagram of the general software structure is shown in Figure 3.1. The C25 Digital Signal Processor (DSP) executes the sensor signal demodulation program while synchronously pulsing the laser diode. The PC provides the interface between the user and the DSP by executing the DSP's debugger or a C-program which can access the information in the DSP's external data memory area.


Figure 3.1 Block diagram of the general software structure.

### 3.1.1 C25 Digital Signal Processor Software

The main assembly language program in the DSP is responsible for modulating the laser diode intensity and demodulating the TSL220 light sensor signals. It is written in Texas Instruments C 2 x assembly language. A listings of the assembly codes and the link
programs are given in Appendix 6 and 7, respectively. A simplified flow chart of the program's operation is shown in Figure 3.2.

Upon reset, the program sets up all of the necessary initialization and house keeping tasks such as: define and label memory locations; define the interrupt vector table; enable the proper interrupts; set the $A / D$ sampling frequency; set up the sine table and sine table pointers; define the digital filter coefficients; and define the laser modulation voltages.


Figure 3.2 Simplified flow chart of the assembly language program.

All of the signal processing takes place in the interrupt service routine. After initialization, the DSP enters into a loop and waits for the interrupt trigger to occur. The interrupt trigger originates from the $\overline{\text { READY }}$ pin of the $A / D$. Its arrival signals the completion of the conversion process whose conversion rate is governed by an external programmable timer mapped to port zero of the DSP.

There are four channels of program code. Each channel is executed, in succession, at 10 KHz . Only one channel can be executed during any one interrupt because the $\mathrm{A} / \mathrm{D}$ can only be sampled one time during each interrupt. Therefore, the external timer must be set to 40 KHz . This sampling rate provides each channel with 25 us quantum of loop time.

The first task of the interrupt service routine is to determine which channel to process during the current interrupt. A description of the responsibility of each channel is as follows: Channel Zero and Channel One are responsible for the synchronous demodulation of their respective sensor module signals (Vmove and Vstat). Vmove is the output signal from the mobile sensor module and Vstat is the output from the stationary sensor module. Channel Two is responsible for the synchronous demodulation of Vpin which is the laser power feedback signal from the laser diode's built-in photodiode monitor. Vpin is passed through a highpass filter and a gain circuit to adjust it to properly fill the A/D's $+/-5$ volt range. Finally, Channel Three is responsible for the modulation of the laser diode and communicating with the host PC

Functionally, the Channel Zero, One, and Two programs are identical. The only difference is that they sample and perform synchronous demodulation on different input signals. A flow chart for the Channel Zero, One and Two Programs is shown in Figure 3.3.


Figure 3.3 Flow chart of Channels Zero, One \& Two.
The first operation of each channel is to set the input Multiplexer (Mux) to the next input channel. The Channel Zero program and the Channel One program read in the sampled signals, Vmove and Vstat, via the A/D from the mobile and the stationary sensor modules respectively. The Channel Two program, on the other hand, reads in Vpin, the sampled feedback signal form the laser diode. A block diagram of the A/D's data acquisition system is shown in Figure 3.4.


Figure 3.4 Block diagram of the $A / D$ 's.

After sampling, synchronous demodulation is performed on each signal. The result is then stored in a memory location where it can be fetched by a C-program running on the host PC and stored on disk. The specific memory locations are described in Section 3.3 and tabulated in Table 3.4. A detailed description of the synchronous demodulation process is given in Section 3.2.

### 3.1.2 PC Software

The interface between the user and the DSP is the host PC which executes either the DSP's debugger or a C-program. The debugger loads and executes the assembly program and can monitor the external memory of the DSP. The C-program can read from and write to the external memory of the DSP. This dual-porter scheme is advantageous because it is easier to perform complex mathematical calculations concurrently with a PC rather than with the fixed-point DSP. A block diagram of the PC interface is shown in Figure 3.5.


Figure 3.5 General block diagram of the PC interface.

### 3.2 Synchronous Demodulation Implementation

The demodulation process converts the input signal D in $=\mathrm{B}+\mathrm{A} \sin (\omega t+\Phi)$ to $A_{-}$out $=\frac{A^{2}}{4}$ where $D_{-}$in represents the three sampled signals from the $A / D$ (Dmove, Dstat, and Dpin). Refer to Section 1.2 for a review of the flow of the signals and their descriptions. The amplitude of the input signal is represented by $A$, the $D C$ bias is represented by $B$, and the phase shift is represented by $\Phi$. A_out represents the output signals from the three demodulation channels (Amove, Astat, and Apin).

There are five major groups of components in this demodulation implementation: a bandpass filter, two multipliers, two lowpass filters, two elliptic filters, two squarers, and a summer. A flow chart of the synchronous demodulation scheme is shown in Figure 3.6 and a description of the symbols used appears in Table 3.1. Each component will be described in detail in the following sections. Refer to Figure 3.6 and Table 3.1 for a description of the variables through out this chapter.


Figure 3.6 Flow chart of synchronous demodulation scheme.

Table 3.1 Description of symbols used in the synchronous demodulation scheme.

| Variable | Description |
| :---: | :---: |
| A | Amplitude of the input sine wave $\mathrm{D}_{\text {_ }}$ in |
| B | DC Bias of the input sine wave $\mathrm{D}_{-}$in |
| C | DC leakage of the output of the bandpass filter Dout_bpf |
| $\Phi$ | Phase shift of the input sine wave D_in |
| S | Digital filter scaling factor |
| D_in | Signal from sensor module or laser feedback sampled at 10 KHz |
| Dout_bpf | Output of bandpass filter |
| Vs | Output of bandpass filter multiplied by $\sin (\omega t)$ |
| Vc | Output of bandpass filter multiplied by $\cos (\omega t)$ |
| Vin_lpf | General input of the lowpass filter (Vs or Vc) |
| Vout_lpf | General output of the lowpass (Vs_lpf or Vc_lpf) |
| Vs_lpf | Output of lowpass filter in $\sin (\omega t)$ path |
| Vc_lpf | Output of lowpass filter in $\cos (\omega t)$ path |
| Vin_ell | General input of the elliptic filter (Vs_lpf or Vs_lpf) |
| Vin_ell | General output of the elliptic (Vs_ell or Vc_ell) |
| Vs_ell | Output of elliptical filter in $\sin (\omega \mathrm{t})$ path |
| Vc_ell | Output of elliptical filter in $\cos (\omega \mathrm{t})$ path |
| A_out | The sum of the squares of the outputs of the elliptical filters |

### 3.2.1 Bandpass Filter

The primary function of the bandpass filter is to remove any $D C$ bias on the input signal and to attenuate, if present, any higher harmonics of the 500 Hz carrier. The bandpass filfer is a second order filter with a center frequency of 500 Hz and a bandwidth of $20 \mathrm{~Hz}(\mathrm{Q}=$ 25). Equations (3.2-1) and (3.2-2) describe the operation of the bandpass filter on the input to the demodulation sequence. The need for attenuating the DC component of $\mathrm{D}_{-}$in will be justified in the next section. Dout_bpf represents the output of the bandpass filter.

$$
\begin{gather*}
D_{-} \text {in }=B+A \sin (\omega t+\Phi)+\text { higher order harmonics }  \tag{3.2-1}\\
\text { Dout_bpf }=A \sin (\omega t+\Phi) \tag{3.2-2}
\end{gather*}
$$

Since the DSP is a fixed-point processor, the filter coefficients must be represented by scaled integers. The scaling factor is then removed at the output of the digital filter by division. For convenience, the scaling factor is always chosen to be a value represented by $2^{S}$, where $S$ is an integer, which can be easily divided out by shifting the value in the accumulator. The larger the scaling factor that can be used, the closer the digital filter realization will be to the theoretical digital filter. The scaling factor implemented for all digital filters in the C 2 x assembly language program is $2^{14}=16384$.

The transfer function of the bandpass filter is shown in (3.2-3) and the difference equation is shown in (3.2-4). The frequency response of the digital bandpass filter is shown in Figure 3.7.

$$
\begin{equation*}
\frac{\text { Dout } \quad \operatorname{bpf}(z)}{D_{-} \operatorname{in}(z)}=\frac{0.006244-0.006244 Z^{-2}}{1-1.89027 Z^{-1}+0.987511 Z^{-2}} \tag{3.2-3}
\end{equation*}
$$

Dout_bpf $(i)=[30970$ Dout_bpf $(i-1)-16179$ Dout_ bpf $(i-2)$

$$
\begin{equation*}
\left.+102 \mathrm{D}_{\_} \text {in }(\mathrm{i})-102 \mathrm{D}_{-} \text {in }(\mathrm{i}-2)\right] / 16384 \tag{3.2-4}
\end{equation*}
$$



Figure 3.7 Frequency and phase response of the bandpass filter.

### 3.2.2 Multipliers

The multipliers demodulate the output of the bandpass filter signal Dout_bpf (3.2-2) by multiplying it by $\cos (\omega t)$ and $\sin (\omega t)$. The resulting signals consist of a DC component, which is directly proportional to the amplitude and phase shift of the input signal, and the
second harmonic of the carrier frequency ( 1 KHz ). Vs and Vc represent the outputs of the $\sin (\omega t)$ and $\cos (\omega t)$ multipliers, respectively. Equations (3.2-7) and (3.2-10) show the result of multiplying Dout_bpf with $\cos (\omega \mathrm{t})$ and $\sin (\omega \mathrm{t})$ respectively.

$$
\begin{align*}
V \mathrm{c} & =\cos (\omega \mathrm{t}) \text { Dout_bpf }  \tag{3.2-5}\\
& =\mathrm{A} \cos (\omega \mathrm{t}) \sin (\omega \mathrm{t}+\Phi)  \tag{3.2-6}\\
& =\frac{\mathrm{A}}{2}[\sin (2 \omega \mathrm{t})+\sin \Phi]  \tag{3.2-7}\\
V \mathrm{~S} & =\sin (\omega \mathrm{t}) \text { Dout_bpf }  \tag{3.2-8}\\
& =\mathrm{A} \sin (\omega \mathrm{t}) \sin (\omega t+\Phi)  \tag{3.2-9}\\
& =\frac{A}{2}[\cos \Phi-\cos (2 \omega t+\Phi)] \tag{32-10}
\end{align*}
$$

It is important that the DC component of $\mathrm{D}_{-}$in be eliminated If there is any DC leakage through the bandpass filter it will cause the primary frequency ( 500 Hz ) to be passed through the synchronous demodulation scheme. To illustrate this effect let the output of the bandpass filter, where the DC component has not been completely attenuated, be given by:

$$
\begin{equation*}
C+A \sin (\omega t+\Phi) \tag{3.2-11}
\end{equation*}
$$

The DC component through the bandpass filter is represented by $C$. When multiplied by the demodulating signals, (3.2-11) produces a carrier frequency component $(500 \mathrm{~Hz})$ in addition to the desired double carrier frequency $(1 \mathrm{KHz})$ and DC component,
(3.2-13) and (3.2-15). Compare (3.2-7) and (3.2-10) when the DC bias has been properly removed to (3.2-13) and (3.2-15), respectively.

$$
\begin{align*}
\mathrm{Vc} & =\cos (\omega \mathrm{t})[\mathrm{C}+\mathrm{A} \sin (\omega \mathrm{t}+\Phi)]  \tag{3.2-12}\\
& =\mathrm{C} \cos (\omega \mathrm{t})+\frac{\mathrm{A}}{2} \sin \Phi+\frac{\mathrm{A}}{2} \sin (2 \omega \mathrm{t}+\Phi)  \tag{3.2-13}\\
\mathrm{Vs} & =\sin (\omega \mathrm{t})[\mathrm{C}+\mathrm{A} \sin (\omega \mathrm{t}+\Phi)]  \tag{3.2-14}\\
& =\mathrm{C} \sin (\omega \mathrm{t})+\frac{\mathrm{A}}{2} \cos \Phi-\frac{\mathrm{A}}{2} \cos (2 \omega \mathrm{t}+\Phi) \tag{3.2-15}
\end{align*}
$$

Figure 3.8 is an example of Vc or Vs if there is DC leakage through the bandpass filter and Figure 3.9 shows the Fast Fourier Transform (FFT) of the signal in Figure 38 Compare Figure 3.9 to the FFT of Vc or Vs when the DC component has been properly attenuated, which is shown in Figure 3.10


Figure 3.8 Demodulated signal with DC leakage through the bandpass filter


Figure 3.9 FFT of signal with DC leakage (Figure 3.8).


Figure 3.10 FFT of Vc or Vs when the DC component has been properly attenuated.

### 3.2.3 Lowpass Filter

An elliptic filter is synthesized to remove the double carrier frequency component ( 1 KHz ) in Vc (3.2-7) and Vs (3.2-10). However, due to finite word length effects, it is necessary to prefilter Vc and Vs by a lowpass filter to avoid signal saturation in the elliptic filter. This is because the amplitude of the double carrier frequency component of $V \mathrm{c}$ and Vs can be much larger than the DC component which must be extracted to calculate the amplitude of the sampled sine wave.

The lowpass filter is a first order Butterworth filter with a cutoff frequency of 100 Hz . The transfer function of the lowpass filter is shown in (3.2-16) and the difference equation is shown in (3.2-17). In (3.2-17), Vin_Ipf represents $V_{s}$ or Vc in Figure 3.6 and Vout_lpf represents Vs_ Ipf, the output of the lowpass filter in the $\sin (\omega \mathrm{t})$ branch, or Vc_ Ipf, the output of the lowpass filter in the $\cos (\omega t)$ branch. The frequency and phase response of the digital low pass filter is shown in Figure 3.11.

$$
\begin{gather*}
\frac{\text { Vout_lpf }(\mathrm{z})}{\text { Vin_}_{-} \operatorname{lpf}(\mathrm{z})}=\frac{0.030468+0.030468 \mathbb{Z}^{-1}}{1-0.939062 Z^{-1}}  \tag{3.2-16}\\
\begin{aligned}
& \text { Vout_ } \operatorname{lpf}(\mathrm{i})= {\left[15386 \text { Vout_ }^{\operatorname{lpf}(i-1)+499 \text { Vin_}_{-} \operatorname{lpf}(\mathrm{i})}\right.} \\
&\left.+499 \text { Vin_ }_{-} \operatorname{lpf}(\mathrm{i}-1)\right] / 16384
\end{aligned}
\end{gather*}
$$



Figure 3.11 Frequency and phase response of the lowpass filter.

### 3.2.4 Elliptic Filter

At this point in the demodulation scheme, the signal consists of the desired DC component and the second harmonic component $(1 \mathrm{KHz})$. The purpose of the elliptic filter is to significantly attenuate the second harmonic component $(1 \mathrm{KHz})$. The operation of the elliptic filter is described in (3.2-18) to (3.2-21). The elliptic filter is a second order filter with 3 db of ripple in the passband, 20 db of attenuation in the stopband and a notch frequency of 1 KHz .

$$
\begin{gather*}
\mathrm{V} c_{-} \operatorname{lpf}=\frac{\mathrm{A}}{2} \sin \Phi+\frac{\mathrm{A}}{20} \sin (2 \omega \mathrm{t})  \tag{3.2-18}\\
\mathrm{V} c_{-} \text {ell }=\frac{\mathrm{A}}{2} \sin \Phi \tag{3.2-19}
\end{gather*}
$$

$$
\begin{gather*}
\mathrm{Vs}_{-} \mathrm{lpf}=\frac{\mathrm{A}}{2} \cos \Phi-\frac{\mathrm{A}}{20} \cos (2 \omega t+\Phi)  \tag{3.2-20}\\
\mathrm{Vs}_{-} \mathrm{ell}=\frac{\mathrm{A}}{2} \cos \Phi \tag{3.2-21}
\end{gather*}
$$

The transfer function and difference equation of the elliptic filter are shown in (3.2-22) and (3.2-23) respectively. In (3.2-24), Vin_ell represents Vs_lpf or Vc_lpf in Figure 3.5 and Vout_ ell represents Vs_ell, the output of the elliptic filter in the $\sin (\omega t)$ branch, or Vc_ ell, the output of the elliptic filter in the $\cos (\omega t)$ branch. The frequency and phase response of the elliptical filter is shown in Figure 3.12.

$$
\begin{equation*}
\frac{\text { Vout_ell }(z)}{\text { Vout_ } \operatorname{lpf}(z)}=\frac{0.100783-0.162863 \mathbb{Z}^{-1}+0.100783 \mathbb{Z}^{-2}}{1-1.795105 \mathbb{Z}^{-1}+0.849777 \mathbb{Z}^{-2}} \tag{3.2-22}
\end{equation*}
$$

$$
\begin{align*}
\text { Vout_ell(i) }= & {[-29411 \text { Vout_ell }(\mathrm{i}-1)+13923 \text { Vin_ell }(\mathrm{i}-2)+} \\
& 1651 \text { Vout_lpf(i)-2668 Vout_lpf }(\mathrm{i}-1)+ \\
& 1651 \text { Vin_lpf }(\mathrm{i}-2)] / 16384 \tag{3.2-23}
\end{align*}
$$



Figure 3.12 Frequency and phase response of the elliptic filter.

### 3.2.5 Squarer and Summer

The final step in the synchronous demodulation scheme, as shown in Figure 3.6, is performed by the squarer and summer blocks. The outputs of each elliptic filter are squared (3.2-24) and (3.2-25) and then summed together $(3.2-26)$ to remove any dependency on $\Phi$, the signal phase.

$$
\begin{gather*}
V_{c_{-}} \text {ell }{ }^{2}=\frac{\mathrm{A}^{2}}{4} \sin ^{2} \Phi  \tag{3.2-24}\\
\text { Vs_ell }^{2}=\frac{\mathrm{A}^{2}}{4} \cos ^{2} \Phi  \tag{3.2-25}\\
V_{-} \text {out }=V_{c_{-}} e l l^{2}+V_{s_{-}} \text {ell }{ }^{2}=\frac{A^{2}}{4} \tag{3.2-26}
\end{gather*}
$$

The result is a signal that is a function of the amplitude of the input signal ( $\mathrm{D} \_$in) only.

### 3.2.6 Filter Development

During software development each filter was individually tested with both sinusoidal and constant signals. While the processing of sinusoidal signals was correct, it was found that, with a constant input, the software produced a small steady-state error. The problem was traced to both the elliptic filters and the lowpass filters. It was found that the elliptic filter generated a constant offset error for all non-zero initial conditions. The source of the problem was found by executing the filter assembly code manually. The result of this method indicated an error in the way the DSP saves negative numbers after shifting. For Example, suppose that the value to be saved is $\frac{-3068}{4096}$. Dividing by 4096 is equivalen: to shifting the accumulator value by four bits to the left and saving the high sixteen bits. The expected answer is zero but the actual answer, as executed by the assembly code, is -1 . Figure 3.13 describes the process by which the DSP shifts and saves the values in the accumulator and the resulting error.


