

© Yingfeng Shan, 2008  
All Rights Reserved

FABRICATION AND INTEGRATED FEEDFORWARD AND FEEDBACK  
CONTROL OF IONIC POLYMER-METAL COMPOSITE ACTUATORS

A thesis submitted in partial fulfillment of the requirements for the degree of  
Master of Science in Engineering  
at Virginia Commonwealth University

by

YINGFENG SHAN

Mechanical Engineering, Virginia Commonwealth University

Major Director:

Dr. Kam K. Leang

Assistant Professor, Mechanical Engineering

Virginia Commonwealth University

Richmond, Virginia

August, 2008

## **Acknowledgment**

Firstly, I would like to express my sincere appreciation to my advisor, Kam Leang, for his guidance, energy, inspiration, and encouragement during those long hours of discussion. Secondly, I thank my parents, older brother, and younger sister for their unconditional love, enthusiasm, and support. Lastly, I acknowledge my lab mates, Uğur Arıdoğan and Robert Riddle, for their encouragement as well as the many hours spent discussing and sharing ideas.

## Dedication

To my parents, Zhanjun Shan and Xinrong Zhang.

## Table of Contents

<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>xvi</b>
<b>Chapter 1: Introduction</b>	<b>1</b>
1.1 Thesis Goal and Objectives . . . . .	1
1.2 Contribution . . . . .	3
1.3 Thesis Overview . . . . .	4
<b>Chapter 2: Ionic Polymer-Metal Composites (IPMCs)</b>	<b>5</b>
2.1 IPMC Material Structure and Characteristics . . . . .	5
2.2 History of IPMCs . . . . .	7
2.3 Applications of IPMCs . . . . .	9
2.4 Challenges with Using IPMCs for Positioning . . . . .	14
<b>Chapter 3: Review of Control Methods for IPMC Actuators</b>	<b>17</b>
3.1 Feedback Control Approaches . . . . .	17
3.2 Feedforward Control Approaches . . . . .	19
3.3 Integrated Feedback and Feedforward Control . . . . .	20
3.4 Summary . . . . .	21
<b>Chapter 4: Integrated Feedforward and Feedback Controller Design</b>	<b>23</b>
4.1 Inversion-Based Feedforward Controller Design . . . . .	24
4.2 Feedback Controller Design . . . . .	28

4.3	Integrated Feedforward and Feedback Controller Design . . . . .	28
<b>Chapter 5: The Experimental IPMC Actuator System</b>		<b>29</b>
5.1	Fabrication of IPMCs . . . . .	29
5.2	The Experimental System . . . . .	34
5.3	Open-Loop Step Responses of a Selected IPMC Actuator . . . . .	44
5.4	Controllers Implementation . . . . .	46
<b>Chapter 6: Experimental Results and Discussion</b>		<b>53</b>
6.1	Feedforward Tracking Results . . . . .	53
6.2	Feedback Tracking Results . . . . .	55
6.3	Integrated Feedforward and Feedback Controller Tracking Results . .	56
<b>Chapter 7: Conclusions and Future Works</b>		<b>60</b>
<b>Bibliography</b>		<b>60</b>
<b>Appendix A: A Biaxial Shape Memory Alloy Actuated Cell/Tissue Stretching System</b>		<b>75</b>
A.1	Introduction . . . . .	75
A.2	Overview of the Design . . . . .	76
A.3	Design Considerations . . . . .	79
A.4	Experimental Results and Discussion . . . . .	89
A.5	Conclusions . . . . .	91
A.6	Future Work . . . . .	93
<b>Appendix B: Low-Cost Optoelectronic Sensors for Sub-Micro-Level Position Measurement and Control</b>		<b>94</b>
B.1	Introduction . . . . .	94
B.2	Reflective IR Sensors . . . . .	98
B.3	IR Sensor Performance . . . . .	99

B.4 Example Application: Observer-Based Feedback Control of Piezopositioner . . . . .	112
B.5 Conclusions . . . . .	120
<b>Appendix C: Matlab Program for Optimal Inversion</b>	<b>123</b>
<b>Appendix D: C Program for Data Acquisition and Control</b>	<b>128</b>
D.1 Feedback controller implementation . . . . .	128
D.2 Feedforward controller implementation . . . . .	136
D.3 Integrated feedforward and feedback controller implementation . . . .	144

## List of Figures

2.1	Movement of cations with water molecules inside of IPMCs. . . . .	7
2.2	Chemical structure of Nafion, which includes fluorocarbons, oxygen, sulfonate groups, and a mobile cation $X^+$ that can be hydrogen or sodium. The $K$ is usually 5 to 11 and the $L$ is usually 1 (Shahinpoor and Kim, 2001). . . . .	8
2.3	An example application of IPMCs as tactile sensor for teleoperation (Bar-Cohen et al., 2002a). . . . .	10
2.4	An example application of IPMCs as artificial muscle for swimming fish-like underwater microrobots (Guo et al., 2006). . . . .	12
2.5	An experimentally measured relaxation behavior in an IPMC, where the dashed line is a 1 V step input applied to the IPMC actuator and the solid line is the displacement response of the IPMC actuator. . . .	15
2.6	The dynamic effects in an IPMC actuator showing the mismatch between the desired response and the measured response at different frequencies. . . . .	15
3.1	Block diagram of a feedback control system, where $y_{ref}$ is the reference output signal, $y$ is the measured output, and $e$ is the tracking error, that is $e = y_{ref} - y$ . . . . .	17
3.2	Block diagram of a feedforward control system, where $y_d$ is the desired output signal, $y$ is the measured output, and $u_{ff}$ is the feedforward input applied to the system. . . . .	19

- 4.1 A block diagram of an integrated feedforward and feedback controller. A desired output trajectory  $y_d$  is applied to the feedforward controller. The feedforward controller generates a feedforward input  $u_{opt}$  as well as a reference output  $y_{opt}$ . The feedforward input  $u_{opt}$  is applied directly to the IPMC actuator. The output of the IPMC actuator is  $y$  and the tracking error  $e = y_{opt} - y$  is used as an input to the feedback controller to generate a feedback input  $u$ . . . . . 23
- 4.2 Block diagram of the inversion-based feedforward controller. A desired output trajectory  $y_d$  is applied to the inverse model. The inverse model generates a feedforward input  $u_{ff}$  which is then applied to the IPMC actuator. The output of the IPMC is  $y$ . . . . . 24
- 5.1 The chemicals used to fabricate IPMC actuators, where the Nafion liquid (Ion Power, type 117) was used to cast Nafion membranes, and the other chemicals, including deionized water, 1 molar  $H_2SO_4$ ,  $Pt(NH_3)_4Cl_2$  powders, and  $NaBH_4$  granules, were used for plating electrodes on the surfaces of the Nafion membranes. . . . . 30
- 5.2 Steps to fabricate IPMCs. Steps 1-4: The process for casting the Nafion polymer membrane from liquid Nafion solution. Step 5: A sample of the custom-made IPMCs after depositing platinum electrodes. . . . . 31
- 5.3 (a) and (c) A sample of the custom-made Nafion membrane compared to (b) a commercially made Nafion membrane. . . . . 32

- 5.4 Photographs of the IPMCs made from commercial and custom-made Nafion membranes. (a) Type I IPMCs were made from two commercially available Nafion membranes with different thickness. The thicknesses of these two samples were  $248\ \mu\text{m}$  and  $324\ \mu\text{m}$ , and the thickness of the platinum electrodes layers on each side of the IPMCs was proximate  $35\ \mu\text{m}$ . (b) Type II IPMCs were made from two custom-made Nafion membranes with different thickness. The thicknesses of these two samples were  $450\ \mu\text{m}$  and  $1.07\ \text{mm}$ , and the thickness of the platinum electrodes layers on each side of the IPMCs was also proximate  $35\ \mu\text{m}$ . (c) Comparison between the Type I IPMCs and the Type II IPMCs. (d) SEM images of the custom-made IPMCs that show the Nafion polymer membrane and the platinum electrodes layers. . . . . 35
- 5.5 (a) and (b) Photographs of experimental system where the IPMC actuator is mounted in cantilever configuration and displacement is measured by a laser displacement sensor along the y-axis as shown (SUNX micro laser displacement sensor LM10, model No. ANR12511 with sensor controller, model No. ANR5232). In the experiment, the IPMC actuator was mounted  $55\ \text{mm}$  away from the sensor front to make sure the bending movement of the IPMC actuator can be captured by the sensor. (c) The block diagram of the experimental system. . . . . 37
- 5.6 The laser displacement sensor. (a) Front view and side view of the sensor, where the emitter emits laser light at the target and the receiver collects the light reflected by the target. The sensor has its own controller, which connects to the laser unit. (b) The sensor configured to detect target displacement. (c) The calibration system used to calibrate the sensor's response. . . . . 38

- 5.7 The sensor calibration results which show the sensor output voltage versus target displacement. (a) Results of four trials and (b) the average of the four trials. . . . . 40
- 5.8 Laser displacement sensor's measurement range. (a) Measurement center of the sensor is 50 mm away from the front of the sensor. When the target is at this point, the sensor output voltage is zero. (b) Range of no response is a range around the measurement center with length of 7 mm. When target moves in this range, the sensor has no response. (c) linear range I and II. When target move in these ranges, the sensor output is nearly linear with respect to the displacement of the IPMC actuator. (d) Target position II and III mean that when the target moves outside of the range between target position II and III, the sensor output will exceed the measurable level of the DAQ system. . . . 41
- 5.9 Circuits for the IPMC actuation and filter for sensor output. (a) Voltage amplifier circuit, where the op-amp and power transistors TIP33 and TIP34 form a closed-loop to equate  $\hat{u}$  to  $u$ . (b) A filter circuit was used to process the sensor output to minimize the effects of noise at high frequency. . . . . 42
- 5.10 Open-loop performance of four custom-made IPMC actuators. (a) The input signal versus time shows the voltages applied to each of the IPMCs. These voltages were chosen because they ensured that each actuator achieved almost the same output displacement. (b) The measured output displacement (in volts) versus time shows the performances of the four IPMC actuators. (c) Inset figure shows the high frequency response of the four IPMC actuators. . . . . 44

5.11	Open-loop step responses of a selected IPMC actuator. (a) The input signal (in volts) versus time. (b) The measured output displacement (normalized) versus time shows the transient behavior and the low frequency relaxation behavior. (c) Inset figure shows more closely the high frequency transient. The 5% settling time of the transient was approximately 170 ms. . . . .	45
5.12	Frequency response $G(j\omega)$ of the IPMC actuator, where the $G_k$ is the gain margin. . . . .	47
5.13	Steps to implement the feedforward method. . . . .	48
5.14	The tracking error and input weights $Q(j\omega)$ and $R(j\omega)$ respectively. .	49
5.15	The process for the feedforward input computation. Part I was used to treat the model $G(j\omega)$ of the IPMC actuator system and to find the optimal inversion from the system model $G_{ff}(j\omega)$ . This part was based on the $Q(j\omega)$ and $R(j\omega)$ weights in frequency domain. Part II was used to treat the given desired tracking trajectory $y_d(t)$ , <i>i.e.</i> , to convert to the frequency domain $Y_d(j\omega)$ . Finally, in Part III the frequency response function $G(j\omega)$ and $G_{ff}(j\omega)$ from Part I along with the $Y_d(j\omega)$ from Part II were used to find the time domain signal $u_{opt}(t)$ and $y_{opt}(t)$ . . . . .	50
5.16	The process reaction curve for a step response of the IPMC actuator, where $L$ is the time delay, $R$ is the slope of the tangent line at the indicated location and $l$ is the time constant. . . . .	51
5.17	The performance of the designed PI feedback controller compared to the performance of the open-loop step response of the IPMC actuator.	52
6.1	Feedforward control inputs relative to the desired outputs. (a) Feedforward input for 1 Hz tracking; (b) Feedforward input for 10 Hz tracking; (c) Feedforward input for 18 Hz tracking. . . . .	54

6.2	Feedforward control results. (a) 1 Hz tracking; (b) 10 Hz tracking; (c) 18 Hz tracking. . . . .	55
6.3	Feedback controller results. (a) 1 Hz tracking; (b) 10 Hz tracking; (c) 18 Hz tracking. . . . .	57
6.4	Integrated feedforward and proportional-integral (PI) feedback controller results. (a) 1 Hz tracking; (b) 10 Hz tracking; (c) 18 Hz tracking.	59
A.1	The shape memory alloy (SMA)-based cell stretching platform. Left: Cells are cultured on an elastic membrane, for example polydimethylsiloxane (PDMS), then the membrane is stretched using SMA wire actuators. Right: $SMA_1$ and $SMA_2$ strain the membrane along the longitudinal direction while $SMA_3$ and $SMA_4$ strain it along the transverse direction. . . . .	77
A.2	Left: Detailed view of the SMA-based cell stretching platform. Four SMA actuators, two in the $x$ -direction and two in the $y$ -direction, stretch the membrane biaxially. Optical sensors are used to calibrate and measure the deflection of the cantilever. Also, they are used in the feedback control system to achieve accurate stretching of the membrane. Right: Each actuator is made by wrapping a single SMA wire to create a parallel set of actuators of length $L_2$ . Current is applied to heat all the wires simultaneously, and the heat causes contraction, which bends a cantilever and subsequently stretches the membrane. As the wires are cooled, the restoring force in the cantilever stretches the SMA wires back to their initial state and the strain in the membrane is relieved. . . . .	80
A.3	Cantilever and SMA wire actuator. When heated, the SMA exerts a force $F$ which in turn deflects the cantilever by $\Delta x$ . The deflection is transmitted to the membrane attached to the cantilever. . . . .	83

A.4	Three membranes considered for FEA analysis of the strain distribution.	83
A.5	Finite element analysis results: strain distribution in the three example PDMS membranes. Membrane is stretched in the longitudinal ( $x$ ) direction. (a) - (c) unconstrained along transverse direction, therefore the membrane contracts. . . . .	85
A.6	Finite element analysis results: (a) and (b) constrained along the transverse direction, therefore zero strain in the $y$ -direction. . . . .	86
A.7	Experimental results of example $SMA_1$ : (a) Open-loop step response showing heating and cooling behavior of SMA. (b), (c), and (d), applied input current vs. time, measured deflection of cantilever beam, and hysteresis curve (deflection (output) vs. applied current (input)), respectively; (e) open-loop tracking error showing distortion due to hysteresis effect. . . . .	87
A.8	Feedback control (PI) of SMA actuator for stretching PDMS membrane: (a) The feedback control system for each SMA actuator; (b) Open-loop and closed-loop . . . . .	88
A.9	Experimental results: uniaxial stretching of PDMS membrane. . . . .	91
A.10	Experimental results: biaxial stretching of PDMS membrane. Arrow indicates the direction of strain. . . . .	92
B.1	Infrared reflective sensors: (a) diffused, (b) focused, and (c) configured to detect target distance ( <i>i.e.</i> , gap) $d$ . . . . .	95
B.2	The calibration system. . . . .	101
B.3	Reflective displacement sensor circuit. . . . .	103
B.4	Sensor calibration: (a) Sensor output voltage $V_s$ versus target distance $d$ . (b) normalized sensitivity (derivative of $V_s$ with respect to $d$ ) versus target distance $d$ . . . . .	104

B.5	The effect of target area on the IR sensor's output. (a1) with inset (a2) shows the response of the diffused IR sensor for different target surface areas; likewise, (b1) with inset (b2) shows the response of the focused IR sensors under similar conditions; and (c) shows the maximum variation in the output response of the diffused and focused IR sensor relative to the response for the nominal-sized target. . . . .	107
B.6	The experimental bimorph piezoactuator. An input voltage causes the piezo to bend along the $x$ axis. On one side of the actuator is an inductive sensor to measure the displacement. On the opposite side is a low-cost IR optoelectronic sensor (No. 4, QRB1113) which faces a white target (Kodak paper #KOD1527795) attached to the piezoactuator. Both the inductive and IR sensor measure the displacement of the piezoactuator along the $x$ axis. . . . .	108
B.7	(a1) - (a3) Unfiltered inductive and (b1) - (b3) IR sensor output versus time to determine the IR sensor's resolution. . . . .	110
B.8	(a) and (b) The measured frequency response (magnitude and phase vs. frequency) of the experimental piezopositioner. Solid line is the response measured by the IR sensor; dashed line denotes the response measured by the inductive sensor. (c) Power spectral densities of the output signals for the IR sensor (solid line) and the commercial inductive sensor (dash line). . . . .	111
B.9	Controller block diagrams: (a) observer-based full-state feedback controller and (b) state-feedback inner loop with integral output controller (outer loop) and feedforward input. Open-loop responses: (c) Transient response and creep behavior. The settling time is approximately 200 ms. (d) hysteresis behavior. . . . .	113

B.10	The measured frequency response (magnitude and phase vs. frequency) of the piezoactuator measured with IR sensor and inductive sensor passed through the filter $G_f(s)$ . Solid line is the response measured by the IR sensor; dashed line denotes the response measured by the inductive sensor. Also shown (dash-dot line) is the second-order model $G(s)$ for comparison. . . . .	115
B.11	Observer results: (a) and (b) are the estimates of the states $x_1$ and $x_2$ , respectively, that comparing IR and inductive sensor outputs for small input $u = 400$ mV; (c) estimates of the state $x_1$ for large input $u = 3.1$ V. . . . .	118
B.12	Tracking results of state-feedback controller. . . . .	120
B.13	(a)-(c) State-feedback controller with integral control, displacement and input vs. time. (d) Comparison of hysteresis effect for open-loop (dash line) and closed-loop (solid line). . . . .	121

## List of Tables

3.1	Review of control methods for IPMC actuators. . . . .	22
6.1	Feedforward control results: maximum and root-mean-squared tracking error relative to range for without and with inverse feedforward (FF) control. . . . .	56
6.2	The proportional-integral (PI) feedback control results: maximum and root-mean-squared tracking error relative to range for without and with the PI feedback (FB) control. . . . .	58
6.3	Feedforward and proportional-integral (PI) feedback control results: maximum and root-mean-squared tracking error relative to range for PI feedback only (FB only) control and integrated feedforward control and PI feedback (FF & FB). . . . .	58
B.1	Low-cost optoelectronic sensors considered in this study. . . . .	102

## **Abstract**

### FABRICATION AND INTEGRATED FEEDFORWARD AND FEEDBACK CONTROL OF IONIC POLYMER-METAL COMPOSITE ACTUATORS

by Yingfeng Shan

A thesis submitted in partial fulfillment of the requirements for the degree of Master  
of Science in Engineering at Virginia Commonwealth University

Virginia Commonwealth University, 2008

Major Director: Dr. Kam K. Leang  
Assistant Professor, Mechanical Engineering

Ionic polymer-metal composites (IPMCs) are innovative materials that offer combined sensing and actuating ability in a lightweight and flexible package. As such, they have been exploited in robotics and a wide variety of biomedical devices. For example, they can be used as sensors for teleoperation, as actuators for positioning in active endoscopes, as injectors for drug delivery, and as fins for propelling aquatic robots. One of the main challenges of IPMC-based actuators is controlling their movements. In particular, these actuators exhibit relaxation behavior, nonlinearities, and dynamic effects which make precision positioning challenging. Precise control of the motion of IPMC actuators is needed to optimize their performance for emerging applications. This thesis considers the fabrication of IPMC actuators and application of an integrated feedforward and feedback controller to improve their performance. Specifically, a model-based feedforward controller was used to account for dynamic effects

and a feedback controller was integrated with the feedforward controller to provide robustness and to minimize the nonlinearities and unmodeled dynamics. Experimental results are presented showing significant improvement in the tracking performance using the proposed controller. For example, the feedforward controller reduced the tracking error by over 75% compared to the case without dynamic compensation. In addition, by adding a proportional-integral feedback controller, the tracking error was less than 10% at an operating frequency of 18 Hz.

# Chapter 1: Introduction

## 1.1 Thesis Goal and Objectives

The main goal of this thesis is to precisely control the movement of ionic polymer-metal composite (IPMC) actuators. Specifically, this work focuses on using an integrated feedforward and feedback controller to account for relaxation behavior, nonlinearities, and vibrational dynamics in IPMCs.

To date, IPMCs have been considered in a wide variety of applications that range from underwater robotics to biomedical devices (Shahinpoor et al., 1998; Bar-Cohen et al., 2000; Shahinpoor and Kim, 2005). As actuators, IPMCs offer several unique advantages compared to other active materials such as piezoelectrics (Moheimani and Fleming, 2006) and shape memory alloys (Waram, 1993). For example, IPMCs can be driven with low voltage; they operate in water; and they are lightweight, flexible, and can be easily cut and shaped. However, when they are used as actuators they undergo relaxation, oscillations caused by dynamic effects, nonlinear response, and changes in the system parameters (Lee et al., 2006). All of these effects can cause significant positioning error when IPMC actuators are used in applications. Therefore, precise control of their movements is needed and this thesis focuses on controlling IPMCs, in particular, custom-made IPMC actuators.

There are two main objectives to achieve the goal:

**1. Fabricate IPMCs:** First, IPMC actuators were fabricated for this project. Currently, IPMCs are only commercially available from a small number of vendors (such as Environmental Robotics, Inc.), and they are relatively expensive to purchase. For example, the least expensive IPMC package costs 99 USD, which includes two pieces of IPMCs with dimensions  $0.5\text{ cm} \times 2\text{ cm} \times 0.2\text{-}0.3\text{ mm}$ . The commercial IPMCs come

in limited thicknesses, too. It is noted that the thickness affects the force that an IPMC actuator can generate. Specifically, thicker IPMCs generate more force and higher force output broadens the applications of IPMC actuators. Therefore, because of the high cost and limited available thickness of commercially-made IPMCs, IPMCs were fabricated in the lab for this project.

The fabricated IPMCs consisted of a Nafion polymer membrane sandwiched between two layers of platinum metal electrodes. The metal electrodes were electrochemically deposited on the Nafion membrane surfaces. Water was used as the polar solvent to induce actuation upon the application of an electric field (voltage). The following tasks were performed to fabricate IPMC actuators:

- a) *Casting the Nafion polymer membranes.* The polymer membrane was created by a casting process using commercially available Nafion liquid solution from the vendor Ion Power. The polymer is required for making IPMCs. Although Nafion membrane can be purchased from Ion Power or Dupont, custom-thickness Nafion membrane is not commercially available. The casting process is one approach to fabricate Nafion membranes for making IPMC actuators. Nafion membrane is a conducting polymer that allows the free movement of ions within its structure. The IPMC actuators made from custom-made, thicker Nafion membrane can generate more force, as well as store more water for longer operating life in air.
- b) *Plating platinum metal electrodes on the Nafion membrane.* The metal electrodes plated on the surfaces of the Nafion membrane allow voltage to be applied to control the bending of the IPMC actuator. The electrodes were deposited on the membrane surfaces through a chemical reduction or electro-plating process. Several different metals can be used as electrodes for IPMCs, for example silver, copper, and platinum. Different metals give IPMCs different properties (Chung et al., 2006; Kim and Shahinpoor, 2000). In this work, platinum was

used to create the electrodes because it has low resistance, is inert, and resists oxidization.

**2. Design and implement a controller:** An integrated feedforward and feedback controller was designed and implemented to control the movement of the custom-made IPMC actuators. An inversion-based feedforward controller (Devasia et al., 1996; Leang and Devasia, 2007) was used to compensate for the dynamic effects at high-operating speed and a proportional-integral (PI) feedback controller was used to minimize the effects of unmodeled dynamics, nonlinearities, and relaxation. Additionally, the feedback controller provided robustness. The tasks included:

- a) *Designing the inversion-based feedforward controller and feedback controller for the IPMC actuator.* In this task, an analysis was done to determine the feedforward input to control the IPMC actuator. Next, a linear vibrational dynamics model was determined from the measured frequency response of the IPMC actuator and the model was used to design and simulate the response of the controller. Then, a PI feedback controller was designed, tuned using the Ziegler-Nichols method (Franklin et al., 2005), and implemented.
- b) *Applying the integrated feedforward and feedback controller to an experimental IPMC system and evaluating its performance.* This task involved creating an experimental IPMC system for implementing and validating the performance of the controller. The system consisted of a custom-made voltage amplifier, laser sensor, and data acquisition system. The results were evaluated and quantified.

## 1.2 Contribution

The contribution of this thesis is the application of an inversion-based feedforward controller for precise motion control of IPMC actuators. This is one of the first work that considers dynamic effects in IPMCs. By adding feedback control, the integrated

controller provided good tracking precision at relatively high operating frequency. In particular, the inversion-based feedforward controller was designed to compensate for the dynamics which are significant at high frequencies. Then the PI feedback controller was designed to take care of unmodeled dynamics, such as nonlinearities and relaxation behavior. Experimental results are presented to demonstrate the effectiveness of the control approach for IPMC actuators.

### **1.3 Thesis Overview**

This thesis is organized as follows. Chapter 2 presents a background of IPMCs. This chapter includes the IPMC's material structure and characteristics, the history of IPMCs, and applications of IPMCs. In Chapter 3, the control methods that have been studied and implemented on IPMC actuators are reviewed. The chapter focuses on feedforward and feedback control methods. Then, Chapter 4 discusses the inversion-based feedforward control method and the feedback controller design. In Chapter 5, the details of the experimental IPMC actuator system are discussed along with the implementation of the controller. The discussion includes a detailed treatment of the IPMC fabrication process, sensor calibration, and control circuit design. In Chapter 6, the experimental results are given and discussed, followed by a brief summary and concluding remarks in Chapter 7. Finally, appendices at the end describe work and sample computer programs.

## Chapter 2: Ionic Polymer-Metal Composites (IPMCs)

In this chapter, the background of ionic polymer-metal composites is presented. First, the IPMC's material structure and its characteristics are discussed. Then the history of the IPMC is described. Finally, applications of IPMCs and the challenges of using IPMC material as actuators are presented.

### 2.1 IPMC Material Structure and Characteristics

Ionic polymer-metal composite consists of a conducting polymer backbone sandwiched between two metal layers (electrodes). When the composite is saturated in a polar solvent such as water and then voltage is applied to the electrodes, the composite bends. The bending is caused by the movement of ions and polar solvent within the polymer – the migration of these ions and molecules induce swelling on one side of the composite and shrinking on the opposite side causing the structure to bend. An oppositely applied voltage causes bending in the opposite direction. Conversely, when the IPMC is deformed, it generates a measurable voltage, and thus IPMCs can be used for sensing applications.

Ionic polymer-metal composites belong in the electroactive polymer (EAP) family. There are two major categories of EAPs. The first is the electrostatic electroactive polymer (Bar-Cohen et al., 1997). The actuation of the electrostatic version of EAPs is caused by electrostatic forces which *squeezes* the polymer. This kind of polymer requires very high voltages for actuation, for example 2 to 10 kV (Bar-Cohen et al., 2001). The other type of EAPs is the ionic electroactive polymer (Bar-Cohen et al., 1999) such as the IPMC. These EAPs require low voltage, for example 2 to 5 V (Bar-Cohen et al., 2002b). This work only considers the ionic version of EAPs,

the IPMC.

Ionic polymer-metal composites have a sandwich-like structure. Typical IPMCs consist of a Nafion ionic polymer membrane sandwiched between inert metal electrodes, such as platinum, gold, or silver. The conducting-polymer membrane allows the free movement of ions inside of the material and it is required for making IPMCs. The metal electrodes are infiltrated into the ionic polymer membrane through a chemical reduction process to form a metal surface on the membrane that allow the application of voltage to control the IPMC material. Instead of Nafion polymer, IPMCs can also be made using Flemion<sup>TM</sup> membrane (Bar-Cohen et al., 2000). Different polymer membranes and metal electrodes give different properties to IPMCs (Nemat-Nasser and Wu, 2003a).

Normally, when a voltage is applied to the metal electrodes of IPMCs and if the polymer membrane is saturated with ionic fluid, such as water, the electric field causes the ions, in this case cations, to migrate and bring water molecules with them from one side of the composite to the other (see Fig. 2.1). The high concentration of molecules on one side causes the IPMC membrane to swell on this side and shrink on the opposite side. As a result, bending towards the cation electrode side occurs as shown in Fig. 2.1. Therefore, the IPMC actuators require ionic fluid to operate. These actuators can also work in air, but only for a limited time as the ionic fluid, for example water, will evaporate with time. The IPMCs have been shown to work well in a humid environment (Shahinpoor et al., 1998). When the polymer membrane is Nafion, a sudden constant electric field applied to the composite causes it to bend immediately toward the anode electrode. But after some time, the composite relaxes back to its initial state. When the backbone polymer is Flemion, the composite actuator does not exhibit as significant of a relaxation behavior compared to the Nafion type (Nemat-Nasser and Wu, 2003a).

When forced to bend, an IPMC will generate a measurable voltage. The sensing ability is caused by the redistribution of ions in the membrane (Bar-Cohen

et al., 2001).

The advantages of IPMC actuators include: require low driving voltage (between 2 to 5 V), can be easily shaped, provide silent motion, has high displacement-to-mass ratio, and operate in aqueous environments. However, the drawback is they require ionic fluid for operation. Additionally, when water is the polar solvent, their behavior (dynamics) can change with time due to evaporation. Also, IPMCs exhibit structural vibration, relaxation, and nonlinearities (Chen et al., 2005; Bar-Cohen et al., 2000; Kang et al., 2007), which make controlling their motion challenging.

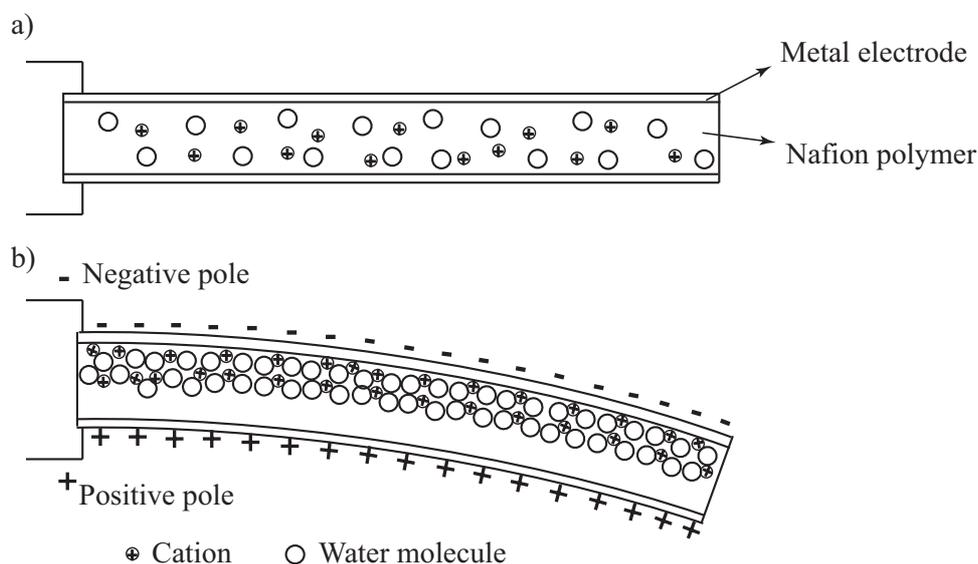


Figure 2.1: Movement of cations with water molecules inside of IPMCs.

## 2.2 History of IPMCs

The main component of IPMC actuators is the conducting-polymer Nafion, which is trademarked by the company Dupont. The chemical structure of Nafion is shown in Fig. 2.2 (Shahinpoor and Kim, 2001). The history of Nafion can be traced back to

the discovery of Teflon, another product of Dupont. In 1938, Dr. Roy J. Plunkett from Dupont's Jackson Laboratory found polytetrafluoroethylene (PTFE), a spontaneously polymerized solid white and waxy substance. The substance was given the registered name Teflon in 1945. This material is inert and slippery, and it contributes to significant advancements in areas including aerospace, communication, electronics, industrial processes, and household applications (Kothera, 2002).

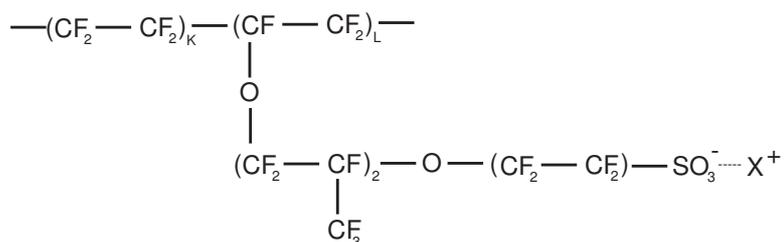


Figure 2.2: Chemical structure of Nafion, which includes fluorocarbons, oxygen, sulfonate groups, and a mobile cation  $X^+$  that can be hydrogen or sodium. The  $K$  is usually 5 to 11 and the  $L$  is usually 1 (Shahinpoor and Kim, 2001).

With further development of Teflon, Walther Grot of Dupont de Nemours discovered sulfonated tetrafluoroethylene copolymer in the late 1960s (Kothera, 2002). Dupont named it Nafion. This new polymer has tetrafluoroethylene (Teflon) as its backbone. It is the first synthetic polymer with ionic properties. Such polymers are called ionomers.

Nafion is based on the Teflon backbone with the acidic sulfonic groups and it exhibits certain desirable characteristics (Kothera, 2002). For example, it has superior conductive properties, which makes it excellent for membrane applications. Also, the pores in the Nafion polymer allow movement of cations but the membrane does not conduct anions or electrons. It is selectively and highly permeable to water, and the degree of hydration of the Nafion polymer directly affects its ion conductivity and overall morphology. It is inert to chemical attack and has a high operating tem-

perature. Nafion can be manufactured with various cationic conductivities (Kothera, 2002).

Because of its excellent chemical, thermal, and mechanical stability, Nafion is used as a proton conductor for proton exchange membrane fuel cells. Also, it has been used in electrochemical devices, chlor-alkali production, metal-ion recovery, water electrolysis, plating, surface treatment of metals, and for making electroactive polymers (Holze and Ahn, 1992). Although Nafion is the key material for making IPMCs, it was not originally invented for electroactive polymer metal composites.

The electromechanical behavior in Nafion was demonstrated in 1990s (Shahinpoor, 1992; Oguro et al., 1992). Researchers found that an imposed electric field can cause the electrophoretic migration of ions inside the hydrated Nafion membrane, and this redistribution of ions in the polymer membrane causes deformation of the polymer membrane. Conversely, Nafion also shows sensing ability, the production of electric charge when the material is deformed (Shahinpoor, 1992).

## **2.3 Applications of IPMCs**

Because of the advantages of the IPMCs, such as low driving voltage, high displacement-to-mass ratio, flexibility, and the ability to work in an aqueous environment, many applications for IPMCs have emerged in the recent years. A brief review is given below.

### **2.3.1 Sensors and Energy Harvesting Devices**

The soft and flexible nature of IPMCs is applied to create tactile sensors in robotics and biomedical devices. One example in robotics is the IPMC tactile sensors for teleoperation. These sensors can be designed into a glove to sense the behavior of the operator's hand for remote control of a slave robot as shown in Fig. 2.3 (Bar-Cohen et al., 2002a). This tactile-sensor based teleoperated system can be used for surgery

or for remote inspection or manufacturing.

The IPMC-based sensors can be used to measure systolic and diastolic blood pressure (BP), pulse rate, and rhythm (Keshavarzi et al., 1999). By installing suitable IPMC sensors on the inner surface of a cuff, both systolic and diastolic BP, pulse rate, and rhythm can be measured. The IPMC sensors can measure the ‘pulse rhythm’ to give a more amplified look at heart irregularities compared to typical pulse rate sensors.

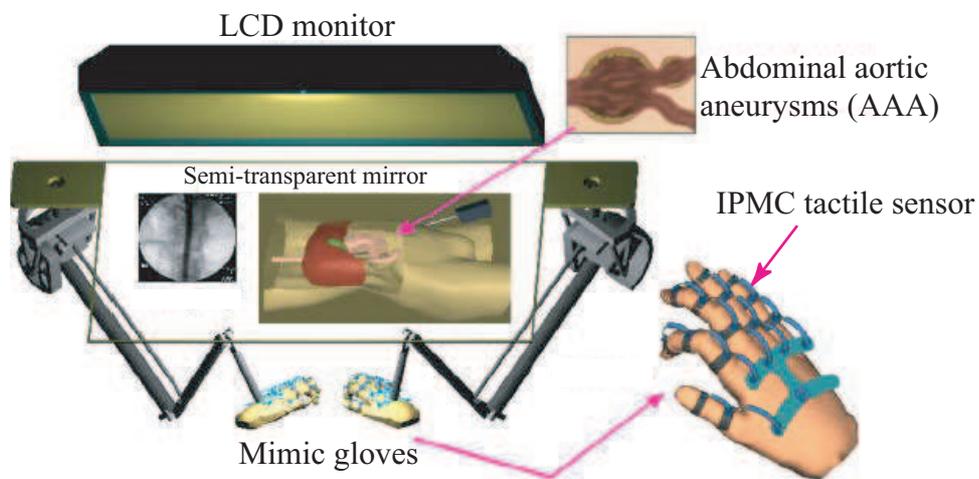


Figure 2.3: An example application of IPMCs as tactile sensor for teleoperation (Bar-Cohen et al., 2002a).

The IPMC material can be applied to create energy harvesting devices (Shahinpoor and Kim, 2004). The power generated is by means of mechanical bending and pressure. An IPMC-based battery was realized by bonding and gluing or laminating thin sheets of paper-thin IPMCs to a flexible or rigid substrate (Shahinpoor and Kim, 2004). Power is generated by the motion of the substrate. The benefit of the IPMC-based batteries is that they are self-powered and can be recharged with moisture.

The IPMC material can also be used as strings in musical instruments. These

IPMC-based strings can generate different frequency signal, which can be translated to different tunes (Shahinpoor and Kim, 2005).

### 2.3.2 Artificial Muscle for Robotics

Ionic polymer-metal composites can be used as artificial muscles in humanoid robots or for biomimetic systems. For example, they can be used to create flapping wings, walking robots, swimming robots, and ventricular or cardiac assistance devices (Nalwa, 2003; Cho et al., 2003; Otis et al., 2003; Lee et al., 2007; Shahinpoor and Kim, 2005; Guo et al., 2006; Lee et al., 2006; Feng and Chen, 2007; Kim et al., 2007). For instance, because IPMCs can be easily cut and shaped, they can be used to mimic bio-inspired forms of locomotion for developing flying robots (Shahinpoor and Kim, 2005; Kim et al., 2007).

Likewise, IPMCs can be used as fins and tails, electrically controlled, to propel aquatic robots in an aqueous environment (Guo et al., 2006). An example swimming fish-like underwater microrobot is shown in Fig. 2.4. The IPMCs were used for the fins and tails to propel the microrobot smoothly in water. This kind of micro-robot is useful in industry for pipe inspection. Microrobots can restrict their work to the affected part and they do not give unnecessary influence on their surroundings. Compared to the traditional mechanical propellers and flippers that replicate undulating motion by means of linkages and other interfacing parts, the wavy motion of IPMC artificial fins and tails has numerous benefits including high efficiency and noiseless propulsion. Also, compared to other active materials, like shape memory alloy actuators, piezoelectric actuators, IPMC actuators provide good response and are generally safer for the human body.

In addition to robotic applications, Shahinpoor and Kim (2005) presented the potential for IPMC artificial muscles to fabricate exo-skeletal human joint for power augmentation. Because IPMCs can bend and generate force, they have the potential to be used as artificial muscles in devices to assist the physically disabled or for

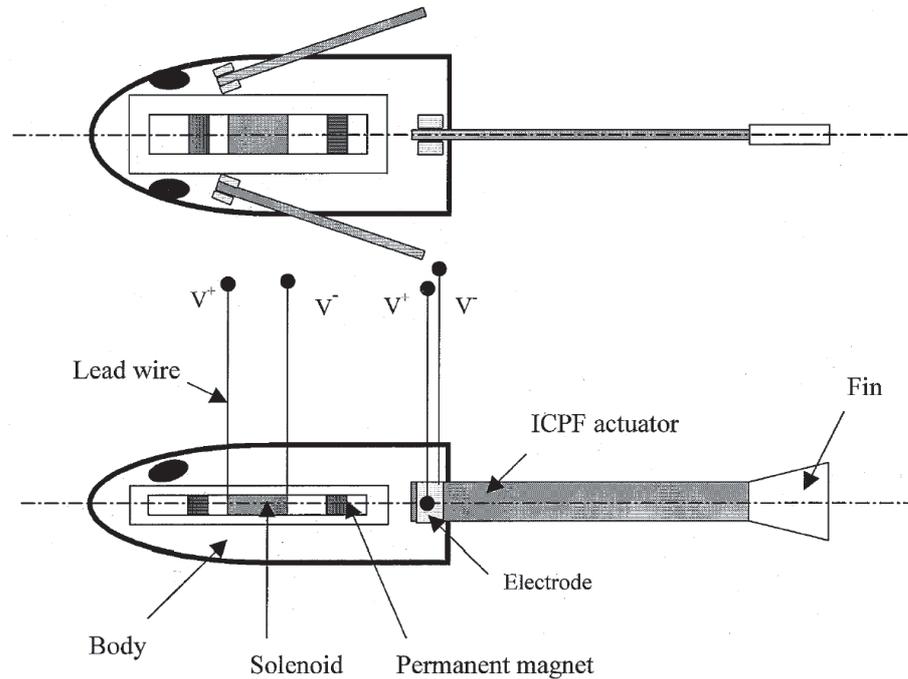


Figure 2.4: An example application of IPMCs as artificial muscle for swimming fish-like underwater microrobots (Guo et al., 2006).

rehabilitation. These exo-skeletal devices, such as joints, can be easily controlled by an operator.

### 2.3.3 Actuators and Positioners

The ability to bend or deform when voltage is applied to IPMCs makes them ideal for actuators and positioners in surgical tools, biomedical devices, and other applications. Some examples are described below.

Micro grippers for laparoscopic surgery can be constructed from IPMC actuators (Feng and Chen, 2007). The laparoscopic surgery is a minimally invasive diagnostic surgical procedure. The procedure involves tiny surgical instruments along

with a luminescent device that is inserted into a human body through tiny incisions, usually 2 to 15 mm long. Currently, the surgical instruments for this kind of surgery are mostly made of steels with predetermined rigid and long shaft structure, such as scissors and graspers. These steel instruments can easily damage the tissue due to improper applied force on the instruments, since it is impossible for the surgeon to directly feel tissue texture and stiffness. Using an IPMC micro gripper instead of a steel surgical tool can limit the damage. The advantage comes from the fact that IPMC's are soft and flexible, yet capable of generating the required level of force.

The IPMC actuators can be used to create a positioning system to mount cell samples (Shahinpoor and Kim, 2005). Such a positioner is created using strips configured in a three-dimensional hollow rectangular tube structure. This tube positioner consists of four independent strips electrically insulated from each other. The speed of the movement of the positioner can also be controlled by sending different frequency signal.

Ionic polymer-metal composite actuators can also be implemented as micropumps (Lee et al., 2007; Cho et al., 2003). Because IPMCs can be operated with low input voltages and can produce large stroke volumes along with controllable flow rates, IPMC is a promising material for micropump applications. Peristaltic pumps can be made from tubular sections of the IPMC membranes with electrodes placed in the appropriate locations. The operation of the volume of the pump can be realized by using a controller to regulate the voltage and the frequency of the input signal (Shahinpoor and Kim, 2005). Compared with other available technologies, using IPMCs to manufacture micropump is convenient.

Another example of IPMC for actuators is in endoscopes (Yoon et al., 2007). An IPMC actuator was used to position an optical fibers for imaging in the human body. The high displacement and flexibility of IPMC actuators give the scanners of the endoscope a large field of view (Yoon et al., 2007). Compared to other active-material actuators, such as SMA actuators and electrostatic EAPs, IPMCs require

lower driving voltage with the benefit of relatively large deformation. In addition, the flexibility of the material does not restrict the motion of the scanner.

In the ventricular or cardiac assisted devices, IPMCs are used as sensors, artificial muscles, and actuators. Artificial ventricular assistant type muscles can be made from IPMCs to help patients with heart abnormalities. Heart compression or arrhythmia devices can be used to support cardiac muscle functions (Shahinpoor and Kim, 2005). The heart compression device can be implanted near the patient's heart and partly sutured to the heart without contacting or interfering with the internal blood circulates. Therefore, thrombosis and similar complications can be avoided, which are common in current artificial heart designs (Shahinpoor and Kim, 2005).

#### **2.4 Challenges with Using IPMCs for Positioning**

There are four main challenges with using IPMCs as actuators: (1) stress relaxation, (2) dynamic effects, (3) nonlinearity, and (4) time-varying behaviors.

When a constant input voltage is applied to an IPMC actuator, it responds by bending to a maximum position. Upon reaching the maximum, it slowly begins to relax towards its original state. The relaxation could be caused by the subsequent water back-flux after excess concentration (Bar-Cohen et al., 2002b), as described in Section 2.1. The behavior is a slow response and makes controlling the positioning of IPMCs in open-loop difficult. Figure 2.5 shows an example of the measured relaxation behavior in an IPMC.

At high frequencies the dynamic effects cause oscillations in the response. The movement-induced oscillations cause significant tracking error. For example, when an IPMC actuator is used to position an optical fiber for endoscopy at high speed (Yoon et al., 2007), movement induced vibrations can cause image distortion. The distortion is a result of the discrepancy between the fiber's actual and desired location at the time information is acquired. The induced vibrations are caused

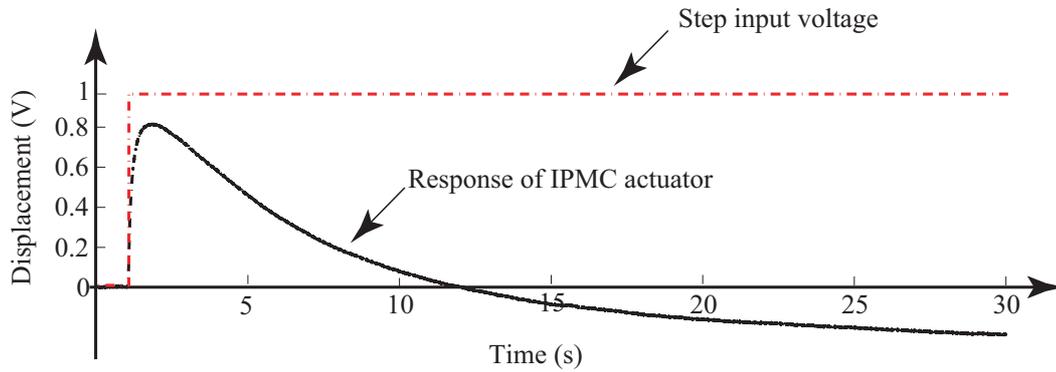


Figure 2.5: An experimentally measured relaxation behavior in an IPMC, where the dashed line is a 1 V step input applied to the IPMC actuator and the solid line is the displacement response of the IPMC actuator.

by exciting the resonant modes of the IPMC actuator. In other applications such as IPMC-based mechanical grippers, robotic arms and fingers, and artificial ventricular muscles, movement-induced vibrations can severely limit performance. Figure 2.6 shows an example of the dynamic effects in an IPMC. At high frequencies, such as 18 Hz, the actual response differs significantly from the desired response.

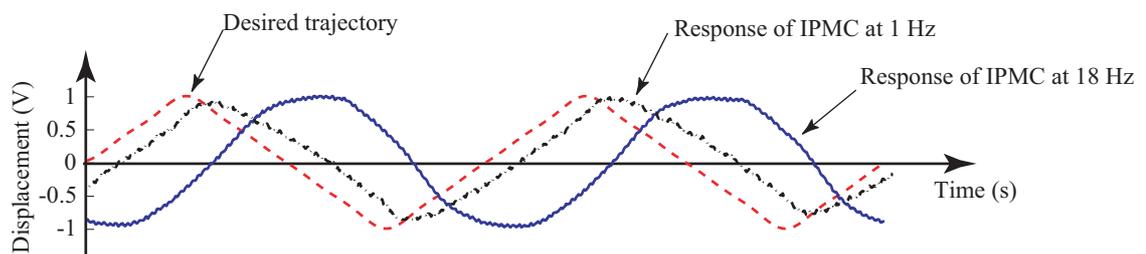


Figure 2.6: The dynamic effects in an IPMC actuator showing the mismatch between the desired response and the measured response at different frequencies.

Nonlinearity and changes in the system parameters also bring challenges in the application of IPMC actuators. The effects come about when IPMC actuators are used in air because the evaporation of the ionic fluid caused by electrolysis and Joule effect leads to changes in the system's parameter. The time-varying behavior makes precision control challenging.

The challenges mentioned above causes significant tracking error. This thesis focuses on minimizing the effects of dynamics, nonlinearities, and relaxation using an integrated feedforward and feedback controller.

## Chapter 3: Review of Control Methods for IPMC Actuators

Due to the challenges of using IPMCs for actuators and positioners, feedback and feedforward controllers have been studied for IPMCs to improve performance. This chapter reviews the feedback and feedforward control methods for controlling IPMC actuators. Table 3.1 summarizes a sample of the different techniques discussed in this chapter.

### 3.1 Feedback Control Approaches

Feedback control is the most widely used controller for IPMC actuators. The block diagram of a standard feedback controller is depicted in Fig. 3.1. The basic idea of feedback control is to use the error between the measured output and desired output as an input signal to a controller to correct for deficiencies in the system's behavior (Franklin et al., 2006).

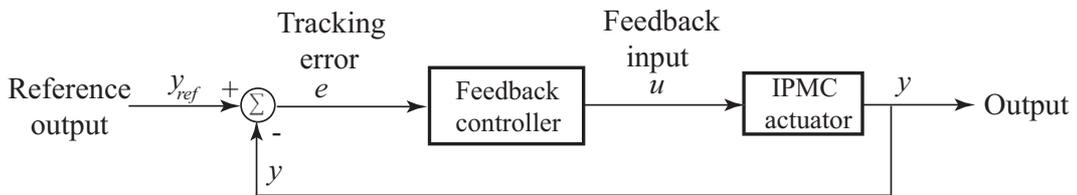


Figure 3.1: Block diagram of a feedback control system, where  $y_{ref}$  is the reference output signal,  $y$  is the measured output, and  $e$  is the tracking error, that is  $e = y_{ref} - y$ .

Specifically, Richardson et al. (2003) integrated a proportional-integral-derivative (PID) controller with a force/position impedance controller for controlling an IPMC actuator. The PID controller was used to achieve a quick response and small

steady-state error. Particularly, the integral term reduced the relaxation behavior compared to the open-loop response, whereas the force/position impedance controller accounted for nonlinear effects. The proposed impedance controller focused on controlling the dynamic relationship between force and displacement. According to their work, the impedance controller was suitable for systems with nonlinear characteristics because of the inertial element that can mimic the physical properties of the system. The results showed that the response time and steady-state error were decreased; however, the integrated controller required a force sensor and a high-gain position controller. The position controller provided robustness to external force disturbances.

Optimal feedback control has also been studied and applied to IPMC actuators. For example, a linear quadratic regulator (LQR) based feedback controller was used for positioning an IPMC actuator (Mallavarapu et al., 2001). The researchers used the controller to minimize the settling time and to constrain the control voltage to less than 2 V. An empirical model was used in the LQR method. The results showed the settling time was reduced from 7 s to 0.95 s, and the control voltage was constrained to less than 2.3 V. However, large overshoot and oscillations were observed.

Lavu et al. (2005) used adaptive intelligent control for IPMC actuators to track a pre-defined path and to adapt to the variations of the IPMC actuator behavior caused by relative humidity. In this work, the IPMC actuator was modeled using a system identification method. The controller was implemented using the pole-placement technique in state-space feedback form. The design was optimized by calculating the cost function to minimize the overshoot, settling time, tracking error, and energy. Their simulation results showed that the root-mean-square error of the tracking error signal was 28.66%.

An  $H_\infty$ -based (Zhou and Doyle, 1998) feedback controller was implemented to overcome the uncertainties and non-repeatability of an IPMC actuator (Kang et al., 2007). There were three types of robust feedback controllers designed:  $H_\infty$ ,  $H_\infty$  with

loop-shaping, and  $\mu$ -synthesis. Their simulation results and discussion showed that these controllers enhanced the performance of the IPMC actuator. It was also shown that the controller achieved faster response and lower overshoot.

In summary, feedback control for IPMC actuators achieves quick response time, small steady-state error, and it can minimize tracking error caused by the relaxation and hysteresis effect. Although feedback control is relatively straightforward to implement, it is often bandwidth-limited (Leang and Devasia, 2007).

### 3.2 Feedforward Control Approaches

In contrast to closed-loop control (feedback), feedforward control is an open-loop process which can be used for IPMCs. A basic feedforward control system is shown in Fig. 3.2. A feedforward controller works by generating a feedforward control input that anticipates the system's behavior and external disturbances. In feedforward control, a desired output signal  $y_d$  is applied to the feedforward controller, and a feedforward input  $u_{ff}$  is generated to control the IPMC actuator system. If the feedforward input  $u_{ff}$  is computed correctly, then the measured output  $y$  equals the desired output  $y_d$ . Feedforward control has been extensively studied for piezoactuators, but limited work has been done for IPMC actuators.

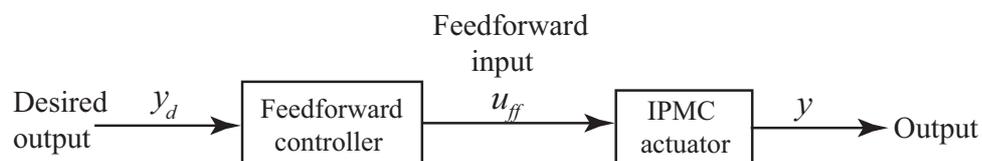


Figure 3.2: Block diagram of a feedforward control system, where  $y_d$  is the desired output signal,  $y$  is the measured output, and  $u_{ff}$  is the feedforward input applied to the system.

Chen et al. (2005) used an inversion-based feedforward controller to account for hysteresis-caused positioning error in an IPMC actuator. The approach was applied to the quasi-static case, *i.e.*, the dynamics were ignored. The feedforward controller was designed by inverting the Preisach hysteresis model for the IPMC actuator. The experimental results showed that the IPMC actuator with using the proposed controller had smaller positioning error compared to a controller based on a model without hysteresis. However, quasi-static inversion-based feedforward controller did not fully address the tracking error caused by nonlinearity in IPMC actuators. This is because the controller was designed based on a static hysteresis model. Additionally, because the dynamic effects were ignored, the precision was limited to quasi-static movements.

Yamakita et al. (2006) used an  $H_\infty$ -based feedforward controller to verify the performance of IPMC actuators doped with different cations like  $Na^+$  and  $TEA^+$ . This control system consisted of an  $H_\infty$  feedforward controller and a third-order Butterworth low-pass filter. The controller was designed based on a linear time invariant (LTI) model of the IPMC actuator. The simulation results showed that the IPMC actuator doped with  $Na^+$  and  $TEA^+$  needed smaller control voltages to achieve the same tracking performance as the  $Na^+$  or  $TEA^+$  doped IPMC actuators. The controller decreased the steady-state error for positioning of IPMC actuators.

### 3.3 Integrated Feedback and Feedforward Control

Feedback and feedforward control has been combined to control IPMC actuators. For instance, Fang et al. (2007) used a PID feedback controller with a nonlinear feedforward compensator to control the bending motion of an IPMC actuator that was fabricated by a new approach. The error due to the relaxation and hysteresis behavior in the IPMC actuator was reduced by this integrated controller. The controller design used a fourth-order linear time-invariant model with a nonlinear gain and a time delay element. Their results showed that the overshoot decreased from 30% to 4.2%, and

the steady-state error was lowered from 15% to 4%. Unfortunately, the rise-time increased from 0.084 to 0.325 s. Additionally, the closed-loop system was operated at low frequencies, such as 0.1 Hz. They also concluded that the PID controller worked well at low frequency and that it overcame the drift problem. On the other hand, the nonlinear compensator accounted mostly for the nonlinear behavior of the actuator at low frequencies.

Kaneda et al. (2003) applied a linear quadratic (LQ) based optimal proportional controller integrated with a nonlinear compensator as a feedforward controller for positioning control of an IPMC actuator. The IPMC actuator was modeled by the Hammerstein method, where a LTI system was put in series with a static nonlinear term. The proportional controller was designed based on LQ optimal servo theory and the nonlinear compensator was generated by inverting the static nonlinearity. Their results showed that the steady state error decreased, but the oscillation increased.

### 3.4 Summary

Feedback controllers can be used to minimize positioning error caused by relaxation, dynamic effects, and nonlinearity in IPMC actuators. In particular, feedback control reduces the response time and steady-state error. Although feedback control is relatively straightforward to implement, it is often bandwidth-limited (Leang and Devasia, 2007). Feedforward control can account for hysteresis during quasi-static movements. However, inversion-based feedforward control for IPMC has not been studied.

Table 3.1: Review of control methods for IPMC actuators.

	<b>Control methods</b>	<b>Groups</b>	<b>Advantages</b>	<b>Disadvantages</b>
	LQR based feedback control	Mallavarapu et al., 2001	Minimized settling time & error	Caused large overshoot & oscillations
	Adaptive control	Mallavarapu et al., 2001	Decreased tracking error & added robustness to humidity changes	Applied in simulation
<b>Feedback control approaches</b>	PID control + impedance control	Richardson et al., 2003	Achieved quick response & small steady-state error	Required high gain position controller
	$H_\infty$ based feedback control	Kang et al., 2007	Achieved fast response & low overshoot	Applied in simulation
<b>Feedforward control approaches</b>	Static hysteresis model inversion feedforward control	Chen et al., 2005	Minimized quasi-state positioning error	Did not consider the dynamic effects
	$H_\infty$ based feedforward control	Yamakita et al., 2006	Decreased steady-state error	Applied in simulation
<b>Feedback + Feedforward control</b>	LQ based feedback control with a nonlinear compensator	Kaneda et al., 2003	Decreased steady-state error	Increased oscillations
	PID feedback control assisted with a nonlinear compensator	Fang et al., 2007	Decreased overshoot & steady-state error	Increased rise time & operated in low frequency

\*The review shows that dynamic effects were not specifically addressed, which is the objective of this thesis.

## Chapter 4: Integrated Feedforward and Feedback Controller Design

This chapter describes the design of an integrated feedforward and feedback controller to compensate for dynamic effects, nonlinearity, and relaxation behavior in IPMCs. The feedforward controller is designed for high speed movements by accounting for the induced structural vibration. A PI feedback controller is used to address the nonlinearity, relaxation, and unmodelled dynamics. The integrated control system is shown in Fig. 4.1. The controller design is described in detail in this chapter.

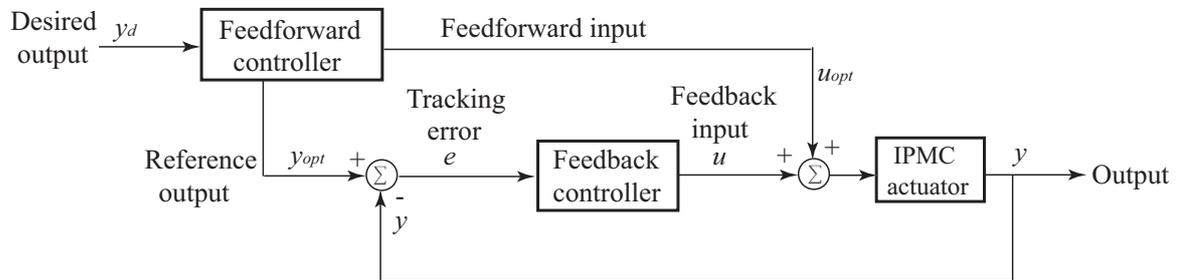


Figure 4.1: A block diagram of an integrated feedforward and feedback controller. A desired output trajectory  $y_d$  is applied to the feedforward controller. The feedforward controller generates a feedforward input  $u_{opt}$  as well as a reference output  $y_{opt}$ . The feedforward input  $u_{opt}$  is applied directly to the IPMC actuator. The output of the IPMC actuator is  $y$  and the tracking error  $e = y_{opt} - y$  is used as an input to the feedback controller to generate a feedback input  $u$ .

#### 4.1 Inversion-Based Feedforward Controller Design

The inversion model-based feedforward control method exploits known information about the system to compensate for effects such as dynamics (Devasia et al., 1996; Croft et al., 2001) and/or hysteresis (Chen et al., 2005). The block diagram for the inversion-based feedforward controller is shown in Fig. 4.2. The basic idea of the inversion-based feedforward method is to invert a model of the system to determine an input that results in achieving a desired trajectory. This approach essentially anticipates the deficit performance of the system. The feedforward technique has been extensively studied for piezoactuators (Croft et al., 2001; Tien et al., 2004) and flexible structures (Dewey et al., 1998). Unlike feedback control, the inversion model-based feedforward approach does not require a sensor, but instead depends on knowing to some degree the input-to-output relationship of the system to be controlled.

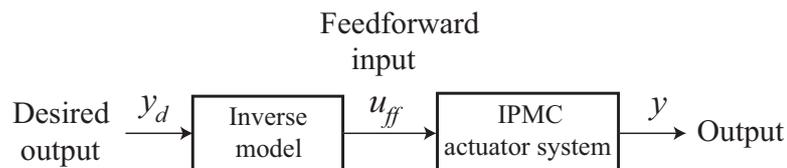


Figure 4.2: Block diagram of the inversion-based feedforward controller. A desired output trajectory  $y_d$  is applied to the inverse model. The inverse model generates a feedforward input  $u_{ff}$  which is then applied to the IPMC actuator. The output of the IPMC is  $y$ .

Let  $G(j\omega)$  be the Fourier transform of the system which relates the input  $U(j\omega)$  to the output  $Y(j\omega)$ . If the model  $G(j\omega)$  of the system is known, then for a given desired trajectory  $Y_d(j\omega)$ , the feedforward input is

$$U_{ff}(j\omega) = G^{-1}(j\omega)Y_d(j\omega). \quad (4.1)$$

where the  $G^{-1}(j\omega)$  is the system inverse (Bayo, 1987; Devasia et al., 1996). It is noted that the desired input  $Y_d(j\omega)$  must be given for the implementation of this inversion-

based feedforward control method, as described in Eq. (4.1). For nonminimum phase systems, a bounded solution to Eq. (4.1) can be determined; however, the inputs may be noncausal (Bayo, 1987; Devasia et al., 1996).

The time domain solution of the inverse feedforward input  $u_{ff}(t)$  is found by the inverse Fourier transform. Although the implementation of Eq. (4.1) is straightforward, the resulting feedforward input may be excessively large, especially if the system has lightly-damped system zeros. The excessively large inputs (*i.e.*, voltage) can damage the IPMC actuator. Additionally, large model uncertainties around the resonant peaks and/or lightly-damped zeros may cause significant error in computing the feedforward input (Devasia, 2002). As a result, the inversion technique Eq. (4.1) may produce unacceptably large error at or around these frequencies.

The issues with modeling errors and dynamics variation can be addressed by the optimal inversion technique (Dewey et al., 1998; Leang and Devasia, 2007). Specifically, an optimized feedforward input  $u_{opt}$  is obtained by minimizing the quadratic cost function (Dewey et al., 1998),

$$J(U) = \int_{-\infty}^{\infty} [U^*(j\omega)R(j\omega)U(j\omega) + e^*(j\omega)Q(j\omega)e(j\omega)] d\omega, \quad (4.2)$$

where  $J(U)$  denotes the system energy, ‘\*’ denotes the complex conjugate transpose,  $U(j\omega)$  is the feedforward input to the IPMC actuator,  $e(j\omega) = Y(j\omega) - Y_d(j\omega)$  is the tracking error,  $Y(j\omega)$  is the output,  $Y_d(j\omega)$  is the desired output, and  $R(j\omega)$  and  $Q(j\omega)$  are non-negative, frequency-dependent real-value weights on the input energy and the tracking error, respectively.

The optimal input that minimizes the cost function  $J(U)$  is found as follows. First, the cost function  $J(U)$  will have minimum value when

$$\frac{dJ(U)}{dU} = 0, \quad (4.3)$$

and

$$\frac{d^2 J(U)}{dU^2} > 0. \quad (4.4)$$

Taking the first and second derivative of  $J(U)$  with respect to  $U$  gives

$$\frac{dJ(U)}{dU} = \int_{-\infty}^{\infty} \left[ \frac{dU^*(j\omega)R(j\omega)U(j\omega)}{dU(j\omega)} + \frac{de^*(j\omega)Q(j\omega)e(j\omega)}{dU(j\omega)} \right] d\omega, \quad (4.5)$$

and

$$\frac{d^2J(U)}{dU^2} = \int_{-\infty}^{\infty} \left[ \frac{d^2U^*(j\omega)R(j\omega)U(j\omega)}{dU(j\omega)^2} + \frac{d^2e^*(j\omega)Q(j\omega)e(j\omega)}{dU(j\omega)^2} \right] d\omega. \quad (4.6)$$

Next, substituting  $Y(j\omega) = G(j\omega)U(j\omega)$  and  $e(j\omega) = Y(j\omega) - Y_d(j\omega)$  into the Eqs. (4.5) and (4.6) gives

$$\begin{aligned} \frac{dJ(U)}{dU} &= \int_{-\infty}^{\infty} \{2R(j\omega)U(j\omega) + 2G^*(j\omega)Q(j\omega)[G(j\omega)U(j\omega) - Y_d(j\omega)]\} d\omega \\ &= \int_{-\infty}^{\infty} \{2[R(j\omega) + G^*(j\omega)Q(j\omega)(G(j\omega))]U(j\omega) \\ &\quad - 2G^*(j\omega)Q(j\omega)Y_d(j\omega)\} d\omega, \end{aligned} \quad (4.7)$$

and

$$\begin{aligned} \frac{d^2J(U)}{dU^2} &= \int_{-\infty}^{\infty} \left[ 2 \frac{dR(j\omega)U(j\omega)}{dU(j\omega)} + 2 \frac{dG^*(j\omega)Q(j\omega)[G(j\omega)U(j\omega) - Y_d(j\omega)]}{dU(j\omega)} \right] d\omega \\ &= \int_{-\infty}^{\infty} [2R(j\omega) + 2G^*(j\omega)Q(j\omega)G(j\omega)] d\omega. \end{aligned} \quad (4.8)$$

Noting that  $dJ(U)/dU = 0$  and  $d^2J(U)/dU^2 > 0$ , the above two equations simplify to:

$$2[R(j\omega) + G^*(j\omega)Q(j\omega)(G(j\omega))]U(j\omega) - 2G^*(j\omega)Q(j\omega)Y_d(j\omega) = 0. \quad (4.9)$$

Because the  $R(j\omega)$  and  $Q(j\omega)$  weights are non-negative,  $G^*(j\omega)Q(j\omega)G(j\omega)$  is also non-negative, then

$$2R(j\omega) + 2G^*(j\omega)Q(j\omega)G(j\omega) > 0. \quad (4.10)$$

Finally, Eq. (4.9) simplifies to

$$[R(j\omega) + G^*(j\omega)Q(j\omega)(G(j\omega))]U(j\omega) = G^*(j\omega)Q(j\omega)Y_d(j\omega), \quad (4.11)$$

which implies the optimal feedforward input is

$$U_{opt}(j\omega) = \left[ \frac{G^*(j\omega)Q(j\omega)}{R(j\omega) + G^*(j\omega)Q(j\omega)G(j\omega)} \right] Y_d(j\omega). \quad (4.12)$$

When  $U_{opt}(j\omega)$  is applied to the IPMC actuator system given by  $G(j\omega)$ , the following modified output is tracked,

$$\begin{aligned} Y_{opt}(j\omega) &= G(j\omega)u_{opt}(j\omega), \\ &= G(j\omega) \left[ \frac{G^*(j\omega)Q(j\omega)}{R(j\omega) + G^*(j\omega)Q(j\omega)G(j\omega)} \right] Y_d(j\omega), \\ &\triangleq G_f(j\omega)Y_d(j\omega). \end{aligned} \quad (4.13)$$

Therefore,  $G_f(j\omega)$  is a filter that modifies the desired trajectory  $Y_d(j\omega)$  based on the  $R(j\omega)$  and  $Q(j\omega)$  weights. So in the experiment, the output signal tracks trajectory  $y_{opt}(t)$  instead of  $y_d(t)$  when  $u_{opt}(t)$  is applied.

The  $R(j\omega)$  and  $Q(j\omega)$  weights can be designed to take into account the input magnitude and the model uncertainties over certain frequency ranges. Generally speaking, the input energy weight  $R(j\omega)$  should be chosen much larger than the tracking error weight  $Q(j\omega)$  at frequencies (a) where there are large model uncertainties and/or (b) around lightly-damped zeros.

In addition to the vibrational dynamics, the behavior of IPMCs may include nonlinearities, like hysteresis (Chen et al., 2005), and relaxation (Nemat-Nasser and Wu, 2003b). These additional effects were not considered in this feedforward design. Particularly, the electro-mechanical model  $G(j\omega)$  is found over a relatively-high frequency range. By doing this, the relaxation behavior that occurs over long periods of time was ignored. Likewise, the effect of hysteresis was not considered. The model  $G(j\omega)$  was obtained over the IPMC actuator's linear range, *i.e.*, by assuming that over relatively small range motion hysteresis effect was negligible (Croft et al., 2001). The implementation of this feedforward method is described in Section 5.4.

To account for unmodeled effects and nonlinearities, the optimal inverse input  $u_{opt}(t)$  and the modified output  $y_{opt}(t)$  were integrated with a feedback controller

to form an integrated controller (see Fig. 4.1). Therefore, the feedback controller provided robustness to variations in the dynamics and accounted for the nonlinearities and relaxation effects. The modified trajectory  $y_{opt}(t)$  became the reference trajectory to the feedback system, *i.e.*,  $y_{ref}(t) = y_{opt}(t)$ .

## 4.2 Feedback Controller Design

A PI feedback controller (see Fig. 3.1) was used to minimize the effects of nonlinearities, relaxation, and other unmodeled dynamics that were not accounted for by the feedforward controller. The transfer function of the PI feedback controller is

$$C(s) = \frac{K_p s + K_i}{s}, \quad (4.14)$$

where  $K_p$  is the proportional gain and  $K_i$  represents the integral gain. The proportional term of the controller can reduce the rise time and decrease the steady-state error, and the integral term has the effect of eliminating the steady-state error. By carefully selecting the gains  $K_p$  and  $K_i$ , the PI controller can improve the performance compared to the open-loop system.

In the thesis, the PI controller gains  $K_p$  and  $K_i$  were tuned experimentally, where the Ziegler-Nichols tuning method (Franklin et al., 2005) was used as starting point. The details of Ziegler-Nichols tuning method is described in Section 5.4.3.