This Value Equals
Negative One

Figure 3.13 Example of the division error in the DSP.

The solution to this problem is the following: If the value in the accumulator to be divided and saved is less than zero, add the denominator minus one to the accumulator before the value is shifted and saved. This will change the hex F's in the high accumulator to zeros. Figure 3.14 shows an example of the division incorporating the correction. To avoid this problem in the other filters the correction for dividing and saving negative numbers was added to all filters through out the synchronous demodulation program.


Figure 3.14 Example of the division with the DSP incorporating the correction.

Besides division error in the elliptic filters, it was also found the lowpass filters also produced a steady state offset error as show in Figure 3.15.


Figure 3.15 Example of the lowpass filter error.

Again, the source of the problem was found by executing the filter assembly code manually. This problem is due to the finite word length effects of the DSP. The output of all of the digital filters must be divided by the scaling factor of the digital filter as described in Section 3.2.1. As the output of the filter approaches the input DC value when testing with a constant input, the change from iteration to iteration becomes smaller and smaller. Eventually, the change from each iteration becomes less than the scaling factor and is lost in the division. The solution of the problem is to gain the input before the lowpass filter

For example, in a fixed point calculation, when two input values 13000 and 13000 +3000 are divided by 4096 , they are both truncated to 3 . With proper scaling ( x 2 for example) the results are 6 and 7 respectively. The effectiveness of gaining the input before the lowpass filter is shown in the data in Table 3.2 .

Table 3.2 Effectiveness of gaining the input of the lowpass filters.

| Filter Value | Actual Value | Fixed Point Value | Percent Error |
| :---: | :---: | :---: | :---: |
| $\frac{13000}{4096}$ | 3.174 | 3 | 5.5 |
| $\frac{13000+3000}{4096}$ | 3.906 | 3 | 23.2 |
| $\frac{2 \times[13000]}{4096}$ | 6.348 | 6 | 5.5 |
| $\frac{2 \times[13000+3000]}{4096}$ | 7.813 | 7 | 10.4 |

### 3.3 Channel Three -- Laser Modulation

A Flow chart of Channel Three is shown in Figure 3.16. The primary responsibility of this channel is to modulate the laser diode at 500 Hz and to copy the outputs of the demodulation routines to the DSP's external memory where it can be accessed by a Cprogram running on the host PC . This channel can also be used for other additional tasks because the specified responsibilities of Channel Three use less than 3us, which is only a fraction of the 25 us allotted for each interrupt service routine.

Since the $A / D$ and $D / A$ share the same external data latch, proper operation of the D/A sample signal requires that the $A / D$ be read each time the interrupt is triggered regardless if the data is needed. This is also critical because the $A / D$ ready signal is used to provide the 25 us timing, as described in section 3.1.1. Therefore, the first operation of channel three is to read the $A / D$.


Figure 3.16 Block diagram of Channel Three.

The laser diode is powered by a sine wave voltage with an amplitude of 0.2 volts added to a 2.6 volt DC bias. Since the Channel Three Program is executed at 10 KHz and the required modulation frequency of the laser diode is 500 Hz , the number of time steps in the sine wave is: $\frac{10 \mathrm{KHz}}{500 \mathrm{~Hz}}=20$ Steps. Therefore, during the initialization part of the program, a sine table with twenty points equally distributed from zero to 342 degrees is constructed by scaling the values of $\sin (\theta)$ by 4096: Values in sine table $=4096 * \sin (\theta)$. In addition, a pointer which points to the first location in the sine table is initialized in auxiliary register one of the DSP . The program also initializes the amplitude ( $\mathrm{AMP}=0.2$ volts is represented by 80 in the DSP), a DC bias of the sine wave (Biasl $=2.6$ volts is represented by 1077 in the $\operatorname{DSP}$ ) and another DC bias which is used to properly adjust the
output voltage of the D/A (Bias2 $=5.0$ volts is represented by 2048 in the DSP). The specifications for the $A / D$ and $D / A$ may be found in the Dalenco-Spry manual.

During the calculation of the laser diode voltage ( $V$ _laser) the program multiplies the preset amplitude (AMP) with the value in the sine table $(\sin (\theta))$ indicated by the pointer and divides out the scaling factor (4096) introduced to define the values in the sine table. The result is summed with two bias values (Bias1 and Bias2). Equation (3.3-1) describes how the laser voltage is calculated as a function of $\theta$.

$$
\begin{equation*}
\text { V_laser }=\frac{80 \times \sin \theta}{4096}+1077+2048 \tag{3.3-1}
\end{equation*}
$$

The value V_laser is then outputted to the D/A and then the amplifier (See Figure 2.5 of Section 2.3 for a description of the laser diode power electronics) which drives the laser diode. If the pointer reaches the end of the table, then it is reset to the top. A block diagram which describes the laser modulation calculation technique is shown in Figure 3.17. Refer to Table 3.3 for a description of the variables involved in laser diode voltage determination.


Figure 3.17 Block diagram of the laser diode voltage determination scheme.

Table 3.3 Description of variables involved in laser diode voltage determination.

| Variable Name | Value in DSP | Description |
| :---: | :---: | :---: |
| $\sin (\theta)$ |  | Value representing the sine wave as a function of $\theta$ |
| Amp | 80 | Amplitude of the sine wave |
| Scaling Factor | 4096 | Integer value used to scale $\sin (\theta)$ |
| Bias1 | 1077 | DC bias of the sine wave |
| Bias2 | 2048 | 5 volt offset of the D/A |

The outputs of the three demodulation channels are saved in variables Amove, Astat, and Apin respectively. After the voltage on the laser diode is calculated, the Channel Three Program copies the memory locations which contain the outputs from the three synchronous demodulation channels from internal memory page 4 to external dual ported memory page 8 on the DSP board where it can be accessed by the host PC. Finally, the channel pointer is updated so that Channel Zero is executed during the next interrupt service routine. Refer to Table 3.4 for the data memory locations of the variables.

Table 3.4: Data memory location summary.

| Variable Name | Data Memory Address Page | Memory Location on Page |
| :---: | :---: | :---: |
| Amove | 4 | 112 |
| Astat | 4 | 115 |
| Apin | 4 | 119 |
| Amove | 8 | 16 |
| Astat | 8 | 56 |
| Apin | 8 | 96 |
| Flag0 | 8 | 20 |
| Flag1 | 8 | 60 |
| Flag2 | 8 | 100 |
| $U$ |  | 120 |

### 3.4 PC Interface

The host PC is a 286 computer with a math co-processor. The PC can execute either the DSP's debugger or a C-program which can perform the following tasks: 1) Sample the data from the DSP and save it on a floppy disk for analysis or 2) Execute a control program by sampling position data from the DSP, calculating a control law, and return the control signal (U) to memory location 120 in data memory page 8 in the DSP.

### 3.4.1 Data Sampling and Saving

The process of taking a data sample was automated as much as possible to save time. The DSP assembly program, which has been described in detail in the previous sections, runs
continuously. A simplified block diagram of the data sampling and saving scheme is shown in Figure 3.18 and a flow chart of the data-saving C-program is shown in Figure 3.19.


Figure 3.18 Block diagram of the data sampling process.

After the necessary initialization, the C-program continuously examines the data-ready flags (Flag0, Flag1, and Flag2) from the DSP which signal that the demodulation outputs from Channels Zero, One and Two, respectively, are ready. Refer to Table 3.4 for the memory addresses of the data flags. The outputs of the Channel Zero, One and Two programs (Amove, Astat, and Apin) are sampled, when available, and normalized. The normalized variables (Nstat and Npin) and calculated as follows: Nstat $=\sqrt{\frac{\text { Astat }}{\text { Amove }}}$ and Npin $=\sqrt{\frac{\text { Apin }}{\text { Amove }}}$. These data are ported to Matlab for analysis. Refer to Appendix 8 for the C-program codes.


Figure 3.19 Flow chart of the data saving C-program.

### 3.4.2 Control Program

Once the sensor system has been calibrated and characterized, it is deployed in a flexible beam control experiment. A C-program was written to control the position of the flexible beam which was described in Section 2.7. A block diagram of the control system is shown in Figure 3.20 and a flow chart of the control algorithm is shown in Figure 3.21. See Appendix 9 for the C-program codes. A description of the variables used in the control program is shown in Table 3.5 .


Figure 3.20 Block diagram of the control program scheme.

Table 3.5 Description of control variables.

| Variable Name | Description |
| :---: | :---: |
| Y | Position Signal at the $\mathrm{i}^{\text {ih }}$ step |
| Y ReF | Reference Position |
| E | Error (Y $\mathrm{Y}_{\mathrm{N}}-\mathrm{Y}_{\text {ReF }}$ ) at the $\mathrm{i}^{\text {it }}$ step |
| U | Command Signal at the $\mathrm{i}^{\text {th }}$ step |
| D | Output of LPF at the $\mathrm{i}^{\text {ih }}$ step |
| NT | Output of Integrator at the $\mathrm{i}^{\text {th }}$ step |
| $\mathrm{K}_{\mathrm{P}}$ | Position Gain |
| $\mathrm{K}_{\mathrm{D}}$ | Derivative Gain |
| $\mathrm{K}_{\mathrm{I}}$ | Integral Gain |



Figure 3.21 Flow chart of the control C-program.

There are two sections in the control program. The first section displays the real-time position of the beam's end point on the monitor continually so the beam can be set to the same initial position before every test for consistency in testing. After the beam is set to the desired starting position, the program runs the control loop which is executed for 20 seconds, and saves the position, error and control signal data on a floppy disk.

A listing of the control variables and their respective descriptions is shown in Table
3.5. The four control laws used are as follows:

1) Position control law (3.4-1), Figure 3.22 .
2) Position and Derivative control law (3.4-4), Figure 3.24 .
3) Position, Derivative and Integral control law with constant gains (3.4-6), Figure 3.26.
4) Position, Derivative and Integral control law with variable gains (3.4-14), Figures 3.27.

$$
\begin{equation*}
U(i)=K_{P}\left(Y(i)-Y_{R E F}\right) \tag{3.4-1}
\end{equation*}
$$



Figure 3.22 Flow chart of the Kp control.

To reduce the noise in the derivative signal, a first order lowpass filter with a cutoff frequency of 40 Hz was synthesized. The transfer function of the lowpass filter is shown in (3.4-2) and the difference equation is given in (3.4-3). The frequency and phase response of the lowpass filter are plotted in Figure 3.23.

$$
\begin{gather*}
\frac{D(z)}{E(z)}=\frac{0.5095+0.5095 Z^{-1}}{1-0.7548 Z^{-1}}  \tag{3.4-2}\\
D(i)=0.7548 D(i-1)+0.5095 E(i)+0.5095 E(i-1) \tag{3.4-3}
\end{gather*}
$$



Figure 3.23 Frequency and phase response of the control program's lowpass filter

$$
\begin{equation*}
U(i)=K_{P}\left[Y(i)-Y_{R E F}\right]+K_{D}[D(i)-D(i-1)] \tag{3.4-4}
\end{equation*}
$$



Figure 3.24 Flow chart of the $\mathrm{Kp}, \mathrm{Kd}$ control.

To overcome the motor friction in the flexible beam setup, an integrator was synthesized to improve the steady state error of the position of the beam. A block diagram of the system which incorporates the integrator is shown in Figure 3.25. The equation for the integrator is shown in (3.4-5).


Figure 3.25 Block diagram of the motor control system incorporating the integrator.

$$
\begin{gather*}
\operatorname{INT}(\mathrm{i})=\mathrm{INT}(\mathrm{i}-1)+\mathrm{E}(\mathrm{i})  \tag{3.4-5}\\
\mathrm{U}(\mathrm{i})=\mathrm{K}_{\mathrm{P}}\left[\mathrm{Y}(\mathrm{i})-\mathrm{Y}_{\mathrm{REF}}\right]+\mathrm{K}_{\mathrm{D}}[\mathrm{D}(\mathrm{i})-\mathrm{D}(\mathrm{i}-1)]+\mathrm{K}_{1}[\mathrm{NT}(\mathrm{i})] \tag{3.4-6}
\end{gather*}
$$



Figure 3.26 Flow chart of the $\mathrm{Kp}, \mathrm{Kd}$, Ki control with constant gains.

To improve the dynamic response of the flexible beam system, a variable gain control program was implemented. If the error signal is large, the PID gains will be small while if the error signal is small the PID gains will be large. This implementation will help
prevent oscillation of the flexible beam if the error signal is large but will improve the steady state error of the position of the flexible beam when the error signal is small.

To prevent division by zero in the variable gain equations, the current error value $(E(i))$ is replaced by the previous error value $(E(i-1))$ if it is zero. The equations for the variable PID gains are shown in (3.4-8), (3.4-10) and (3.4-12). Each gain is also upper and lower bounded by (3.4-9), (3.4-11) and (3.4-13).

$$
\begin{gather*}
\text { If } E(i)=0 \text { then } E(i)=E(i-1)  \tag{3.4-7}\\
K_{P V}(i)=K_{P} / E(i)  \tag{3.4-8}\\
K_{P} / 2<K_{P V}<2 K_{P}  \tag{3.4-9}\\
K_{D V}(i)=K_{D} / E(i)  \tag{3.4-10}\\
K_{D} / 2<K_{D V}<2 K_{D}  \tag{3.4-11}\\
K_{I V}(i)=K_{I} / E(i)  \tag{3.4-12}\\
K_{I} / 10<K_{I V}<10 K_{I}  \tag{3.4-13}\\
U(i)=K_{P V}\left[Y(i)-Y_{R E F}\right]+K_{D V}[D(i)-D(i-1)]+K_{\text {IV }}[I N T(i)] \tag{3.4-14}
\end{gather*}
$$



Figure 3.27 Flow chart of the $\mathrm{Kp}, \mathrm{Kd}$, Ki control with variable gains.

## CHAPTER 4

## TESTING AND DATA

### 4.1 Signal Time Stability

A number of tests were performed to determine, quantitatively, the characteristics of the sensor modules. These tests include: 1) Time stability, 2) Short range accuracy, and 3) Long range accuracy. These test results are now discussed. Refer to Section 1.2 for a review of the signal flow and a description of signal labels:

### 4.1.1 Time Stability of Pstat

The signal Nstat is created by normalizing the intensity of the incident light on the mobile sensor module (Amove) using the stationary sensor module (Astat). Pstat is the average of the integration of Nstat over one minute. To check the time stability of Pstat, a drift test is performed at the same position over several minutes. Figure 4.1 shows the stability of Pstat at 12.7 mm and Figure 4.2 shows the stability of Pstat at 0 mm . Stability of the signal can be assessed by calculating the trends and the standard deviation.


Figure 4.1 Time stability of Pstat at 12.7 mm .

$$
\begin{aligned}
& \text { Mean }(\text { Pstat })=0.9728 \\
& \operatorname{Std}(\text { Pstat })=0.0018
\end{aligned}
$$

Position uncertainty $=0.95 \mathrm{~mm}$


Figure 4.2 Time stability of Pstat at 0 mm .

$$
\begin{gathered}
\text { Mean (Pstat })=0.7719 \\
\text { Std (Pstat })=0.0006 \\
\text { Position uncertainty }=0.23 \mathrm{~mm}
\end{gathered}
$$

These tests provide a verification of the performance of the sensor: There were no detectable trends in the system during the period of testing.

### 4.1.2 Time Stability of Dpin \& Ppin

The signal Dpin is created by sampling the analog feedback voltage from the laser diode at 10 KHz . The values of Dpin are the DSP's representative values of the sampled voltage. To
check the time stability, one minute averages of Dpin were recorded over a period of 16 minutes as shown in Figures 4.3.


Figure 4.3 Time stability of Dpin.

$$
\text { Mean }(\text { Dpin })=5201.1
$$

$$
\operatorname{Std}(\text { Dpin })=5.1
$$

The signal Npin is created by normalizing the intensity of the incident light on the mobile sensor module (Amove) using the feedback signal from the laser diode (Apin). To check the time stability of Npin, a drift test is performed at the same position over several minutes. Each point plotted in Figure 4.4 is the average of the integration of the Npin over one minute. Stability of the signal can be assessed by calculating the standard deviation.


Figure 4.4 Time stability of Ppin.

Mean $($ Ppin $)=0.9307$
$\operatorname{Std}(\operatorname{Ppin})=0.0026$

Position uncertainty $=1.37 \mathrm{~mm}$

### 4.2 Calibration and Verification Procedures

The purpose of the test is to determine the ability of the system to predict the position of the mobile sensor module. Table 4.1 describes the symbols used in the calibration and verification procedures.

Table 4.1 Description of symbols used in the calibration and verification procedures.

| Variable Name | Description |
| :---: | :---: |
| AL | Actual location (mm) |
| EL | Error in predicted location (mm) |
| PL | Predicted location (mm) |
| P | Pstat or Ppin position signal |
| P_test | Set of P data to predict location |
| P_calibration | Set of P data to calculate calibration polynomial |
| Polynomial | Calibration polynomial |

The following steps summarize the calibration and verification procedures:
Step 1) Create a calibration curve: Sample $P$ at several positions ( $P$ _calibration) to create a table of known locations versus $P$. Generate a polynomial using Matlab to fit the calibration data points (4.2-1):

$$
\begin{equation*}
\text { Polynomial }=f(A L, P \text { _calibtation }) \tag{4.2-1}
\end{equation*}
$$

Step 2) Check the quality of the calibration curve: Predict the position using the calibration polynomial and the calibration position data (4.2-2). Plot the known positions Vs the predicted positions and compare. If the curves are approximately the same the calibration polynomial may be used to perform further position prediction tests.

$$
\begin{equation*}
\text { PL }=\text { Polynomial (P_calibration) } \tag{4.2-2}
\end{equation*}
$$

Step 3) Perform prediction test: Take another set of position data (P_test), at known locations, and predict the position using the previously determined calibration polynomial
(4.2-3). To determine the ability of the polynomial to predict the position of the mobile sensor module correctly, plot the difference between the predicted and actual positions (4.2-4).

$$
\begin{gather*}
\mathrm{PL}=\text { Polynomial(P_test) }  \tag{4.2-3}\\
\mathrm{EL}=\mathrm{AL}-\mathrm{PL} \tag{4.2-4}
\end{gather*}
$$

Flow charts of the calibration and position predicting steps are shown in Figures 4.5 and 4.6 , respectively. The hardware setup used for the testing is described in Section 2.6.