## 4.3 Integrated Feedforward and Feedback Controller Design

An integrated feedforward and feedback controller was achieved by combining the inversion-based feedforward controller and the PI feedback controller as shown in Fig. 4.1. The implementation of this integrated controller is described in Section 5.4. The experimental results of the inversion-based feedforward controller, the PI feedback controller, and the integrated feedforward and feedback controller are described in Chapter 6. Before the implementation of these controllers, IPMCs were fabricated for all experiments.

## Chapter 5: The Experimental IPMC Actuator System

This chapter describes the IPMC fabrication process, the experimental IPMC system, and the controller implementations.

### 5.1 Fabrication of IPMCs

The IPMC actuators used in all experiments were custom-fabricated in the lab. Alternatively, they can be purchased from Environmental Robots Inc. ([www.environmental-robots.com](http://www.environmental-robots.com)). However, commercially available IPMC actuators are expensive and limited to select thicknesses. The fabrication of IPMCs consists of two tasks which includes casting Nafion polymer membrane and plating the platinum metal electrodes on the membrane surfaces. The details of the fabrication are described below.

#### 5.1.1 Casting Nafion Polymer Membranes

A Nafion polymer membrane is required for making IPMCs. Nafion polymer membranes are available from Dupont with thickness of  $25.4\ \mu\text{m}$  (1 mil),  $50.8\ \mu\text{m}$  (2 mil),  $88.9\ \mu\text{m}$  (3.5 mil),  $127\ \mu\text{m}$  (5 mil),  $177.8\ \mu\text{m}$  (7 mil) and  $254\ \mu\text{m}$  (10 mil), but thicker membranes are not commercially available. The casting method (Kim and Shahinpoor, 2002) was used to create thicker membranes.

The liquid Nafion solution used in the casting process was Nafion 117 (15% Nafion by weight). It is noted that 1 g of such Nafion solution can form a 0.15 g membrane. Figure 5.1 shows the solution purchased from Ion Power ([www.ion-power.com](http://www.ion-power.com)). The casting process is described in Fig. 5.2 and also outlined in the following steps:

**Step 1:** The liquid Nafion solution was poured into a Pyrex glass mold to dry in open air. The liquid was poured carefully to avoid forming bubbles. By choosing



Figure 5.1: The chemicals used to fabricate IPMC actuators, where the Nafion liquid (Ion Power, type 117) was used to cast Nafion membranes, and the other chemicals, including deionized water, 1 molar  $H_2SO_4$ ,  $Pt(NH_3)_4Cl_2$  powders, and  $NaBH_4$  granules, were used for plating electrodes on the surfaces of the Nafion membranes.

different molds, different shapes can be achieved. In this experiment, two molds were used. One was a circular mold with diameter of 55.5 mm, and the other was a rectangular mold with dimension of 53 mm  $\times$  146 mm.

**Step 2:** The liquid Nafion in the mold was left to dry in air for two days.

**Step 3:** The resulting membrane was carefully released from the mold to avoid damaging the membrane. Then the edge of the membrane was trimmed to remove

edge effects.

**Step 4:** Afterwards, the membrane underwent thermal treatment. In particular, the membrane was heated at  $140^{\circ}\text{C}$  under light pressure of  $1.24 \times 10^4$  Pa (1.798 psi) for 30 min. The heat treatment made the membrane stiffer.

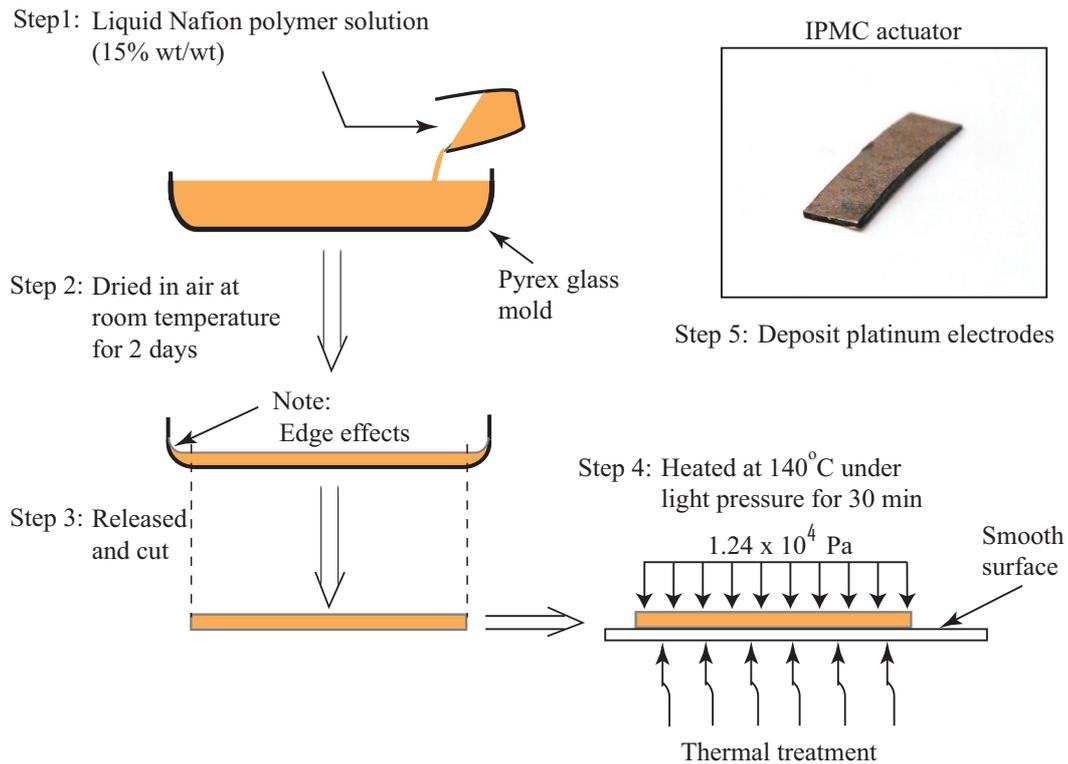


Figure 5.2: Steps to fabricate IPMCs. Steps 1-4: The process for casting the Nafion polymer membrane from liquid Nafion solution. Step 5: A sample of the custom-made IPMCs after depositing platinum electrodes.

Two different thicknesses were created. One was  $380 \mu\text{m}$  thick and the other was approximately 1 mm thick. Figure 5.3(a) and (c) show a sample the custom-casted Nafion membrane compared to a commercially available sample in Fig. 5.3(b).

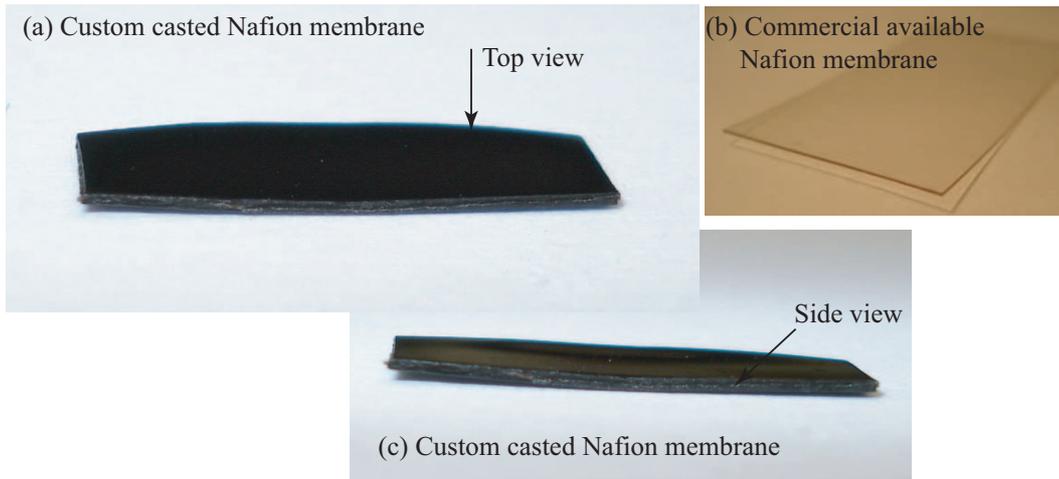


Figure 5.3: (a) and (c) A sample of the custom-made Nafion membrane compared to (b) a commercially made Nafion membrane.

### 5.1.2 Plating Platinum Electrodes via Chemical Reduction Process

Electrode plating was the last step of the fabrication process — Step 5 in Fig. 5.2. This process is essentially a chemical reduction or electro-plating process. In this thesis work, the platinum metal electrode was plated through a chemical reduction process, which was a complex and multi-step process. Finally, two types of IPMC actuators were made: Types I which were based on the commercially available Nafion membranes from Dupont with thickness of  $177.8 \mu\text{m}$  (7 mil) and  $254 \mu\text{m}$  (10 mil), and Types II which were based on thicker membranes custom-made as described above with thickness of approximately  $380 \mu\text{m}$  and 1 mm.

Platinum electrodes were electro-chemically deposited on the surfaces of the Nafion membranes. The precious metal was used because of the acidic nature of the ionic membrane. Additionally, the metal does not oxidize when subjected to water or applied voltage – oxidation corrodes the metal and reduces the conductivity of the IPMC. The electrode plating began with material preparation, then the treating

process.

The materials used included: Nafion membranes (commercially available from Dupont and custom-made membranes), tetramine platinum chloride ( $Pt(NH_3)_4Cl_2$ ), sodium borohydride ( $NaBH_4$ ), 1 molar sulfuric acid ( $H_2SO_4$ ), and deionized water. These materials are shown in Fig. 5.1.

A tetramine platinum chloride ( $Pt(NH_3)_4Cl_2$ ) solution was prepared by mixing tetramine platinum chloride powder with deionized water. The solution had a concentration of 2 mg of platinum in every ml of solution. A 1% (wt/wt) sodium borohydride ( $NaBH_4$ ) solution was also created by mixing deionized water with sodium borohydride powder. The electrode plating steps are:

**Step 1:** First, the Nafion membranes were cleaned in an ultrasonic bath with deionized water for 30 minutes.

**Step 2:** Then, the pieces were boiled in 1 molar sulfuric acid ( $H_2SO_4$ ) for an additional 30 minutes to remove impurities and ions.

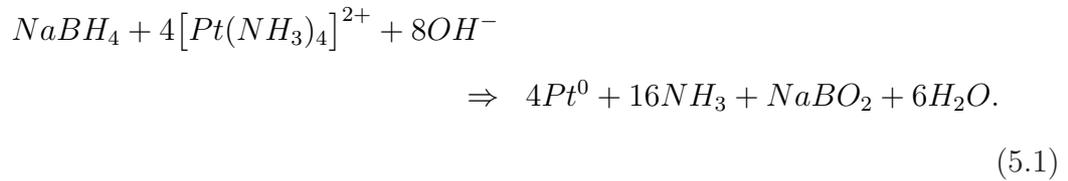
**Step 3:** Again, the membranes were removed from the acid solution and boiled in deionized water for another 30 minutes. This process further cleaned the membranes and saturated them with water.

**Step 4:** After the cleaning process, the membranes were soaked in the  $Pt(NH_3)_4Cl_2$  solution for approximately 16 hours at room temperature with occasional stirring. The amount of the solution used in this step met the minimum requirement of 3 mg of platinum for every  $cm^2$  of the surface area of the Nafion membrane (K.Oguro, 1991). For example, no less than 90 ml of  $Pt(NH_3)_4Cl_2$  solution was used for  $60\text{ cm}^2$  Nafion membrane. Excess amount of the platinum solution is encouraged.

**Step 5:** Finally, the platinum layer was formed by the chemical reaction between

$Pt(NH_3)_4Cl_2$  and the reducing agent  $NaBH_4$  (Kim and Shahinpoor, 2003).

The reaction is



First, the membranes were washed with deionized water. Then then they were soaked vertically in deionized water in test tubes at 40°C. At this temperature, 4 ml of sodium borohydride solution for each membrane sample with 60 cm<sup>2</sup> surface areas was added every 30 minutes for 7 times. During this period, the temperature was maintained at 40°C, and the quantity of the reagent was kept in proportion to the area of the sample. Afterwards, the temperature of the solution was gently raised to 60°C, and then 40 ml of the solution was added for each 60 cm<sup>2</sup> sample. After treating at 60°C for two hours, the samples (now with a platinum electrode layer) were taken out and washed in deionized water.

One layer of platinum was created by going through the above Steps 1-5. More layers were created by repeating the steps described above. For the experimental IPMCs in this thesis, five layers of platinum were created. A thicker platinum layer lowers the resistance of the electrodes (Lee et al., 2006), and lower surface electrode resistance enhances the performance of IPMC actuators (Shahinpoor and Kim, 2000; Punning et al., 2007). The samples of the Type I and Type II IPMCs are shown in Fig. 5.4. The thickness of the platinum metal electrode layer was 35 μm.

## 5.2 The Experimental System

The experimental system is presented in this section. The discussion includes an overview of the experimental fixture, the laser displacement sensor and its calibration,

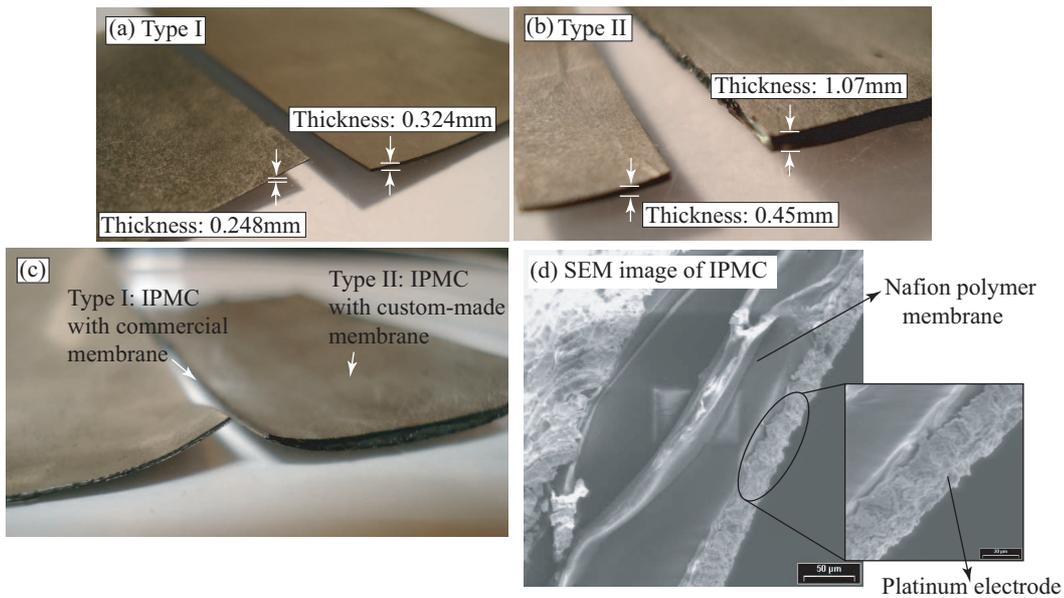


Figure 5.4: Photographs of the IPMCs made from commercial and custom-made Nafion membranes. (a) Type I IPMCs were made from two commercially available Nafion membranes with different thickness. The thicknesses of these two samples were  $248 \mu\text{m}$  and  $324 \mu\text{m}$ , and the thickness of the platinum electrodes layers on each side of the IPMCs was proximate  $35 \mu\text{m}$ . (b) Type II IPMCs were made from two custom-made Nafion membranes with different thickness. The thicknesses of these two samples were  $450 \mu\text{m}$  and  $1.07 \text{mm}$ , and the thickness of the platinum electrodes layers on each side of the IPMCs was also proximate  $35 \mu\text{m}$ . (c) Comparison between the Type I IPMCs and the Type II IPMCs. (d) SEM images of the custom-made IPMCs that show the Nafion polymer membrane and the platinum electrodes layers.

the design of a voltage amplifier and filter circuits, and the performance testing of the custom-made IPMC actuators.

### 5.2.1 Experimental Setup

To apply the controller for the precise positioning of IPMC actuators, an experimental setup was built as shown in Fig. 5.5. The apparatus included a custom-designed stainless steel fixture for clamping the IPMC actuator as shown in Fig. 5.5(a) and (b), and a laser displacement sensor which was used to measure the displacement of

the IPMC actuator. The system also contained a custom-designed voltage amplifier circuit for driving the actuator, a filter circuit for processing the sensor output signal, and a desktop computer with 12-bit digital-to-analog converter board (DAQ/ADC). The DAQ/ADC board was used to implement the feedforward and feedback controllers as well as to collect output signal from the laser displacement sensor. The closed-loop bandwidth of the computer and data acquisition system is 20 kHz.

### 5.2.2 Sensor Calibration

The sensor used in the experiment was a noncontact reflective laser displacement sensor (SUNX micro laser sensor LM10). The model number is ANR12511 and the sensor controller model number is ANR5232. This noncontact laser displacement sensor consists of a coupled laser generating pair as the emitter and a receiver as shown in Fig. 5.6(a). The emitter and the receiver part were mounted in a reflective configuration. The light source is a laser diode, with a wavelength of 650 nm. The emitter radiates light to a target's surface and the reflected light is received by the receiver. The sensor was used to measure the bending motion of the IPMC actuator. The resolution of this sensor depends upon the operating frequency of the target. Therefore, when the sensor is set at different response frequencies, the resolution changes. In this experiment, the response frequency of the sensor was set at 10 Hz, which gave a resolution of 1  $\mu\text{m}$ , the highest resolution of the sensor.

For proximity sensing, the orientation of the emitting/detecting plane of the laser displacement sensor should be parallel to the target's surface as depicted in Fig. 5.6(b). The sensor emitter radiates laser light which subsequently reflects off the target surface and the reflected light is captured by the receiver. The target was arranged such that it moved in the measuring range of the laser displacement sensor, which is 20 mm around the measurement center. When the distance between the front of the sensor and the target's surface was 50 mm, the target was at the center of the measurement range. This location was known as measurement center of the

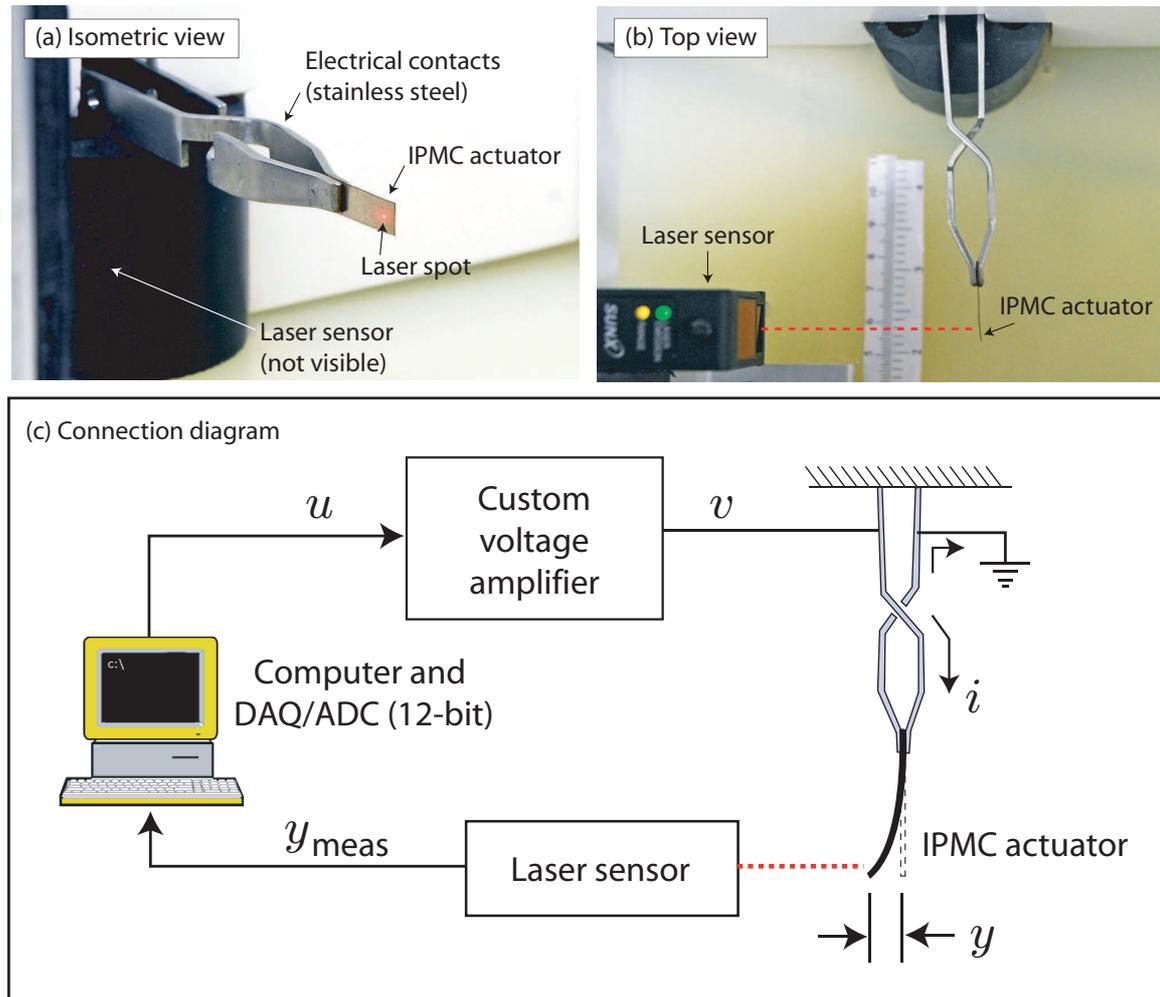


Figure 5.5: (a) and (b) Photographs of experimental system where the IPMC actuator is mounted in cantilever configuration and displacement is measured by a laser displacement sensor along the  $y$ -axis as shown (SUNX micro laser displacement sensor LM10, model No. ANR12511 with sensor controller, model No. ANR5232). In the experiment, the IPMC actuator was mounted 55 mm away from the sensor front to make sure the bending movement of the IPMC actuator can be captured by the sensor. (c) The block diagram of the experimental system.

sensor with a sensor voltage of zero.

To check the behavior of the laser displacement sensor, an experiment was

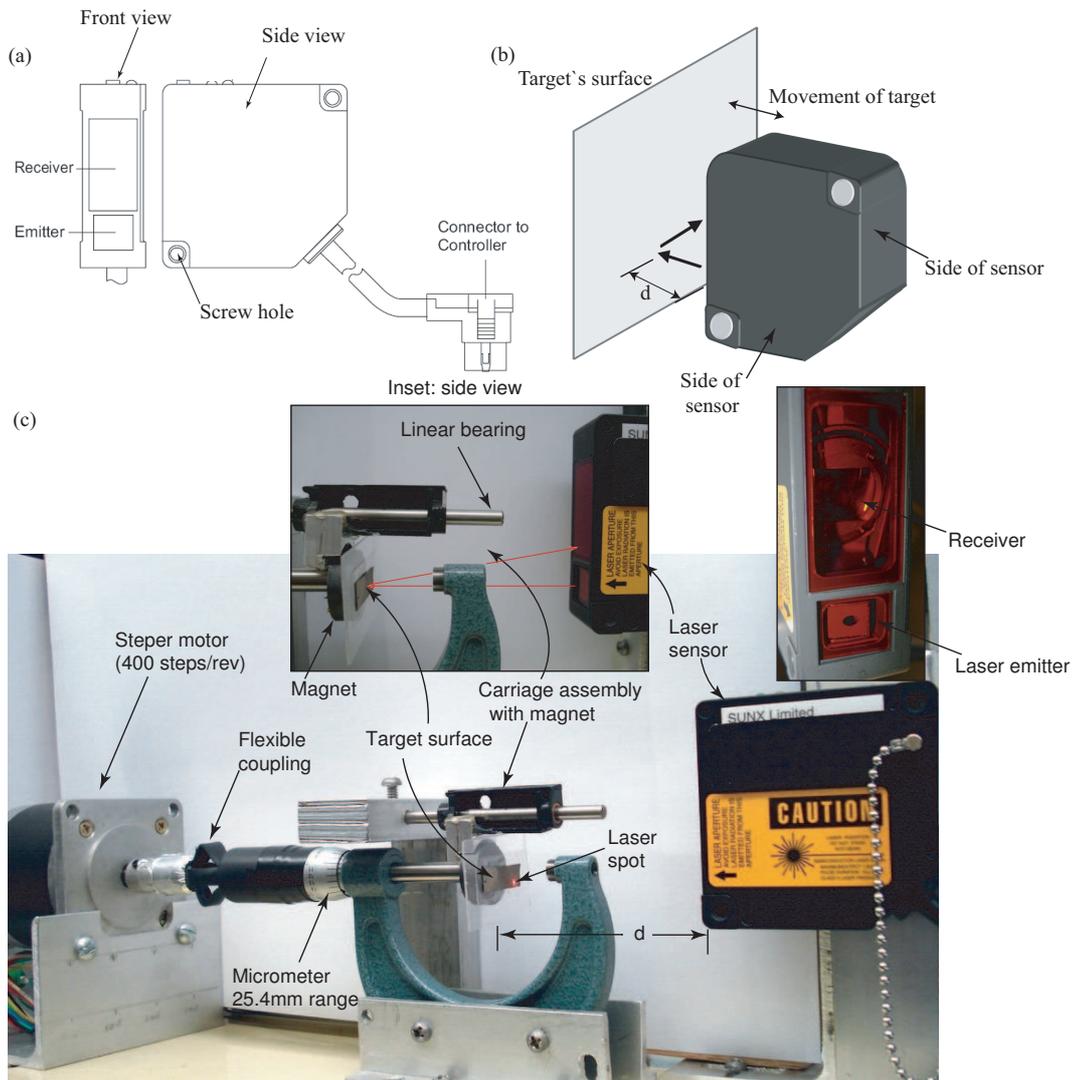


Figure 5.6: The laser displacement sensor. (a) Front view and side view of the sensor, where the emitter emits laser light at the target and the receiver collects the light reflected by the target. The sensor has its own controller, which connects to the laser unit. (b) The sensor configured to detect target displacement. (c) The calibration system used to calibrate the sensor's response.

conducted to calibrate its response by creating a calibration curve. This method determines the experimental relation between the detector output voltage and the

target's displacement, and the results can give some information about the sensor's performance. The test system built to calibrate the sensor is shown in Fig. 5.6(c). The system consisted of a micrometer [25.4 mm (1 inch) range] for positioning the target in front of the laser displacement sensor. The target was a thin and rigid rectangular plate covered with a strip of IPMC. The target was also attached to the micrometer shaft through a magnetic contact. The magnetic contact made the target move with the micrometer shaft, but with out rotation. The displacement of the micrometer shaft was driven by a stepper motor. In particular, the micrometer shaft was connected to the motor through a flexible rubber coupler. The stepper motor moved 400 steps per revolution, which was  $0.9^\circ$  each step. Therefore, the target moved  $1.56 \mu\text{m}$  per step relative to the sensor.

The sensor was calibrated four times in the operating range of 12.7 mm (about 0.5 inch). During calibration, the sensor was fixed and the target moved relative to the sensor's measurement center (a position that is 50 mm away from the sensor). The output voltages  $V_s$  of the sensor versus the target displacement  $d$  are shown in Fig. 5.7. From the experimental results, it can be seen that when the distance between the target and sensor is in the range of 50 mm to 53.5 mm, the sensor cannot detect the motion of the target. But when the distance is larger than 53.5 mm, the sensor responds to the movement of the target and the response between the sensor output and the target's displacement is approximately linear. Similarly, when the distance between the target and sensor is in the range of 50 mm to 46.5 mm, the sensor cannot detect the movement of the target. But when the distance is smaller than 46.5 mm, the sensor responds to the movement of the target and the response between the sensor output and the target's displacement is also approximately linear. So the range from 46.5 mm to 53.5 mm is called the *no-response* range and the range outside of this is called the *linear-working range* of the laser displacement sensor.

The measurement range of the sensor is shown in Fig. 5.8. When an IPMC actuator moves in the *no-response* range, the sensor output is not related to the

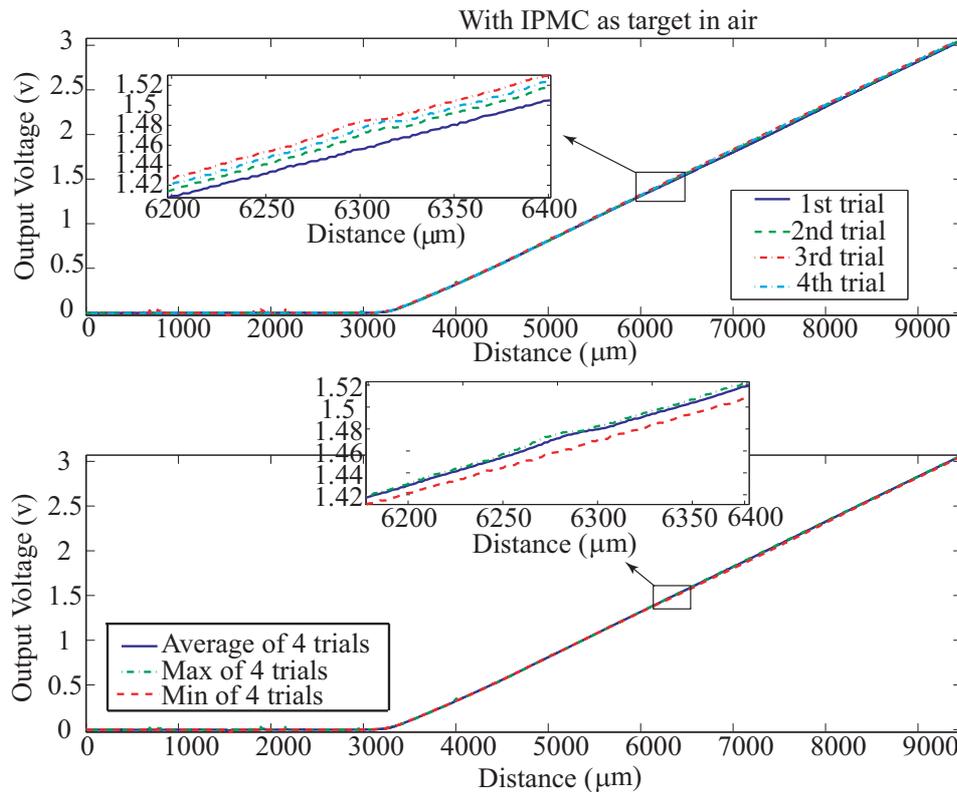


Figure 5.7: The sensor calibration results which show the sensor output voltage versus target displacement. (a) Results of four trials and (b) the average of the four trials.

displacement. When the actuator is moving in the *linear range I and II*, the sensor output is approximately linear with respect to the displacement. The experiments were conducted in range II.

### 5.2.3 IPMC Voltage Amplifier and Sensor Filter Circuits

The IPMC actuators can be driven using a voltage amplifier, current amplifier, or charge amplifier circuit (Robinson, 2005). A voltage amplifier was designed to maintain a particular voltage on the IPMC's electrodes (see example in Fig. 5.9). A current amplifier controls the current passing through the IPMC. Finally, a charge amplifier

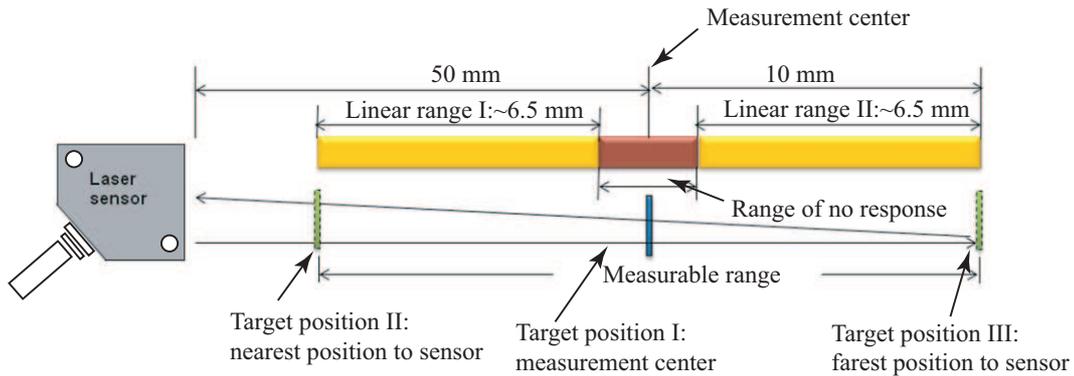


Figure 5.8: Laser displacement sensor's measurement range. (a) Measurement center of the sensor is 50 mm away from the front of the sensor. When the target is at this point, the sensor output voltage is zero. (b) Range of no response is a range around the measurement center with length of 7 mm. When target moves in this range, the sensor has no response. (c) linear range I and II. When target move in these ranges, the sensor output is nearly linear with respect to the displacement of the IPMC actuator. (d) Target position II and III mean that when the target moves outside of the range between target position II and III, the sensor output will exceed the measurable level of the DAQ system.

controls the charge delivered to the IPMC.

A custom-designed voltage amplifier circuit as shown in Fig. 5.9 was used to control all the IPMCs. This design consists of a closed-loop circuit using an op-amp and two power transistors, TIP33 and TIP34. In practice, an input signal  $u$  was sent directly from the computer to the op-amp. The signal was then compared to the voltage across the electrodes of the IPMC actuator. The high gain of the op-amp in the negative feedback configuration caused the circuit to maintain the desired input voltage  $u(t)$ .

The sensor output filter circuit [Fig. 5.9(b)] was used to process the output signal of the laser displacement sensor. It included a bias term; the bias was used to zero the output signal. A low-pass filter minimized the effects of noise. The gain of the filter circuit was set to 5, with cutoff frequency of 159.15 Hz

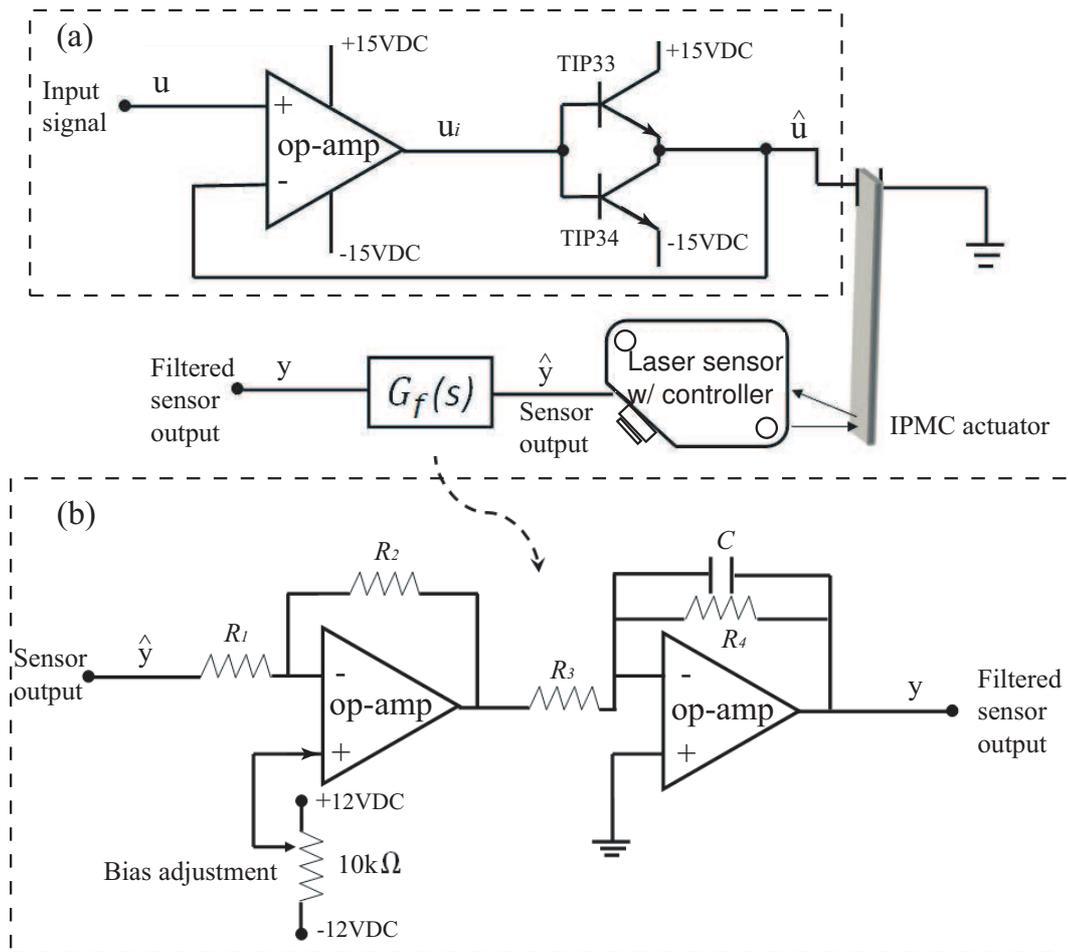


Figure 5.9: Circuits for the IPMC actuation and filter for sensor output. (a) Voltage amplifier circuit, where the op-amp and power transistors TIP33 and TIP34 form a closed-loop to equate  $\hat{u}$  to  $u$ . (b) A filter circuit was used to process the sensor output to minimize the effects of noise at high frequency.