Figure 4.5 Step one of testing -- Calibration.


Figure 4.6 Step three of testing -- Position prediction.

### 4.3 Short-Range Test Using Pstat

The purpose of the short-range test is to determine the system's ability to predict the position of the mobile sensor module over a range of 12.7 mm using Pstat, which is the intensity signal normalized by the intensity signal of the stationary sensor module. Refer to section 2.5 for a description of the test setup. The position repeatability of the mobile sensor module is 0.02 mm . The calibration and prediction data were taken as described in Section 4.3.

The 2 nd order polynomial generated from the calibration data for the short range test is shown in (4.3-1). The calibration curve for the short-range test is shown in Figure 4.7. The solid line is constructed by linearly interpolating the verification data taken at positions indicated by the asterisks. The dashed line, on the other hand, is generated by (4.3-1). The error between the predicted and the actual position, which is calculated using the second set of data $P$ _test as in (4.2-3) and (4.2-4), is plotted in Figure 4.8.

$$
\begin{equation*}
\text { Location }(\mathrm{mm})=101.1071 \text { Pstat }^{2}-21.2036 \text { Pstat }-41.1373 \tag{4.3-1}
\end{equation*}
$$



Figure 4.7 Calibration curve for the short-range test.


Figure 4.8 Location prediction error for the short-range test.

$$
\begin{aligned}
\text { Mean }(\text { Error }) & =0.10 \mathrm{~mm} \\
\text { Std }(\text { Error }) & =0.53 \mathrm{~mm}
\end{aligned}
$$

From the above standard deviation the location of the mobile sensor module can be predicted to within an uncertainty of approximately 0.5 mm .

### 4.4 Long-Range Test Using Pstat

The system was also tested for its ability to predict the position of the mobile sensor module over a range of approximately 600 mm . The ptatform with the mobile sensor module mounted on it was adjusted, by hand, along an aluminum bar with distance markings scribed on it. This enabled testing over a greater distance range but increased the uncertainty in the accuracy of positioning the platform to about 2 mm . The method for determining the calibration polynomial and prediction data set is the same as for the short range test. However, three calibration data sets were taken for three separate calibration polynomials. The order of the calibration polynomials has also been increased to the 4th power to better fit the data over the expanded operating range.

The three polynomial, generated form the three sets of calibration data shown in (4.4-1), (4.4-2) and (4.4-3). The three calibration curves for the long-range test are shown in Figures 4.9, 4.11 and 4.13. The solid lines are constructed by linearly interpolating the verification data taken at positions indicated by the asterisks. The dashed lines, on the other hand, are generated by (4.4-1), (4.4-2) and (4.4-3) respectively. Figures 4.10, 4.12
and 4.14 show the error between the predicted and the actual position from the three calibration polynomials.

$$
\begin{equation*}
\text { Location }(\mathrm{mm})=-0.0226 \mathrm{Pstat}^{4}+0.8816 \text { Pstat }^{3}-12.7834 \mathrm{Pstat}^{2}+ \tag{4.4-1}
\end{equation*}
$$

110.6928 Pstat - 18.0409


Figure 4.9 Calibration curve \#1 for the long-range test using Pstat.


Figure 4.10 Location prediction error from the first calibration polynomial.

$$
\begin{gather*}
\text { Mean (Error) }=-8.3 \mathrm{~mm} \\
\text { Std }(\text { Error })=5.1 \mathrm{~mm} \\
\text { Location }(\mathrm{mm})=-0.0164 \text { Pstat }^{4}+0.6944 \text { Pstat }^{3}-11.1875 \text { Pstat }^{2}+ \\
107.6895 \text { Pstat }^{2}-15.9818 \tag{4.4-2}
\end{gather*}
$$



Figure 4.11 Calibration curve \#2 for the long-range test using Pstat


Figure 4.12 Location prediction error from the second calibration polynomial.

Mean $($ Error $)=4.0 \mathrm{~mm}$
Std $($ Error $)=4.2 \mathrm{~mm}$

Location (mm) $=-0.0175$ Pstat $^{4}+0.7356$ Pstat $^{3}-11.7290$ Pstat $^{2}+$
110.2947 Pstat - 18.6185


Figure 4.13 Calibration curve \#3 for the long-range test using Pstat.

Long Range Pstat Test Usin (4.4-8)- Actual Loc. vs Prediction Error


Figure 4.14 Location prediction error from the third calibration polynomial.

$$
\begin{gathered}
\text { Mean }(\text { Error })=-3.8 \mathrm{~mm} \\
\mathrm{Std}(\text { Error })=5.0 \mathrm{~mm}
\end{gathered}
$$

From the above three standard deviations ( $5.1 \mathrm{~mm}, 4.2 \mathrm{~mm}$, and 5.0 mm ) the location of the mobile sensor module can be predicted to within an uncertainty of approximately 5 mm with the long range test using Pstat.

### 4.5 Long-Range Test Using Ppin

The long range test is repeated using the laser diode output power signal (Apin) as the normalization signal instead of Amove. This will free up the stationary sensor module
which was previously used as the normalization signal, for use as another mobile sensor module. The test is otherwise the same as the previous long-range test.

For this test, two calibration polynomials were generated, as shown in (4.5-1) and (4.5-2). The tests of the calibration polynomials are shown in Figures 4.15 and 4.17. Note that the coefficients for the calibration polynomials when using Ppin are significantly different from the coefficients for the calibration polynomials when using Pstat: (4.4-1), (4.4-2) and (4.4-3). This is the result of a difference in the amplitudes of the signals used to normalize the intensity of the incident light on the mobile sensor module.

$$
\begin{align*}
\text { Location }(\mathrm{mm})= & -0.0012 \mathrm{Ppin}^{4}+0.0976 \mathrm{Ppin}^{3}-3.0817 \mathrm{Ppin}^{2}+ \\
& 55.1808 \mathrm{P} \operatorname{pin}-0.7590
\end{align*}
$$



Figure 4.15 Calibration curve \#1 for the long-range test using Ppin.


Figure 4.16 Location prediction error from the first calibration polynomial.

$$
\begin{gather*}
\text { Mean }(\text { Error })=3.7 \mathrm{~mm} \\
\text { Std }(\text { Error })=4.0 \mathrm{~mm} \\
\text { Location }(\mathrm{mm})=-0.0013 \mathrm{Ppin}^{4}+0.1083 \mathrm{Ppin}^{3}-3.3126 \mathrm{Ppin}^{2}+ \\
56.7829 \text { Ppin }-3.7504 \tag{4.5-2}
\end{gather*}
$$



Figure 4.17 Calibration curve \#2 for the long-range test using Ppin


Figure 4.18 Location prediction error from the second calibration polynomial.

$$
\begin{aligned}
& \text { Mean }(\text { Error })=4.7 \mathrm{~mm} \\
& \text { Std }(\text { Error })=3.9 \mathrm{~mm}
\end{aligned}
$$

From the above three standard deviations ( 4.0 mm , and 3.9 mm ) the location of the mobile sensor module can be predicted to within an uncertainty of approximately 4 mm with the long range test using Ppin, which is approximately the same as when using Pstat with the long range test.

### 4.6 Summary of Test Results

Table 4.2 shows a summary of the test results discussed in this chapter
Table 4.2 Summary of test results.

| Test | Location Repeatability <br> $(+/-\mathrm{mm})$ | Range (mm) | Resolution (mm) |
| :---: | :---: | :---: | :---: |
| Short range w/ Pstat | 0.02 | 25.4 | 0.5 |
| Long range w/Pstat | 2 | 600 | 5 |
| Long range w/ Ppin | 2 | 600 | 4 |

From Table 4.2, it is clear that the ability of the system to predict the location of the mobile sensor module is much greater when used in the short range test where the resolution is approximately 0.5 mm that with the long range test where the resolution is approximately 5 mm . Some possible explanations are as follows:

1) The location repeatability for the short range test, when the verner table is used, is 0.02 mm while the position repeatability for the long range test, when the mobile sensor module is positioned by hand, is 2 mm .
2) The amplitude of the position signal from the mobile sensor module is small when at the maximum distance from the laser diode during the long range test ( 600 mm ). This will lower the signal-to-noise ratio and hence the sensor resolution.
3) When the mobile sensor module is in the maximum distance range from the laser diode ( 200 mm to 600 mm ) the amount of background noise signal, the majority of which comes from the room lights, increases.

## CMAPTER 5

## FLEXIBLE BEAM EXPERIMENT

### 5.1 Control Implementation

The objective of the flexible beam control experiment is to position the end of the flexible beam at a desired location, in minimum time and with minimum oscillation, using the output position signal from the sensor module as the position feedback. Refer to Chapter 2 for the discussion of the hardware implementation of the sensor module and Section 2.6 for a diagram of the flexible beam setup. The three performance indices used to monitor the relative responses of the control experiment for a variety of control laws and control gains are: 1) $\sum \mathrm{E}(\mathrm{i})^{2}$ sum of the squared position error, 2) the final steady state position error of the end of the flexible beam, and 3) the $5 \%$ settling time of the flexible beam. Refer to Section 3.4.2 for a description of the four control laws implemented.

To perform each test, the end of the beam is located at $+1-20 \mathrm{~cm}$ from the neutral command position, as the initial positions. The beam is then commanded to move to the neutral position ( 0 mm ) by the control program which also records the performance indices previously mentioned.

### 5.2 Test Results

The results of these four control laws are now summarized.

### 5.2.1 Position Control with Constant Gain

The first control law implemented was position control. See (3.4-1) and Figure 3.22 for a review of the position control law.

To test the effects of the position control law the position gain $K_{p}$ was varied and the three performance indices used to monitor the relative responses of the control experiment were recorded, plotted, and compared. The test results are tabulated in Tables 5.2 and 5.3 for, respectively, initial conditions of 20 cm and -20 cm .

The following is a listing of typical time plots of the $Y$ and $U$ signals:

Table 5.1: Listing of typical time plots of the $Y$ and $U$ signals for $K_{p}$ tests.

| Figure | Initial Position (cm) | $\mathrm{K}_{\mathrm{p}}$ |
| :---: | :---: | :---: |
| 5.1 | +20 | 5 |
| 5.2 | +20 | 10 |
| 5.3 | +20 | 20 |
| 5.4 | +20 | 30 |
| 5.5 | -20 | 10 |
| 5.6 | -20 | 20 |
| 5.7 | -20 | 30 |
| 5.8 |  |  |

Table 5.2 Summary of results: Initial position $=+20 \mathrm{~cm}, \mathrm{~K}_{\mathrm{P}}$ control

| $K_{P}$ | Avg |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\Sigma E(i)^{2}$ | Std |  |  |  |  |  |
| $\Sigma E(i)^{2}$ | Avg Final <br> Error (cm) | Std Final <br> Error (cm) | Avg Settling <br> Time (sec.) | Std Settling <br> Time (sec.) |  |  |
| 5 | 46.8 | 1.8 | -2.18 | 0.37 | N. A. | N. A. |
| 10 | 108.2 | 12.4 | 6.10 | 0.79 | N. A. | N. A. |
| 20 | 75.2 | 6.37 | -1.78 | 0.63 | N.A. | N. A. |
| 30 | N.A. | N. A. | N. A. | N.A. | N.A. | N. A. |



Figure 5.1 Typical $Y$ and $U$ time plots for $K_{p}=5$.


Figure 5.2 Typical Y and U time plots for $\mathrm{K}_{\mathrm{p}}=10$.


Figure 5.3 Typical Y and U time plots for $\mathrm{K}_{\mathrm{p}}=20$.


Figure 5.4 Typical $Y$ and $U$ time plots for $K_{P}=30$.

From the summary of the $K_{p}$ test results in Table 5.2, when the initial position is $+20 \mathrm{~cm}, \mathrm{~K}_{\mathrm{F}}=20$ produces the best results when measured with performance indices $\Sigma E(i)^{2}$ and final position error. Due to the friction in the motor positioning system, no value of $K_{P}$, when the initial position is +20 cm will produce a final error within the $+/-5 \%$ range $(1.0 \mathrm{~cm})$, which can be seen as the lack of settling times in Table 5.2 . If $\mathrm{K}_{\mathrm{P}}$ is too much greater than 20, the system will become unstable as shown in Figure 5.4.

Table 5.3 Summary of results: Initial position $=-20 \mathrm{~cm}, K_{p}$ control

| $\mathrm{K}_{\mathrm{P}}$ | Avg |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\Sigma \mathrm{E}(\mathrm{i})^{2}$ | $\Sigma \mathrm{E}(\mathrm{i})^{2}$ | Avg Final <br> Error (cm) | Std Final <br> Error (cm) | Avg Settling <br> Time (sec.) | Std Settling <br> Time (sec.) |  |
| 5 | 314 | 20.8 | 12.30 | 0.41 | N. A. | N. A. |
| 10 | 47.6 | 2.7 | 4.12 | 0.38 | N. A. | N. A. |
| 20 | 29.4 | 5.8 | 1.00 | 0.73 | 9.0 | 3.9 |
| 30 | 87.2 | 9.6 | -0.52 | 1.10 | 10.5 | 1.3 |



Figure 5.5 Typical Y and U time plots for $\mathrm{K}_{\mathrm{p}}=5$.


Figure 5.6 Typical $Y$ and $U$ time plots for $K_{P}=10$.


Figure 5.7 Typical Y and U time plots for $\mathrm{K}_{\mathrm{P}}=20$.


Figure 5.8 Typical Y and U time plots for $\mathrm{K}_{\mathrm{p}}=30$.
From the summary of the $K_{p}$ test results in Table 5.3, when the initial position is $20 \mathrm{~cm}, \mathrm{~K}_{\mathrm{p}}=20$ will produce the best results when measured with performance indices $\sum \mathrm{E}(\mathrm{i})^{2}$ and settling time, however, $\mathrm{K}_{\mathrm{p}}=30$ produces the best results when measured by the final position error.

### 5.2.2 Position and Derivative Control with Constant Gain

As discussed in Section 3.4.2, a position and derivative ( $\mathrm{K}_{\mathrm{p}}$ and $\mathrm{K}_{\mathrm{D}}$ ) control law was synthesized to improve the response of the flexible beam. See (3.4-4) and Figure 3.24 for a review of the position and derivative control law.

To test the effects of a variety of derivative gains, the $K_{p}$ gain was set to 10 and the derivative gain $K_{D}$ was varied. The results of the tests when the initial position of the
flexible beam is +20 cm are shown in Table 5.5 and when the initial position is -20 cm in Table 5.6. The following is a listing of typical time plots of the $Y$ and $U$ signals with $K_{p}$ to 10 :

Table 5.4: Listing of typical time plots of the $Y$ and $U$ signals for $K_{p} K_{p}$ tests.

| Figure | Initial Position (cm) | $K_{D}$ |
| :---: | :---: | :---: |
| 5.9 | +20 | 5 |
| 5.10 | +20 | 10 |
| 5.11 | +20 | 20 |
| 5.12 | +20 | 30 |
| 5.13 | -20 | 10 |
| 5.14 | -20 | 20 |
| 5.15 | -20 | 30 |
| 5.16 | -20 |  |

Table 5.5 Summary of results: Initial position $=+20 \mathrm{~cm}, \mathrm{~K}_{\mathrm{p}}, \mathrm{K}_{\mathrm{D}}$ control, $\mathrm{K}_{\mathrm{P}}=10$

| $\mathrm{K}_{\mathrm{D}}$ | Avg |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\Sigma \mathrm{E}(\mathrm{i})^{2}$ | $\Sigma \mathrm{E}(\mathrm{i})^{2}$ | Error (cm) | Avg Final | Std Final <br> Error (cm) | Avg Settling <br> Time (sec.) | Std Settling <br> Time (sec.) |
| 5 | 62.0 | 2.3 | -4.14 | 0.43 | N A. | N. A. |
| 10 | 46.4 | 12.4 | -2.66 | 1.54 | N.A. | N. A. |
| 20 | 37.2 | 2.2 | -0.31 | 1.44 | 2.3 | 0.2 |
| 30 | 37.0 | 1.8 | -0.27 | 0.60 | 2.5 | 0.3 |



Figure 5.9 Typical Y and U time plots for $\mathrm{K}_{\mathrm{D}}=5$.


Figure 5.10 Typical $Y$ and $U$ time plots for $K_{D}=10$.


Figure 5.11 Typical Y and U time plots for $\mathrm{K}_{\mathrm{D}}=20$.


Figure 5.12 Typical $Y$ and $U$ time plots for $K_{D}=30$.

From the summary of test results in Table $5.5, \mathrm{~K}_{\mathrm{D}}=30$ produces the best results, which are slightly better than those obtained with $\mathrm{K}_{\mathrm{D}}=20$, when measured with performance indices $\Sigma E(i)^{2}$ and the final position error. $K_{D}=20$ produces the best results when measured by the settling time, but only slightly better than $K_{D}=30$. Therefore, it is likely that the optimum $K_{D}$, when $K_{P}$ is to 10 and the initial position is +20 cm , is approximately 25 .

Table 5.6 Summary of results: Initial position $=-20 \mathrm{~cm}, \mathrm{~K}_{\mathrm{P}} \mathrm{K}_{\mathrm{D}}$ control, $\mathrm{K}_{\mathrm{P}}=10$

| $K_{D}$ | Avg |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\Sigma E(\mathrm{i})^{2}$ | Std <br> $\Sigma(\mathrm{i})^{2}$ | Avg Final <br> Error (cm) | Std Final <br> Error (cm) | Avg Settling <br> Time (sec.) | Std Settling <br> Time (sec.) |  |
| 5 | 27.6 | 4.5 | -1.94 | 0.66 | N. A. | N. A. |
| 10 | 21.8 | 1.7 | -1.09 | 0.26 | N.A. | N. A. |
| 20 | 22.7 | 1.5 | -0.68 | 0.65 | 2.2 | 0.3 |
| 30 | 20.3 | 1.6 | -0.06 | 0.90 | 2.2 | 0.7 |



Figure 5.13 Typical Y and U time plots for $\mathrm{K}_{\mathrm{D}}=5$.


Figure 5.14 Typical $Y$ and $U$ time plots for $K_{D}=10$.


Figure 5.15 Typical Y and U time plots for $\mathrm{K}_{\mathrm{D}}=20$.


Figure 5.16 Typical $Y$ and $U$ time plots for $K_{D}=30$.