#### 5.2.4 Performance of IPMC Actuators

Experiments were conducted to evaluate the performance of the custom-made IPMC actuators prior to implementing the controllers. The open-loop responses were evaluated. Four custom-made IPMCs were cut into  $7\text{ mm} \times 24\text{ mm}$  strips. These IPMC strips were used as actuators, and they were clamped at one end (cantilever configu-

ration) using the custom-made stainless steel fixture depicted in Figs. 5.5(a) and (b). The other end of the actuator was unconstrained and free to move.

In the first experiment, a step voltage signal  $u$  from the desktop computer and DAC card was applied to the voltage amplifier circuit as shown in Fig. 5.9(a). The IPMC's reponse along the  $y$ -direction shown was measured by the noncontact laser displacement sensor (see Fig. 5.5). The sensor was positioned at approximately  $d = 55$  mm away from the IPMC actuator in the linear range II (see Fig. 5.8). The sensor output after the filter circuit [shown in Fig. 5.9(b)] was collected using the 12-bit DAQ board and processed by the desktop computer.

Five trials were conducted for each IPMC actuator using the step input. The output was recorded over a 30 s time interval. After each trial the IPMC actuator was quickly put back into deionized water to ensure that it was saturated with water. The average of the open-loop step responses (in volts) versus time are shown in Fig. 5.10.

From the results, it can be seen that the four custom-made IPMC actuators made from the casted Nafion membranes and the commercially available membranes performed as desired. Also, the results show that the IPMC actuators share the common behaviors that included relaxation. Upon application of a step voltage, the IPMC initially responded with a quick change in displacement toward a maximum value. Upon reaching the maximum value, the IPMC slowly relaxed. This behavior has been observed by other researchers (Shahinpoor et al., 1998; Bhat and Kim, 2004; Tan and Baras, 2005). Specifically, when these IPMC actuators have the same displacement, the thicker one requires more voltage to activate as indicated in Fig. 5.10(a). Also the thicker IPMC has a longer settling time as shown in Fig. 5.10(b). As described by Kim and Shahinpoor (2002), the thickness affects the dynamics of the actuator. Based on these results, the custom-fabricated Type II IPMC with thickness of  $450 \mu\text{m}$  was selected for implementing the controllers.

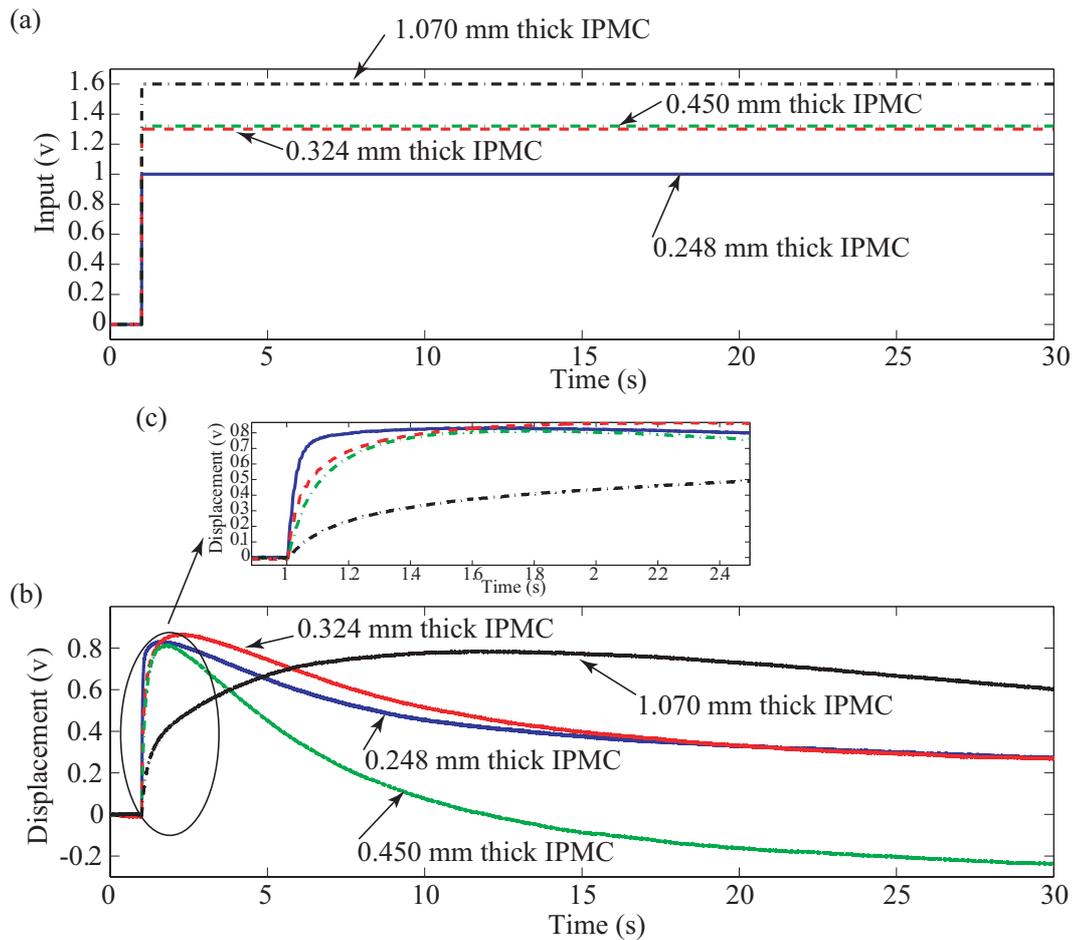


Figure 5.10: Open-loop performance of four custom-made IPMC actuators. (a) The input signal versus time shows the voltages applied to each of the IPMCs. These voltages were chosen because they ensured that each actuator achieved almost the same output displacement. (b) The measured output displacement (in volts) versus time shows the performances of the four IPMC actuators. (c) Inset figure shows the high frequency response of the four IPMC actuators.

### 5.3 Open-Loop Step Responses of a Selected IPMC Actuator

Before the application of the feedforward and feedback controllers, several experiments on the open-loop response of the candidate Type II actuator were performed. In these experiments, step inputs with three different magnitudes were applied. The objective

was to check the dynamic behavior of the actuator for designing the PI controller.

The open-loop step responses for the selected IPMC actuator are shown Fig. 5.11. The three input voltages were 0.5 V, 1.0 V and 2.5 V. The corresponding measured displacements (outputs), normalized with respect to the peak value ( $\pm 2.0$  mm), are shown in Fig. 5.11(b). In the figure, it is apparent that the 3 different inputs resulted in approximately the same 5% settling times at  $t_s = 170$  ms.

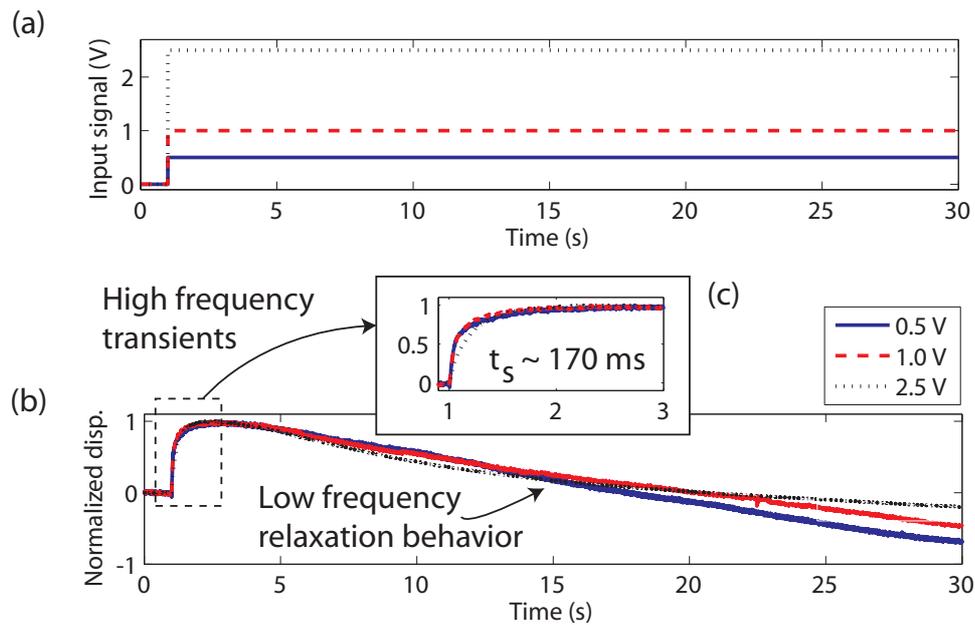


Figure 5.11: Open-loop step responses of a selected IPMC actuator. (a) The input signal (in volts) versus time. (b) The measured output displacement (normalized) versus time shows the transient behavior and the low frequency relaxation behavior. (c) Inset figure shows more closely the high frequency transient. The 5% settling time of the transient was approximately 170 ms.

This work was concerned with the high-frequency behavior (vibrational dynamics) of the IPMC shown in the inset plot, Fig. 5.11(c). The relaxation effect was referred to as the low-frequency behavior and it was not directly considered other than using the PI controller to minimize the effect. In all experiments the IPMC was operated at relatively high frequency ( $\geq 1$  Hz) to avoid the effects of relaxation.

Moreover, the motion of the actuator was centered about the origin.

## 5.4 Controllers Implementation

To implement the inversion-based feedforward controller and the PI feedback controller, it is important to model the dynamic behavior of the IPMC actuator. As presented in Chapter 4, the inversion-based feedforward controller is based on the transfer function  $G(j\omega)$ , and the design of the gains  $K_p$  and  $K_i$  of the PI feedback controller are based on the gain margin of the open-loop system (Franklin et al., 2006). So the implementation of the controllers starts with modelling the dynamics of IPMC actuator.

### 5.4.1 Dynamics Modelling of IPMC

The frequency response of the IPMC actuator, *i.e.*,  $G(j\omega)$ , was measured using a dynamic signal analyzer (DSA, Hewlett Packard model 35670A). (Other methods to obtain the frequency response curve from measured input-output data can be used.) A sinusoidal input voltage  $u$ , with a fixed amplitude and varying frequency, was applied to drive the IPMC actuator. The magnitude of the input was maintained at 500 mV for protecting the IPMC actuator for damage due to electrolysis. The displacement of the IPMC actuator was captured by the laser displacement sensor. The output of the sensor was fed back to the DSA to construct the frequency response plots (magnitude and phase versus frequency). The frequency range was 1 Hz to 1 kHz and the resulting frequency response is shown in Fig. 5.12.

### 5.4.2 Feedforward Controller Implementation

The feedforward control method described above requires the knowledge of  $G(j\omega)$ . A model can be obtained by curve fitting the measured frequency response of the IPMC actuator over an appropriate frequency range. For example, the Matlab sys-

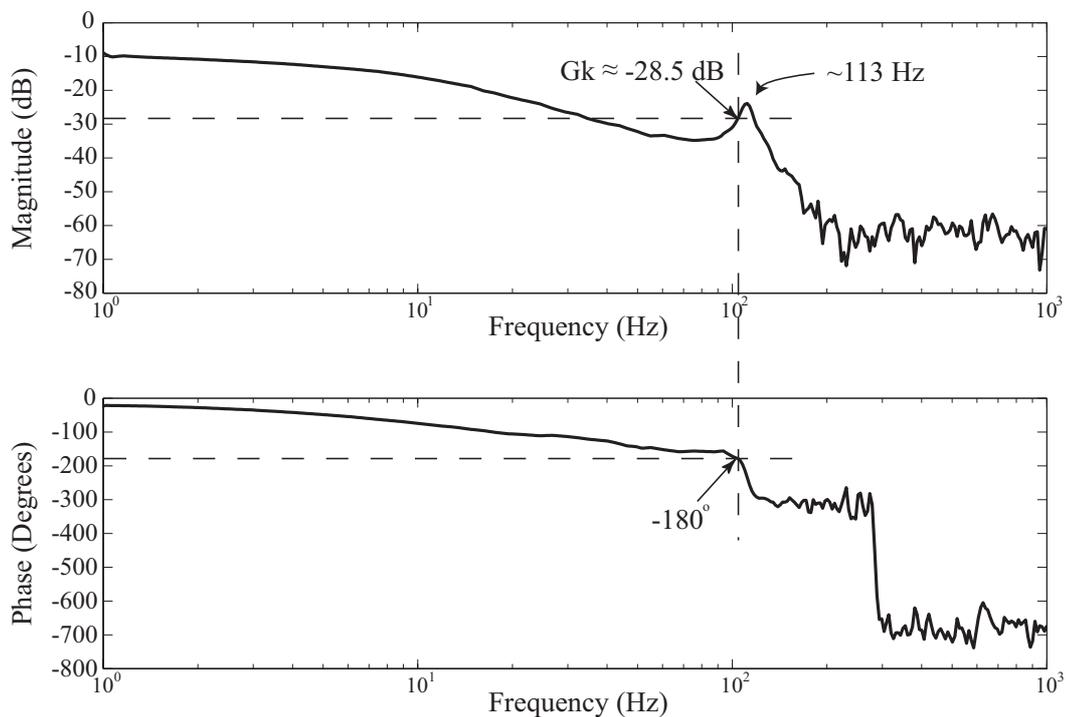


Figure 5.12: Frequency response  $G(j\omega)$  of the IPMC actuator, where the  $G_k$  is the gain margin.

tem identification package can be used to identify the model  $G(j\omega)$  from measured input-output data. Then the model can be inverted to find the feedforward input as described above.

Rather than curve fit the measured frequency response curves to find  $G(j\omega)$ , the measured frequency response data was used directly in the inversion feedforward scheme Eq. (4.12, 4.13). This approach eliminated the time-intensive step of identifying the model  $G(j\omega)$ . Furthermore, the modeling process step can introduce modeling errors. Figure 5.13 outlines the simplified implementation process.

First, the frequency response for  $G(j\omega)$  was obtained using the DSA as described above. The frequency response shows a dominantly resonant peak at approxi-

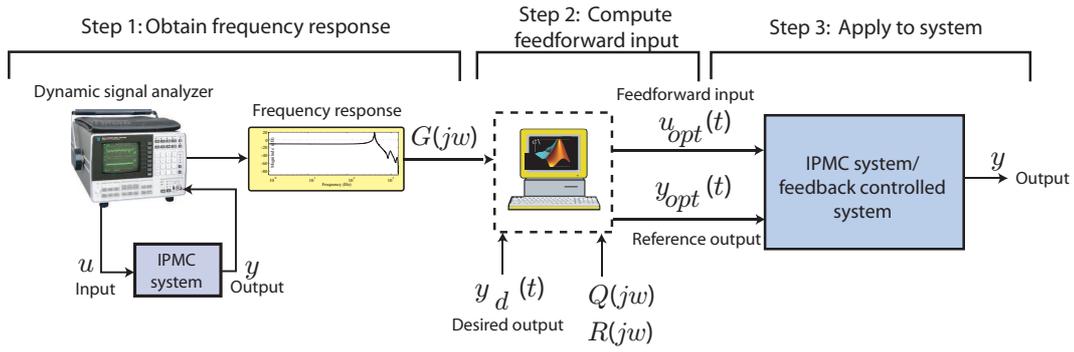


Figure 5.13: Steps to implement the feedforward method.

mately 113 Hz. Therefore, when the IPMC was operated at relatively high frequencies, oscillations in the output response can occur. The objective was to compensate for these effects using the feedforward method.

The frequency response was used directly to determine the optimal feedforward input  $U_{opt}(j\omega)$  from Eq. (4.12). First, the specified desired output trajectory  $y_d(t)$  was transformed into the frequency domain using the fast Fourier transform (FFT) algorithm in Matlab. The tracking error and input weights  $Q(j\omega)$  and  $R(j\omega)$ , respectively, were chosen based on the frequency response of the IPMC actuator shown in Fig. 5.12. For example,  $Q = 1$  and  $R = 0$  for  $\omega \in [0, 130]$  Hz and  $Q = 0$  and  $R = 1$  for  $\omega > 130$  Hz as shown in Fig. 5.14. The weights were chosen such that the computed feedforward input magnitude was less than 5 V.

Application of optimal inversion feedforward scheme Eqs. (4.12, 4.13) produced the feedforward input  $U_{opt}(j\omega)$  and reference trajectory  $Y_{opt}(j\omega)$ . The time-domain solutions  $u_{opt}(t)$  and  $y_{opt}(t)$  were found by the inverse Fourier transform function in Matlab. The process for this feedforward input computation is outlined in Fig. 5.15, and the Matlab code for the feedforward input calculation is shown in Appendix C.

Finally, the feedforward input  $u_{opt}(t)$  and reference output  $y_{opt}(t)$  were applied to the closed-loop controlled IPMC system shown in Fig. 4.2. A custom C-program

was used to implement the control scheme and is presented in Appendices D.2 and D.3.

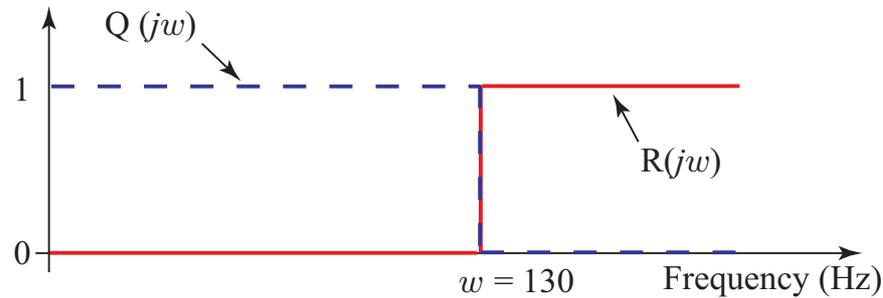


Figure 5.14: The tracking error and input weights  $Q(j\omega)$  and  $R(j\omega)$  respectively.

#### 5.4.3 PI Feedback Controller Implementation

The PI controller described above was implemented on a desktop computer using a custom-written  $C$  program as shown in Appendix D.1. The digital controller had a closed-loop bandwidth of 10 KHz.

The proportional gain  $K_p$  was found using the Ziegler-Nichols tuning method (Franklin et al., 2006), based on the gain margin of the open-loop system. The integral gain  $K_i$  was found experimentally. The Ziegler-Nichols method required the knowledge of the process reaction curve, which is the high frequency part of the step response data of the selected IPMC actuator as shown in Fig. 5.16. In the Ziegler-Nichols tuning method, the proportional gain  $K_p = 0.9/RL$ . Additionally, the gain margin limits the selection of  $K_p$ . If  $K_p$  is larger than the gain margin, the closed loop system may become unstable.

The process reaction curve of the open-loop step response of the selected IPMC actuator is shown in Fig. 5.16. The process reaction curve was taken as an average of the normalized measured displacements (outputs) shown in Fig. 5.11(b). In this

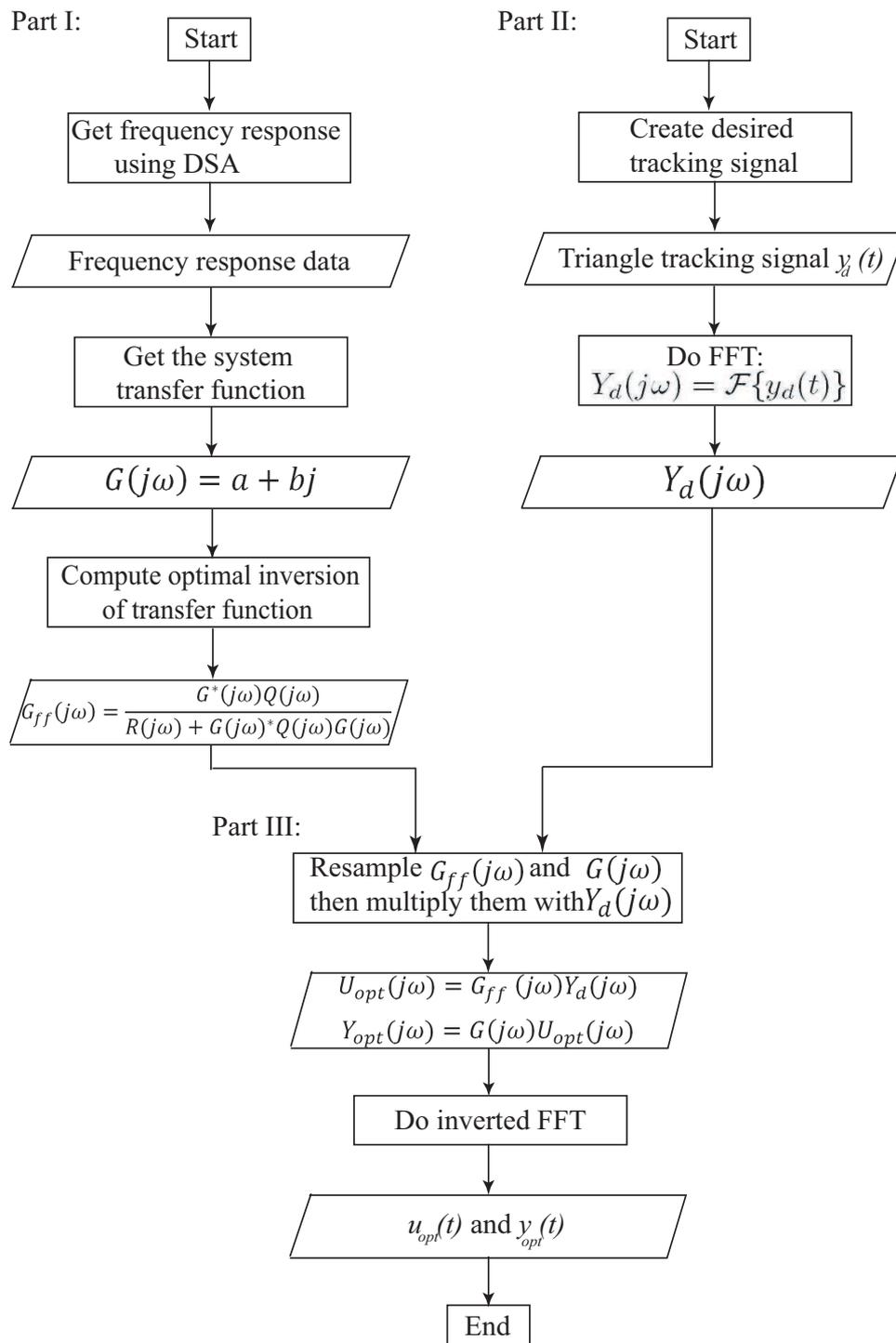


Figure 5.15: The process for the feedforward input computation. Part I was used to treat the model  $G(j\omega)$  of the IPMC actuator system and to find the optimal inversion from the system model  $G_{ff}(j\omega)$ . This part was based on the  $Q(j\omega)$  and  $R(j\omega)$  weights in frequency domain. Part II was used to treat the given desired tracking trajectory  $y_d(t)$ , *i.e.*, to convert to the frequency domain  $Y_d(j\omega)$ . Finally, in Part III the frequency response function  $G(j\omega)$  and  $G_{ff}(j\omega)$  from Part I along with the  $Y_d(j\omega)$  from Part II were used to find the time domain signal  $u_{opt}(t)$  and  $y_{opt}(t)$ .

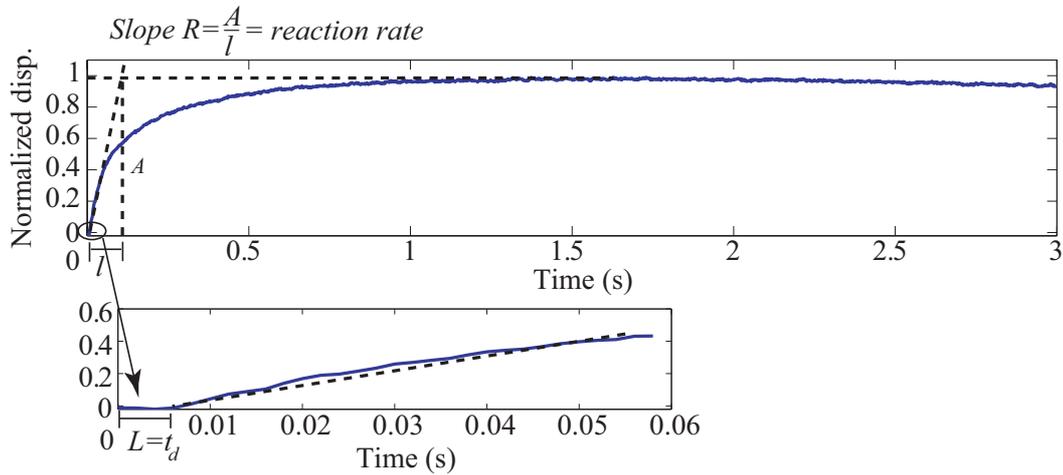


Figure 5.16: The process reaction curve for a step response of the IPMC actuator, where  $L$  is the time delay,  $R$  is the slope of the tangent line at the indicated location and  $l$  is the time constant.

figure, it is apparent that the slope of the tangent line of the process reaction curve  $R \approx 5.5$ , and the time delay  $L \approx 7$  ms. Therefore, the  $K_p \approx 23.4$ .

From the frequency response results of the IPMC actuator in Fig. 5.12, it can be seen that the gain margin  $G_k = -28.5dB$ , therefore, the gain  $G_n = 26.6$ . This gain is a reference for selecting  $K_p$ .

The starting point for tuning the proportional gain of the PI controller was between  $K_p \in [23.4, 26.6]$ . Finally, the proportional and integral gains were tuned experimentally to 24 and 0.7, respectively. The performance of this PI feedback controller with  $K_p = 24$  and  $K_i = 0.7$  for the step response of the IPMC actuator is shown in Fig. 5.17.

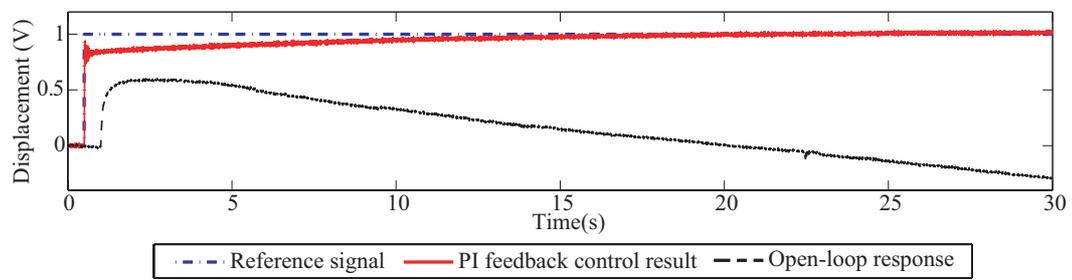


Figure 5.17: The performance of the designed PI feedback controller compared to the performance of the open-loop step response of the IPMC actuator.

## Chapter 6: Experimental Results and Discussion

In this chapter, the experimental results for the feedforward, feedback, and the integrated controller are presented and discussed. The discussion begins with the feedforward controller results, followed by the feedback controller results, then ending with the integrated controller results.

### 6.1 Feedforward Tracking Results

The feedforward method was applied to track the desired triangle trajectories at 1 Hz, 10 Hz, and 18 Hz; and the results are shown in Fig. 6.2. The desired trajectories were chosen as triangle signals. The computed feedforward inputs are shown in Fig. 6.1, and they were less than 5 V. The plots in Fig. 6.2 show the normalized displacement of the IPMC actuator with respect to time corresponding to the feedforward inputs. The optimized desired triangle trajectory  $y_{opt}$  is denoted by the dash line. The dash-dot line represents the response of the IPMC actuator without feedforward compensation, that is, by applying the input

$$u(t) = \frac{1}{G(0)}y_{opt}(t), \quad (6.1)$$

where  $G(0)$  is the DC gain of the IPMC system. In all three cases, plots (a) through (c), the tracking errors of the uncompensated case were significant. Table 6.1 lists the magnitude of the maximum tracking error,

$$e_{max}(\%) = \left[ \frac{|\hat{y} - \hat{y}_{opt}|}{\max(\hat{y}_{opt}) - \min(\hat{y}_{opt})} \right] \times 100\%, \quad (6.2)$$

where  $\hat{y} = y/\max(y)$  and  $\hat{y}_{opt} = y_{opt}/\max(y_{opt})$  are the normalized measured and desired output, respectively. Also listed are the root-mean-squared error values, defined

as

$$e_{rms}(\%) = \left[ \frac{\sqrt{\frac{1}{T} \int_0^T [\hat{y}(t) - \hat{y}_{opt}(t)]^2 dt}}{\max(\hat{y}_{opt}) - \min(\hat{y}_{opt})} \right] \times 100\%. \quad (6.3)$$

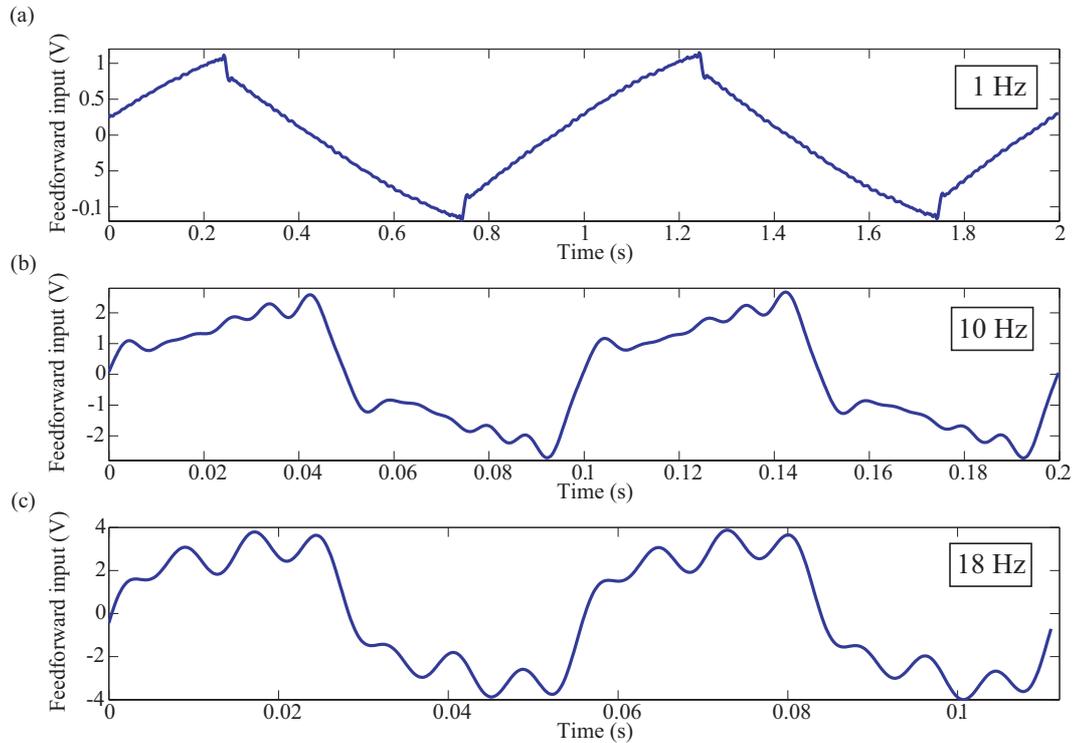


Figure 6.1: Feedforward control inputs relative to the desired outputs. (a) Feedforward input for 1 Hz tracking; (b) Feedforward input for 10 Hz tracking; (c) Feedforward input for 18 Hz tracking.

The results show clearly that the feedforward controller was able to improve the tracking response of the IPMC by at least 70% compared to the uncompensated case. Although the improvement was significant, at the 18 Hz operating frequency, the maximum tracking error with feedforward control was quite large at 17.54%. The large tracking error at high frequency could have been attributed to unmodelled dynamic effects. To further improve the response of the feedforward controller, a feedback controller was designed and integrated with the feedforward controller.

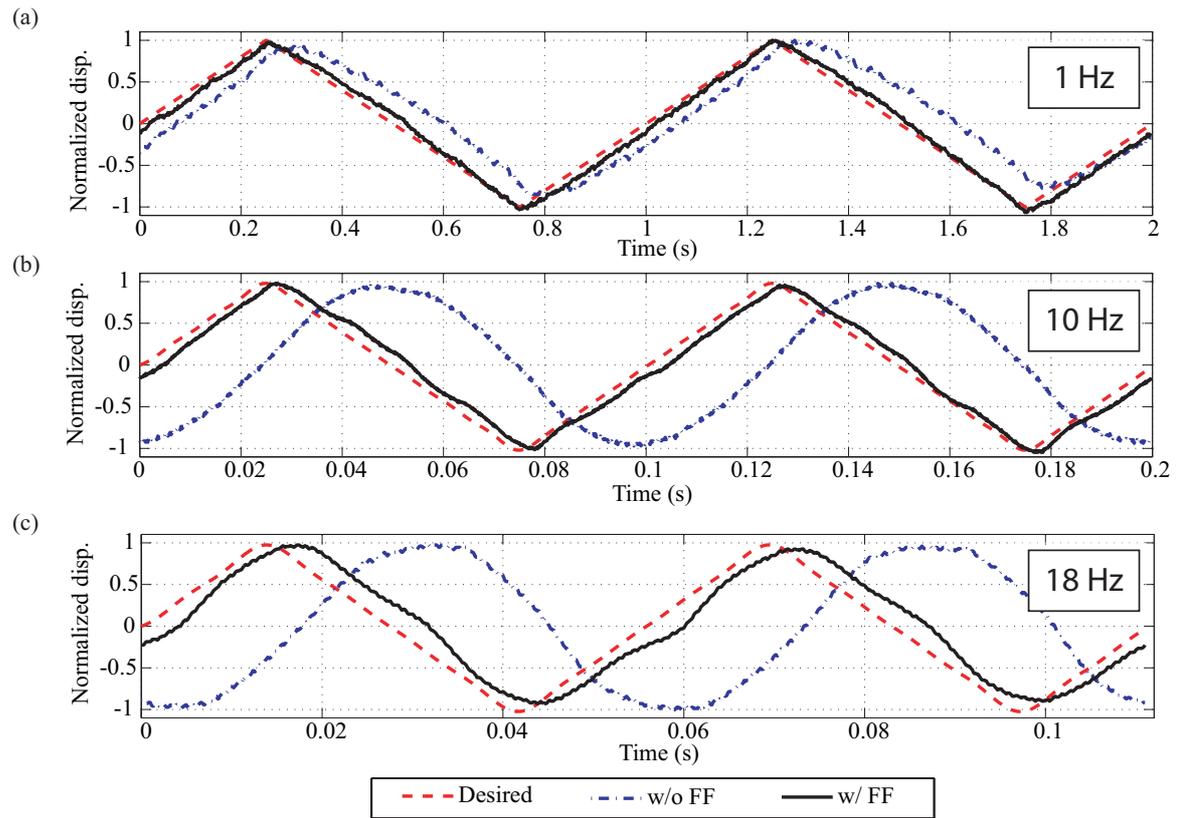


Figure 6.2: Feedforward control results. (a) 1 Hz tracking; (b) 10 Hz tracking; (c) 18 Hz tracking.

## 6.2 Feedback Tracking Results

The feedforward controller compensated for the dynamic effects; however, residual error remained due to unmodelled effects. To minimize the residual error, a PI feedback controller  $C(s) = K_p + \frac{K_i}{s}$ , was combined with the feedforward controller (see Fig. 4.1). In the experiment, the constants were tuned experimentally to  $K_p = 24$  and  $K_i = 0.7$ , as mentioned in the above chapter. One of the benefits of feedback control was it provided robustness to parameter or dynamics variation. This section discusses the results of the performance of the PI feedback controller without feedforward control.

Table 6.1: Feedforward control results: maximum and root-mean-squared tracking error relative to range for without and with inverse feedforward (FF) control.

<b>Traj. freq. (Hz)</b>	$e_{max}$ (%) [w/o <b>FF</b> ]	$e_{max}$ (%) [w/ <b>FF</b> ]	<b>Improvement (%)</b>
1	25.52	7.29	71.43
10	60.32	10.19	83.11
18	79.85	17.54	78.03

<b>Traj. freq. (Hz)</b>	$e_{rms}$ (%) [w/o <b>FF</b> ]	$e_{rms}$ (%) [w/ <b>FF</b> ]	<b>Improvement (%)</b>
1	14.94	4.07	72.76
10	43.48	5.77	86.73
18	55.08	10.64	80.68

The performance of the feedback controller is shown in Fig. 6.3, and immediately the results shows good tracking performance at low frequency (1 Hz). However, as the frequency increases, the PI feedback controller's performance degraded. Table 6.2 shows that maximum error at 18 Hz was over 33%.

It is clear that the feedback controller out performs the open-loop response (without inversion feedforward). But at high frequency, the feedback controller was unable to provide good performance as indicated by the large ( $>5\%$ ) tracking error.

### 6.3 Integrated Feedforward and Feedback Controller Tracking Results

When the feedback controller was combined with the feedforward controller, the error was reduced considerably compared to both the uncompensated case shown in Fig. 6.2 and the feedback-only case shown in Fig. 6.3. At 18 Hz tracking frequency, the maximum tracking error for the integrated controller was just over 7%. Table 6.3 lists the tracking results for the integrated controller. Comparing the results of Table 6.1 and Table 6.3, the feedback controller reduced the maximum tracking error of the

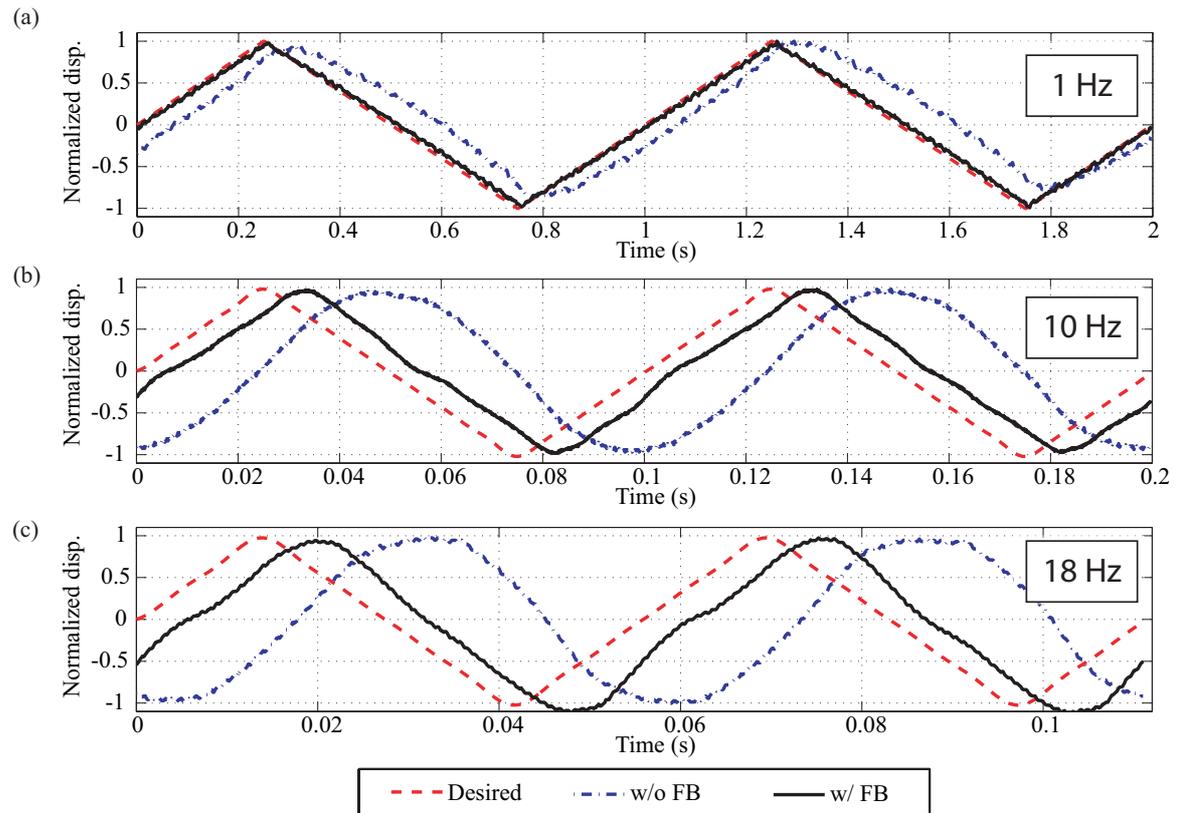


Figure 6.3: Feedback controller results. (a) 1 Hz tracking; (b) 10 Hz tracking; (c) 18 Hz tracking.

feedforward controller at 18 Hz by over 50%. Therefore, the integrated controller allows precise tracking control of the IPMC at relatively high operating frequency.

Table 6.2: The proportional-integral (PI) feedback control results: maximum and root-mean-squared tracking error relative to range for without and with the PI feedback (FB) control.

<b>Traj. freq. (Hz)</b>	$e_{max}$ (%) [w/o FB]	$e_{max}$ (%) [w/ FB]	<b>Improvement (%)</b>
1	25.52	4.88	80.88
10	60.32	18.82	68.80
18	79.85	33.79	57.68

<b>Traj. freq. (Hz)</b>	$e_{rms}$ (%) [w/o FB]	$e_{rms}$ (%) [w/ FB]	<b>Improvement (%)</b>
1	14.94	2.35	84.27
10	43.48	14.47	66.72
18	55.08	20.53	62.73

Table 6.3: Feedforward and proportional-integral (PI) feedback control results: maximum and root-mean-squared tracking error relative to range for PI feedback only (FB only) control and integrated feedforward control and PI feedback (FF & FB).

<b>Traj. freq. (Hz)</b>	$e_{max}$ (%) [FB only]	$e_{max}$ (%) [FF & FB]	<b>Improvement (%)</b>
1	4.88	3.12	36.07
10	18.82	5.06	73.11
18	33.79	7.87	76.71

<b>Traj. freq. (Hz)</b>	$e_{rms}$ (%) [FB only]	$e_{rms}$ (%) [FF & FB]	<b>Improvement (%)</b>
1	2.35	0.87	62.98
10	14.47	2.11	85.42
18	20.53	2.86	86.07

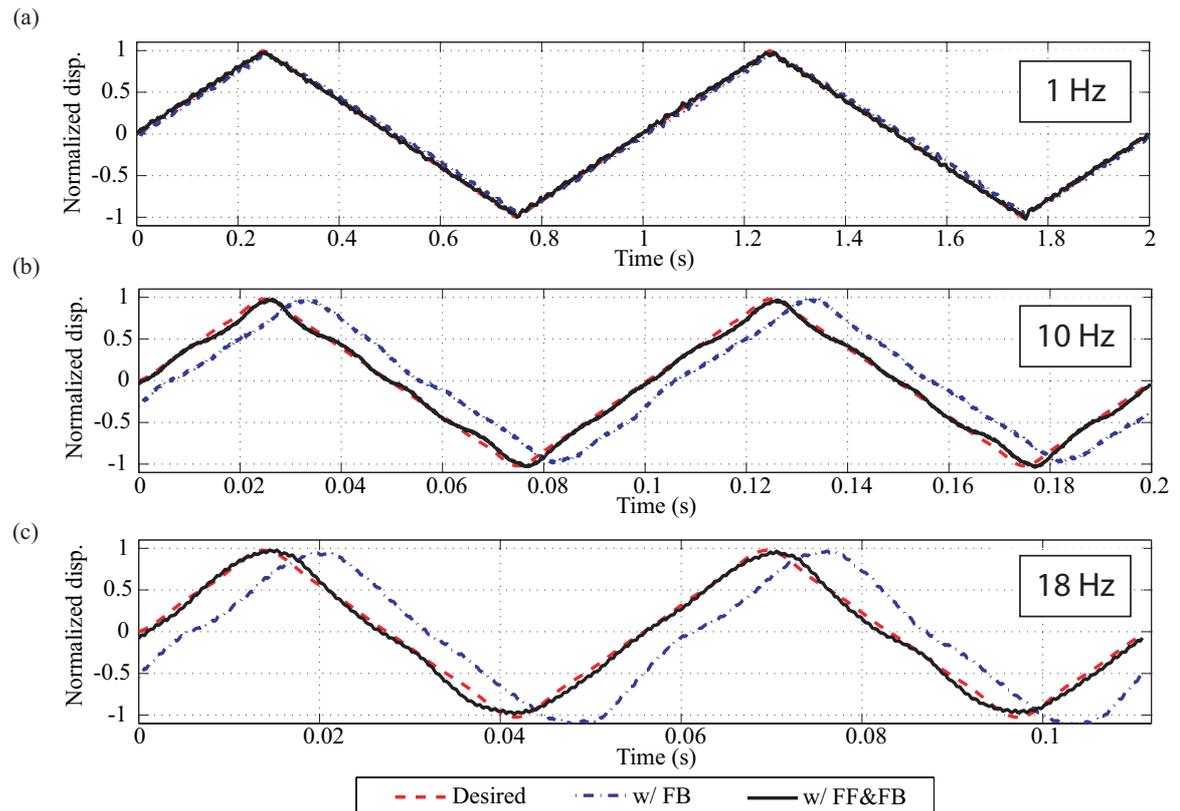


Figure 6.4: Integrated feedforward and proportional-integral (PI) feedback controller results. (a) 1 Hz tracking; (b) 10 Hz tracking; (c) 18 Hz tracking.

## Chapter 7: Conclusions and Future Works

This thesis presented the fabrication of IPMC actuators and application of feedforward and feedback control to account for dynamics, relaxation, and nonlinearities. The fabrication process included a casting process to create custom-made Nafion membranes for making IPMC actuators. This method enabled the fabrication of thicker IPMCs, and these IPMCs can generate more force and store more ionic fluid for enhanced performance compared to thinner actuators.

An inversion-based feedforward controller was used to account for the linear dynamics and a PI feedback controller was used to provide robustness and to account for unmodeled dynamics and nonlinearity. A feedforward controller was implemented by using the measured frequency response of the IPMC actuator, thus eliminating the process of identifying the dynamics model. Experimental results showed that the feedforward controller compensated for the dynamics at relatively high operating frequency. The performance of the feedforward controller was further improved by augmenting a PI feedback controller, an over 50% reduction of the tracking error at 18 Hz. Therefore, the integrated feedforward and feedback controller allows precise positioning of IPMC actuators at relative high frequency.

The future work includes incorporating nanocomposites into the design of IPMCs and nonlinear modeling and control. The nanocomposites can potentially enhance the actuating ability of the IPMC actuators. This work considered a linear model in the inversion method, but the IPMC actuator is a nonlinear system; therefore, the next step is to consider the nonlinear model and inversion process.

## Bibliography

- B. Ando and S. Graziani. A new IR displacement system based on noise added theory. In *IEEE Conference on Instrumentation and Measurement Technology*, pages 482 – 485, Budapest, Hungary, 2001.
- T. Ando, T. Uchihashi, N. Kodera, A. Miyagi, R. Nakakita, H. Yamashita, and M. Sakashita. High-speed atomic force microscopy for studying the dynamic behavior of protein molecules at work. *Jap. J. Appl. Physics*, 45(3B):1897 – 1903, 2006.
- D. Armani, C. Liu, and N. Aluru. Re-configurable fluid circuits by PDMS elastomer micromachining. In *12th International Conference on MEMS*, pages 222 – 227, Orland, FL, 1999.
- A. Ashkin, J. M. Dziedzic, J. E. Bjorkholm, and S. Chu. Observation of a single-beam gradient force optical trap for dielectric particles. *Optics Letters*, 11(5):288–290, 1986.
- G. Bao and S. Suresh. Cell and molecular mechanics of biological materials. *Nature Materials*, 2:715 – 725, 2003.
- Y. Bar-Cohen, T. Xue, B. Joffe, S.-S. Lih, M. Shahinpoor, J. Simpson, J. Smith, and P. Willis. Electroactive polymers (EAP) low-mass muscle actuators. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 3041, pages 697 – 701, San Diego, CA, 1997.
- Y. Bar-Cohen, S. Leary, M. Shahinpoor, J. O. Harrison, and J. Smith. Electro-active polymer (EAP) actuators for planetary applications. In *Proceedings of SPIE - The*

- International Society for Optical Engineering*, volume 3669, pages 57 – 63, Newport Beach, CA, 1999.
- Y. Bar-Cohen, S. Leary, A. Yavrouian, K. Oguro, S. Tadokoro, J. Harrison, J. Smith, and J. Su. Challenges to the transition to the practical application of ipmc as artificial muscle actuators. In *Electroactive Polymers (EAP). Symposium (Materials Research Society Symposium Proceedings)*, volume 600, pages 13 – 20, Boston, MA, 2000.
- Y. Bar-Cohen, S. Sherrit, and S.-S. Lih. Characterization of the electromechanical properties of EAP materials. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 4329, pages 319 – 327, Newport Beach, CA, 2001.
- Y. Bar-Cohen, X. Bao, S. Sherrit, and S.-S. Lih. Characterization of the electromechanical properties of ionomeric polymer-metal composite (IPMC). In *Proceedings of the SPIE Smart Structures and Materials Symposium*, volume 4695, pages 286–293, San Diego, CA, 2002a.
- Y. Bar-Cohen, X. Bao, S. Sherrit, and S.-S. Lih. Characterization of the electromechanical properties of ionomeric polymer-metal composite (IPMC). In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 4695, pages 286 – 293, San Diego, CA, 2002b.
- C. Bartoletti, R. Buonanni, L. G. Fantasia, R. Frulla, W. Gaggioli, and G. Sacerdoti. The design of a proximity inductive sensor. *Meas. Sci. Technol.*, 9:1180 – 1190, 1998.
- L. K. Baxter. *Capacitive Sensors Design and Application*. IEEE Press Series on Electronics Technology, New Jersey, 1997.
- E. Bayo. A finite-element approach to control the end-point motion of a single-link flexible robot. *Journal of Robotic Systems*, 4:63–75, 1987.

- M.-C. Belanger and Y. Marois. Hemocompatibility, biocompatibility, inflammatory and in vivo studies of primary reference materials low-density polyethylene and polydimethylsiloxane: a review. *Journal of Biomedical Materials Research*, 58(5): 467 – 477, 2001.
- G. Benet, F. Blanes, J. E. Simo, and P. Perez. Using infrared sensors for distance measurement in mobile robots. *Robotics and Autonomous Systems*, 40:255 – 266, 2002.
- N. D. Bhat and W.-J. Kim. Precision position control of ionic polymer-metal composite. In *Proceeding of the American Control Conference*, pages 740 – 745, Boston, Massachusetts, USA, 2004.
- C. V. C. Bouten, M. M. Knight, D. A. Lee, and D. L. Bader. Compressive deformation and damage of muscle cell subpopulations in a model system. *Annals of Biomedical Engineering*, 29:153–163, 2001.
- S. H. Chang, C. K. Tseng, and H. C. Chien. An ultra-precision  $XY\theta_z$  piezo-micropositioner part i: design and analysis. *IEEE Trans. Ultrasonics, Ferroelectrics, and Frequency Cont.*, 46(4):897 – 905, 1999.
- C. S. Chen, M. Mrksich, S. Huang, and G. M. Whitesides. Geometric control of cell life and death. *Science*, 276:1425 – 1428, 1997.
- Z. Chen, X. Tan, and M. Shahinpoor. Quasi-static positioning of ionic polymer-metal composite (IPMC) actuators. In *Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 60 – 65, Monterey, California, USA, 2005.
- S. Cho, S. Lee, B. Kim, and J. Pak. A study on the fabrication of a membrane type micro-actuator using IPMC (Ionic polymer-metal composite) for micro-pump

- application. *Transactions of the Korean Institute of Electrical Engineers, C*, 52(7): 298 – 304, 2003.
- C. K. Chung, P. K. Fung, Y. Z. Hong, M. S. Ju, C. C. K. Lin, and T. C. Wu. A novel fabrication of ionic polymer-metal composites (IPMC) actuator with silver nano-powders. *Sensors and Actuators B (Chemical)*, 117:367 – 375, 2006.
- D. Croft, D. McAllister, and S. Devasia. High-speed scanning of piezo-probes for nano-fabrication. *ASME J. Manuf. Sci. and Eng.*, 120(3):617–622, 1998.
- D. Croft, G. Shed, and S. Devasia. Creep, hysteresis, and vibration compensation for piezoactuators: atomic force microscopy application. *ASME J. Dyn. Syst., Meas., and Control*, 123:35–43, 2001.
- Z. J. Davis, G. Abadal, O. Hansen, X. Borise, N. Barniol, F. Perez-Murano, and A. Boisen. AFM lithography of aluminum for fabrication of nanomechanical systems. *Ultramicroscopy*, 97(1-4):467–472, 2003.
- S. Devasia. Should model-based inverse inputs be used as feedforward under plant uncertainty? *IEEE Trans. Autom. Contr.*, 47(11):1865–1871, 2002.
- S. Devasia, D. Chen, and B. Paden. Nonlinear inversion-based output tracking. *IEEE Trans. Automatic Control*, 41(7):930–942, 1996.
- S. Devasia, E. Eleftheriou, and S. O. R. Moheimani. A survey of control issues in nanopositioning. *IEEE Trans. Contr. Syst. Tech.*, 15(5):802 – 823, 2007.
- J. S. Dewey, K. K. Leang, and S. Devasia. Experimental and theoretical results in output-trajectory redesign for flexible structures. *ASME J. Dyn. Syst., Meas., and Control*, 120:456–461, 1998.
- F. Di Palma, M. Doucet, C. Boachon, A. Guignandon, S. Peyroche, B. Forest, C. Alexandre, A. Chamson, and A. Rattner. Physiological strains induce differ-

- entiation in human osteoblasts cultured on orthopaedic biomaterial. *Biomaterials*, 24(18):3139 – 3151, 2003.
- C. A. Dickinson and J. T. Wen. Feedback control using shape memory alloy actuators. *J. Intel. Mat. Syst. Struct.*, pages 242–250, 1998.
- P. H. Dietz, W. S. Yerazunis, and D. L. Leigh. Very low-cost sensing and communication using bidirectional LEDs. In *International Conference on Ubiquitous Computing (UbiComp)*, 2003.
- B. Fang, M. Ju, and C. Lin. A new approach to develop ionic polymer-metal composites IPMC actuator: Fabrication and control for active catheters systems. *Sensors and Actuators A: Physical*, 137(2):321 – 329, 2007.
- G. Feng and R. Chen. Universal concept for fabricating arbitrary shaped IPMC transducers and its application on developing accurately controlled surgical devices. In *Proceedings of the 2nd IEEE International Conference on Nano/Micro Engineered and Molecular Systems, IEEE NEMS 2007*, pages 622–625, Bangkok, Thailand, 2007.
- G. F. Franklin, J. D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Upper Saddle River, New Jersey, fifth edition, 2005.
- G. F. Franklin, J. D. Powell, and A. Emani-Naeini. *Feedback control of dynamic systems*. 5th edition, 2006.
- G. Giovannetti, V. Hartwig, R. Francesconi, L. Landini, and A. Benassi. Study for a portable IR sensor to detect the blood temperature during coronary bypass implantation. *Review of Scientific Instruments*, 76:84302–1–5, 2005.
- F. K. Glenister, R. L. Coppel, A. F. Cowman, N. Mohandas, and B. M. Cooke. Contributions of parasite proteins to altered mechanical properties of malaria-infected red blood cells. *Blood*, 99(3):1060 – 1063, 2002.

- S. Guo, Y. Ge, L. Li, and S. Liu. Underwater swimming micro robot using IPMC actuator. In *Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation, June 25 - 28*, pages 249 – 254, Luoyang, China, 2006.
- A. E. Holman, C. D. Laman, P. M. L. O. Scholte, W. Chr. Heerens, and F. Tuinstra. A calibrated scanning tunneling microscope equipped with capacitive sensors. *Rev. Sci. Instr.*, 67(6):2274–2280, 1996.
- R. Holze and J. C. Ahn. Advances in the use of perfluorinated cation-exchange membranes in integrated water electrolysis and hydrogen - oxygen fuel systems. *Journal of Membrane Science*, 73(1):87 – 97, 1992.
- S. Hosoe. Laser interferometric system for displacement measurement with high precision. *Nanotechnology*, 2:88 – 95, 1991.
- C. T. Hung, D. R. Henshaw, C. C. Wang, R. L. Mauck, F. Raia, G. Palmer, P. H. Chao, V. C. Mow, A. Ratcliffe, and W. B. Valhmu. Mitogen-activated protein kinase signaling in bovine articular chondrocytes in response to fluid flow does not require calcium mobilization. *J Biomech*, 33(1):73 – 80, 2000.
- K. L. D. Jong, H. C. MacLeod, P. R. Norton, N. O. Petersen, and M. F. Jasnin. Atomic force microscopy compatible device for stretching cells and absorbed proteins. *Rev. Sci. Instr.*, 77:023701, 2006.
- Y. Kaneda, N. Kamamichi, M. Yamakita, K. Asaka, and Z. W. Luo. Control of linear artificial muscle actuator using IPMC. *SICE 2003 Annual Conference*, 2:1650 – 1655, 2003.
- S. Kang, J. Shin, S.J. Kim, H. J. Kim, and Y. H. Kim. Robust control of ionic polymer-metal composites. *Smart Materials and Structures*, 16:2457 – 2463, 2007.

- P. Kejik, C. Kluser, R. Bischofberger, and R. S. Popovic. A low-cost inductive proximity sensor for industrial applications. *Sensors and Actuators A*, 110:93 – 97, 2004.
- A. Keshavarzi, M. Shahinpoor, K. J. Kim, and J. W. Lantz. Blood pressure, pulse rate, and rhythm measurement using ionic polymer-metal composite sensors. volume 3669, pages 369 – 376, 1999.
- H. I. Kim, D. K. Kim, and J. H. Han. Study of flapping actuator modules using IPMC. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 6524, page 65241A, San Diego, CA, 2007.
- K. J. Kim and M. Shahinpoor. The effect of surface-electrode resistance on the performance of ionic polymer-metal composite (IPMC) artificial muscles. *Smart Materials and Structures*, 9(4):543 – 551, 2000.
- K. J. Kim and M. Shahinpoor. A novel method of manufacturing three-dimensional ionic polymer-metal composites (IPMCs) biomimetic sensors, actuators and artificial muscles. *Polymer*, 43(3):797 – 802, 2002.
- K. J. Kim and M. Shahinpoor. Ionic polymermetal composites: II. manufacturing techniques. *Smart Mater. Struct.*, 12:65 – 79, 2003.
- K.Oguro. *Recipe for Creating the Electrode for IPMC*. Osaka National Research Institute, AIST, Japan, 1991.
- C. Kothera. Micro-manipulation and bandwidth characterization of ionic polymer actuators. Technical report, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2002.
- A. A. Kuijpers, R. J. Wiegerink, G. J. M. Krijnen, T. S. J. Lammerink, and M. Elwenspoek. Capacitive long-range position sensor for microactuators. In *IEEE 17th*

- International Conference on Micro Electro Mechanical Systems*, pages 544 – 547, 2004.
- B. C. Lavu, M. P. Schoen, and A. Mahajan. Adaptive intelligent control of ionic polymer-metal composites. *Smart Mater. Struct.*, 14:466 – 474, 2005.
- K. K. Leang and S. Devasia. Design of hysteresis-compensating iterative learning control for piezo positioners: application to atomic force microscopes. *Mechatronics*, 6(3-4):141 – 158, 2006.
- K. K. Leang and S. Devasia. Feedback-linearized inverse feedforward for creep, hysteresis, and vibration compensation in AFM piezoactuators. *IEEE Trans. Contr. Syst. Tech.*, 15(5):927 – 935, 2007.
- D. V. Lee and S. A. Velinsky. Analysis and experimental verification of a three-dimensional noncontacting angular motion sensor. *IEEE/ASME Trans. on Mechatronics*, 12(6):612 – 622, 2007.
- S. J. Lee, M. J. Han, S. J. Kim, J. Y. Jho, H. Y. Lee, and Y. H. Kim. A new fabrication method for IPMC actuators and application to artificial fingers. *Smart Materials and Structures*, 15(5):1217 – 1224, 2006.
- S. J. Lee, K. J. Kim, and H. C. Park. Design and performance analysis of a novel IPMC-driven micropump. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 5759, pages 439–446, San Diego, CA, 2007.
- C. S. Loh, H. Yokoi, and T. Arai. Improving heat sinking in ambient environment for the shape memory alloy (SMA). In *IEEE/RSJ International Conference on Robots and Systems (IROS 2005)*, pages 3560 – 3565, 2005.
- K. Mallavarapu, K. M. Newbury, and D. Leo. Feedback control of the bending response of ionic polymer-metal composite actuators. In *Proceedings of the SPIE -*

- The International Society for Optical Engineering*, volume 4329, pages 301 – 310, Newport Beach, CA, 2001.
- A. F. Md. Nor and E. W. Hill. Noise power spectral density in single-strip NiFeCoCu GMR sensors. *IEEE Trans. on Magnetics*, 38(5):2696 – 2699, 2002.
- S. O. R. Moheimani and A. J. Fleming. *Piezoelectric transducers for vibration control and damping (advances in industrial control)*. Springer, 2006.
- H. S. Nalwa. *Handbook of Polyelectrolytes and Their Applications*. American Scientific Publishers, California, 2003.
- NASA. Improved unidirectional cell-stretching device. Technical report, NASA, Lyndon B. Johnson Space Center, 2000.
- S. Nemat-Nasser and Y. Wu. Comparative experimental study of ionic polymer-metal composites with different backbone ionomers and in various cation forms. *Journal of Applied Physics*, 93(9):5255 – 5267, 2003a.
- S. Nemat-Nasser and Y. Wu. Comparative experimental study of ionic polymer-metal composites with different backbone ionomers and in various cation forms. *Journal of Applied Physics*, 93(9):5255 – 5267, 2003b.
- K. Oguro, Y. Kawami, and H. Takenaka. Bending of an ion-conducting polymer film-electrode composite by an electric stimulus at low voltage. *Journal of Micromachine Society*, 5:27 – 30, 1992.
- Optek Technology Inc. Application bulletin 204. Technical report, Optek, 1989.
- M. Otis, R. Bernier, Y. Pasco, H. Menard, H. Semmaoui, M. Jarry, and R. Fontaine. Development of an hexapod BioMicRorobot with Nafion-Pt IPMC microlegs. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology*, volume 4, pages 3423 – 3426, 2003.

- R. E. Palmer, A. J. Brady, and D. P. Roos. Mechanical measurements from isolated cardiac myocytes using a pipette attachment system. *Am. J. Physiol.*, 270(C697–C704), 1996.
- Ph. A. Passeraub, P. A. Besse, S. Hediger, Ch. de Raad, and R. S. Popovic. High-resolution miniaturized inductive proximity sensor: characterization and application for step-motor control. *Sensors and Actuators A*, 68:257 – 262, 1998.
- R. Perez, J. Agnus, C. Clevy, A. Hubert, and N. Chaillet. Modeling, fabrication, and validation of a high-performance 2-DoF piezoactuator for micromanipulation. *IEEE/ASME Trans. on Mechatronics*, 10(2):161 – 171, 2005.
- B. T. Phong. Illumination for computer generated pictures. *Comm. ACM*, 18(6):311 – 317, 1975.
- A. Punning, M. Kruusmaa, and A. Aabloo. Surface resistance experiments with IPMC sensors and actuators. *Sensors and Actuators A (Physical)*, 133(1):200 – 9, 2007.
- R. C. Richardson, M. C. Levesley, M. D. Brown, J. A. Hawkes, K. Watterson, and P. G. Walker. Control of ionic polymer-metal composites. *IEEE/ASME Transactions on Mechatronics*, 8(2):245 – 253, 2003.
- W. J. Robinson. Charge control of ionic polymers. Technical report, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2005.
- S. Saha, L. Ji, J. J. De Pablo, and S. P. Palecek. Inhibition of human embryonic stem cell differentiation by mechanical strain. *Journal of Cellular Physiology*, 206:126 – 137, 2006.
- N. Sanyal, B. Bhattacharyya, and S. Munshi. An analog non-linear signal conditioning circuit for constant temperature anemometer. *Measurement*, 39:308 – 311, 2006.

- J. L. Schaffer, M. Rizen, G. J. L'Italien, A. Benbrahim, J. Megerman, L. C. Gerstenfeld, and M. L. Gray. Device for the application of a dynamic biaxially uniform and isotropic strain to a flexible cell culture membrane. *Journal of Orthopaedic Research*, 12:709 – 719, 1994.
- G. Schitter and A. Stemmer. Fast closed loop control of piezoelectric transducers. *J. Vac. Sci. Technol. B*, 20(1):350–352, 2002.
- G. Schitter and A. Stemmer. Identification and open-loop tracking control of a piezoelectric tube scanner for high-speed scanning-probe microscopy. *IEEE Control Systems Technology*, 12(3):449 – 454, 2004.
- G. Schitter, G. E. Fantner, P. J. Thurner, J. Adams, and P. K. Hansma. Design and characterization of a novel scanner for high-speed atomic force microscopy. In *4th IFAC Symposium on Mechatronic Systems*, Heidelberg, 2006.
- U. S. Schwarz, N. Q. Balaban, D. Riveline, A. Bershadsky, B. Geiger, and S. A. Safran. Calculation of forces at focal adhesions from elastic substrate data: the effect of localized force and the need for regularization. *Biophysical Journal*, 83(3):1380 – 1394, 2002.
- M. Shahinpoor. Conceptual design, kinematics and dynamics of swimming robotics structures using ionic polymeric gel muscles. *Smart Materials and Structures*, 1(1):91 – 94, 1992.
- M. Shahinpoor and K. J. Kim. The effect of surface-electrode resistance on the performance of ionic polymer-metal composite (IPMC) artificial muscles. *Smart Materials and Structures*, 9(4):543 – 551, 2000.
- M. Shahinpoor and K. J. Kim. Ionic polymer-metal composites: I. fundamentals. *Smart Mater. Struct.*, 10:819 – 833, 2001.

- M. Shahinpoor and K. J. Kim. Ionic polymer-metal composites: III. modeling and simulation as biomimetic sensors, actuators, transducers, and artificial muscles. *Smart Materials and Structures*, 13:1362 – 1388, 2004.
- M. Shahinpoor and K. J. Kim. Ionic polymer-metal composites: IV. industrial and medical applications. *Smart Mater. Struct.*, 14:197 – 214, 2005.
- M. Shahinpoor, Y. Bar-Cohen, J. O. Simpson, and J. Smith. Ionic polymer-metal composites (IPMCs) as biomimetic sensors, actuators and artificial muscles – a review. *Smart Mater. Struct.*, 7:R15 – R30, 1998.
- J. E. Shingley, C. E. Mischke, and Richard G. Budynas. *Mechanical engineering design*. McGraw Hill, New York, 2004.
- C. Shiu, Z. Zhang, and C. R. Thomas. A novel technique for the study of bacterial cell mechanical properties. *Biotechnology Techniques*, 13:707 – 713, 1999.
- B. M. Spiegelman and C. A. Ginty. Fibronectin modulation of cell shape and lipogenic gene expression in 3T3-adipocytes. *Cell*, 35:657 – 666, 1983.
- K. K. Tan and S. C. Ng. Computer-controlled piezoactuator for cell manipulation. *IEE Proceedings Nanobiotechnology*, 150(1):15 – 20, 2003.
- X. Tan and J. S. Baras. Adaptive identification and control of hysteresis in smart materials. *IEEE Trans. Automatic Control*, 50(6):827–839, 2005.
- S. M. Tanaka. A new mechanical stimulator for cultured bone cells using piezoelectric actuator. *Journal of Biomechanics*, 32:427 – 430, 1999.
- G. Tchoupo and K. K. Leang. Hysteresis compensation for high-precision positioning of a shape memory alloy actuator using integrated iterative-feedforward and feedback inputs. In *American Control Conference*, pages 4246 – 4253, New York City, 2007.

- S. Tien, Q. Zou, and S. Devasia. Preview-based optimal inversion for output tracking: application to scanning tunneling microscopy. *IEEE Trans. Contr. Syst. Tech.*, 12(3):375 – 386, 2004.
- T. Truper, A. Kortschack, M. Jahnisch, H. Hulsen, and S. Fatikow. Transporting cells with mobile microrobots. *IEE Proceedings Nanobiotechnology*, 151(4):145 – 150, 2004.
- M.-S. Tsai and J.-S. Chen. Robust tracking control of a piezoactuator using a new approximate hysteresis model. *ASME J. Dyn. Syst., Meas., Control*, 125:96–102, 2003.
- V. Vogel and M. Sheetz. Local force and geometry sensing regulate cell functions. *Nature Reviews Molecular Cell Biology*, 7:265 – 275, 2006.
- T. Waram. *Actuator design using shape memory alloys*. T. C. Waram, 2nd edition, 1993.
- Y. Wu, S. Culter, Q. Zou, G. Pannozzo, K. K. Leang, and S. Devasia. Modeling and control of smart actuators: integrating nano/bio technology into the me curriculum. In *ASEE Annual Conference and Exposition*, Pittsburgh, PA, USA, 2008.
- M. Yamakita, A. Sera, N. Kamamichi, K. Asaka, and Z. Luo. Integrated design of IPMC actuator/sensor. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 2006, pages 1834 – 1839, Orlando, FL, 2006.
- W. J. Yoon, P. G. Reinhall, and E. J. Seibel. Analysis of electro-active polymer bending: A component in a low cost ultrathin scanning endoscope. *Sensors and Actuators A*, 133(2):506 – 517, 2007.
- K. Zhou and J. C. Doyle. *Essentials of robust control*. Prentice-Hall, Inc., 1998.