From the summary of test results in Table $5.6, \mathrm{~K}_{\mathrm{D}}=30$ produces the best results when measured with performance indices $\Sigma \mathrm{E}(\mathrm{i})^{2}$, the final position error, and the settling time.

### 5.2.3 Position, Derivative and Integral Control with Constant Gain

As discussed in Section 3.4.2, an integrator was synthesized to overcome the motor friction in the flexible beam setup and to improve the steady state error of the position of the beam. See (3.4-6) and Figure 3.26 for a review of the control law. The equation for the integrator is shown in (3.4-5).

To test the effects of a variety of integrator gains, the $K_{P}$ and $K_{D}$ gains were both set to 10 and the integrator gain $\mathrm{K}_{1}$ was varied. The results of the tests when the initial position of the flexible beam is +20 cm is shown in Table 5.8 and when the initial position is -20 cm is shown in Table 5.9 .

The following is a listing of typical time plots of the Y and U signals with $\mathrm{K}_{\mathrm{P}}$ and $K_{D}$ set to 10 :

Table 5.7: Listing of typical time plots of the $Y$ and $U$ signals for constant gains tests.

| Figure | Initial Position (cm) | $\mathrm{K}_{\mathrm{I}}$ |
| :---: | :---: | :---: |
| 5.17 | +20 | 0.005 |
| 5.18 | +20 | 0.01 |
| 5.19 | +20 | 0.02 |
| 5.20 | -20 | 0.04 |
| 5.21 | -20 | 0.005 |
| 5.22 | -20 | 0.02 |
| 5.23 | -20 | 0.04 |
| 5.24 |  |  |

Table 5.8 Summary of results: Initial position $=+20 \mathrm{~cm}$, Constant $K_{P} K_{D} K_{1}$ control $-K_{P}=10, K_{D}=10$.

| $\mathrm{K}_{\mathrm{I}}$ | Avg |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\Sigma \mathrm{E}(\mathrm{i})^{2}$ | $\Sigma \mathrm{E}(\mathrm{i})^{2}$ | Error (cm) | Avg Final | Std Final <br> Error (cm) | Avg Settling <br> Time (sec.) | Std Settling <br> Time (sec.) |
| 0.005 | 55.5 | 5.1 | 0.43 | 0.34 | 10.9 | 1.3 |
| 0.01 | 36.4 | 2.7 | -2.07 | 0.19 | N.A. | N.A. |
| 0.02 | 32.6 | 2.1 | -0.39 | 0.29 | 7.9 | 7.2 |
| 0.04 | 48.8 | 9.8 | 0.21 | 0.71 | 8.6 | 3.1 |



Figure 5.17: Typical Y and U time plots for $\mathrm{K}_{\mathrm{I}}=0.005$ constant gain.


Figure 5.18: Typical $Y$ and $U$ time plots for $\mathrm{K}_{\mathrm{I}}=0.01$ constant gain.


Figure 5.19: Typical $Y$ and $U$ time plots for $K_{I}=0.02$ constant gain.


Figure 5.20: Typical $Y$ and $U$ time plots for $\mathrm{K}_{\mathrm{I}}=0.04$ constant gain.

From the summary of test results in Table $5.8, \mathrm{~K}_{1}=0.02$ produces the best results when measured with performance indices $\Sigma E(i)^{2}$ and settling time but $K_{1}=0.04$ produces the best results when measured by the final position error. Therefore, it is likely that the optimum $K_{1}$, when $K_{p}$ and $K_{D}$ are set to 10 and the initial position is +20 cm , is approximately 0.03 .

Table 5.9 Summary of results: Initial position $=-20 \mathrm{~cm}$, Constant $K_{P} K_{D} K_{1}$ control $-K_{P}=10, K_{D}=10$

| $\mathrm{K}_{1}$ | Avg |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\Sigma \mathrm{E}(\mathrm{i})^{2}$ | $\Sigma \mathrm{E}(\mathrm{i})^{2}$ | Error (cm) | Avror (cm) | Time (sec.) | Time (sec.) |  |
| Erd |  |  |  |  |  |  |
| 0.005 | 22.5 | 0.6 | -0.15 | 0.85 | 2.0 | 0.1 |
| 0.01 | 25.8 | 0.8 | 1.16 | 0.26 | N. A. | N. A. |
| 0.02 | 27.3 | 1.5 | 0.36 | 0.22 | 15.7 | 1.5 |
| 0.04 | 29.6 | 1.7 | -0.12 | 0.34 | 7.8 | 4.4 |



Figure 5.21: Typical Y and U time plots for $\mathrm{K}_{\mathrm{i}}=0.005$ constant gain.


Figure 5.22: Typical Y and U time plots for $\mathrm{K}_{1}=0.01$ constant gain


Figure 5.23: Typical Y and U time plots for $\mathrm{K}_{1}=0.02$ constant gair


Figure 5.24: Typical $Y$ and $U$ time plots for $K_{I}=0.04$ constant gain

From the summary of test results in Table 5.9 , when the initial position is -20 cm , $\mathrm{K}_{\mathrm{I}}=0.005$ produces the best results when measured with performance indices $\Sigma \mathrm{E}(\mathrm{i})^{2}$ and settling time. However, due to friction and backlash in the DC motor system, the results when $K_{1}=0.005$ may be unusually optimistic.

If the test results when $\mathrm{K}_{\mathrm{I}}=0.005$ are disregarded, $\mathrm{K}_{\mathrm{I}}=0.01$ produces the best results when measured with the performance index $\Sigma \mathrm{E}(\mathrm{i})^{2}$ but $\mathrm{K}_{\mathrm{I}}=0.04$ produces the best results when measured by the final position error and the settling time.

### 5.2.4 Position, Derivative and Integral Control with Variable Gain

To attempt to improve on the response of the flexible beam system with constant gain, a control law with variable gains was implemented as described in Chapter 3.4.2. The variable gain equations are shown in (3.4-7) to (3.4-14) and the block diagram of the variable gain control law is shown in Figure 3.27. The results of the tests when the initial position of the flexible beam is +20 cm is shown in Table 5.11 and when the initial position is -20 cm is shown in Table 5.12.

The following is a listing of typical time plots of the $Y$ and $U$ signals with $K_{P}$ and $K_{D}$ set to 10 :

Table 5.10: Listing of typical time plots of the $Y$ and $U$ signals for variable gains tests.

| Figure | Initial Position (cm) | $\mathrm{K}_{1}$ |
| :---: | :---: | :---: |
| 5.25 | +20 | 0.005 |
| 5.26 | +20 | 0.01 |
| 5.27 | +20 | 0.02 |
| 5.28 | -20 | 0.04 |
| 5.29 | -20 | 0.005 |
| 5.30 | -20 | 0.02 |
| 5.31 | -20 | 0.04 |
| 5.32 |  |  |

Table 5.11 Summary of results: Initial position $=+20 \mathrm{~cm}$, Variable $K_{P} K_{D} K_{I}$ control $-K_{P}=10, K_{D}=10$.

| $\mathrm{K}_{\mathrm{I}}$ | Avg |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\Sigma \mathrm{E}(\mathrm{i})^{2}$ | Std <br> $\Sigma \mathrm{E}()^{2}$ | Avg Final <br> Error (cm) | Std Final <br> Error (cm) | Avg Settling <br> Time (sec.) | Std Settling <br> Time (sec.) |  |
| 0.005 | 63.0 | 5.6 | 0.26 | 0.14 | 9.4 | 1.1 |
| 0.01 | 44.0 | 4.6 | -2.21 | 0.19 | N.A. | N.A. |
| 0.02 | 33.8 | 2.5 | -0.31 | 0.31 | 3.0 | 0.1 |
| 0.04 | 45.8 | 0.8 | 0.10 | 0.30 | 4.1 | 0.2 |



Figure 5.25: Typical $Y$ and $U$ time plots for $K_{I}=0.005$ variable gain.


Figure 5.26: Typical Y and U time plots for $\mathrm{K}_{\mathrm{I}}=0.01$ variable gain.


Figure 5.27: Typical $Y$ and $U$ time plots for $K_{I}=0.02$ variable gain.


Figure 5.28: Typical $Y$ and $U$ time plots for $K_{I}=0.04$ variable gain.

From the summary of test results in Table $5.11, \mathrm{~K}_{\mathrm{I}}=0.02$ produces the best results when measured with performance indices $\Sigma \mathrm{E}(\mathrm{i})^{2}$ and settling time but $\mathrm{K}_{1}=0.04$ produces the best results when measured by the final position error. Therefore, it is likely that the optimum $K_{\mathrm{l}}$, when $\mathrm{K}_{\mathrm{p}}$ and $\mathrm{K}_{\mathrm{D}}$ are set to 10 and the initial position is +20 cm , is approximately 0.03 for the variable gain tests.

Table 5.12 Summary of results: Initial position $=-20 \mathrm{~cm}$, Variable $K_{P} K_{D} K_{I}$ control $-K_{P}=10, K_{D}=10$.

| $\mathrm{K}_{\mathbf{1}}$ | Avg |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\Sigma \mathrm{E}(\mathrm{i})^{2}$ | $\Sigma \mathrm{E}(\mathrm{i})^{2}$ | Avg Final |  |  |  |  |
| Error (cm) | Std Final |  |  |  |  |  |
| Error (cm) | Avg Settling <br> Time (sec.) | Std Settling <br> Time (sec.) |  |  |  |  |
| 0.005 | 22.2 | 3.3 | 0.74 | 0.32 | 2.3 | 0.4 |
| 0.01 | 24.0 | 2.0 | 1.08 | 0.20 | 5.3 | 4.9 |
| 0.02 | 24.6 | 1.3 | 0.48 | 0.26 | 15.8 | 2.5 |
| 0.04 | 25.8 | 1.9 | -0.28 | 0.83 | 11.7 | 1.5 |



Figure 5.29: Typical $Y$ and $U$ time plots for $K_{1}=0.005$ variable gain.


Figure 5.30: Typical $Y$ and $U$ time plots for $K_{I}=0.01$ variable gain.


Figure 5.31: Typical Y and U time plots for $\mathrm{K}_{\mathrm{I}}=0.02$ variable gain.


Figure 5.32: Typical $Y$ and $U$ time plots for $K_{I}=0.04$ variable gain.

From a comparison of the data obtained during the constant and variable gain tests for the initial position of -20 cm (Tables 5.9 and 5.12 ), it is determined that the optimum $\mathrm{K}_{\mathrm{I}}$ is the same for both conditions: If the test results when $\mathrm{K}_{\mathrm{I}}=0.005$ are disregarded, $\mathrm{K}_{1}=$ 0.01 will produce the best results when measured with the performance indices $\Sigma E(i)^{2}$ and settling time but $\mathrm{K}_{\mathrm{I}}=0.04$ produces the best results when measured by the final position error.

From a comparison of the summary of the test results between the constant and variable gain control laws, Tables 5.8 and 5.11, when the initial position is +20 cm there is no improvement in the performance indices. However, when the initial position is -20 cm , the results of which are summarized in Tables 5.9 and 5.12 , there is an approximately $10 \%$ improvement in the performance indices. Also, the initial overshoot of the position is decreased by approximately half when the variable gain is used.

## CHAPTER 6

## CONCLUSIONS

In this thesis work, an optical position sensor has been designed and implemented

### 6.1 Sensor Module Characterization

The characteristics of the sensor module were analyzed using various standard test procedures such as drift stability, short-range repeatability, and long-range repeatability. A summary of these test results are tabulated in Table 6.1 below. Based on these results, the following conclusions are now drawn:

Table 6.1 Summary of test results (See Chapter 4).

| Test | Location Repeatability <br> $(+/-\mathrm{mm})$ | Range (mm) | Resolution (mm) |
| :---: | :---: | :---: | :---: |
| Short range w/ Pstat | 0.02 | 25.4 | 0.5 |
| Long range w/ Pstat | 2 | 600 | 5 |
| Long range w/ Ppin | 2 | 600 | 4 |

1) The system can predict the location of the mobile sensor module to an accuracy of 0.5 mm when used in the short-range test where the known position repeatability is 0.02 mm (Refer to Section 4.3 for a review of the short-range test).
2) The system can predict the location of the mobile sensor module to an accuracy of 5 mm when used in the long-range test where the known position repeatability is
approximately 2 mm and the stationary sensor module output is used as the normalization signal (Refer to Section 4.4 for a review of the long-range test when the stationary sensor module output is used as the normalization signal).
3) The system can predict the location of the mobile sensor module to an accuracy of 4 mm when used in the long-range test where the known position repeatability is approximately 2 mm and the feedback signal from the laser diode's output power is used as the normalization signal (Refer to Section 4.5 for a review of the long-range test when the feedback signal from the laser diode's output power is used as the normalization signal).

### 6.2 Flexible Beam Experiment

Based on the test results of the flexible beam experiment, which tested the ability of the control system (Section 2.7) to position the flexible beam to the command position when the position feedback signal is produced by the mobile sensor module attached to the end of the flexible beam, the following conclusion can be made: The system can consistently position the end of the flexible beam to within 5 mm of the command position in approximately 8 seconds when used with a properly tuned PID controller.

Also note that there is an advantage of using a microprocessor based control system over an analog-circuit based control system for the following reasons: 1) the properties of the control system can be easily changed by editing the control software rather than making changes to analog circuit components, 2) the coefficient values in the control software are stable over time where the component values of an analog circuit can
change due to environmental conditions and over time, and 3) complex logic and arithmetic operations can be easily implemented by a microprocessor.

### 6.3 Future Improvements

The following improvements are proposed for future work to increase the ability of the system to predict the location of the mobile sensor module:

1) By far the greatest improvement to the system would be to incorporate the Time Sampling scheme, as discussed in Section 2.5.1, which would require the use of a floating point processor. This would eliminate the need for the analog electronics required to pre-process the signal from the sensor modules and enable the DSP to sample to time between the data pulses from the TSL220 light sensor to an accuracy of 0.1 uS .
2) The use of a floating point processor would also improve the realization of the digital filters, which can only be approximated with a fixed point processor.
3) Since the output frequency of the TSL220 light sensor is determined by an external capacitor for a given incident light intensity, it is critically important that the value of the capacitor remain very stable. This can either be accomplished with an active monitor and compensator for the capacitance or a forthcoming version of the TSL220 light sensor that no longer requires an external capacitor to control the output frequency.
4) All digital monitoring of the laser diode's output power feedback signal would remove the need for the analog highpass filter circuit (Figure 2.7) and improve the stability of the monitored feedback signal.
5) A variable lens setup which can create either a rapidly diverging cone of light from the laser diode would optimize the system for the short range discussed in Section 4.3 or a slowly diverging cone of light which would optimize the system for the long range tests discussed in Sections 4.4 and 4.5 .
6) Further optimization of the control program for the flexible beam experiment and the incorporation of the dynamics of the flexible beam to realize a compensator in the control program would improve the system's dynamics.
7) The use of multiple sensor modules on the flexible beam to monitor and compensate for the flexure in the flexible beam would further improve the system's dynamics.

## APPENDIX 1: SPECIFICATION SHEET OF THE TSL220 LIGHT TO FREQUENCY CONVERTER

- High-Resolution Conversion of Light Intensity to Frequency
- Wide Dynamic Range . . . 118 dB
- Varlable (and Single) Supply Range . . . 5 V to 10 V
- High Linearlty . . . Typically Within $2 \%$ of FSR ( $C=100 \mathrm{pF}$ )
- High Sensltivity . . . Can Detect Change of $0.01 \%$ of FSA
- CMOS Compatlble Output for Digltal Processing
- Minimum Extarnal Components
- Microprocessor Compatiblo


## description

The TSL220 consists of a large-area photodicde and a current-to-frequency corverter. The output voltage is a pulse rain and its frequency is directly proportional to the light Intensity (irradiance) on the photodiode. The output is CMOSt compatible and its frequency may be measured using pulse counting, period timing, or integration techniques. The TSL220 is ideal for light-sensing applications requiting wide dynamic range, high sensitivity, and high noise immunity. The output frequency range is determined by an extemal capacitor; hence, the desired output frequency is adjustable for a given light intensity at the input. The TSL220 is characterized for operation over the temperature range of $-25^{\circ} \mathrm{C}$ to $70^{\circ} \mathrm{C}$.
mechanical data
The photodiode and current-to-frequency converte- are packaged in a clear plastic 8 -pin dual-in-line package. The active chip area is typically $4,13 \mathrm{~mm}^{2}\left(0.0064 \mathrm{in}^{2}\right)$.

t Use of LSTTL logic famlies may require a $3300-\Omega$ pulldown resistor on the output.

## PARAMETER MEASUREMENT INFORMATION



TEST CIRCUT


OUTPUT WAVEFORM

Figure 1. Switching Times
NOTE: OUDOU wavetom is monitored on an cscilloscope with the feltowing charactenstics: $\mathrm{Fi}_{\mathrm{i}} \times 1 \mathrm{M} \Omega . \mathrm{C}_{\mathrm{j}} \times 6.5 \mathrm{pF}$.


## UGHT-TO-FREQUENCY CONVERTER

functional block dagram

absolute maximun ratings over operating free-air temperature range (unless otherwise noied)
Supply vcltage, $V_{C c}$ (see Note 1) ............................................................................... 12 V


Lead temperature 1.5 mm ( $1 / 16$ inch) trom case tor 10 seconds ................................... . . $200^{\circ} \mathrm{C}$

HOTE : All vonsge values are vith respoct to GND (pin 5).
recommended operating conditions

|  | MIM | NOM | Max | UNTT |
| :---: | :---: | :---: | :---: | :---: |
| Sudpiy vorrge, VCC | 4 | 5 | $: 01$ | $v$ |
| Outur trequency, $t_{0} \quad(\mathrm{C} \times 100 \mathrm{ph}$ |  |  | 7501 | kitz |
| Opmating fro-er tempershur rango. TA | -25 |  | 701 | \% |

electical charecteristics at $V_{C C}=5 \mathrm{~V}, T_{A}=25^{\circ} \mathrm{C}$ (see Figure y)

| PARAMETER |  | TEST CO | Ions | Min | n'p | max | Ux11 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| vom | Peak outout votagy | R E - 50 ko |  | 3 | 4 |  | $v$ |
| Icc | Supply curten | $\mathrm{C} \times 100 \mathrm{oF}$. | Enwo |  | 7.5 | 101 | $m$ |

operating characieristics at $V_{C C}=5 \mathrm{~V}, \mathrm{~T}_{\mathrm{A}}=25^{\circ} \mathrm{C}$ (see Figure 1)

| PARAMETER |  | TEST CONOITIONS |  |  | нHI: | TYP | MAX | UNIT |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| to Output frequency |  | $E_{e}+125$ w/er ${ }^{2}$ | $\lambda=880 \mathrm{~nm}$. | $\mathrm{C}=100 \mathrm{pF}$ | 50 | 150 | 250 | $\mathrm{kH}_{\mathrm{k}}$ |
|  |  | $E_{\text {e }}=0$. | $\mathrm{C}=100$ of |  | 0 | 1 | 501 | $\mathrm{Hz}_{2}$ |
| Ww | Output pulse duration | C. 470 pF |  |  |  | : |  | ${ }^{4}$ |
| t | Outun putse nso tume | C= 100 pr |  |  |  | 20 |  | $n 5$ |
| 4 | Outout pulse lall hme | $C=100 \mathrm{pF}$ |  |  |  | 125 |  | $n \mathrm{n}$ |

## ISGHT-TO-FREQUENCY CONVERTER



LIGHT-TO-FREQUENCY CONVERTER

## TYPICAL CHARACTERISTICS



Figure 8


Flgure 9


Figure 10

## APPENDIX 2: SPECIFICATION SHEET OF THE PANASONIC LN9705 LASER DIODE





Hote 1 . ith is defined at ibe cross point of current axis and
the line conecting the paints of $\mathrm{l}=\mathrm{W}$ and 3 aki.
Hote 2. Sampling inspection by lot.
Hote 3. fincle of 502 peak intensity (funh)
Hote 4 . 7 is defined by the slope of the fine conerting the point of lok and 3 rk .




## APPENDIX 3: TIME SAMPLING ASSEMBLY CODE

Table A3.1: Description of memory locations in the Time Sampling code.

| Memory Location | Memory Page | Description |
| :---: | :---: | :---: |
| 2 | 0 | DSP Timer |
|  | 8 | Saved Timer Values |

int0_ser
;interrupt 0 handeler
;the interrupt zero is triggered whenever the TSL220
; outputs a pulse [time proportional to light intensity]
;timer stuff
;save the old timer value in "timer"
ldpk $>0$
lac $>2$
ldpk $>8$
larp $>1$
sacl *+
;reset the timer
ldpk $>0$
lalk $>$ ffff
$\mathrm{sacl}>2$
jump out of the interrupt routine
eint
ret

## APPENDIX 4A

## DESCRIPTION OF INVERSION ASSEMBLY LANGUAGE PROGRAM VARIABLES

Table A4.1 Symbol name cross reference.

| Symbol in Figure 2.17 | Equivalent variable in DSP program |
| :---: | :---: |
| $\mathrm{I}_{\mathrm{E}}$ | ec_hi \& ec lo |
| F | fc |
| $\mathrm{T}_{\mathrm{U}}$ | T |
| $2^{N}$ | N |

Table A4.2 Description of assembly program variables.

| Program Variable | Description |
| :---: | :--- |
| ticker | Number of inversion iterations executed |
| fc | Current frequency value |
| fp | Previous frequency value |
| ec_lo | Low two bytes of the current error |
| ec_hi | Hi two bytes of the current error |
| ep_lo | Low two bytes of the previous error |
| ep_hi | Hi two bytes of the previous error |
| $\mathrm{e}_{-}$temp | Temporary error signal |
| N | Target bound |
| T | Time signal to be inverted |

## APPENDIX 4B

## INVERSION ASSEMBLY PROGRAM

| ;i.asm |  |  |
| :--- | :--- | :--- |
| t_in | .set | 1011 |
| f_last | .set | 16352 |
| loops | .set | 20 |
| Ki | .set | 6 |
| Edivide | .set | 7 |

;declare memory locations

```
.bss ticker,1
.bss fc,1
.bss fp,1
.bss ec_lo,l
.bss ec_hi,1
.bss ep_lo,1
.bss ep_hi,I
.bss e_temp,1
.bss N,l
.bss T,1
.bss xl,1
.bss x2,1
```

;interrupt flow table
;the "vectors" section is a label to link with the link.cmd
; file to load the interrupt flow table at address Oh
.sect "vectors"
b start
b int0_ser
bintl_ser
;the "text" section is a label to link with the link.cmd
; file to load the main program at address 50 h
.text
start: ;start of program
dint
ldpk 8
lrlk 1, 1280
zac
lrlk 2, 1024
lrlk 3, 512
again: larp 2
sacl *+
larp 3
banz again
lack loops
sacl ticker
$; N=2 \wedge 24$
lalk 256
$\operatorname{sacl} \mathrm{N}$
lalk $t$ in
sacl T
lark 7, 20
lalk f last
sacl fc
loop: save the old values
lac fc
sacl fp
larp 1
$\mathrm{sacl}^{*+}$
lac ec_lo
sacl ep_lo
lacec_hi
sacl ep_hi
load the previous error in the acc

```
lac ep_lo
addh ep_hi
;"multiply" by Ki
;Ki=2^(-n)
;shift left by 16-n bits
;Ki= 2^(-13)
sach e_temp, Ki
;calculate f(i)=f(i-1)+Ki* e(i-1)
;e_temp = Ki*e(i-1)
lac e_temp
add fp
sacl fc
;load the acc with 2^N
zac
addh N
ccalculate e(i)=2^N-T*f(i)
It fc
mpy T
spac
;save the current error
sacl ec_lo
sach ec_hi
;check the see if error is sufficiently small to stop
; if error < 2^12 the loop will stop
bgz no_neg
neg
no_neg: sach e_temp, Edivide
    lac e_temp
    bz stop_loop
;check to see if max loops reached
```

lac ticker
subk 1
sacl ticker
larp 7
banz loop
int0 ser:
intl_ser:
stop_loop:
b stop_loop .end


PMI MODEL \#U12M4 DC PANCAKE MOTOR WITH DUAL SHAFT<br>Peak Torque: 1315.50 z-in<br>Continuous Torque: 121.5 oz -in<br>Peak Torque: 84.5 amps<br>Continuous Current: 8.13 amps<br>Torque Constant: 15.6\% oz-in/amp<br>No Load Speed: 3000 APM<br>Terminal Vohage: 43.4VDC<br>Inertia: 0.019 oz-in/sec/sec<br>Inductance: < 100 mh<br>Dimensions: $5.5^{\prime \prime}$ dia. $\times 2.1^{\prime \prime} \mathrm{L}$., Shaft Front $0.5^{\prime}$ dia. $\times 1-1 / 4^{4} \mathrm{~L}$., Rear $0.5^{3}$ dia. $\times 1-1 / 4^{\mathrm{n}} \mathrm{L}$.<br>Weight: 8 Lbs.<br>USED<br>Stock No. DM-253 . . . . \$175.00

## APPENDLX 6A

## DESCRIPTION OF ASSEMBLY LANGUAGE PROGRAM VARLABLES

Table A6.1 Symbol name cross reference.

| Symbol in Figure 1.1 | Equivalent variable in DSP program |
| :---: | :---: |
| Dmove | bpf_in_nm0_0 |
| Dstat | bpf in_nm0_1 |
| Dpin | bpf_in_nm0_2 |
| Amove | a_squ_out_0 |
| Astat | a_squ_out_1 |
| Apin | a_squ_out_2 |

Table A6.2 Description of assembly program variables.

| Program Variable | Description |
| :--- | :--- |
| amplitude | amplitude of the laser diode modulation signal. |
| bias | DC Bias of the laser diode modulation signal. |
| v_laser | DSP laser diode modulation signal voltage |
| h800 | 800 hex storage location |
| dummy | dummy variable |
| dummyl | dummy variable |

Table A6.2
(Continued)

| Program Variable | Description |
| :---: | :---: |
| flag | dummy loop decision flag |
| ad_in_0 | A/D 0 storage variable |
| ad_in_l | A/D l storage variable |
| ad_in_2 | A/D 2 storage variable |
| bpf_in_nm0_0 | input(i) of the Channel 0 bandpass filter |
| bpf_in_nml_0 | input(i-1) of the Channel 0 bandpass filter |
| bpf_in_nm2_0 | input(i-2) of the Channel 0 bandpass filter |
| bpf_out_nm0_0 | output(i) of the Channel 0 bandpass filter |
| bpf_out_nml_0 | output(i-1) of the Channel 0 bandpass filter |
| bpf_out_nm2_0 | output(i-2) of the Channel 0 bandpass filter |
| bpf_in_nm0_1 | input(i) of the Channel 1 bandpass filter |
| bpf_in_nm1_1 | input(i-1) of the Channel 1 bandpass filter |
| bpf_in_nm2_1 | input(i-2) of the Channel 1 bandpass filter |
| bpf_out_nm0_1 | output(i) of the Channel 1 bandpass filter |
| bpf_out_nml_1 | output(i-1) of the Channel 1 bandpass filter |
| bpf_out_nm2_1 | output(i-2) of the Channel 1 bandpass filter |
| bpf _in_nm0_2 | input(i) of the Channel 2 bandpass filter |
| bpf_in_nml_2 | input(i-1) of the Channel 2 bandpass filter |
| bpf_in_nm2_2 | input(i-2) of the Channel 2 bandpass filter |

Table A6.2
(Continued)

| rogram Variable | Description |
| :--- | :--- |
| bpf_out_nm0_2 | output(i) of the Channel 2 bandpass filter |
| bpf_out_nm1_2 | output(i-1) of the Channel 2 bandpass filter |
| bpf_out_nm2_2 | output(i-2) of the Channel 2 bandpass filter |
| bpf_coeff_in_nm0 | input coefficient(i) of the bandpass filters |
| bpf_coeff_in_nm1 | input coefficient(i-1) of the bandpass filters |
| bpf_coeff_in_nm2 | input coefficient(i-2) of the bandpass filters |
| bpf_coeff_out_nm0 | output coefficient(i) of the bandpass filters |
| bpf_coeff_out_nm1 | output coefficient(i-1) of the bandpass filters |
| bpf_coeff_out_nm2 | output coefficient(i-2) of the bandpass filters |
| s_out_0 | output of the Channel 0 sin multiplier |
| c_out_0 | output of the Channel 0 cos multiplier |
| s_out_l | output of the Channel l sin multiplier |
| c_out_1 | output of the Channel l cos multiplier |
| s_out_2 | output of the Channel 2 sin multiplier |
| c_out_2 | output of the Channel 2 cos multiplier |
| lpfcos_in_nm0_0 | input(i) of the Channel 0 cos lowpass filter |
| lpfcos_in_nm1_0 | input(i-1) of the Channel 0 cos lowpass filter |
| lpfcos_out_nm0_0 | output(i) of the Channel 0 cos lowpass filter |

Table A6.2
(Continued)

| Program Variable | Description |
| :--- | :--- |
| lpfsin_in_nm0_0 | input(i) of the Channel 0 sin lowpass filter |
| lpfsin_in_nml_0 | input(i-1) of the Channel 0 sin lowpass filter |
| lpfsin_out_nm0_0 | output(i) of the Channel 0 sin lowpass filter |
| lpfsin_out_nm1_0 | output(i-1) of the Channel 0 sin lowpass filter |
| lpfcos_in_nm0_1 | input(i) of the Channel 1 cos lowpass filter |
| lpfcos_in_nml_1 | input(i-1) of the Channel 1 cos lowpass filter |
| lpfcos_out_nm0_1 | output(i) of the Channel 1 cos lowpass filter |
| lpfcos_out_nml_l | output(i-1) of the Channel 1 cos lowpass filter |
| lpfsin_in_nm0_1 | input(i) of the Channel 1 sin lowpass filter |
| lpfsin_in_nml_l | input(i-1) of the Channel 1 sin lowpass filter |
| lpfsin_out_nm0_1 | output(i) of the Channel 1 sin lowpass filter |
| lpfsin_out_nml_1 | output(i-1) of the Channel 1 sin lowpass filter |
| lpfcos_in_nm0_2 | input(i) of the Channel 2 cos lowpass filter |
| lpfcos_in_nml_2 | input(i-1) of the Channel 2 cos lowpass filter |
| lpfcos_out_nm0_2 | output(i) of the Channel 2 cos lowpass filter |
| lpfcos_out_nml_2 | output(i-1) of the Channel 2 cos lowpass filter |
| lpfsin_in_nm0_2 | input(i) of the Channel 2 sin lowpass filter |
| lpfsin_in_nml_2 | input(i-1) of the Channel 2 sin lowpass filter |
| lpfsin_out_nm0_2 | output(i) of the Channel 2 sin lowpass filter |

Table A6.2 (Continued)

| Program Variable | Description |
| :---: | :---: |
| Ipfsin_out_nml_2 | output(i-1) of the Channel $2 \sin$ lowpass filter |
| lpf_coeff_in_nm0 | input coefficient(i) of the lowpass filters |
| lpf_coeff_in_nml | input coefficient( $\mathrm{i}-1$ ) of the lowpass filters |
| lpf_coeff_out_nm0 | output coefficient(i) of the lowpass filters |
| lpf_coeff_out_nml | output coefficient(i-1) of the lowpass filters |
| ellcos_in_nm0_0 | input(i) of the Channel 0 cos bandpass filter |
| ellcos_in_nm1_0 | input(i-1) of the Channel 0 cos bandpass filter |
| ellcos_in_nm2_0 | input(i-2) of the Channel 0 cos bandpass filter |
| ellcos_out_nm0_0 | output(i) of the Channel 0 cos bandpass filter |
| ellcos_out_nml_0 | output( $\mathrm{i}-1$ ) of the Channel 0 cos bandpass filter |
| ellcos_out_nm2_0 | output(i-2) of the Channel 0 cos bandpass filter |
| ellcos_in_nm0_1 | input(i) of the Channel 1 cos bandpass filter |
| ellcos_in_nml_l | input(i-1) of the Channel 1 cos bandpass filter |
| ellcos_in_nm2_1 | input(i-2) of the Channel 1 cos bandpass filter |
| ellcos_out_nm0_1 | output(i) of the Channel 1 cos bandpass filter |
| ellcos_out_nml_l | output( $\mathrm{i}-1$ ) of the Channel 1 cos bandpass filter |
| ellcos_out_nm2_1 | output(i-2) of the Channel 1 cos bandpass filter |
| ellcos_in_nm0_2 | input(i) of the Channel 2 cos bandpass filter |
| ellcos_in_nml_2 | input(i-1) of the Channel 2 cos bandpass filter |

Table A6.2
(Continued)

| Program Variable | Description |
| :---: | :---: |
| ellcos_in_nm2_2 | input(i-2) of the Channel 2 cos bandpass filter |
| ellcos_out_nm0_2 | output(i) of the Channel 2 cos bandpass filter |
| ellcos_out_nml_2 | output(i-1) of the Channel 2 cos bandpass filter |
| ellcos_out_nm2_2 | output(i-2) of the Channel 2 cos bandpass filter |
| ellsin_in_nm0_0 | input(i) of the Channel 0 sin bandpass filter |
| ellsin_in_nml_0 | input( $\mathrm{i}-1)$ of the Channel $0 \sin$ bandpass filter |
| ellsin_in_nm2_0 | input(i-2) of the Channel $0 \sin$ bandpass filter |
| ellsin_out_nm0_0 | output(i) of the Channel 0 sin bandpass filter |
| ellsin_out_nml_0 | output( $\mathrm{i}-1)$ of the Channel 0 sin bandpass filter |
| ellsin_out_nm2_0 | output( $\mathrm{i}-2)$ of the Channel 0 sin bandpass filter |
| ellsin_in_nm0_1 | input(i) of the Channel 1 sin bandpass filter |
| ellsin_in_nml_1 | input(i-1) of the Channel I sin bandpass filter |
| ellsin_in_nm2_1 | input(i-2) of the Channel 1 sin bandpass filter |
| ellsin_out_nm0_1 | output(i) of the Channel 1 sin bandpass filter |
| ellsin_out_nml_l | output(i-1) of the Channel 1 sin bandpass filter |
| ellsin_out_nm2_1 | output( $\mathrm{i}-2)$ of the Channel I $\sin$ bandpass filter |
| ellsin_in_nmo_2 | input(i) of the Channel 2 sin bandpass filter |
| ellsin_in_nm1_2 | input( $\mathrm{i}-1)$ of the Channel 2 sin bandpass filter |
| ellsin_in_nm2_2 | input(i-2) of the Channel 2 sin bandpass filter |

Table A 6.2
(Continued)

| Program Variable | Description |
| :--- | :--- |
| ellsin_out_nm0_2 | output(i) of the Channel 2 sin bandpass filter |
| ellsin_out_nml_2 | output(i-1) of the Channel 2 sin bandpass filter |
| ellsin_out_nm2_2 | output(i-2) of the Channel 2 sin bandpass filter |
| ell_coeff_in_nm0 | input coefficient(i) of the elliptic filters |
| ell_coeff_in_nml | input coefficient(i-1) of the elliptic filters |
| ell_coeff_in_nm2 | input coefficient(i-2) of the elliptic filters |
| ell_coeff_out_nm0 | output coefficient(i) of the elliptic filters |
| ell_coeff_out_nm1 | output coefficient(i-1) of the elliptic filters |
| ell_coeff_out_nm2 | output coefficient(i-2) of the elliptic filters |
| a_squ_hi_0 | hi l6 bits of the output of the Channel 0 squarer |
| a_squ_lo_o | lo 16 bits of the output of the Channel 0 squarer |
| a_squ_out_0 | output of the Channel 0 squarer ported to D/A |
| a_squ_hi_1 | hi 16 bits of the output of the Channel I squarer |
| a_squ_lo_1 | lo 16 bits of the output of the Channel 1 squarer |
| a_squ_out_1 | output of the Channel I squarer ported to D/A |
| a_squ_hi_2 | hi 16 bits of the output of the Channel 2 squarer |
| a_squ_lo_2 | output of the Channel 2 squarer ported to D/A |
| aux_laser_pointer | tem reg storage for the laser diode voltage pointer |