Q. Zou, K. K. Leang, E. Sadoun, M. J. Reed, and S. Devasia. Control issues in high-speed AFM for biological applications: collagen imaging example. *Asian Journal of Control*, 6(2):164–178, 2004.

## Appendix A: A Biaxial Shape Memory Alloy Actuated Cell/Tissue Stretching System

This chapter presents work that was published in the ASME 2007 International Mechanical Engineering Congress and Exposition, November, 2007, Seattle, WA. Authors of the paper are: Yingfeng Shan, Jacob Dodson, Sheena Abraham, John E. Speich, Raj Rao and Kam K. Leang.

### A.1 Introduction

Mechanical cues (such as strain and force) affect the state and behavior of cells, for example morphology, differentiation, and apoptosis (Spiegelman and Ginty, 1983; Chen et al., 1997; Bao and Suresh, 2003; Vogel and Sheetz, 2006). For instance, by simulating fetal breathing movement using a mechanical stretcher (Flexercell Strain Unit), the exposed fetal alveolar epithelial cells underwent differentiation and maturation (Di Palma et al., 2003). Likewise, mechanical stimulation from fluid flow has been reported to activate a specific signaling pathway in bovine articular chondrocytes (Hung et al., 2000). Stem cell differentiation has also been linked to mechanical strain (Saha et al., 2006).

To apply mechanical perturbations, a number of approaches have been proposed, such as designs that provide uniaxial stretching, while others provide equibiaxial (isotropic) strain (Schaffer et al., 1994); however, there is lack of a device with the ability for controlled anisotropic (multi-axial) strain. The contribution of this paper is presenting the design of an instrument to stretch cells cultured on top of a flexible membrane in two independent directions – longitudinal ( $x$ ) and transverse ( $y$ ) – for studies in cell biology. The instrument exploits the strain-recovery ability

of shape memory alloy (SMA) actuators (Waram, 1993) to deform a membrane, and the proposed design realizes high-resolution stretching via a feedback control system which compensates for the nonlinear behavior of the SMA actuators.

Controlled biaxial stretching is needed because certain cells, for example endothelial cells that line the inside of blood vessels, experience multi-axial (circumferential and longitudinal) stress and strain. The system of interest is designed to explore the influence of controlled biaxial stretching on the morphology of human fibroblasts. Additionally, the platform will be used to study the effects of controlled mechanical perturbations (in conjunction with other signals) on cell-fate decisions.

## A.2 Overview of the Design

To achieve controlled biaxial stretching of a membrane, the proposed design relies on four SMA actuators. Figure A.1 presents the concept of the instrument, where two sets of SMA actuators are arranged in an orthogonal configuration to mechanically stretch a membrane. As shown, an actuator is attached to an edge (four actuator/edge combination total) and the movement of each actuator is independently controlled. The SMA actuators are activated and controlled by varying the temperature of the shape memory material, *e.g.*, by heating the SMA using electric current (Joule Effect) to cause a phase transformation (from martensite to austenite) within the material that induces contraction (Waram, 1993). By using sensors to measure the deformation in the two directions, a control system can be augmented to precisely control the spatial (as well as temporal) precision of the instrument. One unique advantage of the proposed configuration is the ability for relatively large strain, in excess of 10%, simply by varying the ratio of the length of the membrane ( $L_1$ ) to the length of the actuator (*e.g.*,  $L_2$ ).

Existing stretching assays utilize a variety of mechanisms to apply mechanical strain on cells. For example, cells can be stretched using the micropipette aspiration

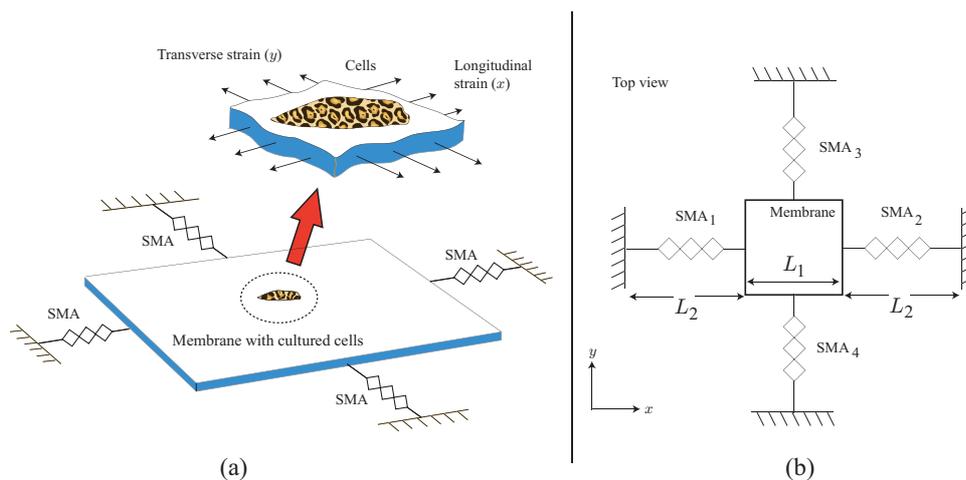


Figure A.1: The shape memory alloy (SMA)-based cell stretching platform. Left: Cells are cultured on an elastic membrane, for example polydimethylsiloxane (PDMS), then the membrane is stretched using SMA wire actuators. Right:  $SMA_1$  and  $SMA_2$  strain the membrane along the longitudinal direction while  $SMA_3$  and  $SMA_4$  strain it along the transverse direction.

technique (Palmer et al., 1996; Glenister et al., 2002), where two pressurized pipettes pull from opposite sides of the cell in a uniaxial fashion. In this method the pulling action is heavily concentrated on a small area (the diameter of the pipette). Furthermore, it is difficult to stretch a cluster of cells where each cell in the group experiences essentially the same mechanical perturbation. A similar approach that locally “pokes” a cell has also been used (Shiu et al., 1999; Bouten et al., 2001; Truper et al., 2004; Tan and Ng, 2003). The laser trap or optical tweezer is a non-contact approach to cell manipulation (Ashkin et al., 1986). This approach uses light to trap and move dielectric spheres attached to the cells of interest, but such a system is better suited to a small cluster of cells. Membrane-based devices are one of the most common to stretch cells (Bao and Suresh, 2003; Schaffer et al., 1994). One of the advantages of this method is simplicity – cells are cultured on top of an elastic membrane and then the membrane is stretched to transfer the mechanical perturbation (provided the shear stress at the cell-to-substrate interface is negligible). A number of devices

have been developed to exploit this concept, for example the commercially available Flexcell (Flexcell Int. Corp), piezoelectric-based stretchers (NASA, 2000), as well as fluid-based and those using pistons to deform a membrane (Schaffer et al., 1994). A recently proposed design was based on a lead-screw – turning the screw by hand caused the membrane to stretch (Jong et al., 2006). In general these systems deliver uniaxial or uniform biaxial strain and they provide limited control of the resolution of the applied strain (approximately 0.33% of the range of motion). Although the piezoelectric-based system can provide nano-scale resolution, one drawback of using piezos is they have relatively low strain (0.1 to 0.3% (Shahinpoor et al., 1998)) and require large electric fields for actuation (Tanaka, 1999).

In contrast to membrane-based methods used to stretch cells, the proposed design utilizes SMA actuators uniquely arranged to stretch a membrane/tissue biaxially. One of the advantages of SMA-based actuators is they provide relatively large strains (8%) compared to other smart material based actuators, for example piezoelectric (0.5%) (Shahinpoor et al., 1998). Furthermore, over this range, their movements can be precisely controlled (Tchoupo and Leang, 2007). Precise control of the strain over large ranges is needed for accurate studies of the effects of mechanical perturbation on cells. SMA actuators can also exert a significant amount of force per unit area, as much as 600 million Newtons per square meter (approximately 40 tons per square inch), compared to electroactive ceramics and polymers which deliver roughly one-tenth of this value. (The electroactive polymers such as ionic polymer metal composites (Shahinpoor et al., 1998) can provide large strain ( $> 10\%$ ); however, they deliver limited amount of force.) Also, SMA actuators require relatively low voltage range (tens of volts), whereas piezoelectric actuators operate in the hundred to thousand-voltage range. Furthermore, being a smart material, SMA have no moving parts which can wear, and they do not suffer from friction and stiction effects.

### A.3 Design Considerations

#### A.3.1 The Mechanical Design

Cells are mechanically stretched by culturing them on top of an elastic membrane, such as polydimethylsiloxane (PDMS), then the membrane is stretched to transfer the mechanical perturbation to the cells. (This design assumes that an appropriate interface exists between cells and the membrane to enable the transfer of strain.) Four SMA actuators (one-way type) — two in the  $x$ -direction and two in the  $y$ -direction — stretch the membrane biaxially (Fig. A.1). In our preliminary design each actuator (*e.g.*,  $SMA_1$ ) is constructed by wrapping a single SMA wire around small pins, back and forth, as shown in the Fig. A.2, to create a parallel set of  $n$  number of actuators with effective length  $L_2$ . One important aspect is the number of wraps,  $n$ , determines the total force exerted by the actuator. Wires, as opposed to SMA springs or plates, were chosen because they are readily available, inexpensive (approx. \$20US per meter, Flexinol®), and relatively easy to work with — that is, they can be cut to any length, easily bent to accommodate a desired configuration, and they are compact.

As illustrated in Fig. A.2, each set of parallel actuators share a common conductive point at their ends: one point is the fixed rigid support ( $A$ ) and the other point is the top of a cantilever beam ( $B$ ), where this fixture is also conductive and attaches to the membrane. Electric current is applied to heat (via Joule Effect) each set of parallel SMA wires. An increase in temperature (above the transition temperature) causes the wires to contract (up to 8%), therefore exerting a force to bend the cantilever. Subsequently, the bending of the cantilever transfers the strain to the membrane which is connected to the cantilever. When the wires are cooled (either by free or forced convection), the restoring force in the cantilever stretches the one-way SMA wires back to their initial length ( $L_2$ ) and the strain in the membrane is relieved.

In the proposed design, the conductive fixture at the end of the cantilever is designed with considerably more mass compared to the SMA wire actuators to act

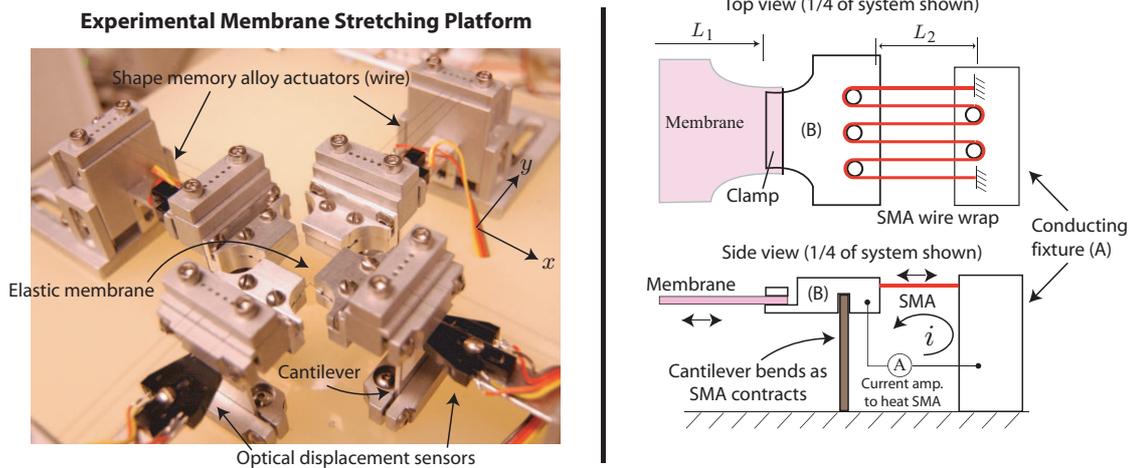


Figure A.2: Left: Detailed view of the SMA-based cell stretching platform. Four SMA actuators, two in the  $x$ -direction and two in the  $y$ -direction, stretch the membrane biaxially. Optical sensors are used to calibrate and measure the deflection of the cantilever. Also, they are used in the feedback control system to achieve accurate stretching of the membrane. Right: Each actuator is made by wrapping a single SMA wire to create a parallel set of actuators of length  $L_2$ . Current is applied to heat all the wires simultaneously, and the heat causes contraction, which bends a cantilever and subsequently stretches the membrane. As the wires are cooled, the restoring force in the cantilever stretches the SMA wires back to their initial state and the strain in the membrane is relieved.

as a heat sink. This is done to ensure that minimal thermal energy is transferred to the attached flexible membrane – a significant increase in temperature can potentially affect the specimen.

By using a sensor to measure the behavior of each set of SMA actuators (*e.g.*,  $SMA_1$ ,  $SMA_2$ ,  $SMA_3$ , and  $SMA_4$  shown in Figs. A.1 and A.2), precise strain along the longitudinal ( $x$ ) and transverse ( $y$ ) directions can be achieved using a control system. Additionally, the control system will enable precise tracking of user-defined strain profiles for dynamic studies.

One distinct advantage of this design is the ability to amplify the strain on the membrane by simply varying the ratio of the lengths of the membrane,  $L_1$ , and the SMA actuator,  $L_2$ . Therefore, relatively large percent strain can be achieved.

For example, in the longitudinal direction, where we consider  $SMA_1$  and  $SMA_2$  (see Fig. A.1(b)), the total distance between the two outermost conducting fixtures,  $L_2 + L_1 + L_2$ , remains constant; therefore, strain in the membrane  $\epsilon_1$  must equal the combined strain  $\epsilon_2$  of the two SMA actuators,  $SMA_1$  and  $SMA_2$ , *i.e.*,

$$\Delta L_1 = 2\Delta L_2. \quad (\text{A.1})$$

Rewriting (A.1) in terms of  $L_1$  and  $L_2$  defines the relationship between the strains  $\epsilon_1$  and  $\epsilon_2$ :

$$\epsilon_1 L_1 = 2\epsilon_2 L_2, \quad \Rightarrow \quad \epsilon_1 = 2\epsilon_2 \frac{L_2}{L_1}. \quad (\text{A.2})$$

Hence, the strain in the membrane  $\epsilon_1$  is proportional to the ratio  $L_2/L_1$ , and by simply increasing this ratio, we can mechanically amplify the strain in the membrane. For example, if the membrane and the SMAs have the same length ( $L_1 = L_2$ ), then the strain in the membrane  $\epsilon_1$  is equal to twice the strain of the SMAs. Assuming a nominal actuator percent strain of 5%, the membrane will strain 10% when the membrane and SMA wire actuators are the same length. Therefore, strain in excess of 10% can be easily achieved by making  $L_2/L_1 > 1$ . Also, the response time of SMA is relatively fast – they can contract in less than one second.

### A.3.2 Recovery Force Using a Cantilever

Above a critical temperature (typically 70 to 90° C), the one-way SMA actuator transitions from the martensite to the austenite phase and as a result, the material contracts – the contraction is exploited to stretch the membrane. However, when the wire is brought to low temperature a reversal of state occurs (austenite back to martensite) but a deformation force is required to recover its original length. Such a force can be generated using a variety of methods which include springs, counter weights, or cantilevers (Waram, 1993). Because of simplicity, we investigate the use of a vertically mounted cantilever, as shown in Fig. A.3, to provide the needed recovery force.

For a given cantilever, the number of parallel SMA wires,  $n$ , must be properly chosen to impart sufficient force such that the connected SMA actuator exercises its full range (typically 5% of the wire length  $L_2$ ). For a prismatic beam with fixed/free end conditions, assuming small angle deflections, the maximum deflection at the free end of the cantilever beam is proportional to the applied force  $F$  (Shingley et al., 2004):

$$\Delta x = \frac{Fh_2^3}{3EI}, \quad (\text{A.3})$$

where  $E$  is Young's modulus,  $I$  is the mass moment of inertia, and  $h_2$  is the distance measured from the base of the cantilever to the point of the applied force  $F$  (see Fig. A.3). In our design the length of the SMA actuator is  $L_2$ . Assuming a typical strain of 5% of  $L_2$  (*i.e.*,  $\Delta x = 0.05L_2$ ), then the total number of SMA wires,  $n$ , needed for a given cantilever with mass moment of inertia  $I$  is

$$n \triangleq \text{ceil} \left( \frac{F}{f_{rec}} \right) = \text{ceil} \left( \frac{0.15}{f_{rec}} \frac{EIL_2}{h_2^3} \right), \quad (\text{A.4})$$

where  $f_{rec}$  is the nominal force generated by a given diameter SMA wire (value provided by manufacturer as 'recommended' recovery force).

The deflection of each cantilever beam was measured using noncontacting infrared proximity sensors (Optek QRB1113). The sensor has a gain of 106  $mV/100 \mu m$ .

### A.3.3 Membrane Design Using Finite Element Analysis

The membrane material of choice is polydimethylsiloxane (PDMS, *e.g.*, Sylgard 184 silicone elastomer from Dow Corning), a flexible material commonly used to culture cells because of its ability to absorb proteins for cell attachment (Jong et al., 2006; Schwarz et al., 2002). (We note that the proposed design is compatible with other types of membrane material.) Finite element analysis (FEA, Cosmos package) was conducted to study the strain distribution on the three example membrane geometries shown in Fig. A.4. The geometries include a square with (I) rectangular, (II) radial

corners, and (III) chamfered corners removed. The membranes were strained along the longitudinal and transverse directions by pulling on the edges. The Young's modulus for PDMS is presented by (Armani et al., 1999), and it ranges between  $3.6 \times 10^5 Pa$  and  $8.7 \times 10^5 Pa$ .

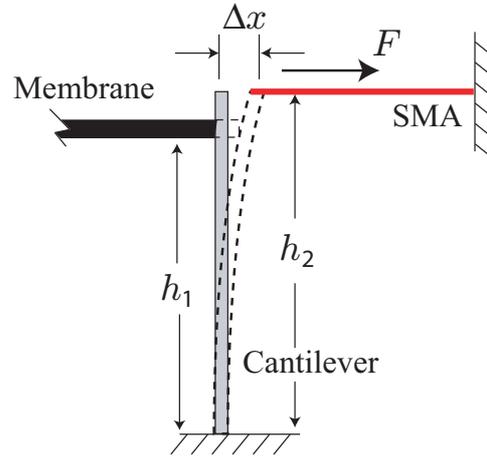


Figure A.3: Cantilever and SMA wire actuator. When heated, the SMA exerts a force  $F$  which in turn deflects the cantilever by  $\Delta x$ . The deflection is transmitted to the membrane attached to the cantilever.

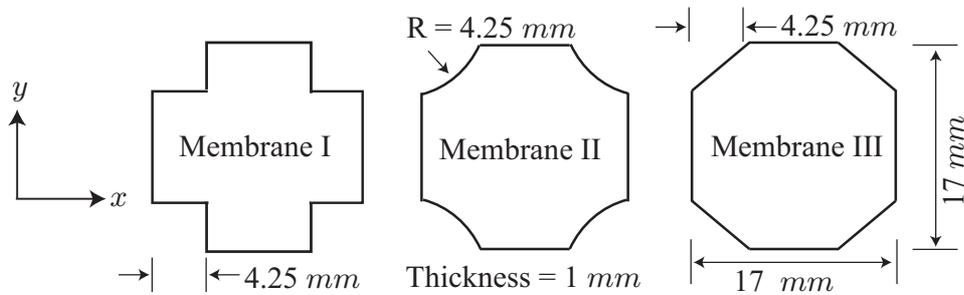


Figure A.4: Three membranes considered for FEA analysis of the strain distribution.

The results for the strain distributions for the example membranes area shown in Figs. A.5 and A.6; in particular, when the membrane is stretched in the  $x$ -direction and the  $y$ -direction is unconstrained (Figs. A.5(a) - (c)), the membrane contracts due

to the nonzero Poisson's ratio (0.49) as expected. We note that there exists a region at the center of the membrane (roughly 10% of the total surface area of the membrane) where the strain is uniform along the  $x$ -direction. Therefore, cells that are attached to this region will experience a uniform strain in the  $x$ -direction and likewise a uniform contraction in the  $y$ -direction.

On the other hand, when the membrane is constrained in the  $y$ -direction as it is being stretched in the  $x$ -direction (Figs. A.6(a) and (b)), our analysis shows that the contraction due to Poisson's ratio can be compensated. For example, Fig. A.6(b) shows nonzero strain in the  $y$ -direction due to the constraint. Therefore, the amount of strain in the longitudinal and transverse direction can be controlled by simply controlling the stretching of the membrane using the SMA actuators. Hence, controlled biaxial stretching can be achieved as indicated by the FEA results.

#### A.3.4 Control System Design

Although SMA can achieve relatively large strain, one of the main challenges is loss in positioning precision due to the effects of hysteresis (Dickinson and Wen, 1998). To compensate for hysteresis error, a feedback control system was designed to ensure precise straining of the membrane. Without control, the open-loop response (without membrane attached) for the SMA-based system shows nearly 30% distortion due to hysteresis error as shown in Fig. A.7, plot (e).

A proportional-integral (PI) feedback controller shown in Fig. A.8 was used to compensate for the hysteresis error. In the Laplace domain, the controller takes the form:

$$C(s) = \frac{n(s)}{d(s)} = k_p + \frac{k_i}{s}, \quad (\text{A.5})$$

where  $K_p$  and  $K_i$  are the proportional and integral gains, respectively. The gains were tuned experimentally to  $k_p = 200$  and  $k_i = 0.9$  by comparing the step response of the closed-loop system for different gain values. A comparison of the performance

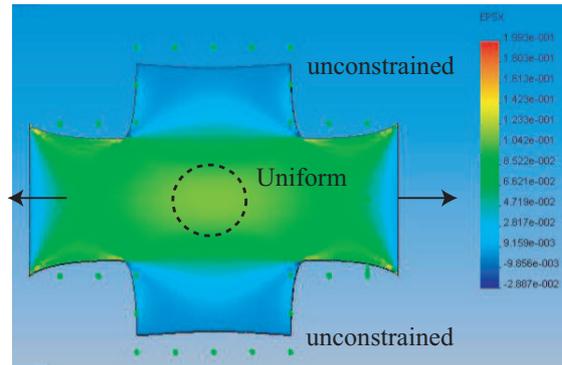
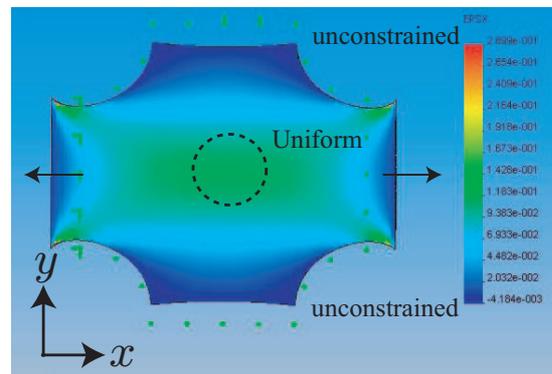
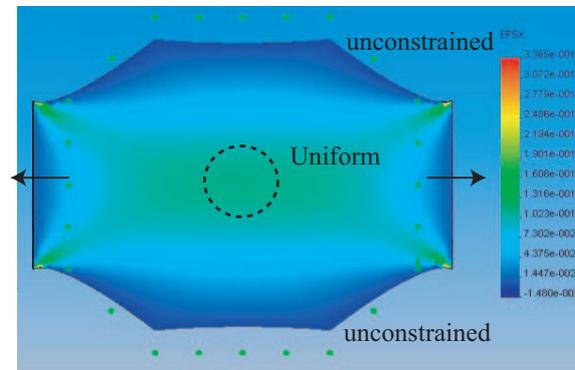
(a) Membrane I: Strain in  $x$ -direction;  $y$  unconstrained(b) Membrane II: Strain in  $x$ -direction;  $y$  unconstrained(c) Membrane III: Strain in  $x$ -direction;  $y$  unconstrained

Figure A.5: Finite element analysis results: strain distribution in the three example PDMS membranes. Membrane is stretched in the longitudinal ( $x$ ) direction. (a) - (c) unconstrained along transverse direction, therefore the membrane contracts.

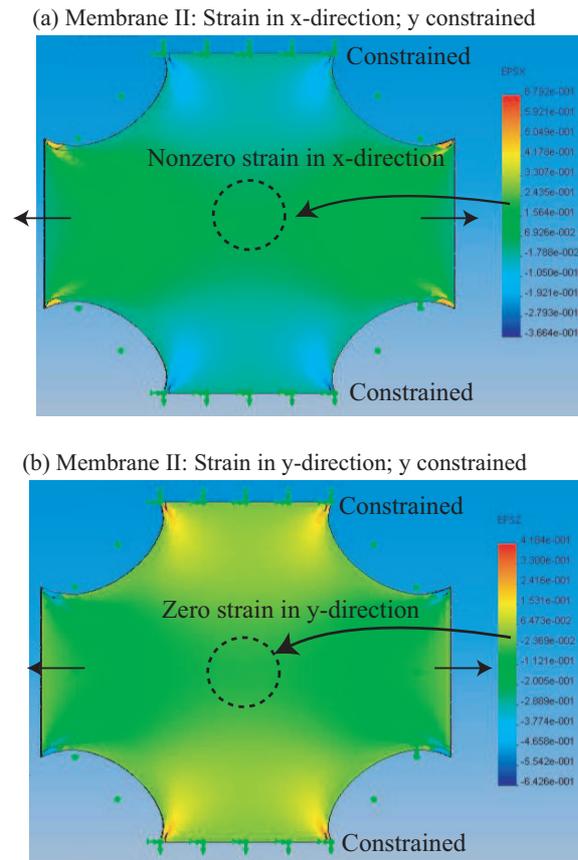


Figure A.6: Finite element analysis results: (a) and (b) constrained along the transverse direction, therefore zero strain in the  $y$ -direction.

of the PI control system with the open-loop response for  $SMA_1$  is shown in Fig. A.8. (We note that the other response of the other SMA actuators are similar and omitted for brevity.) In Fig. A.8, plot (b), the PI controller improved the response time of the actuator during contraction (*i.e.*, stretching) from 3.60 s for the open-loop case to 0.76 s. Also, by using the feedback control system, the SMA actuator can track a user defined stretching profile with significantly better precision compared to the open-loop case. For example, experiments were run to demonstrate tracking of a trajectory that transitions from zero to 1000  $\mu m$  in a one-second time period. The desired trajectory, the response of the open-loop system, and the response of the PI

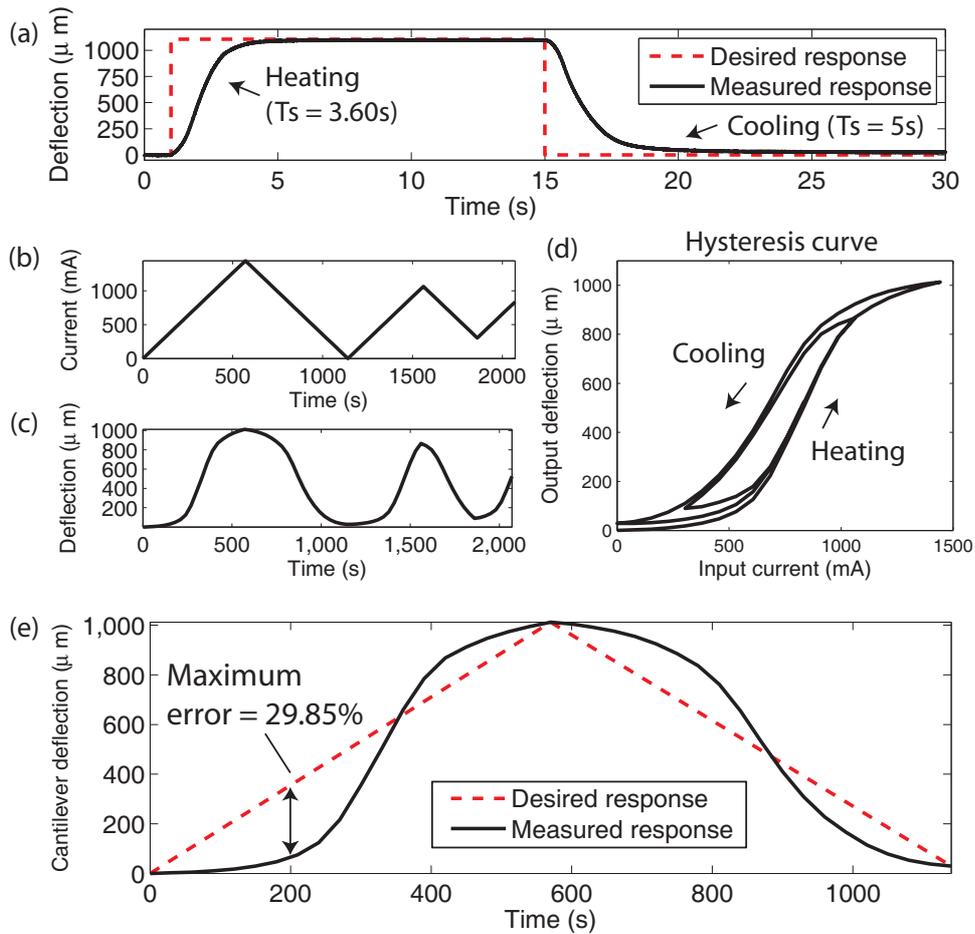


Figure A.7: Experimental results of example  $SMA_1$ : (a) Open-loop step response showing heating and cooling behavior of SMA. (b), (c), and (d), applied input current vs. time, measured deflection of cantilever beam, and hysteresis curve (deflection (output) vs. applied current (input)), respectively; (e) open-loop tracking error showing distortion due to hysteresis effect.

controller are shown in Fig. A.8, plot(c). In plot (d), the tracking errors between the open-loop and closed-loop cases are compared; the error is significantly lower using PI control, 2.7% versus 93.7% without the PI controller. We conclude that precision control of the SMA actuator can be achieved. Although the settling time was improved using PI control by over a factor of 4 (from 3.60 s open-loop to 0.76 s closed-loop), additional bandwidth can be achieved by using smaller diameter wires

and a heat sink to improve cooling (Loh et al., 2005).

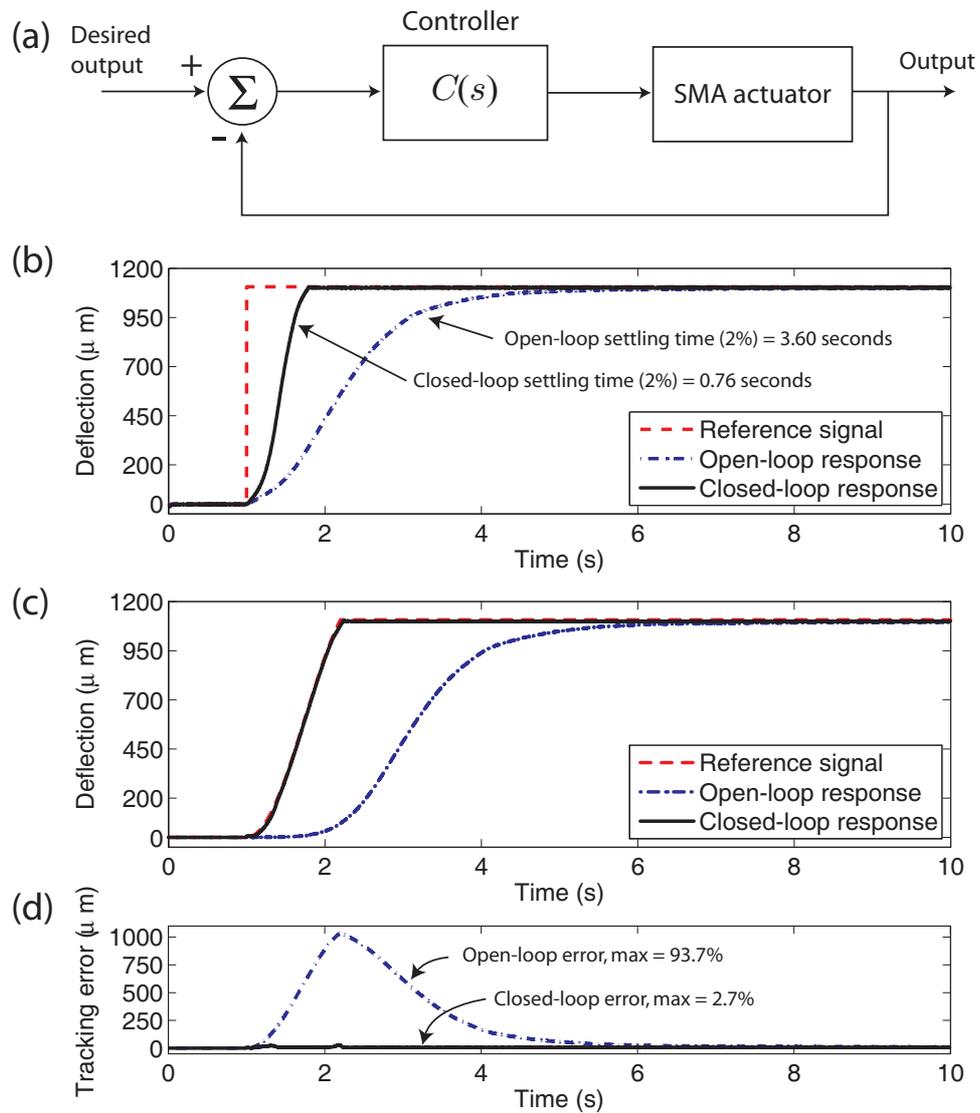


Figure A.8: Feedback control (PI) of SMA actuator for stretching PDMS membrane: (a) The feedback control system for each SMA actuator; (b) Open-loop and closed-loop

### A.3.5 Considerations for in Vitro Studies

The stretching platform is designed for real time monitoring of cells as they are subjected to controlled stretching conditions. In particular, the instrument's footprint was designed to be compatible with inverted light microscope (phase contrast) by allowing light to project from the bottom side of the instrument. Also, the system can be integrated with an AFM system to enable high-resolution measurement of morphology, changes in surface stiffness, as well as expression of proteins. Additionally, video-rate AFM (Schitter et al., 2006; Ando et al., 2006) can be used to capture in real-time the behavior of cells during the stretching process. The platform can be housed in an environmental chamber; likewise, the membrane can be submersed in a liquid environment, such as saline solution, during stretching.

## A.4 Experimental Results and Discussion

In this section we discuss the preliminary results to demonstrate stretching of an example PDMS membrane.

### A.4.1 Fabrication of PDMS Membrane

The PDMS membrane (Belanger and Marois, 2001) for the stretching platform is made using Sylgard 184 Silicon Elastomer, produced by Dow Corning. Sylgard 184 comes in two parts, the base and the curing agent. They are mixed together using various ratios, for example, 10 : 1, that is, ten parts base to one part curing agent, by weight. The ratio determines the stiffness of the membrane. The membranes are created in three-inch diameter petri dishes. Three pipettes are used when mixing and measuring the PDMS: (1) one for the curing agent, (2) one for the base, and (3) another for the PDMS mixture. The curing agent is weighed in a 100 *ml* beaker. Base is then added to the curing agent to obtain the desired concentration. The base and curing agent are stirred together using a glass rod for 3 to 4 minutes to ensure

uniform mixture. The PDMS mixture is then dispensed into the petri dishes using a sterile pipette. Afterwards, the petri dish is covered and allowed to cure at elevated temperature ( $75^{\circ}\text{C}$ ) for 1 hour 15 minutes.

#### A.4.2 Uniaxial Stretching of PDMS Membrane

We demonstrate qualitatively biaxial stretching of a PDMS membrane. These initial experiments do not involve live cells. After curing, a 1 *mm* thick membrane was cut into a desired geometry, for example, shape Membrane I (see Fig. A.4). Then, the membrane was secured in the stretching system. A CCD camera with resolution of 2  $\mu\text{m}$  was used to capture video and still-images of the stretching process. These results were used to qualitatively assess the initial performance of the instrument.

The first experiment was to demonstrate uniaxial stretching (along the  $x$ -direction) of the membrane. The membrane was stretched using two SMA actuators,  $SMA_1$  and  $SMA_2$ . The membrane was unconstrained in the transverse direction (along  $y$ -axis). To simulate cells attached to the membrane and also to provide a visual mark for estimating the amount of stretch, black ink was placed on the membrane and monitored by the CCD camera. Figure A.9 shows the uniaxial test. The ink marks are shown as vertical lines, and a small 37  $\mu\text{m}$  diameter wire was positioned on top of the membrane for size reference. From the images, the estimated strain was 6.25%. This value was obtained by comparing the relative spacing between the vertical ink marks before and after stretching. These preliminary results suggest the possibility that cells cultured on top of the membrane can be stretched.

#### A.4.3 Biaxial Stretching of PDMS Membrane

In the second experiment, we demonstrate biaxial stretching (along the  $x$  and  $y$  directions). Similar to the previous experiment, black ink was placed on the membrane and monitored by the CCD camera. In this experiment, the round ink marks were used as shown in Figure A.10. Also, a small 37  $\mu\text{m}$  diameter wire was positioned

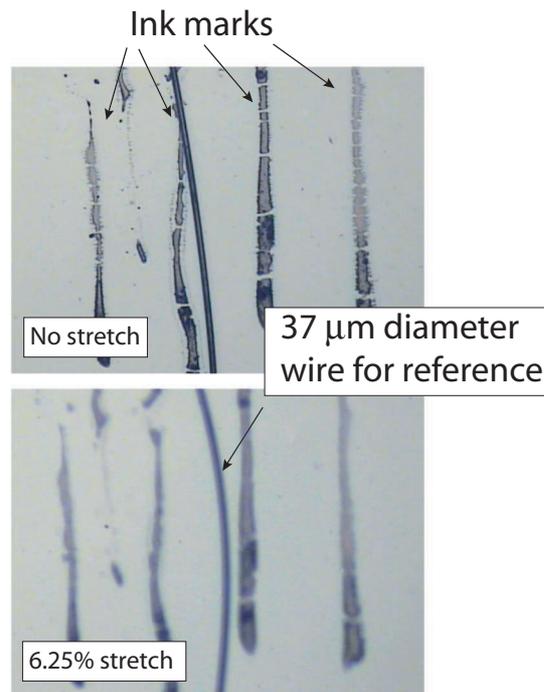


Figure A.9: Experimental results: uniaxial stretching of PDMS membrane.

on top of the membrane for size reference. The images shown in Figure A.10 were acquired at approximately 1 s intervals. The arrows indicate which SMA actuator was activated. As shown the figure by the relative positions of the ink marks with respect to the fixed camera between frames (a) to (i), there is evidence of the membrane being stretched in two directions. We also note the ink marks elongated as the membrane as stretched. In this case, the maximum percent strain (along the longitudinal direction) was estimated to be over 6% by comparing before and after images.

## A.5 Conclusions

The design of a biaxial cell stretching system based on SMA actuators was presented along with experimental results. The design utilizes four SMA actuators arranged in an orthogonal direction to stretch a flexible membrane with cells cultured on top in

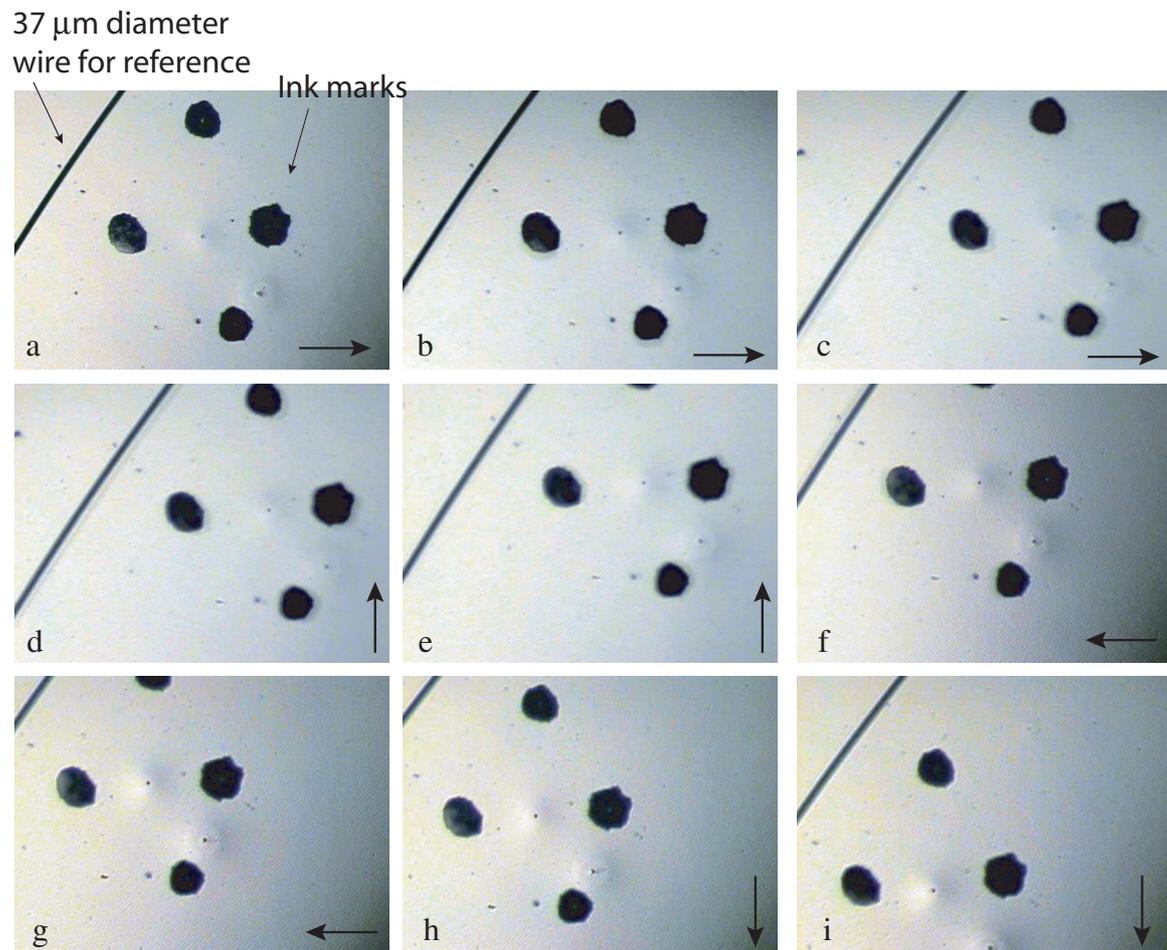


Figure A.10: Experimental results: biaxial stretching of PDMS membrane. Arrow indicates the direction of strain.

the longitudinal and transverse direction. We showed the application of a feedback controller to enable high-precision stretching by compensating for the nonlinear hysteresis behavior in SMA actuators. Preliminary results were presented to show over 5% uniaxial and biaxial stretching of an example PDMS membrane.

## **A.6 Future Work**

The preliminary experiments provide encouraging results and future work includes stretching of fibroblast cells and monitoring the effects using both optical microscopy (phase contrast) and the atomic force microscope. For example, we will consider human foreskin fibroblasts cultured on the PDMS surface that is coated with  $10 \mu\text{g}/\text{ml}$  human fibronectin (Sigma, MO) to support cell adhesion. The platform will be used in long-term studies the effects of controlled mechanical perturbations (in conjunction with other signals) on cell-fate decisions.

## Appendix B: Low-Cost Optoelectronic Sensors for Sub-Micro-Level Position Measurement and Control

This chapter presents an article submitted for publication to the IEEE Transaction on Mechatronics. Authors of the paper are Yingfeng Shan, John E. Speich, and Kam K. Leang.

### B.1 Introduction

The measurement and control of mechatronic systems, especially those involving *active* material-based actuators such as piezoelectric actuators (piezoactuators or piezopositioners), over small displacements are extremely important in emerging micro/nanotechnology (Devasia et al., 2007). Piezoactuators displace over the micro and nano length scales with extremely high resolution (Perez et al., 2005), and their ability for fine movements has been exploited in scanning probe microscopy (SPM)-based systems, such as the atomic force microscope, to position a sample relative to a probe tip for imaging (Leang and Devasia, 2006) and in fabricating micro/nano-sized objects and features (Davis et al., 2003). Precision positioning is achieved by measuring the displacement of the piezoactuator and applying control techniques (Devasia et al., 2007). Therefore, the ability to measure the fine motion of these actuators is extremely important for ensuring high-performance operation of SPMs, as well as other piezo-based positioning systems. With the continued growth of emerging micro/nanotechnology, low-cost methods to sense the movement of these actuators is important from an economic and commercial standpoint. The contribution of this article is a simple method to measure displacements down to the sub-micro level using readily available and inexpensive infrared reflective sensors typically used in printers

and optical encoders (see Fig. B.1). These sensors are not specifically designed for this purpose, but it is shown they can be implemented to measure micro-scale displacement using a small number of discrete components. One unique advantage of these sensors is they are lower in cost compared to commercially available inductive and capacitive sensors and interferometers. The latter class of sensors is typically priced over hundreds of dollars compared to the proposed low-cost alternative at less than 10 USD. The following study quantifies the operating range, resolution, linear distortion, and bandwidth of the sensors. The performance is compared with a commercially available inductive sensor, and the results show comparable performance over the same operating conditions. To illustrate their application, an example sensor is used in the design of a state-feedback controller to control the movement of an experimental piezopositioner. Such sensors are also attractive for educational laboratory experiments in mechatronics and controls courses (Wu et al., 2008).

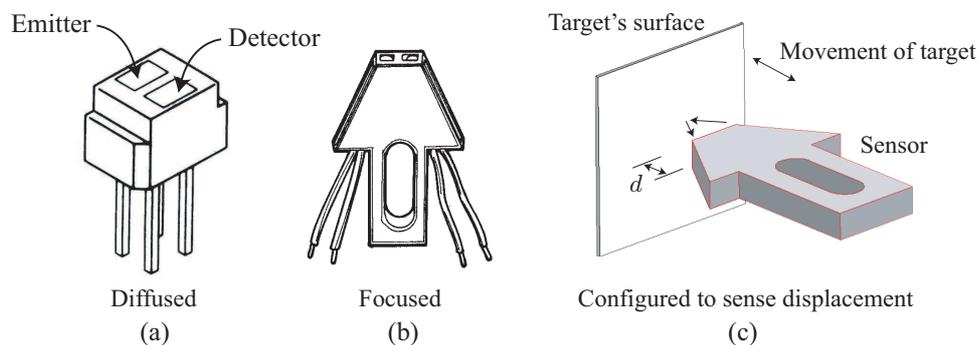


Figure B.1: Infrared reflective sensors: (a) diffused, (b) focused, and (c) configured to detect target distance (*i.e.*, gap)  $d$ .

The micro/nanoscale movements of a piezopositioner or other fine-positioning device can be measured using a wide variety of sensors, including both contacting and noncontact types. For instance, contact-type sensors include (inductive) linear variable differential transducers (LVDTs) (Chang et al., 1999) and (resistive) strain

gauges (Schitter and Stemmer, 2002; Tsai and Chen, 2003). To sense displacement, these sensors either make physical contact with or are attached to the object of interest. In some cases, however, physical contact may not be practical, for example, in hard-to-reach places or when the object is fragile and prone to damage through physical contact. Likewise, physical contact may impede (or alter) the natural behavior of the device.

Noncontact sensors are useful in situations where physical contact between the sensor and the device is not possible and/or beneficial. These sensors operate on capacitive (Baxter, 1997), inductive (Bartoletti et al., 1998), magnetic (Lee and Velinsky, 2007), or optical principles (Hosoe, 1991). For example, capacitive sensors have been used in SPMs (Schitter and Stemmer, 2004; Holman et al., 1996) and with microactuators (Kuijpers et al., 2004) to detect translational and rotational motion. In principle, capacitive sensors have very high-resolution ( $< 0.01$  nm); however, they tend to be sensitive to surface irregularities, changes in temperature, and humidity (Baxter, 1997). Furthermore, the range of detectable motion of capacitive sensors is relatively short. The inductive sensor is frequently used to measure displacements of piezopositioners (Croft et al., 1998) as well as other actuators, such as stepper motors (Passeraub et al., 1998). These sensors work under the electrical principle of inductance, when a ferrous or non-ferrous metallic object passes through the electromagnetic field of a coil wound around a ferrous material, the displacement induces current which is related to the relative position between the object and the coil (Kekjik et al., 2004; Bartoletti et al., 1998). These sensors have relatively high-resolution (nanometer) and good bandwidth (tens of kHz), with the added advantage of being immune to dirt, water, and lubricating oil. A laser interferometer is another commonly used displacement sensor for nano-scale applications (Hosoe, 1991). This type of sensor provides high-resolution measurement over large range, but like capacitive and inductive sensors, they are generally expensive (over hundreds of dollars) and require special signal-processing circuitry to operate.

Reflective optical proximity sensors offer comparable performance to inductive and capacitive sensors in terms of resolution and bandwidth. In an optical sensor, a source emits light which subsequently reflects off a target's surface and the reflected light is sensed by a detector. The intensity of the reflection is related to an object's distance from the detector (Benet et al., 2002). Optical sensors can be relatively inexpensive (tens of dollars) compared to their capacitive and inductive counterparts (over hundreds of dollars). Additionally, these sensors can be used for other applications such as to detect the blood temperature during coronary bypass implantation (Giovannetti et al., 2005), as well as for transmitting/receiving data (Dietz et al., 2003).

The contribution of this work is investigating the use of low-cost (<10 USD), commercially available IR optical sensors (optosensors) for detecting micro to sub-micro-level displacements. These sensors are not specifically designed for this application, but rather they are commonly used as optical switches in printers and optical encoders (Optek Technology Inc., 1989). Their use as submicro-level proximity sensors for detecting the motion of piezopositioners or other fine-positioning actuators has not been explored. By carefully calibrating and quantifying their performance, it is shown that the sensors can be used to measure and control the displacement of a piezopositioner.

The remainder of this paper is organized as follows. Section B.2 describes the principle of operation for the reflective optosensor, and Section B.3 presents the experimental results which highlight the performance of the sensor in terms of operating range, resolution, linear distortion, and bandwidth. With an understanding of the sensor's performance, Section B.4 describes the application of an example sensor in the design of feedback control system for a bimorph piezoelectric actuator, and concluding remarks follow in Section B.5.

## B.2 Reflective IR Sensors

The reflective IR sensors considered in this article consist of a coupled infrared optical pair — transmitter (*e.g.*, IRLED) and detector (*e.g.*, phototransistor) — mounted in a reflective configuration. They are designed to recognize the presence or absence of an infrared reflective surface as shown in Fig. B.1(c). They are often used in printers and copiers for sensing the presence of paper or as optical switches and encoders (Optek Technology Inc., 1989).

To sense proximity, the emitting/detecting plane of the IR sensor is oriented (usually parallel) to a target's surface as depicted in Fig. B.1(c). The emitter radiates IR light which subsequently reflects off a target's surface and the reflected light falls on the nearby detector. When the *gap* (distance between the front of the sensor and target's surface, known as target distance and denoted by  $d$ ) is zero, no light can escape from the emitter, thus the detector senses no light. As the gap increases, the surface of the target is illuminated and the reflected light strikes the detector. The detector output is related (nonlinearly) to the gap distance  $d$ . The range of detectable motion for these types of sensors can be over 20 cm (Ando and Graziani, 2001). However, this investigation focuses on using sensors that operate within 2.0 cm range. The smaller-range sensors are preferred because they are more sensitive to motion in the preferred range of operation for piezoactuators.

The IR sensors are conveniently packaged in a plastic housing and readily available in two types as shown in Fig. B.1: (a) diffused and (b) focused. The diffused package is configured such that the optical transmitter radiates in a pattern parallel to the optical detector. The diffused sensor can recognize objects in a wider variety of distances. In contrast, the focused package is designed with convex lenses and the transmitter and detector pointing at the same point at a specified distance in front of the device package. This design enhances the ability of an infrared surface at or about the specified distance to reflect a sufficient signal (Optek Technology Inc., 1989).

The optical emitter can be an IRLED or a vertical cavity surface emitting laser (VCSEL). A VCSEL has a tight area of light emission thus making the focused distance more critical than with a LED. The VCSEL also has a higher light intensity which makes it easier for the optical detector to recognize the presence of a reflecting surface. Additionally, VCSELs generate more optical light (power in milli-Watts, mW) than the LED, requiring a much lower drive current. The optical detector can either be a photodiode, a phototransistor, a photodarlington, or a photologic device. The rise time of the detectors are ordered as follows: photodiode < photodarlington < phototransistor. Although photodiodes offer very fast rise times (as fast as 10 ps), they require more complex circuitry for operation.

### B.3 IR Sensor Performance

Five commercially available sensors as single-unit purchases were examined as listed in Table B.1. There are essentially two methods to estimate the distance of an object in front of a sensor: model-based approach or calibration curve. The former approach involves modeling the sensor's behavior, for example the inverse square law (Benet et al., 2002) or the Phong Model (Phong, 1975). The model-based approach requires the identification of parameters, and uncertainty in any of the values will result in uncertainty in the estimated distance. The second option is to obtain and use a calibration curve – this is the method used in this investigation. This method determines an empirical relationship between the gap distance  $d$  and the detector output voltage to estimate proximity. A commercially available inductive sensor (Kaman SMU0-9000-15N001) was used for comparison. According to the manufacturer's specifications, the inductive sensor has a range of 500  $\mu\text{m}$  with a resolution of 5 nm at 100 Hz and a bandwidth of approximately 10 kHz.

### B.3.1 Sensor Calibration, Operating Range, and Sensitivity

Each IR sensor listed in Table B.1 was first calibrated to determine its behavior (output voltage,  $V_s$ ) as a function of target distance  $d$ . Of the sensors listed in Table B.1: one was a diffused type and four were the focused type (see Fig. B.1). These sensors cost less than 10 USD/each, and they require basic circuitry for operation.

A test system was constructed to calibrate each IR sensor, as well as the reference inductive sensor. A photograph of the experimental calibration system is shown in Fig. B.2, where the inset photo shows additional details of the target carriage assembly. The setup consists of a veneer-scale micrometer [25.4 mm (1 inch) range] for positioning a target in front of a subject sensor fixed at location  $P$ . The target was a thin and rigid rectangular plate covered with white 90% reflective paper (Kodak paper #KOD1527795). The target was also attached to a carriage which travels on a linear bearing; the carriage made contact with the micrometer shaft via magnetic contact. The magnetic contact enables the micrometer shaft to rotate relative to the carriage. It is noted that if the target rotates as it translates toward/away from the sensor, the slightest wobble in the target was detected by the sensor as a change in distance  $d$ , which appears as oscillations in the sensor's output signal. By using the magnetic contact with the carriage assembly that travels on the linear bearing, the target translates without rotating relative to the front of the sensor – clockwise and counter clockwise rotation decreases and increases, respectively, the gap distance  $d$ . One full revolution of the micrometer translates the target  $625\ \mu\text{m}$  (0.025 inches) relative to the sensor. The translational resolution was determined by a stepper motor attached to the micrometer shaft via a flexible coupler. With a 400 steps per revolution ( $0.9^\circ$  per step) stepper motor, with each step the target translated  $1.56\ \mu\text{m}$  relative to the sensor.

In the circuit diagram shown in Fig. B.3, the current supplied to the infrared LED emitter is limited by resistor  $R_e$ . The detector is configured as a voltage divider,

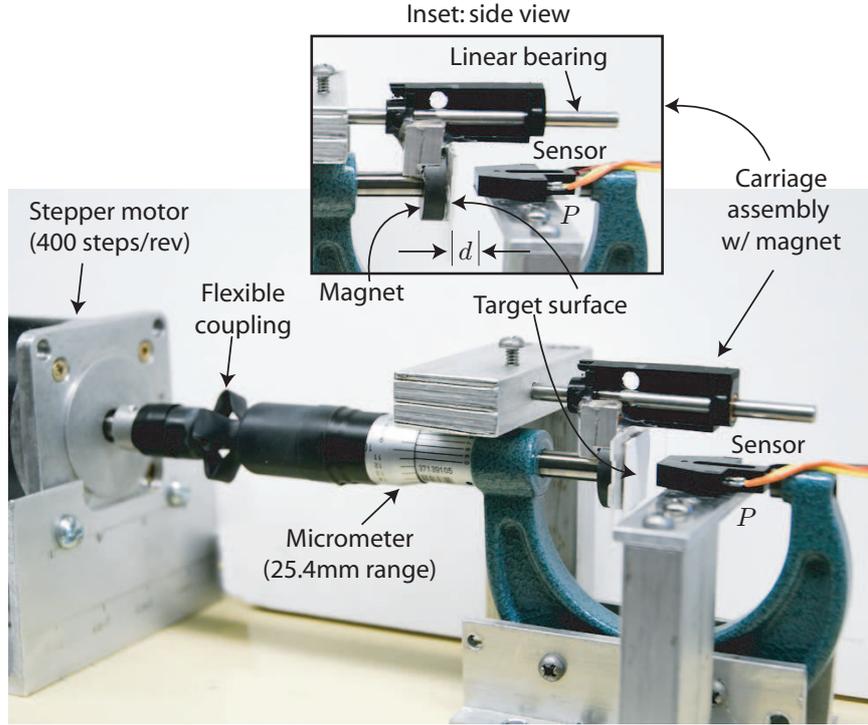


Figure B.2: The calibration system.

and without light the output of the detector  $V_s$  is approximately the supply voltage (+5 VDC). When light is detected, the sensor's output  $V_s$  varies between the supply voltage and ground. When mentioned, the sensor output  $V_s$  was low-pass filtered through  $G_f(s)$ :

$$G_f(s) \triangleq \frac{\hat{V}_s(s)}{V_s(s)} = \frac{a}{s + b}; \quad \text{units} \left[ \frac{\text{Volts}}{\text{Volts}} \right], \quad (\text{B.1})$$

where  $a = R_2/(R_1R_3C)$ ,  $b = 1/(CR_4)$ , and “ $s$ ” is the Laplace variable. The filter reduces the effects of high-frequency noise and it produces the filtered signal  $\hat{V}_s$  (in volts).

Each sensor was calibrated ten-times over a range of 12,000  $\mu\text{m}$  (0.5 inch) in increments of 1.58  $\mu\text{m}$ . The average results are reported, where the maximum

Table B.1: Low-cost optoelectronic sensors considered in this study.

Sensor Part No.	Type	Detector	$I_f^\dagger$	$\max\{t_r, t_f\}^\ddagger$	$R_c^\S$
1. Optek OPB703	Focused	Phototransistor; no lens	18 mA	200 $\mu s$ ( $R_c = 15 k\Omega$ )	15 k $\Omega$
2. Optek OPB704	Focused	Phototransistor; blue polysulfone lens (dust protection)	18 mA	200 $\mu s$ ( $R_c = 15 k\Omega$ )	15 k $\Omega$
3. Optek OPB706	Diffused	Phototransistor	18 mA	30 $\mu m$ ( $R_c = 1.2 k\Omega$ )	1.2 k $\Omega$
4. Fairchild QRB1113	Focused	Phototransistor	18 mA	8 $\mu s$ ( $R_c = 100\Omega$ )	4.7 k $\Omega$
5. Fairchild QRB1134	Focused	Phototransistor	18 mA	8 $\mu s$ ( $R_c = 100\Omega$ )	3.9 k $\Omega$

$^\dagger$ Emitter forward DC current;

$^\ddagger t_r$  and  $t_f$  denote rise and fall time, respectively, provided by manufacturer;

$^\S$ Detector load resistance value used in circuit diagram shown in Fig. B.3.

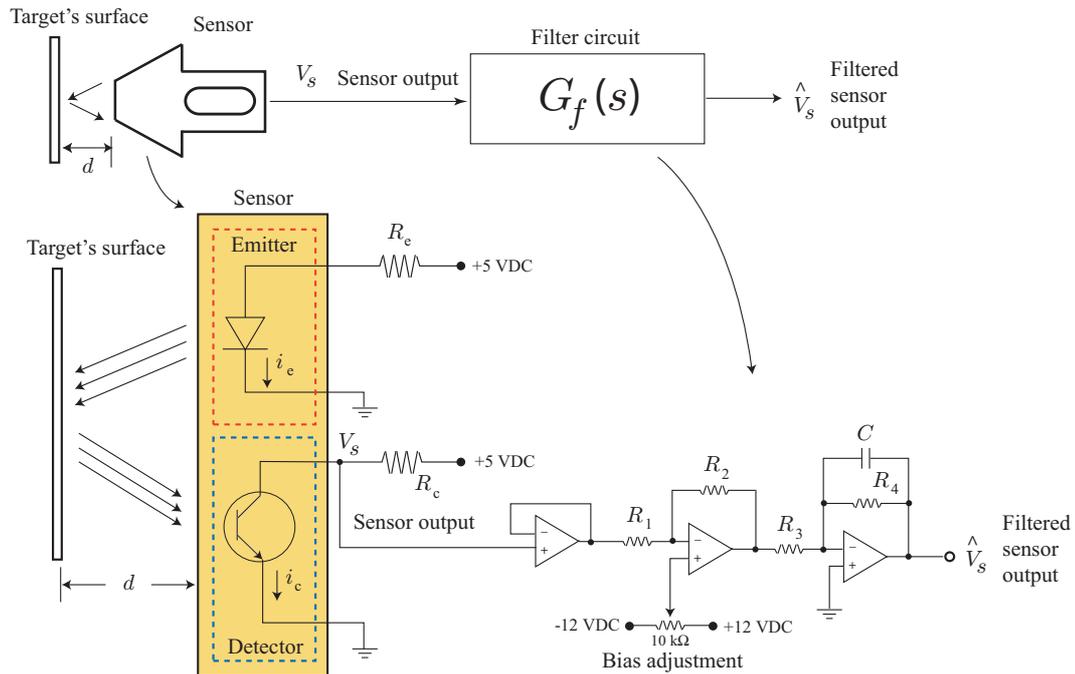


Figure B.3: Reflective displacement sensor circuit.

standard deviation for the output signals of the diffused and focused sensors were 8.1 mV and 64.0 mV, respectively (noise floor of the electric signal was  $\approx 10$  mV). The output voltage  $V_s$  versus the target distance  $d$  for each IR sensor is shown in Fig. B.4, plot (a), and Fig. B.4, plot (b) shows the normalized sensitivity (derivative of  $V_s$  with respect to  $d$ ) versus target distance  $d$ . The normalized scale in plot (a) only identifies the useable range of the sensor, and is not to compare sensitivity magnitudes between sensors. The sensitivity plot (b) shows the range in which the sensor's output is most responsive to changes in the gap separation  $d$ . Also, this information can be used to determine the optimum operating range, for example, in terms of sensitivity, sensor No. 4 (Table B.1) is most effective in the neighborhood of  $d = 1.5$  mm. It is noted that each sensor has a point of maximum output response, for example, sensor No. 4 the distance is  $d = 3.81$  mm. It is not advisable to use the sensor in this region because

the phase of the output signal may reverse in direction beyond this point. Instead, it is preferable to use the sensor in the region where the sensor is most sensitive to change in displacement, such as  $d = 1.5$  mm as shown in Fig. B.4(b). The results also indicate that the diffused package (No. 3, OPB706) at the point of maximum sensitivity is approximately 6-times more sensitive than the focused version (*e.g.*,  $9.9 \text{ mV}/\mu\text{m}$  for diffused compared with  $1.6 \text{ mV}/\mu\text{m}$  for focused). Additionally, the effective sensing range of the diffused version is approximately 5-times less compared to the focused package (*cf.* results in Figs. B.4(a) and (b)). Furthermore as shown in Fig. B.4(a), as the gap distance increases beyond the point where the minimum value of the detector output occurs, the detector output exhibits the typical  $1/d$  behavior.

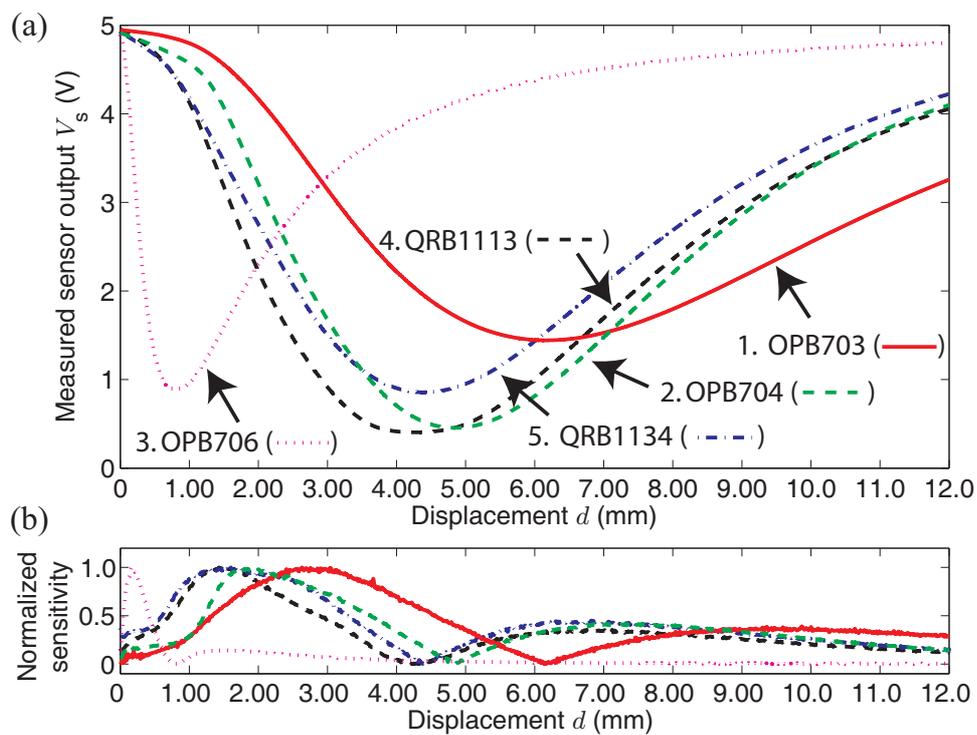


Figure B.4: Sensor calibration: (a) Sensor output voltage  $V_s$  versus target distance  $d$ . (b) normalized sensitivity (derivative of  $V_s$  with respect to  $d$ ) versus target distance  $d$ .

It is noted that the phototransistor detectors exhibit a broad spectral response, with a peak in the infrared range (900 to 1000 nm). The IR LED's peak emission occurs in a narrow band of the infrared range to complement the phototransistor's peak response to minimize the effect of ambient light. However, the effects of ambient light should be carefully considered in application.

### B.3.2 Linear Distortion

The linear distortion (quantified as a percentage, where the smaller the value the more linear the response) was calculated over a range centered about the maximum sensitivity of each sensor (see Fig. B.4, plot (b) that shows the distance  $d$  in front of the sensor where the maximum occurs). The range of interest relative to the point of maximum, denoted by  $d_0$ , was  $d_0 \pm 200 \mu\text{m}$ . This range is of interest because piezoelectric actuators are typically operated over this range, therefore, studying the behavior of each sensor over this range gives valuable information. Each sensor's output voltage  $V_s$  versus target distance relative to  $d_0$  was fit to a linear equation of the form

$$V_s(d) = md + b, \quad (\text{B.2})$$

where  $m$  and  $b$  are constants. The constants  $m$  and  $b$  were determined by least-squares fit for each IR sensor. Afterwards, the percent of linear distortion was determined by comparing the measured sensor output  $V_s$  with its linear fit Eq. (B.2), *i.e.*,

$$\% \text{ linearity} = \frac{\max |V_s(d) - (md + b)|}{|\max(V_s) - \min(V_s)|} \times 100 \quad (\text{B.3})$$

The results are summarized as follows: 1. OPB703, 0.35%; 2. OPB704, 0.73%; 3. OPB706, 1.64%\*; 4. QRB1113, 0.44%; 5. QRB1134, 0.18%; Inductive sensor, 4.51% (2.35%)\*. It is noted that for sensor No. 3 and the inductive sensor, the "\*" denotes that the percentage was calculated over  $d_0 + [0, 100] \mu\text{m}$  range.

The percent of linear distortion is less than 1% of full-scale ( $\pm 200 \mu\text{m}$ ), with the exception of sensor No. 3 (diffused package). The results show that the IR sensor

has lower linear distortion than the inductive sensor. For improved measurement precision, calibration can be done by curve fitting the sensor's output response using high-order polynomials and creating a look-up table to relate the output voltage with the distance  $d$ . Also, the linear distortion can be reduced by modifying the sensor's output using nonlinear amplifiers, for example as described in (Sanyal et al., 2006).

### B.3.3 The Effect of Target Size

The sensors' responses to different target sizes were compared to the responses from a sufficiently-large reference size. The diffused sensor has area  $6\text{ mm} \times 4.3\text{ mm}$  and the focused has area  $9.5\text{ mm} \times 5.3\text{ mm}$ . The reference size is referred to as the *nominal* target size with dimensions of  $25.4 \times 25.4\text{ mm}^2$  ( $645.2\text{ mm}^2$ ). Furthermore, the sensor was centered relative to the target for all experiments. The results are shown in Fig. B.5 where plots (a1) and (a2) show the response of the diffused IR sensor (No. 3) for different target surface areas; likewise, plots (b1) and (b2) are for the focused IR sensor (No. 4) under similar conditions. Plot (c) compares the maximum difference in the sensor output from one size to the next and it is reported as the percent of the total range. The diffused sensor requires the target to be relatively large compared to the sensor's area, over 6-times larger. On the other hand, the focused sensor can be used with smaller target surface area, and the target's surface area should be at least double the surface area of the focused sensor.