Table A 6.2
(Continued)

| Program Variable | Description |
| :---: | :---: |
| aux_laser_reset | temp aux reg storage for the laser diode reset pointer |
| aux_cos_pointer | temp aux reg storage for the cos pointer |
| aux_cos_reset | temp aux reg storage for the reset cos pointer |
| aux_sin_pointer | temp aux reg storage for the sin pointer |
| aux_sin_reset | temp aux reg storage for the reset sin pointer |
| out_lo_old_0 | hi 16 bits of the output average of the Channel 0 squarer |
| out_hi_old_0 | lo 16 bits of the output average of the Channel 0 squarer |
| out_avg_0 | output avg of the Channel 0 squarer ported to host PC |
| avged_out_0 | Output avg of the Channel 0 squarer ported to host PC |
| out_lo_old_1 | hi 16 bits of the output average of the Channel 1 squarer |
| out_hi_old_1 | lo 16 bits of the output average of the Channel 1 squarer |
| out_avg_1 | output avg of the Channel I squarer ported to host PC |
| avged_out_ | output avg of the Channel 1 squarer ported to host PC |
| out_lo_old_2 | hi 16 bits of the output average of the Channel 2 squarer |
| out_hi_old_2 | lo 16 bits of the output average of the Channel 2 squarer |
| out_avg_2 | output avg of the Channel 2 squarer ported to host PC |
| avged_out_2 | output avg of the Channel 2 squarer ported to host PC |
| avg_out_lo_old_0 | previous lo 16 bits of the output avg of Channel 0 |
| avg_out_hi_old_0 | previous hi 16 bits of the output avg of Channel 0 |

Table A6.2
(Continued)

| Program Variable | Description |
| :--- | :--- |
| avged_out_avg_0 | output avg of the Channel 0 squarer ported to host PC |
| avged_avged_out_0 | output avg of the Channel 0 squarer ported to host PC |
| avg_out_lo_old_1 | previous lo 16 bits of the output avg of Channel 1 |
| avg_out_hi_old_1 | previous hi 16 bits of the output avg of Channel 1 |
| avged_out_avg_1 | output avg of the Channel 1 squarer ported to host PC |
| avged_avged_out_1 | output avg of the Channel 1 squarer ported to host PC |
| avg_out_lo_old_2 | previous lo 16 bits of the output avg of Channel 2 |
| avg_out_hi_old_2 | previous hi 16 bits of the output avg of Channel 2 |
| avged_out_avg_2 | output avg of the Channel 2 squarer ported to host PC |
| avged_avged_out_2 | output avg of the Channel 2 squarer ported to host PC |
| command | flexible beam command signal from C-program |

## APPENDIX 6B

## DSP ASSEMBLY LANGUAGE PROGRAM

;assembly program
;assembly.asm
;Set up labels
f_ext_timer $\quad$ set $-2571 \quad ; 40 \mathrm{k} \mathrm{Hz}$

| ;Laser Voltage | *** | DO NOT CHANGE THESE VALUES!!!!!! ${ }^{* * *}$ |
| :--- | :--- | :--- |
| amp | set 80 |  |
| bias | set | 3125 |
| number_average | set | 99 |

;Sin wave logistics variables

| sin_start | set | 900 h |
| :--- | :--- | :--- |
| reset_ticker | .set | 19 |
| display_tick_start | .set | 117 |

;Gains \& Divisors

| hpfgain | .set | 1 | $;$ gain $=2^{\wedge} \mathrm{x}$ |
| :--- | :--- | :--- | :--- |
| hpfdiv | .set | 2 | $; 4=$ divide by 4096 for unscaling |
| shift | .set | 4 | $; 4=$ divide by 4096 for unscaling |
| Ipfgain | .set | 0 | $;$ gain $=2^{\wedge} \mathrm{X}$ |
| lpfdiv | .set | 2 | $; 4=$ divide by 4096 for unscaling |
| ellgain | .set | 0 | $;$ gain $=2^{\wedge} \mathrm{X}$ |
| elldiv | .set | 2 | $; 4=$ divide by 4096 for unscaling |
| outdiv | .set | 5 |  |

;Interrupt Enable Variable
int_on .set 2 ;enable interrupt(s)
;AD \& D/A Mux Controls

| out_0 | set | 0 | ;latch value for $D / A$ chan zero |
| :--- | :--- | :--- | :--- |
| out_1 | set | 8 | ;latch wavue for $D / A$ chan one |
| in_0 | set | 0 |  |
| in_1 | set | 1 |  |
| in_2 | set | 2 |  |


| ;Port map |  |  |  |
| :--- | :--- | :--- | :--- |
| ext_timer | set | 0 | ;timer $=$ port 0 |
| latch | set | 1 | ;latch $=$ port 1 |
| input | set | 2 | ;input a/d $=$ port 2 |
| timer_loc | set | 2 | ;timer location |
| output | set | 3 | ;output $d / a=$ port 3 |
| timer_per | set | 3 | ;timer starting value |
| int_mask | set | 4 | ;interrupt mask = port 4 |

;Define Variables and memory locations
;Page 4
;Laser variables and stuff

| amplitude | .set | 0 |
| :--- | :--- | :--- |
| bias | set | 1 |
| v_laser | set | 2 |
| h800 | set | 3 |
| dummy | set | 4 |
| dummy 1 | set | 5 |
| flag | set | 6 |

;A/D Input variables
ad_in_0 0 set 7
ad_in_1 set 8
ad_in_2 .set 9

| ;HPF variables |  |  |
| :--- | :--- | :--- |
| bpf_in_nm0_0 | .set | 10 |
| bpf_in_nml_0 | .set | 11 |
| bpf_in_nm2_0 | .set | 12 |
| bpf_out_nm0_0 | .set | 13 |
| bpf_out_nm1_0 | .set | 14 |
| bpf_out_nm2_0 | .set | 15 |

bpf_in_nm0_1 set 16
bpf_in_nml_1 .set 17
bpf_in_nm2_1 $\quad$ set 18
bpf_out_nm0_1 . set 19
bpf_out_nml_1 .set 20
bpf_out_nm2_1 set 21
bpf_in_nm0_2 set 22
bpf_in_nml_2 set 23
bpf_in_nm2_2 24
bpf_out_nm0_2 .set 25

| bpf_out_nml 2 | set | 26 |
| :---: | :---: | :---: |
| bpf_out_nm2_2 | set | 27 |
| bpf_coeff_in_nm0 | set | 28 |
| bpf_coeff_in_nml | set | 29 |
| bpf_coeff_in_nm2 | . Set | 30 |
| bpf_coeff_out_nm0 | . Set | 31 |
| bpf_coeff_out_nml | .set | 32 |
| bpf_coeff_out_nm2 | .set | 33 |
| ; Sin \& Cos Multipli |  |  |
| s_out 0 | set | 34 |
| c_out_0 | set | 35 |
| s_out_l | set | 36 |
| c_out_1 | . set | 37 |
| s_out_2 | . set | 38 |
| c_out_2 | set | 39 |
| ;LPF variables |  |  |
| lpfcos_in_nmo_0 | .set | 40 |
| Ipfcos_in_nml_0 | . set | 41 |
| Ipfcos_out nm0 0 | . set | 42 |
| lpfcos_out_nml__0 | . set | 43 |
| Ipfsin_in_nmo o | .set | 44 |
| lpfsin_in_nml_0 | . set | 45 |
| lpfsin_out_nmo_0 | set | 46 |
| 1pfsin_out_nml_0 | set | 47 |
| lpfcos_in_nm0_l | .set | 48 |
| lpfcos_in_nml_1 | .set | 49 |
| Ipfcos_out_nm0_1 | set | 50 |
| lpfcos_out_nml_1 | set | 51 |
| lpfsin_in_nm0_1 | set | 52 |
| lpfsin_in_nml_1 | .set | 53 |
| lpfsin_out_nmo_1 | set | 54 |
| lpfsin_out_nml_1 | .set | 55 |
| lpfoos_in_nm0_2 | .set | 56 |
| lpfcos _in_nml_2 | .set | 57 |
| Ipfoos_out_nm0 2 | . set | 58 |
| lpfcos_out_nml_2 | . set | 59 |
| lpfsin_in_nm0_2 | .set | 60 |
| lpfsin_in_nml_2 | . set | 61 |
| lpfsin_out_nm0_2 | . set | 62 |
| lpfsin_out_nml_2 | . set | 63 |


| lpf_coeff_in_nm0 | .set | 64 |
| :--- | :--- | :--- |
| lpf_coeff_in_nml | .set | 65 |
| lpf_coeff_out_nm0 | .set | 66 |
| lpf_coeff_out_nml | .set | 67 |

## ;ELL variables

```
ellcos_in_nm0_0 .set 68
```

ellcos_in_nml_0 .set 69
ellcos_in_nm2_0 set 70
ellcos_out_nm0_0 set 71
ellcos_out_nml_0 set 72
ellcos_out_nm2_0 set 73
ellcos_in_nm0_1 set 74
ellcos_in_nml_l set 75
ellcos_in_nm2_1 set 76
ellcos_out_nm0_1 . set 77
ellcos_out_nml_1 .set 78
ellcos_out nm2_1 .set 79
ellcos_in_nm0_2 set 80
ellcos_in_nml_2 .set 81
ellcos_in_nm2_2 . set 82
ellcos_out_nm0_2 .set 83
ellcos out nml 2 .set 84
ellcos_out_nm2_2 . set 85
ellsin_in_nm0_0 .set 86
ellsin_in_nml_0 .set 87
ellsin_in_nm2_0 .set 88
ellsin_out_nm0_0 .set 89
ellsin_out_nml_0 .set 90
ellsin_out_nm2 0 .set 91
ellsin_in_nm0_1 .set 92
ellsin_in_nml_1 set 93
ellsin_in_nm2_1 set 94
ellsin_out_nm0_1 .set 95
ellsin_out_nml_1 .set 96
ellsin_out_nm2_1 set 97
ellsin in_nm0_2 set 98
ellsin_in_nml_2 set 99
ellsin_in_nm2_2 .set 100

| ellsin_out_nm0_2 | .set | 101 |
| :--- | :--- | :--- |
| ellsin_out_nm1_2 | .set | 102 |
| ellsin_out_nm2_2 | .set | 103 |
|  |  |  |
| ell_coeff_in_nm0 | .set | 104 |
| ell_coeff_in_nm1 | .set | 105 |
| ell_coeff_in_nm2 | .set | 106 |
| ell_coeff_out_nm0 | set | 107 |
| ell_coeff_out_nm1 | .set | 108 |
| ell_coeff_out_nm2 | .set | 109 |

;Square \& Summer variables

| a_squ_hi_0 | set | 110 |
| :--- | :--- | :--- |
| a_squ_lo_0 | set | 111 |
| a_squ_out_0 | set | 112 |
| a_squ_hi_1 | .set | 113 |
| a_squ_lo_1 | set | 114 |
| a_squ_out_1 | set | 115 |
| a_squ_hi_2 | set | 117 |
| a_squ_lo_2 | .set | 118 |
| a_squ_out_2 | set | 119 |

;Auvilary Register [Pointers \& Resets] Load \& Save Locations
aux_laser_pointer .set 120
aux_laser_reset .set 121
aux_cos_pointer .set 122
aux_cos_reset .set 123
aux_sin_pointer .set 124
aux_sin_reset .set 125
; Page 5
;Average variables
out_lo_old_0 .set 0
out_hi_old_0 . set 1
out_avg_0 set 2
avged_out_0 .set 3
out_lo_old_1 .set 4
out_hi_old_1 .set 5
out_avg_1 set 6
avged_out_ set 7

```
out_lo_old_2 .set 8
out_hi_old_2 set 9
out_avg_2 set 10
avged_out_2 .set 11
avg_out_lo_old_0 .set 12
avg_out_hi_old_0 set 13
avged_out_avg_0 .set 14
avged_avged_out_0 .set 15
avg_out_lo_old_1 .set 16
avg_out_hi_old_1 .set 17
avged_out_avg_1 .set 18
avged_avged_out_1 .set 19
avg_out_lo_old_2 .set 20
avg_out_hi_old_2 .set 21
avged_out_avg_2 .set 22
avged_avged_out_2 .set 23
command .set 24
```

;interrupt flow table
;the "vectors" section is a label to link with the link.cmd
; file to load the interrupt flow table at address Oh
.sect "vectors"
b start ;external reset interrupt
b int0_ser ;external interrupt zero
bintl_ser ;external interrupt one
;the "text" section is a label to link with the link.cmd
; file to load the main program at address 50 h
.text
start: ;start of program
;sets the interrupt mask - turn on INT 1
dint
ldpk 0
lack int_on
sacl int_mask
;set up external timer for adc operation ;this interrupt sets the "resampling" rate
ldpk 4
lalk f_ext_timer
sacl dummy
out dummy, ext_timer
;set up latch to choose $\mathrm{A} / \mathrm{D}=0 \mathrm{D} / \mathrm{A}=0$
lalk
sacl dummy
out dummy, latch
;clear the memory locations
call clear
;call the set up $\sin \& \cos$ look-up routine
call setup_sin
;set up sin \& cos auxilary registers and tickers
call sin_logistics
;load the HPF coefficients
call hpf_coeff
;load the LPF coefficients
call lpf_coeff
;load the ELL coefficients
call ell_coeff
;load laser power values
call laser_setup
;load 800 hex for D/A outputs
lalk 2048

```
    sacl h800
    ;setup "flag" to choose which channe! to nin
    ;If Acc ==0 --- will execute channel zero
    ;f Acc = 1 --- will execute channel one
    ;If Acc ==2 --- will execute channel two
    ;If Acc == 3 --- will execute channel three
    lark 0, 1000h
    lark 7, number_average
    zac
    sacl flag
    larp 1
    eint
main_loop:;ifflag == 0 than go into demod loop
    ;if flag ==1 than don't go into demod loop
    idle
    b main_loop
;********************
intO ser:
    eint
    ret
```


; The demodulation loops and the laser modulation calculations are ; all performed in interrupt number one server.
intl_ser:
;Load auxilary pointer to 1 for pointers and tickers
larp 1
;Decide which channel to run
ldpk 4
lac flag
bz channel_0
subk 1
bz channel_1
lac flag
subk 2
bz channel_2
b channel_3
; Program Channel 0
channel_0:
;set up latch to choose $\mathrm{A} / \mathrm{D}=1 \mathrm{D} / \mathrm{A}=0$
lalk 10
sacl dummy
out dummy, latch
Read in data from A/D
in ad_in 0 , input
lac ad_in_0
sub h800
sacl ad_in_0

[^1];execute the difference equation
It bpf_out_nm2_0
mpy bpf_coeff_out_nm2
Itd bpf_out_nml_0
mpy bpf_coeff_out_nml
apac
It bpf_in_nm2_0
mpy bpf_coeff_in_nm2
ltd bpf_in_nml_0
mpy bpf_coeff_in_nml
ltd bpf_in_nm0_0
mpy bpf_coeff_in_nm0
apac
bgz no_add_hpf_0
addk 4095
no_add_hpf_0: sach bpf_out_nm0_0, hpfdiv
dmov bpf_out_nm0_0
mult_cos_0:
;figure out $\cos$
lar 1, aux_cos_pointer
It bpf_out_nm0_0
mpy *
pac
sach c_out_0, shift
mult_sin_0:
;figure out sin
;the sub_in_val is from the cos signal

```
        lar 1,aux_sin_pointer
        It bpf_out_nm0_0
        mpy *
        pac
        sach s_out_0, shift
;*******************
lpf_cos_0:
    ;load the "input"
    ; add gain to the
    lac c_out_0, lpfgain
    sacl lpfcos_in_nm0_0
    zac
        ;execute the difference equation
    It lpfcos_out_nml 0
    mpy lpf_coeff_out_nml
    apac
    It lpfcos_in_nml_0
    mpy lpf_coeff_in_nml
    Itd lpfcos_in_nm0_0
    mpy lpf_coeff_in_nm0
    apac
    bgz noadd_lpfcos_0
    addk 4095
noadd_lpfcos_0: sach lpfcos_out_nm0_0, lpfdiv
    dmov lpfcos_out_nm0_0
lpf_sin_0:
    ;ioad the "input"
    ; add gain to the
```

```
    lac s_out_0, lpfgain
    sacl lpfsin_in_nm0_0
    zac
    ;execute the difference equation
    lt lpfsin_out_nml_0
    mpy lpf_coeff_out_nml
    apac
    1t lpfsin_in_nml_0
    mpylpf_coeff_in_nml
    ltd lpfsin_in_nm0_0
    mpy lpf_coeff_in_nm0
    apac
    bgz noadd_lpfsin_0
    addk 4095
noadd_lpfsin_0: sach lpfsin_out_nm0_0, lpfdiv
    dmov Ipfsin_out_nm0_0
ell_cos_0:;(125e0 + 4071el + 4045fl)/4096
    ;load the "input"
    ; add gain to the
    lac lpfcos_out_nm0_0, ellgain
    sacl ellcos_in_nm0_0
    zac
    ;execute the difference equation
    It ellcos_out_nm2_0
    mpy ell_coeff_out_nm2
    Itd ellcos_out_nml_0
    mpy ell_coeff_out_nml
```

```
    apac
    It ellcos_in_nm2_0
    mpy ell_coeff_in_nm2
    ltd ellcos_in_nml_0
    mpy ell_coeff_in_nml
    ltd ellcos_in_nm0_0
    mpy ell_coeff_in_nm0
    apac
    bgz noadd_ellcos_0
    addk 4095
noadd_ellcos_0: sach ellcos_out_nm0_0, elldiv
    dmov ellcos_out_nm0_0
ell_sin_0: ;(125g0 + 407lgl + 4045hl)/4096
    ;load the "input"
    ; add gain to the
    lac lpfsin_out_nm0_0, ellgain
    sacl ellsin_in_nm0_0
    zac
    ;execute the difference equation
    It ellsin_out_nm2_0
    mpy ell_coeff_out_nm2
    ltd ellsin_out_nml_0
    mpy ell_coeff_our_nml
    apac
    It ellsin_in_nm2_0
    mpy ell_coeff_in_nm2
    Itd ellsin_in_nml_0
```

```
    mpy ell_coeff_in_nml
    Itd ellsin_in_nm0_0
    mpy ell_coeff_in_nm0
    apac
    bgz noadd_ellsin_0
    addk 4095
noadd_ellsin_0: sach ellsin_out_nm0_0, elldiv
    dmov ellsin_out_nm0_0
summer_0:
```

;NOTE -- when outputting the answer to the DAC the value ; must be less that $2^{\wedge} 12$ because of the limit of the DAC

$$
; \text { HOWEVER }- \text { the internal answer can remain at } 2^{\wedge} 16 \text { resolution }
$$

sqra ellcos_out_nm0_0
zac
sqra ellsin_out_nm0_0
apac
sach a_squ_hi_0
sacl a_squ_lo_0
sach a_squ_out_0, outdiv
lac a_squ_out_0
; out a_squ_out_0, output
skip_0:
;Set flag to implement next channel
lack 1
sacl flag
eint
ret

## ;********************

## ;Program Channel 1

## channel_1:

;set up latch to choose $\mathrm{A} / \mathrm{D}=2 \mathrm{D} / \mathrm{A}=0$
lalk 11
sacl dummy
out dummy, latch
;Read in data from the $A / D$
in ad_in_1, input
lac ad_in_1
sub h800
sacl ad_in_1
;Band Pass Filter
bpf_1: ;load the "input"
; add gain to the
lac ad_in_1, hpfgain
sacl bpf_in_nm0_1
zac
;execute the difference equation
It bpf_out_nm2_1
mpy bpf_coeff_out_nm2
lid bpf_out_nmi_1
mpy bpf_coeff_out_nml
apac
It bpf_in_nm2_1
mpy bpf_coeff_in_nm2

```
    ltd bpf_in_nml_l
    mpy bpf_coeff_in_nml
    Itd bpf_in_nm0_1
    mpy bpf_coeff_in_nm0
    apac
    bgz no_add_hpf_1
    addk 4095
no_add_hpf_1: sach bpf_out_nm0_1, hpfdiv
    dmovbpf_out_nm0_1
;*******************
mult_cos_1:
    ;figure out cos
    lar 1, aux cos_pointer
    It bpf_out_nm0_1
    mpy *
    pac
    sach c_out_1, shift
********************
mult_sin_1:
    ;figure out sin
    ;the sub_in_val is from the cos signal
    lar 1, aux_sin_pointer
    lt bpf_out_nm0_1
    mpy *
    pac
    sach s_out_1, shift
lpf_cos_1
    ;load the "input"
    ; add gain to the
```

```
    lac c_out_1, Ipfgain
    sacl Ipfcos_in_nm0_1
    zac
    ;execute the difference equation
    It lpfcos_out_nml l
    mpy lpf_coeff_out_nml
    apac
    It lpfcos_in_nmI_l
    mpylpf_coeff_in_nml
    ltd lpfcos_in_nm0_l
    mpy lpf_coeff_in_nm0
    apac
    bgz noadd_lpfcos_l
    addk 4095
noadd_lpfcos_1: sach Ipfcos_out_nm0_1, lpfdiv
    dmov lpfcos_out_nmO_1
lpf_sin_1:
    ;load the "input"
        ; add gain to the
    lac s_out_1, lpfgain
    sacl lpfsin_in_nmO_l
    zac
    ;execute the difference equation
    It lpfsin_out_nml_l
    mpylpf_coeff_out_nml
    apac
    It lpfsin_in_nml_l
```

```
mpy lpf_coeff_in_nml
ltd Ipfsin_in_nmo_1
mpy lpf_coeff_in_nm0
apac
noadd_lpfsin_1
addk 4095 ;add to acc short immediate
noadd_lpfsin_1: sach 1 pfsin_out_nm0_1, 1pfdiv
dmov lpfsin_out_nmO_1
, \({ }^{* * * * * * * * * * * * * * * * * * * ~}\)
ell_cos_1: ; \((125 \mathrm{e} 0+407 \mathrm{lel}+4045 \mathrm{fl}) / 4096\)
;load the "input"
; add gain to the
lac Ipfcos_out_nm0_1, ellgain
sacl ellcos in_nm0_1
zac
;execute the difference equation
It ellcos_out_nm2_1
mpy ell_coeff_out_nm2
ltd ellcos_out_nml_1
mpy ell_coeff_out_nml
apac
It ellcos_in_nm2_1
mpy ell_coeff_in_nm2
ltd ellcos_in_nml_1
mpy ell_coeff_in_nml
Itd ellcos_in_nm0_]
mpy ell_coeff_in_
apac
bgz noadd_ellcos_1
```

addk 4095
noadd_ellcos_1 : sach ellcos_out_nm0_1, elldiv
dmovellcos_out_nm0_1

ell_sin_1: ; $(125 \mathrm{~g} 0+407 \mathrm{lg} 1+4045 \mathrm{hl}) / 4096$
;load the "input"
; add gain to the
lac lpfsin_out_nm0_1, ellgain
sacl ellsin_in_nm0_1
zac
;execute the difference equation
It ellsin_out_nm2_1
mpy ell_coeff_out_nm2
Itd ellsin_out_nml_1
mpy ell_coeff_out_nml
apac
It ellsin_in_nm2_1
mpy ell_coeff_in_nm2
ltd ellsin_in_nml_1
mpy ell_coeff_in_nml
Itd ellsin_in_nm0_1
mpy ell_coeff_in_nm0
apac
bgz noadd_ellsin_1
addk
noadd_ellsin_1: sach ellsin_out_nm0_1, elldiv
dmovellsin_out_nm0_1
summer_1:
;NOTE -- when outputting the answer to the DAC the value ; must be less that $2^{\wedge} 12$ because of the limit of the DAC ;HOWEVER -- the internal answer can remain at $2^{\wedge} 16$ resolution
sqra ellcos_out_nm0_1
zac
sqra ellsin_out_nm0_1
apac
sach a_squ_hi_1
sacl a_squ_lo_1
sach a_squ_out_1, outdiv
skip_1:

Set flag to implement next channel
lack 2
sacl flag
eint
ret

## ;Program Channel 2

channel_2:
; set up latch to choose $\mathrm{A} / \mathrm{D}=0 \mathrm{D} / \mathrm{A}=1$
lalk 8
sacl dummy
out dummy, latch
;Read in data from the $A / D$
in ad_in_2, input
lac ad_in_2
sub h800
sacl ad_in_2

```
*******************
;Band Pass Filter
bpf_2: ;load the "input"
    ; add gain to the
        lac ad_in_2, hpfgain
        sacl bpf_in_nm0_2
    zac
        ;execute the difference equation
        lt bpf_out_nm2_2
        mpy bpf_coeff_out_nm2
        ltd bpf_out_nml_2
        mpy bpf_coeff_out_nml
        apac
        It bpf_in_nm2_2
        mpy bpf_coeff_in_nm2
        ltd bpf_in_nml_2
        mpy bpf_coeff_in_nml
        ltd bpf_in_nm0_2
        mpy bpf_coeff_in_nm0
        apac
    bgz no_add_hpf_2
    addk 4095
no_add_hpf_2: sach bpf_out_nm0_2, hpfdiv
    dmov bpf_out_nm0_2
;*******************
mult_cos_2:
```

lar 1, aux_cos_pointer
It bpf_out_nm0_2
mpy*+
sar l, aux_cos_pointer
pac
sach c_out_2, shift
;check weather to reset the cos_ref pointer
lar 1, aux_cos_reset
banz skip_c
;Reset the Pointers
lalk sin_start
sacl aux_cos_pointer
lalk reset_ticker
sacl aux_cos_reset
b dont_reset_cos_again

```
skip_c: sar l, aux_cos_reset
dont_reset_cos_again
********************
mult_sin_2:
;figure out \(\sin\)
;the sub_in_val is from the cos signal
lar 1, aux_sin_pointer
It bpf_out_nmO_2
mpy *+
sar 1, aux_sin_pointer
pac
sach s_out_2, shift
;check weather to reset the sin_ref pointer
lar 1, aux_sin_reset
banz skip_s
```

;Reset the pointers
lalk sin_start

```
sacl aux_sin_pointer
lalk reset ticker
sacl aux_sin_reset
b dont_reset_sin_again
skip_s: sar l, aux_sin_reset dont_reset_sin_again:
lpf_cos_2:
;load the "input"
; add gain to the
lac c_out_2, lpfgain
sacl lpfcos_in_nm0_2
zac
;execute the difference equation
It lpfcos_out_nm1_2
mpy lpf_coeff_out_nml
apac
It lpfcos_in_nml_2
mpy lpf_coeff_in_nml
Itd lpfcos_in_nm0_2
mpy lpf_coeff_in_nm0
apac
bgz noadd_lpfcos_2
addk 4095
noadd_lpfcos_2: sach lpfcos_out_nm0_2, Ipfdiv
dmov lpfcos_out_nm0_2
lpf_sin_2:
```

;load the "input"

```
    ; add gain to the
    lac s_out_2, lpfgain
    sacl lpfsin_in_nm0_2
    zac
    ;execute the difference equation
    It lpfsin_out_nml_2
    mpy lpf_coeff_out_nml
    apac
    It lpfsin_in_nml_2
    mpylpf_coeff_in_nml
    ltd Ipfsin_in_nmO_2
    mpylpf_coeff_in_nm0
    apac
    bgz noadd_lpfsin_2
    addk 4095
noadd_lpfsin_2: sach lpfsin_out_nm0_2, lpfdiv
    dmov lpfsin_out_nmO_2
    lpfsin_out_nm0_2
ell_cos_2:;(125e0+407lel + 4045fl)/4096
;oad the "input"
; add gain to the
lac lpfcos_out_nm0_2, ellgain
sacl ellcos_in_nm0_2
zac
;execute the difference equation
It ellcos_out_nm2_2
mpy ell_coeff_out_nm2
```

```
Itd ellcos_out_nml 2
mpy ell_coeff_out_nml
apac
It ellcos_in_nm2_2
mpy ell_coeff_in_nm2
ltd ellcos_in_nml_2
mpy ell_coeff_in_nml
ltd ellcos_in_nm0_2
mpy ell_coeff_in_nm0
apac
bgz noadd_ellcos 2
addk 4095
noadd_ellcos_2: sach ellcos_out_nm0_2, elldiv
    dmov ellcos_out_nm0_2
ell_sin_2:;(125g0 + 407lgl + 4045hl)/4096
;load the "input"
; add gain to the
lac lpfsin_out_nm0_2, ellgain sacl ellsin_in_nm0_2
zac
execute the difference equation
It ellsin_out_nm2_2
mpy ell_coeff_out_nm2
Itd ellsin out nml 2
mpy ell_coeff_out_nml
apac
It ellsin_in_nm2_2
mpy ell_coeff_in_nm2
```

```
    Itd ellsin_in_nm1_2
    mpy ell_coeff_in_nml
    Itd ellsin_in_nm0_2
    mpy ell_coeff_in_nm0
    apac
    bgz noadd_ellsin_2
    addk 4095
noadd_ellsin_2: sach ellsin_out_nm0_2, elldiv
    dmov ellsin_out_nm0_2
```

summer_2:
;NOTE -- when outputting the answer to the DAC the value ; must be less that $2^{\wedge} 12$ because of the limit of the DAC ;HOWEVER -- the internal answer can remain at $2 \wedge 16$ resolution
sqra ellcos_out_nm0_2
zac
sqra ellsin_out_nm0_2
apac
sach a_squ_hi_2
sacl a_squ_lo_2
sach a_squ_out_2, outdiv
skip_2:
;Set flag to implement next channel
lack 3
sacl flag
eint
ret

## ;Program Channel 3

```
channel_3:
    ;set up latch to choose A/D =1 D/A =1
    lalk }
    sacl dummy
    out dummy, latch
    ;Read in data from the A/D
    n dummy, input
laser: ;routine to power the laser
    ;calculate the voltage on the laser
    lar 1, aux laser_pointer
    lt amplitude
    mpy *+
    sar l, aux_laser pointer
    pac
    sach v_laser, shift
    lac v_laser
    add bias
    sacl v_laser
    out v_laser, output
    ;check weather to reset the sin_ref pointer
    lar 1, aux_laser_reset
    banz skip_laser_reset
    ;Reset the pointers
    lalk sin_start
    sacl aux_laser_pointer
    lalk reset_ticker
    sacl aux_laser_reset
    b dont_reset_laser_again
skip_laser_reset:
```

```
    sar 1, aux_laser_reset
dont_reset_laser_again:
    ;$$$$$$$$$$$$$$$$$$$$$
    ;$$$$$$$$$$$8$$$$$$$$$
    ;$$$$$$$$$$$$$$$$$$$$$
;set up latch to choose A/D =0 D/A =0
lalk l
sacl dummy
out dummy, latch
```

lalk 2048
sacl dummy
ldpk 8
out 120 , output ;should be 120
;\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$8\$\$\$
;\$\$\$\$\$\$\$\$\$\$8\$\$\$\$\$\$\$\$\$\$
;\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$

;AVERAGE OUTPUTS
;Average Channel Zero
ldpk 5
zalh out_hi_old_0
adds out_lo_old_0
ldpk 4
adds a_squ_10_0
addh a_squ_hi_0
ldpk 5
sacl out_lo_old_0
sach out_hi_old_0
;Average Channel One
zalh out_hi_old_1 adds out_lo_old_1
ldpk 4
adds a_squ_lo_1
addh a squ_hi_l
ldpk 5
sacl out_lo_old_1
sach out_hi_old_1
;Average Channel Two
zalh out_hi_old_2
adds out_lo_old_2
ldpk 4
adds a_squ_lo_2
addh a_squ_hi_2
ldpk 5
sacl out_lo_old_2
sach out_hi_old_2
larp 7
banz no_divide
divide:
zalh out_hi_old_0
adds out_lo_old_0
sach avged out_0
zalh out_hi_old_1
adds out_lo_old_1
sach avged_out_1
zalh out_hi_old_2
adds out_lo_old_2
sach avged_out_2
lac avged_out_0
ldpk 8
sacl 16
lalk 1
sacl 20
ldpk 5

```
lac avged_out_1
ldpk }
sacl }5
lalk I
sacl }6
ldpk }
lac avged_out_2
ldpk }
sacl }9
lalk 1
sacl 100
idpk 5
zac
sacl out_hi_old_0
sacl out_lo_old_0
sacl out_hi_old_1
sacl out_lo_old_1
sacl out_hi_old_2
sacl out_lo_old_2
lark 7, number_average
```


## ;AVERAGE OUTPUTS -- AGAIN!!!

```
;Average Channel Zero AGAIN
zalh avg_out_hi_old_0 adds avg_out_lo_old_0 adds avged_out_0
sacl avg_out_lo_old_0
sach avg_out_hi_old_0
;Average Channel One AGAIN
zalh avg_out_hi_old_1
adds avg_out_lo_old_1 adds avged_out_1
```

```
avg_out_lo_old_1
sach avg_out_hi_old_l
;Average Channel Two AGAIN
zalh avg_out_hi_old_2
adds avg_out_lo_old_2
adds avged_out_2
sacl avg_out_lo_old_2
sach avg_out_hi_old_2
larp 6
banz no_divide
zalh avg_out_hi_old_0
adds avg_out_lo_old_0
rptk }
sfl
sach avged_avged_out_0
zalh avg_out_hi_old_1
adds avg_out_lo_old_1
rptk 7
sfl
sach avged_avged_out_l
zalh avg_out_hi_old_2
adds avg_out_lo_old_2
rptk 7
sfl
sach avged_avged_out_2
lac avged_avged_out_0
ldpk 8
sacl }2
lalk l
sacl }2
ldpk 5
lac avged_avged_out_1
ldpk }
sacl }6
lalk 1
```

sacl 68
ldpk 5
lac avged_avged_out_2
ldpk 8
sacl 104
lalk 1
sacl 108
ldpk 5
;Zero all the sums
zac
sacl avg_out_hi_old_0
sacl avg_out_lo_old_0
sacl avg_out_hi_old_1
sacl avg_out_lo_old_1
sacl avg_out_hi_old_2
sacl avg_out_lo_old_2
lark 6, number_average
no divide:
no_avg:
bskip_monitor_laser
;Monitor the varriables
ldpk 4
lac a_squ_hi_0
Idpk 8
sacl 0
lalk 1
sacl 4
ldpk 4
lac a_squ_lo_0
ldpk 8
sacl 8
lalk 1
sacl 12
ldpk 4
lac a_squ_hi_1
ldpk 8
sacl 40
lalk 1
sacl 44
ldpk 4
lac a_squ_lo_1
ldpk 8
sacl 48
lalk 1
sacl 52
ldpk 4
lac a_squ_hi_2
ldpk 8
sacl 80
lalk 1
sacl 84
ldpk 4
lac a_squ_lo_2
ldpk 8
sacl 88
lalk 1
sacl 92
ldpk 4
$\operatorname{larp} 1$
skip_monitor laser:
skip_laser:
;Set "flag" to execute channel zero next
lack 0
sacl flag
eint
ret

clear: ;load aux register one with the number of mem to clear
lark $1,200 \mathrm{~h}$
lark 2, 400h
zac
clearl: larp 1
sacl *+
$\operatorname{larp} 2$
banz clearl
lark 1,0
lark 2, 0
ret
setup_sin:
;current output frequency set to 500 Hz
; $\sin$ table scaled by 4096
;shift left by four bits
lark 5, sin_start
larp 5
lalk 0
sacl *+
1266
sacl *+
lalk 2408
sacl *+
lalk 3314
sacl *+
lalk 3896
sacl *+
lalk 4096
sacl *+
lalk 3896
$\mathrm{sacl}^{*+}$
lalk 3314
$\mathrm{sacl}^{*}+$

```
lalk 2408
sacl *+
lalk 1266
sacl *+
lalk 0
sacl *+
lalk -1266
sacl *+
lalk -2408
sacl *+
lalk -3314
sacl *+
lalk -3896
sacl *+
lalk -4096
sacl *+
lalk -3896
sacl *+
lalk -3314
sacl *+
lalk -2408
sacl *+
lalk -1266
sacl *+
ret
```

sin_logistics:
;memory location 900 h is start of sine table
lalk 902h
sacl aux_laser_pointer
lalk 17
sacl aux_laser_reset
lalk 900 h
sacl aux_sin_pointer
lalk 19
sacl aux_sin_reset
lalk 905h
sacl aux_cos_pointer
lalk 14
sacl aux_cos_reset
ret
hpf_coeff:

```
;multiplier \(=2^{\wedge} 14\)
;cutoff low = at ??? Hz
;cutoff hi = at ??? Hz
```

ldpk 4
lalk 102
sacl bpf_coeff_in_nm0
lalk 0
sacl bpf_coeff_in_nml
lalk-102
sacl bpf_coeff_in_nm2
lalk - 16384
sacl bpf_coeff_out_nmo
lalk 30970
sacl bpf_coeff_out_nml
lalk-16179
sacl bpf_coeff_out_nm2
ret
lpf_coeff:
;coefficients set to a low pass filter
;multiplier $=2 \wedge 12=4096$
;cutoff at 500 Hz
Jdpk 4
lalk 499
sacl lpf_coeff_in_nm0
lalk 499
sacl lpf_coeff_in_nml
lalk -16384
sacl lpf_coeff_out_nm0
lalk 15386
sacl lpf_coeff_out_nml
ret
ell_coeff:
;coefficients set to an elliptical filter
; multiplier $=2 \wedge 12=4096$
;cutoff at ??? Hz
;dip at ??? Hz
1dpk 4
lalk 1651
sacl ell_coeff_in_nm0
lalk -2668
sacl ell_coeff_in_nm1
lalk 1651
sacl ell_coeff_in_nm2
lalk -16384
sacl ell_coeff_out_nm0
lalk 29411
sacl ell_coeff_out_nml
lalk - 13923
sacl ell_coeff_out_nm2
ret
laser_setup:
lalk amp
sacl amplitude
lalk bias
sacl bias
ret
end;

## APPENDIX 7

## PROGRAM TO LINK THE ASSEMBLY PROGRAM

```
/* link.cmd */
h.obj /* object file for the testl.asm */
-o h.out /* option to name the output file */
/* see page 8-21 for memory stuff */
/* "TMS320 Fixed-Point Assembly Language Tools" */
/* PAGE 0: ROM: == program space */
/* PAGE l: RAM: = data space */
MEMORY
{
    PAGE 0: ROM1: origin = 0h, length }=30\textrm{h
    PAGE 0: ROM2: origin }=050\textrm{h},\quad\mathrm{ length }=1000\textrm{h
    PAGE 1: RAM: origin =200h, length = 1000h
}
/* see page 8-23 for section stuff */
/* "TMS320 Fixed-Point Assembly Language Tools" */
/* the "vectors" label is linked to the command file by '.sect "vectors"*/
/* in the assembly program to load the interrupt flow table at Oh*/
/* the ".text" label is linked to the assembly program to loaf the*/
/* main program at 50h*/
```


## SECTIONS

```
\{
vectors: \(\quad\) load \(=\) ROMI
.text: load = ROM2
\}
Batch file to compile DSP assembly language program:
File Name = h.asm
```

```
dspa h.asm
```

dspa h.asm
dsplnk -h link.cmd
dsplnk -h link.cmd
dsphex -t h.out

```
dsphex -t h.out
```


## APPENDDX 8A

DESCRIPTION OF DATA SAVING C-PROGRAM VARIABLES

Table A8.1: Symbol cross reference in data saving C-program.

| Symbol in Figure 1.2 | Equivalent variable in C-program |
| :---: | :---: |
| Amove | data_0 |
| Astat | data_1 |
| Apin | data_1 |
| Nstat | data_out |
| Npin | data_out |

Table A8.2 Description of data saving C-program variables.

| C-program variable | Type | Description |
| :--- | :--- | :--- |
| data_out[5000] | double | array to save the output data |
| data_0 | unsigned | Nmove signal from the DSP |
| data_1 | unsigned | Nstat or Npin signal from the DSP |
| flag0 | unsigned | data ready flag for data_0 |
| flagl | int | data ready flag for data_l number of samples to read |
| length | int | current number of samples taken |
| count | int | program sample flag |
| go_in | int | data sampled flag DSP |
| clear | unsigned | DSP memory address of Nmove |
| pointer_addr0 | unsigned | DSP memory address of Nstat or Npin |
| pointer_addrl | unsigned | DSP memory address of Nmove data flag |
| flag_addr0 | unsigned | DSP memory address of Nstat or Npin data flag |
| flag_addrl |  |  |

## APPENDRX 8B

## DATA SAVING C-PROGRAM

```
/* dspl.c */
#include <stdio.h>
#include <math.h>
#include <conio.h>
main()
{
/* Define Variables */
double data_out[5000];
unsigned data 0, data_1, flag0, flagl;
int length, count, go_in, clear;
unsigned pointer_addr0, pointer_addrl;
unsigned flag_addr0, flag_addr1;
FILE *fp;
/* Define variables to save data in Matlab format */
typedef struct
{
    long type; /* type */
    long mrows; /* row dimension */
    long ncols; /* column dimension */
    long imagf; /* flag indicating imag part */
    long namlen; /* name length (including NULL) */
}
Fmatrix;
Fmatrix \(x\);
/* Initialize variables */
pointer_addr0 \(=0 \times 438\);
pointer_addrl \(=0 \times 460\);
flag_addr0 \(=\) pointer_addr0 +4 ;
flag_addrl \(=\) pointer_addrl +4 ;
```

```
length = 2344;
```

```
/* Initialization to store data in Matlab file. */
x.type = 0;
x.mrows = 1;
x.ncols = length;
x.imagf = 0;
x.namlen = strlen("a") + 1;
/* Initialize relevant values to zero. */
count = 0;
flag0 = 0;
go_in = 0;
clear = 0;
clrscr();
/* Enter data taking loop. */
while(go_in == 0)
{
/* Read "data ready" flag from DSP. */
outport(0\times302, flag_addr0);
flag0 = inport(0\times300);
if(flag0== 1)
{
```

    /* Read data from DSP if data is ready. */
    /* Read Nmove */
    outport( \(0 \times 302\), pointer_addr0);
    data_0 \(=\) inport \((0 \times 300)\);
    /* Read Nstat or Npin */
    outport( \(0 \times 302\), pointer_addr1);
    data_1 \(=\) inport \((0 \times 300)\);
    /* Normalize data */
    data_out[count] \(=\operatorname{sqrt}((\) double \()\) data_1 \(1 /(\) double \()\) data_ 0\() ;\)
    gotoxy (1,1);
    printf("\%lf",data_out[count]);
        /* Reset the flag to zero for the next set of data */
        outport( \(0 \times 302\), flag_addr0);
    ```
    outport(0\times300, clear);
        count++;
        if(count == length)
        {
        go_in = 1;
    }
    }
}
/* Save the data into a Matlab readable file */
fp = fopen("b:data mat","w+b");
fwrite(&x, sizeof(Fmatrix), 1, fp);
fwrite("a", sizeof(char), (int)x.namlen, fp);
fwrite(&data_out, sizeof(double), count, fp);
fclose(fp);
```

printf(" $\operatorname{lnD}$ Data have been written to b:data.mat, name: aln");
\}

## APPENDIX 9A

## DESCRIPTION OF CONTROL C-PROGRAM VARIABLES

Table A9.1: Symbol cross reference in control C-program.

| Symbol in Figure 1.2 | Equivalent variable in C-program |
| :---: | :---: |
| Astat | $y_{-}$raw |
| Apin | Vpin |
| Npin | yd |

Table A9.2 Description of control C-program variables.

| C-program variable | Type | Description |
| :--- | :--- | :--- |
| data[7000] | double | array to save the output data |
| pointer_addr | unsigned | DSP memory address of the position signal |
| flag_addr | unsigned | DSP address of the position signal ready flag |
| Vpin_addr | unsigned | DSP memory address of the normalization signal |
| upper | int | upper limit of the D/A output value |
| lower | int | lower limit of the D/A output value |
| count | int | number of 100 Hz loops executed |
| flag |  |  |

Table A9.2
(Continued)

| C-program variable | Type | Description |
| :--- | :--- | :--- |
| go_in | int | program sample flag |
| U | int | command |
| length | int | number of program loops to execute |
| save_length | int | total length of data vector |
| mode | int | instantaneous position or run loop flag |
| run | int | values of control gains input from keyboard |
| scan | float | current position |
| y | float | reference position |
| y_last | float | position gain |
| y_ref | float | derivative gain |
| Kp | float | derivative signal |
| Kd | float | command signal before correction for D/A output |
| delta | current error |  |
| z | current output of lowpass filter |  |
| z_last | previous output of lowpass filter |  |
| al | lowpass filter output coefficient |  |

Table A9.2
(Continued)

| bl | float | lowpass filter input coefficient |
| :--- | :--- | :--- |
| Vcorrection | float | correction value to correct D/A offset error |
| Vpin | float | normalization signal from laser diode |
| y_raw | float | non-normalized pos signal from sensor module |
| Kpo | float | position gain input from keyboard |
| Kp_min | float | maximum value for the position gain |
| Kp_max | Type | Description |
| C-program variable | float | previous error |
| e_last | float | demporary error signal |
| e_temp | float | minimum value for the derivative gain |
| Kdo | floain input from keyboard |  |
| Kd_min | float | maximum value for the derivative gain |
| Kd_max | float | position gain multiplication and division factor |
| Kp_range | integral signal |  |
| K_sum | integral gain |  |
| Ki_min | minimum value for the integral gain |  |
|  | maximum value for the integral gain |  |

Table A9.2
(Continued)

| Kio | float | integral gain input from keyboard |
| :--- | :--- | :--- |
| L1 | float | 4th order coefficient of calibration polynomial |
| L2 | float | 3rd order coefficient of calibration polynomial |
| L3 | float | 2nd order coefficient of calibration polynomial |
| L4 | float | lst order coefficient of calibration polynomial |
| L5 | float | constant coefficient of calibration polynomial |
| yd | float | normalized position signal |
| sum | float | average of all error signals |
| avg |  |  |

## APPENDIX 9B

## CONTROL C-PROGRAM

```
/* control.c */
/* control C-program. */
#include <stdio.h>
#include <math.h>
#include <conio.h>
void main(int argc, char *argv[])
{
/* Define variables. */
double data[7000];
unsigned pointer_addr, flag_addr, Vpin_addr;
int upper, lower, count, flag, go_in, clear, U, length, save_length;
int Vreset, mode, run, scan;
int Kp_power, Kd power;
float y, y last, y ref, Kp, Kd, delta, u temp, e, z, z_last, al, bl;
float Vcorrection, Vpin, y_raw, Kpo, Kp_min, Kp_max, e_last, e_temp;
float Kdo, Kd_min, Kd_max, Kp_range, Kd_range, e_sum, Ki;
float Ki_min, Ki_max, Ki_range, Kio;
float L1,L2, L3, L4, L5, yd, sum, avg;
FILE *fp;
/* Define variables to save data in Matlab format. */
typedef struct
{
        long type;
        long mrows;
        long ncols;
        long imagf;
        long namlen;
}
Fmatrix;
Fmatrix x;
```

```
/* Initialize variables. */
mode = 0; /* mode = 0 for constant gain, = 1 for variable gain */
length =2000; /* number of data samples */
y_ref=70.0; /* command position */
al = 0.7548; /* lpf input gain */
bl=0.5095; /* lpf output gain */
Kpo = (float) atof(argv[1]); /* position gain */
Kdo = (float) atof(argv[2]); /* derivative gain */
Kio = (float) atof(argv[3]); /* integration gain */
Kp_range =2.0; /* position gain range multiplier */
Kd_range =2.0; /* derivative gain range multiplier */
Ki_range = 10.0; /* integration gain range multiplier */
upper = 4095; //* command upper limit */
lower = 0; /* command lower limit */
pointer_addr = 0x438; /* DSP address to read feedback signal */
flag_addr = pointer_addr + 4; /* DSP address of data ready flag */
Vpin_addr = 0x410; /* DSP address to read Vpin signal */
save_length = length * 3; /* length of data vector to save */
Vcorrection = 0x6; , /* hex correction value to fix D/A */
/* Coefficient values for the curve fit.*/
Ll = 3346;
L2 =-5630;
L3 = 3262;
L4 =-835;
L5 = 141.8;
/* Initialization to store data in Matlab file. */
x.type = 0;
x.mrows = 1;
x.ncols = save_length;
x.imagf = 0;
x.namlen = strlen("a") + 1;
/* Calculate the ranges of the gains. */
Kp_max = Kp_range * Kpo;
Kp_min = Kpo/Kp_range;
Kd_max = Kd_range * Kdo;
Kd_min = Kdo / Kd_range;
Ki_max = Ki_range * Kio;
Ki_min = Kio / Ki_range;
/* Initialize relevant values to zero. */
count = 0;
flag= 0;
```

```
go_in = 0;
clear = 0;
y = 0;
y_last = 0;
z=0;
z_last = 0;
data[0] = 0;
data[length] = 0;
data[length*2] = 0;
e_sum = 0;
run = 0;
scan = atoi(argv[4]);
/* Ensure that e is not zero to start. */
e=(Kp_max + Kp_min)/2.0;
clrscr();
/* Enter data-taking loop. */
while (run ==0)
{
if (kbhit())
/* Press any key to take data. */
{
    scan = 1;
}
if (scan == 0)
/* Output the position on the screen to set. */
{
    /* Read non-normalized mobile sensor module signal. */
    outport (0x302, pointer_addr);
    y_raw = inport(0x300);
    /* Read laser diode power feedback signal for normalization. */
    outport (0\times302, Vpin_addr);
    Vpin = inport(0x300);
/* Normalize mobile sensor module signal. */
yd = y_raw / Vpin;
/* Calculate current position. */
y=L1* yd* yd* yd*yd + L2* yd*yd*yd + L3**yd*yd + L4* yd +L5;
/* Output current position to monitor. */
gotoxy(1,1);
```

```
    printf("%lf",y);
}
if (scan == 1)
{
    while(go_in == 0)
    {
        outport(0x302, flag_addr);
        flag = inport(0x300);
        if(flag == 1)
        {
            count++;
                if( count == length-1 )
                {
            go_in=1;
            }
        y_last = y;
```

            /* Read non-normalized mobile sensor module signal. */
            outport ( \(0 \times 302\), pointer_addr);
            \(y_{\text {_raw }}=\operatorname{inport}(0 \times 300)\);
            /* Read laser diode power feedback signal for
                    normalization. */
                outport (0x302, Vpin_addr);
                Vpin \(=\operatorname{inport}(0 \times 300)\);
                /* Normalize mobile sensor module signal. */
                yd = y_raw / Vpin;
                /* Calculate current position. */
                \(y=L 1^{*} y d^{*} y d^{*} y d^{*} y d+L 2^{*} y d^{*} y d^{*} y d+L 3^{*} y d^{*} y d+\)
                    L4*yd +L 5 ;
                data[count] \(=y\);
                /* Output current position to monitor. */
                gotoxy(1,1);
                printf("\%lf", y);
                /* Calculate the error. */
                e_last \(=\mathrm{e}\);
                \(\mathrm{e}=\mathrm{y}-\mathrm{y}\) ref;
                \(\mathrm{e}_{-}\)sum \(=\mathrm{e}_{-}\)sum +e ;
    ```
data[count+length] = e;
e_temp = e;
if(mode = 0)
{
    /* Mode Zero -- Constant Gain. */
    Kp = Kpo;
    Kd=Kdo;
    Ki=Kio;
}
if(mode == 1)
{
    /* Mode One -- Variable Gain. */
    if (e_temp == 0)
    {
        e_temp = e_last;
    }
    if ( e_temp < 0)
    {
        e_temp = -e_temp;
    }
    Kp=Kpo / (e_temp);
    if(Kp>Kp_max )
    {
        Kp=Kp_max;
    }
    if( Kp< Kp_min )
    {
        Kp=Kp_min;
    }
    Kd = Kdo / (e_temp);
    if(Kd> Kd_max )
    {
        Kd = Kd_max;
    }
```

```
    if (Kd< Kd_min)
    {
    Kd=Kd_min;
    }
Ki= Kio / (e_temp);
if(Ki> Ki_max)
{
    Ki= Ki_max;
}
if(Kio < Ki_min);
{
    Ki=Ki_min;
}
}
/* Put error through lowpass filter. */
z_last = z;
z=al*(y+y_last )-bl* z_last;
delta = z-z_last;
/* Calculate command signal. */
u_temp = Kp * e + Kd * delta + Ki * e_sum;
u_temp = u_temp + 0x800 + Vcorrection;
/* Clip command signal to +/- 5volts. */
if(u_temp > upper)
{
    u_temp = upper,
}
if(u_temp < lower)
{
    u_temp = lower;
}
data[count+length*2] = u_temp;
U = u_temp;
/* Output command signal to DSP. */
outport(0\times302, 0\times478);
outport(0x300, U);
```

```
                /* Reset the flag to zero for the next set of data. */
                outport(0x302, flag_addr);
                outport(0\times300, clear);
        }
    }
    run = 1;
    }
}
/* Set the D/A to zero volts. */
Vreset =0x800 + Vcorrection;
outport(0\times302, 0x478);
outport(0x300, Vreset);
sum = 0;
go_in = 0;
count = 0;
/* calculate }\sum\mp@subsup{\textrm{E}}{}{2}*
while(go_in = 0)
{
```

```
    sum \(=\) sum + data \([\) count + length \(] *\) data \([\) count + length \(] ;\)
```

    sum \(=\) sum + data \([\) count + length \(] *\) data \([\) count + length \(] ;\)
    if \((\) count \(=\) length -1\()\)
    if \((\) count \(=\) length -1\()\)
    \{
    \{
        go_in \(=1 ;\)
        go_in \(=1 ;\)
    \}
    \}
    count \(=\) count +1 ;
    count \(=\) count +1 ;
    }
avg = sum/length;
printf("\n%lf",sum);
/* Save the data into a Matlab readable file. */
fp = fopen("b:data.mat","w+b");
fwrite(\&x, sizeof(Fmatrix), 1, fp);
fwrite("a", sizeof(char), (int)x.namlen, fp);
fwrite(\&data, sizeof(double), save_length, fp);

```
fclose(fp);
printf("\nData have been written to b:d.mat, name: aln");

\section*{REFERENCES}

Chang, T. Decentralized Robust Control of Interconnected Resonators.
Davison, E and T. Chang. Decentralized Controller Design Using Parameters Optimization Methods. Control-Theory and Advanced Technology. Vol. 2, No 2, 131-154, 1986.

Davison, E and T. Chang. Decentralized Stabilization and Pole Assignment for General Proper Systems. IEEE Transactions on Automatic Control, June 1990, pp 652-664.

Dorf, Richard. Modern Commol Systems, 5th Edition. New York, NY: Addison-Wesley, 1989

Johnson, Johnny. Introduction to Digital Signal Processing, Englewook Cliffs, NI: PrenticeHall, 1989

Kwong, R. and T. Chang. Development of Control; System Hardware Demonstration for Third Generation Spacecrafts. Department of Electrical Engineering, University of Toronto, Toronto, Ontario June, 1986.

Lewis, Frank. Applied Optimal Control and Estimation: Digital Design and Implementation. Englewook Cliffs, NJ: Prentice-Hall, 1992.

Phillips, Charles and Royce Harbor. Feedback Control Systems, 2nd Edition. Englewood Cliffs, NJ Prentice-Hall, 1991

Ziemer, R and W. Tranter Principles of Communicaton, Systems Modulation and Noise, 3rd Edition. Boston, MA: Houghton-Mifflin Co, 1990.```


[^0]:    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

[^1]:    ;Band Pass Filter
    bpf_0: ;load the "input"
    ; add gain to the input
    lac ad_in_0, hpfgain
    sacl bpf_in_nm0_0
    zac