### B.3.4 Sensor Resolution

The resolution was investigated for one IR sensor, No. 4 (Table B.1), as an illustrative example. The approximate resolution was determined by exploiting the nano-resolution positioning capabilities of a bimorph piezoactuator. Piezoelectric actuators can displace with sub-nanometer resolution, and by measuring the fine displacements of the actuator using the inductive sensor, while simultaneously measuring the same displacement using the low-cost IR sensor, a comparison can be made to infer the

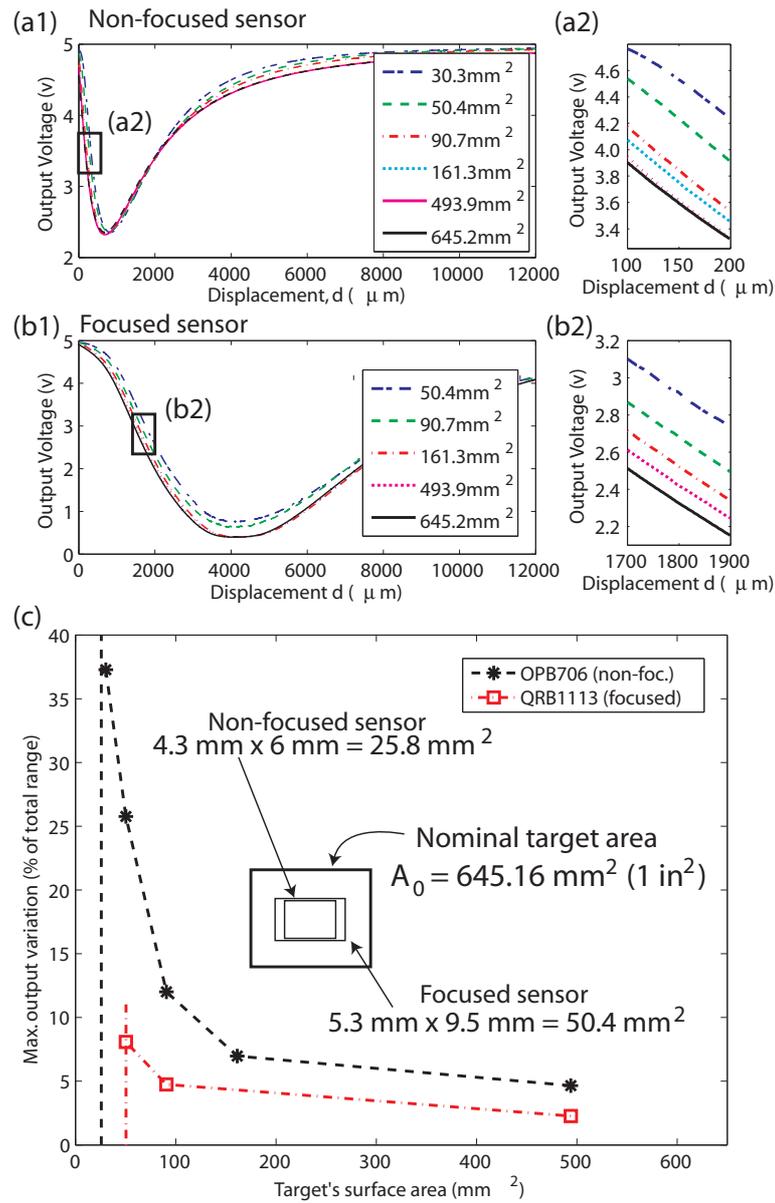


Figure B.5: The effect of target area on the IR sensor's output. (a1) with inset (a2) shows the response of the diffused IR sensor for different target surface areas; likewise, (b1) with inset (b2) shows the response of the focused IR sensors under similar conditions; and (c) shows the maximum variation in the output response of the diffused and focused IR sensor relative to the response for the nominal-sized target.

approximate IR sensor's resolution. The experimental setup is shown in Fig. B.6, where a bimorph piezoactuator is mounted in a rigid fixture. An input voltage  $u$  from a desktop computer was applied to a high-voltage amplifier (Trek, model PZD700) and the output of the amplifier was used to drive the piezoactuator. As a result, the piezoactuator displaces along the  $x$ -direction, where the displacement was measured with the inductive sensor and IR sensor (No. 4). On the side with the IR sensor, white 90% reflective paper with area  $9.5 \text{ mm} \times 5.56 \text{ mm}$  was secured to the piezoactuator as shown in Fig. B.6. The calibration curve in Fig. B.4 was used to determine the displacement of the positioner along the  $x$ -axis.

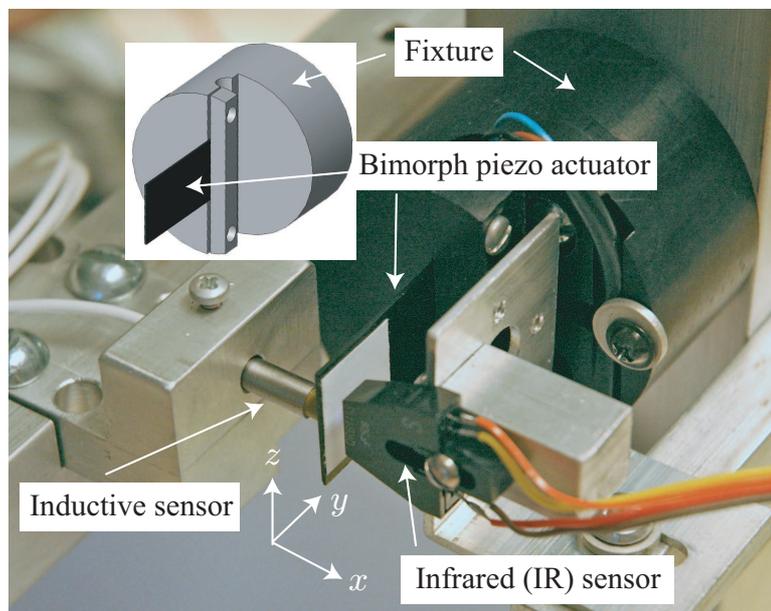


Figure B.6: The experimental bimorph piezoactuator. An input voltage causes the piezo to bend along the  $x$  axis. On one side of the actuator is an inductive sensor to measure the displacement. On the opposite side is a low-cost IR optoelectronic sensor (No. 4, QRB1113) which faces a white target (Kodak paper #KOD1527795) attached to the piezoactuator. Both the inductive and IR sensor measure the displacement of the piezoactuator along the  $x$  axis.

The resolution of the IR sensor was determined by applying a series of staircase inputs to actuate the piezo bimorph, and with each run, the step size was decreased until the IR sensor was unable to detect the piezopositioner's movement. The output of the IR sensor,  $V_s$ , was not filtered, but amplified to maximize the data acquisition's resolution. The response of the inductive and IR sensor are shown in Fig. B.6; the plots (a1) to (a3) on the left column show the response measured by the inductive sensor (resolution 5 nm) and the plots (b1) to (b3) on the right column show the same response measured by the IR sensor. Comparing plots (a3) and (b3), a discernible change in response can be observed, which suggests that the resolution of the IR sensor (No. 4) falls within the sub-micro-level range. With filtering, the resolution can be further improved.

### B.3.5 Dynamic Response

The dynamic response of the IR sensor was examined indirectly to assess its ability to measure the frequency response of the piezoactuator. To do this, a dynamic signal analyzer (DSA, Hewlett Packard model 35670A) applied a sinusoidal input voltage denoted by  $u$  (fixed amplitude, but varying frequency) to drive the piezoactuator. The magnitude of the input was kept small [ $\leq 100$  mV, (*i.e.*,  $< 10\%$  of the maximum displacement range)] to minimize hysteresis. The piezoactuator's movement was measured by the IR and inductive sensor. Each output was then fed back to the DSA to compute the frequency response (magnitude and phase vs. frequency). The results of the tests are shown in Fig. B.8(a) and (b) for both the IR and inductive sensor. It is noted that the inductive sensor has a bandwidth of 10 kHz. The bandwidth of the IR sensors is limited by the rise time of the phototransistors, which can be as low as a few micro-seconds. Therefore, the bandwidth of the IR sensors compares well with the bandwidth of the inductive sensor. In Fig. B.8(a) and (b), the frequency response of the piezoactuator measured by both sensors are nearly identical, with the exception at higher frequency where there is slight discrepancy. The results indicate

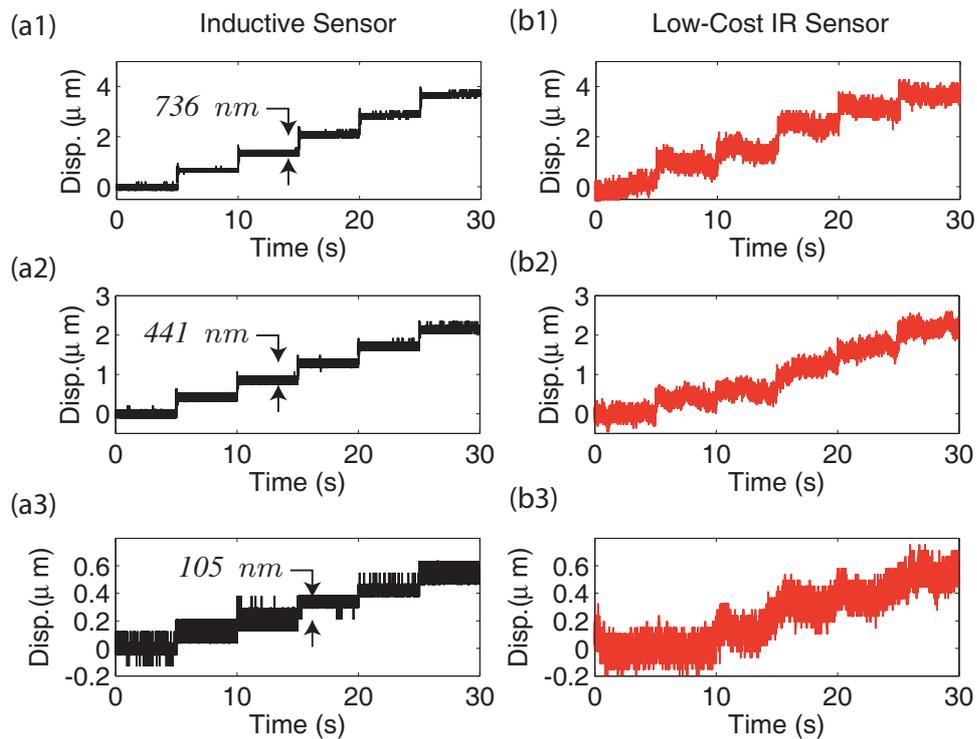


Figure B.7: (a1) - (a3) Unfiltered inductive and (b1) - (b3) IR sensor output versus time to determine the IR sensor's resolution.

that the IR sensor can measure the dynamic response of the piezoactuator and its accuracy compares well with the inductive sensor.

To further evaluate and compare the sensors, the power spectral densities (PSD) of their output signals were measured and they are shown in Fig. B.8(c). For the IR sensor, the measurement was taken at the point labelled  $V_s$  shown in the diagram depicted in Fig. B.3(a). To minimize the effects of interference caused by the main-line power supply, dry-cell batteries were used to power the sensors (Md. Nor and Hill, 2002). Additionally, the PSD curves shown in Fig. B.8(c) were averaged 100 times. The overall results indicate that the inductive sensor has lower frequency content. In particular, below 1 kHz the IR sensor has at least 14 dB more noise

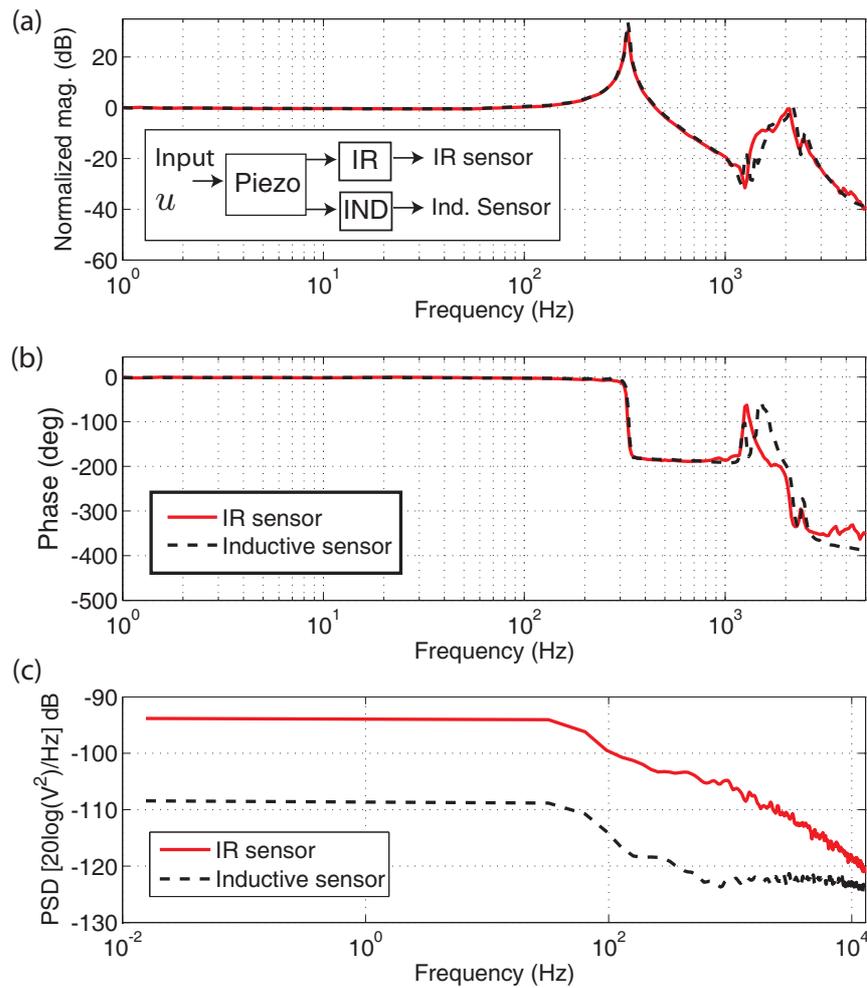


Figure B.8: (a) and (b) The measured frequency response (magnitude and phase vs. frequency) of the experimental piezopositioner. Solid line is the response measured by the IR sensor; dashed line denotes the response measured by the inductive sensor. (c) Power spectral densities of the output signals for the IR sensor (solid line) and the commercial inductive sensor (dash line).

(*i.e.*, 5-times more noise) compared to the inductive sensor per Hz. For low-frequency application filters may be required to reduce the effects of the noise.

## B.4 Example Application: Observer-Based Feedback Control of Piezopositioner

The objective is to demonstrate the application of the IR sensor to control the movement of the example bimorph piezopositioner shown in Fig. B.6. In particular, the two controllers shown in Figs. B.9(a) and (b) were studied. It is noted that the controller designs were not optimized for best performance, but focuses on demonstrating the application of the IR sensor. First, an observer was designed that uses the IR sensor's output to estimate the states of the system. The observer's output allows assessment of the *quality* of the IR sensor measurement. Then the estimated states were used in a state-feedback controller [Fig. B.9(a)] to improve the transient performance such as percent overshoot and settling time. Next, the state-feedback controller was integrated with an integral controller to reduce the steady-state error in the presence of creep and hysteresis effect inherent in piezoactuators (Leang and Devasia, 2007). The integral controller [Fig. B.9(b)] also provides robustness.

### B.4.1 Open-Loop Response of Piezopositioner

Piezoactuators exhibit creep and hysteresis behaviors which make precise control of piezo-based systems a challenge (Leang and Devasia, 2007). For example, when a constant input is applied to a piezoactuator, its displacement (output) consists of high-frequency transients and creep. Figure B.9(c) shows a measured open-loop step response for the experimental bimorph piezoactuator illustrating the two effects using the IR sensor. The transient response settles after approximately 200 ms, at which point creep causes the piezoactuator to slowly drift with time. The high-frequency transient response was captured using a linear second-order model; the model was used to design a state-feedback controller to improve the transient response (percent overshoot and settling time as described in Sections B.4.3 and B.4.3). The creep was neglected in the model, but integral feedback control was applied to minimize this

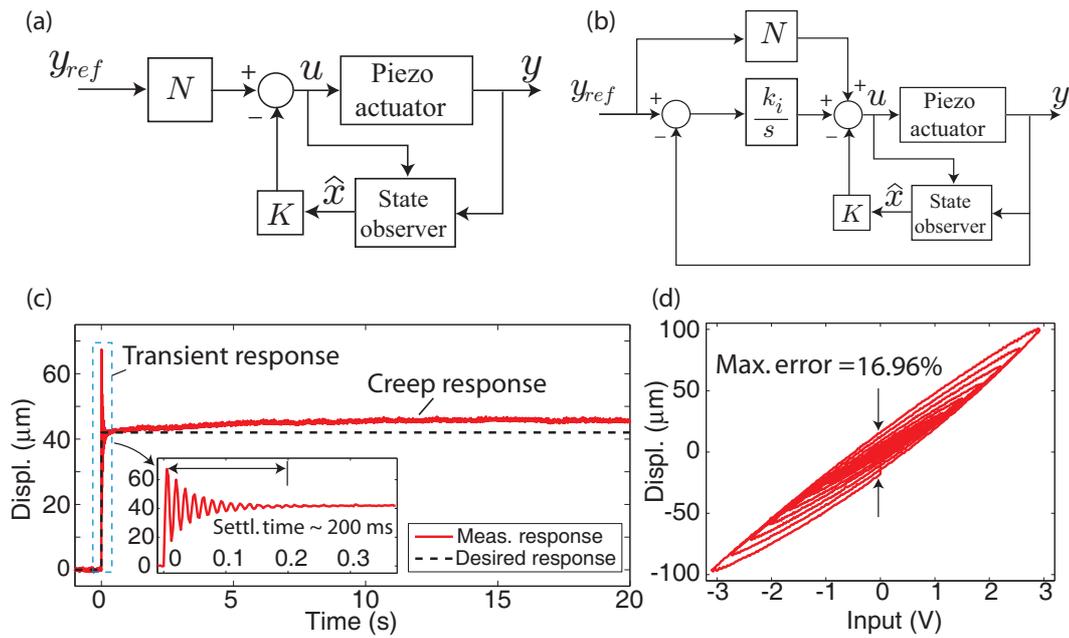


Figure B.9: Controller block diagrams: (a) observer-based full-state feedback controller and (b) state-feedback inner loop with integral output controller (outer loop) and feedforward input. Open-loop responses: (c) Transient response and creep behavior. The settling time is approximately 200 ms. (d) hysteresis behavior.

behavior. In addition to the high-frequency transient and creep effect, piezoactuators exhibit hysteresis. The measured open-loop hysteresis curve (input vs. output plot) for the experimental piezoactuator is shown in Fig. B.9(d). The figure reveals up to 16.96% displacement hysteresis relative to the positioning range ( $198 \mu\text{m}$ ). To minimize hysteresis, high-gain integral feedback control was used as described in Section B.4.3.

#### B.4.2 System Modeling

A (linear) dynamic model that relates the applied input voltage and the output displacement of the piezoactuator was obtained by curve fitting the measured frequency response of the piezoactuator. Before obtaining the frequency response for curve fit-

ting, the IR sensor signal was low-pass filtered through  $G_f(s)$ , the circuit shown in Fig. B.3(a) where  $R_3 = 10 \text{ k}\Omega$ ,  $R_4 = 220 \text{ k}\Omega$ , and  $C = 220 \text{ pF}$ . The cut-off frequency (3.29 kHz) is one-order of magnitude higher than the dominating resonant peak shown in Fig. B.10. The measured frequency response only represents the *linear* vibrational dynamics and the effects of hysteresis and creep were not modeled. The linear dynamics were curve fitted to the measured frequency response using MATLAB software. A second-order transfer function model relating the input voltage to the piezopositioner  $u$  and its output displacement  $x$ , is given by

$$G(s) = \frac{2.673 \times 10^6}{s^2 + 31.42s + 4.222 \times 10^6}; \quad \text{units } \frac{\text{V}}{\text{V}}. \quad (\text{B.4})$$

The model is superimposed on the measured frequency response as shown in Fig. B.10. The model captures the dominant resonant peak near 327 Hz and the higher-order dynamics were neglected in this study; however, in the following sections a higher-order model could have been used. The transfer function model (B.4) was converted to the following canonical form:

$$\dot{x}(t) = \mathbf{A}x(t) + \mathbf{B}u(t) = \begin{bmatrix} -31.42 & 1 \\ -4.222 \times 10^6 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 2.673 \times 10^6 \end{bmatrix} u(t), \quad (\text{B.5})$$

$$y(t) = \mathbf{C}x(t) = [1 \ 0]x(t). \quad (\text{B.6})$$

The above model was used in the observer design and as well to design the state-feedback controller described in Section B.4.2.

#### B.4.3 Controller Designs and Experimental Results

The IR sensor was applied in two controller designs to improve transient and steady-state performance for the experimental piezoactuator. The inductive sensor was not

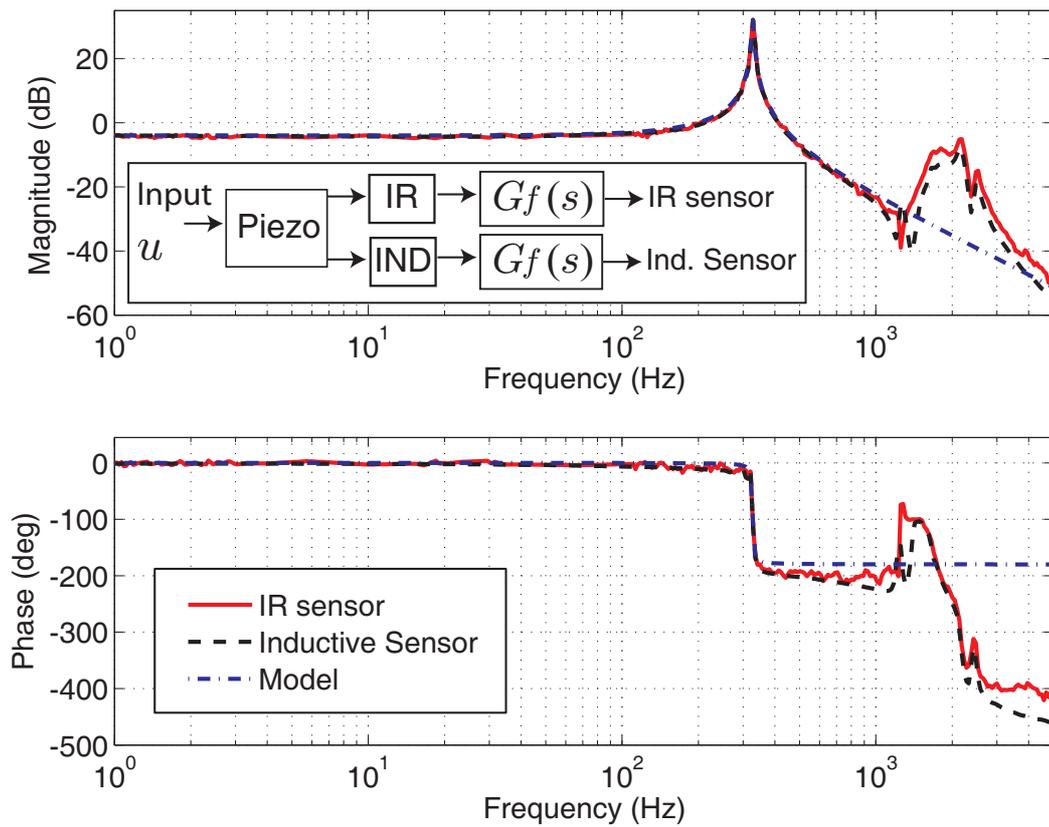


Figure B.10: The measured frequency response (magnitude and phase vs. frequency) of the piezoactuator measured with IR sensor and inductive sensor passed through the filter  $G_f(s)$ . Solid line is the response measured by the IR sensor; dashed line denotes the response measured by the inductive sensor. Also shown (dash-dot line) is the second-order model  $G(s)$  for comparison.

used in feedback and compared to the IR sensor. But rather, the inductive sensor was used to measure the piezoactuator's performance to validate the IR sensor's response.

#### Observer-Based State-Feedback Controller

The control input for the state-feedback system shown in Fig. B.9(a) is given by

$$u(t) = -\mathbf{K}\hat{x}(t) + \mathbf{N}y_{ref}(t), \quad (\text{B.7})$$

where  $\mathbf{K}$  is the state-feedback gain,  $\hat{x}(t)$  is the estimated states from the observer,  $\mathbf{N}$  is the feedforward gain, and  $y_{ref}$  is the reference trajectory (assumed to be constant).

First, the state-feedback gain  $\mathbf{K}$  was obtained using the linear quadratic regulator (LQR) approach subject to the standard cost function which penalizes the states and control effort through weighting matrices  $\mathbf{Q}$  and  $\mathbf{R}$ , respectively (Franklin et al., 2006). The matrices were chosen as  $\mathbf{Q} = \mathbf{C}^T \mathbf{C}$  and  $\mathbf{R} = 1$ , which yielded  $\mathbf{K} = [2.75 \times 10^{-1}, 4.54 \times 10^{-4}]$  and closed-loop poles  $s_{1,2} = -622.1 \pm 2146.7i$ .

Next, the feedforward gain  $\mathbf{N}$  was calculated to achieve the desired steady-state response due to a constant reference  $y_{ref}$ . In the steady-state the error between the reference  $y_{ref}$  and the output  $y$  must vanish, hence

$$\lim_{t \rightarrow \infty} y(t) = \mathbf{C} \lim_{t \rightarrow \infty} x(t) = y_{ss} = \lim_{t \rightarrow \infty} y_{ref}(t) = y_{ref,ss}. \quad (\text{B.8})$$

From the state-space Eq. (B.5), at steady state

$$\dot{x}(t) = 0 = (\mathbf{A} - \mathbf{BK})x_{ss} + \mathbf{BN}y_{ref,ss}, \quad (\text{B.9})$$

where  $x_{ss}(t) = \lim_{t \rightarrow \infty} x(t)$ . Solving the above for  $x_{ss}$  and noting that  $y_{ss} = \mathbf{C}x_{ss}$ , the steady-state output is

$$y_{ss} = \mathbf{C}x_{ss} = -\mathbf{C}(\mathbf{A} - \mathbf{BK})^{-1} \mathbf{BN}y_{ref,ss}. \quad (\text{B.10})$$

By imposing  $y_{ss} = y_{ref,ss}$ , the feedforward gain  $\mathbf{N}$  can be found from (B.10):

$$\mathbf{N} = \frac{-1}{\mathbf{C}(\mathbf{A} - \mathbf{BK})^{-1} \mathbf{B}}. \quad (\text{B.11})$$

The gain  $\mathbf{N}$  exists provided  $\mathbf{C}(\mathbf{A} - \mathbf{BK})^{-1} \mathbf{B} \neq 0$ , and to meet this condition it is assumed that the closed-loop state-feedback system does not have zeros at  $s = 0$ . This assumption is satisfied since for given  $\mathbf{K}$ , the closed-loop poles are  $s_{1,2} = -622.1 \pm 2146.7i$ . The feedforward gain was determined to be  $\mathbf{N} = 1.87$ .

Finally, the observer was designed to estimate the state  $x_2(t)$  from the measured output  $y(t)$  by pole placement. The observer state equation is defined in the usual way (Franklin et al., 2006):

$$\dot{\hat{\mathbf{x}}}(t) = \hat{\mathbf{A}}\hat{\mathbf{x}}(t) + \hat{\mathbf{B}}u(t) + \mathbf{L}y(t), \quad (\text{B.12})$$

where  $\mathbf{L}$  is the observer gain,  $\hat{\mathbf{A}} = \mathbf{A} - \mathbf{BK} - \mathbf{LC}$  and  $\hat{\mathbf{B}} = \mathbf{B}$ . The poles of the observer were placed 10-times further into the left-half plane than the slowest pole of the closed loop system, *e.g.*, the observer poles were chosen as  $s_{1,2} = -6220, -6230$ . The resulting observer gain was  $\mathbf{L} = [1.24 \times 10^4, 3.45 \times 10^7]^T$ .

The observer was implemented using a desktop computer running a custom C program which interfaced to a data acquisition card (12-bit resolution). The sampling rate was 20 kHz. The output  $y(t)$  of the IR sensor was used to estimate the system states  $\hat{\mathbf{x}}(t) = [\hat{x}_1(t), \hat{x}_2(t)]^T$ , where a step input was applied to the piezoactuator. According to the state and output Eqs. (B.5) and (B.6), the first state  $x_1(t)$  can be measured by observing the output  $y(t)$ . Figures B.11(a) and (c) show the measured output  $y(t)$  and the estimated state  $\hat{x}_1(t)$  for two different input values ( $u = 0.4$  and 3.1 V). Likewise, the inductive sensor measurement is also shown for comparison and it was found that the observer estimates the first state with good accuracy. Figure B.11(b) shows an example of the state estimate  $\hat{x}_2(t)$  for the small input case. Based on the results, the second-order approximation of the system was adequate to capture the dominating dynamics of the piezoactuator.

The estimated states  $\hat{\mathbf{x}}(t)$  were used in feedback to improve the settling time and reduce the overshoot compared to the open-loop response. The feedback control law is given by Eq. (B.7), where the feedforward term is  $\mathbf{N} = 1.87$  and the state-feedback gain  $\mathbf{K}$  was defined above. The step response for the observer-based feedback controller are shown in Fig. B.12 for several reference inputs  $y_{ref} = 10 \mu\text{m}$  [plot (a)],  $50 \mu\text{m}$  [plot (b)], and  $100 \mu\text{m}$  [plot (c)]. The relatively large ranges were considered to study the controller's ability to compensate for nonlinearity such as hysteresis. In each of the three plots, the IR sensor's response, the inductive sensor's output, and the measured open-loop response (from the IR sensor) are shown. The open-loop settling time (dotted line) was approximately 200 ms [see Fig. B.11(a)]. In contrast, the settling time of the state-feedback system is 5 ms, over 97% improvement compared

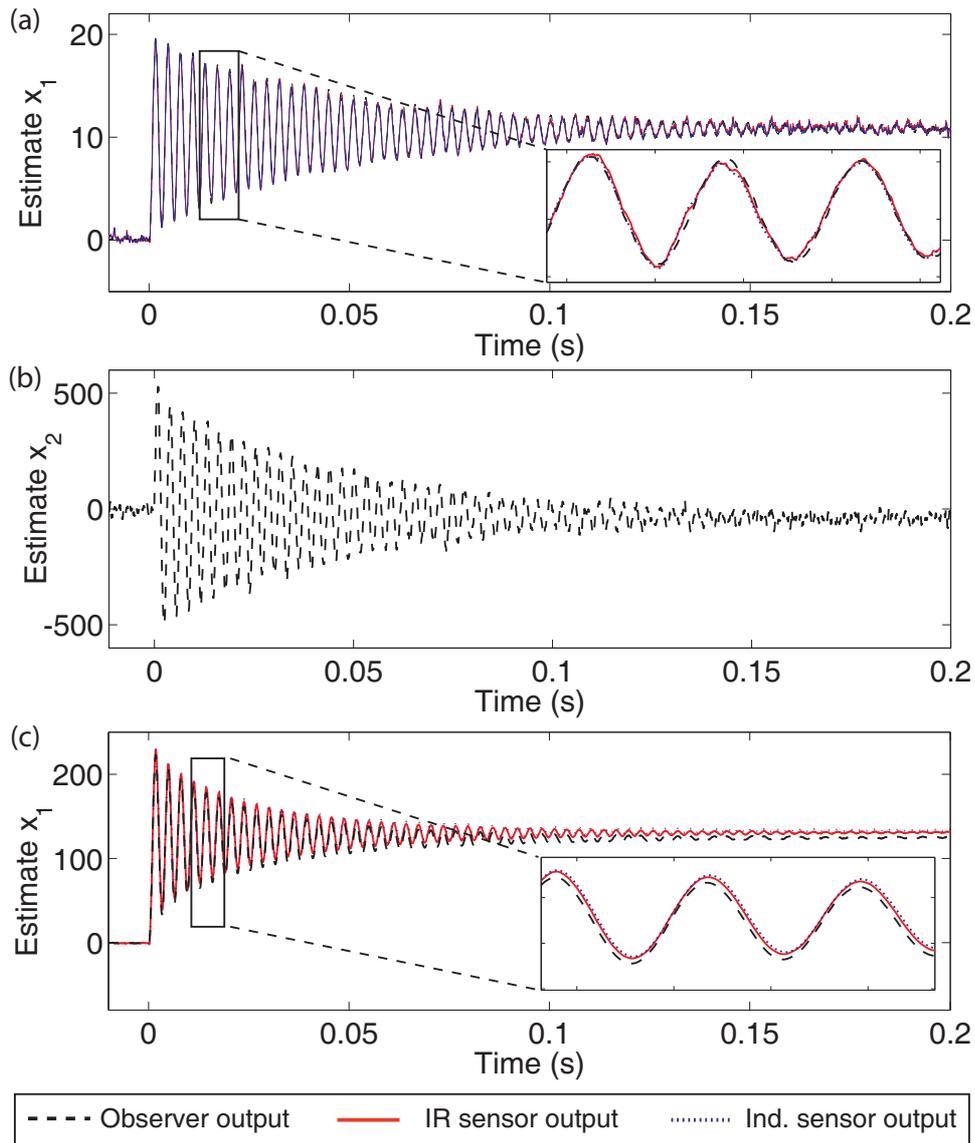


Figure B.11: Observer results: (a) and (b) are the estimates of the states  $x_1$  and  $x_2$ , respectively, that comparing IR and inductive sensor outputs for small input  $u = 400$  mV; (c) estimates of the state  $x_1$  for large input  $u = 3.1$  V.

to the open-loop case. In plots (b) and (c), the effect of creep becomes noticeable with time, especially after the transients have settled. Likewise, when the step reference is 50 and 100  $\mu\text{m}$ , there is significant steady-state error. This is primarily caused by the

hysteresis effect which increases the low-frequency gain of the system [*cf.* Fig. B.9(d)]. Therefore, the computed feedforward gain  $\mathbf{N}$  based on the linear second-order model was much larger than required at 50 and 100  $\mu\text{m}$  (Leang and Devasia, 2007), causing the closed-loop response to settle at a higher value compared to the desired value. To minimize the effects due to creep and hysteresis, it was proposed to combine the state-feedback system with integral control.

### State-Feedback with Integral Control

Integral control was combined with the state-feedback system to minimize the steady state tracking error caused by creep and hysteresis effects. Figure B.9(b) shows the state-feedback controller as the inner loop combined with an outer-loop integral controller and a feedforward path. Examples of other integrated feedback and feedforward controllers for piezoactuators are described in (Zou et al., 2004; Leang and Devasia, 2007). The integral gain was chosen as  $k_i = 600$ .

The experimental results show integral feedback combined with state-feedback control reduces the steady-state response compared to both the open-loop and state-feedback responses. The tracking results shown in Fig. B.13 compare the response for three different step references,  $y_{ref} = 10 \mu\text{m}$  [plot (a)],  $50 \mu\text{m}$  [plot (b)], and  $100 \mu\text{m}$  [plots (c)] (or 0.26, 1.39, and 2.62 V, respectively). It is observed that integral control minimizes the drift error due to creep. Likewise, the hysteresis is also reduced as shown in Fig. B.13(d). Under integral closed-loop control, the maximum output hysteresis was 0.7% of the total displacement range ( $204.2 \mu\text{m}$ ), over 95% improvement compared to the open-loop case. However, as the range increases the overshoot also increases because of the feedforward gain was not designed to consider the hysteresis effect. One solution to minimize the overshoot is to reduce the feedforward gain. In summary, the experimental results demonstrate that the low-cost IR sensor can be used (a) to measure the displacement of a piezoactuator with sub-micro-level precision and (b) for feedback control design to improve the transient and steady-state response.

## B.5 Conclusions

This paper showed that low-cost IR reflective proximity sensors can be used to measure displacements with sub-micro-level accuracy. The experimental results show that the resolution, range, linear distortion, and bandwidth of the IR sensor compares well

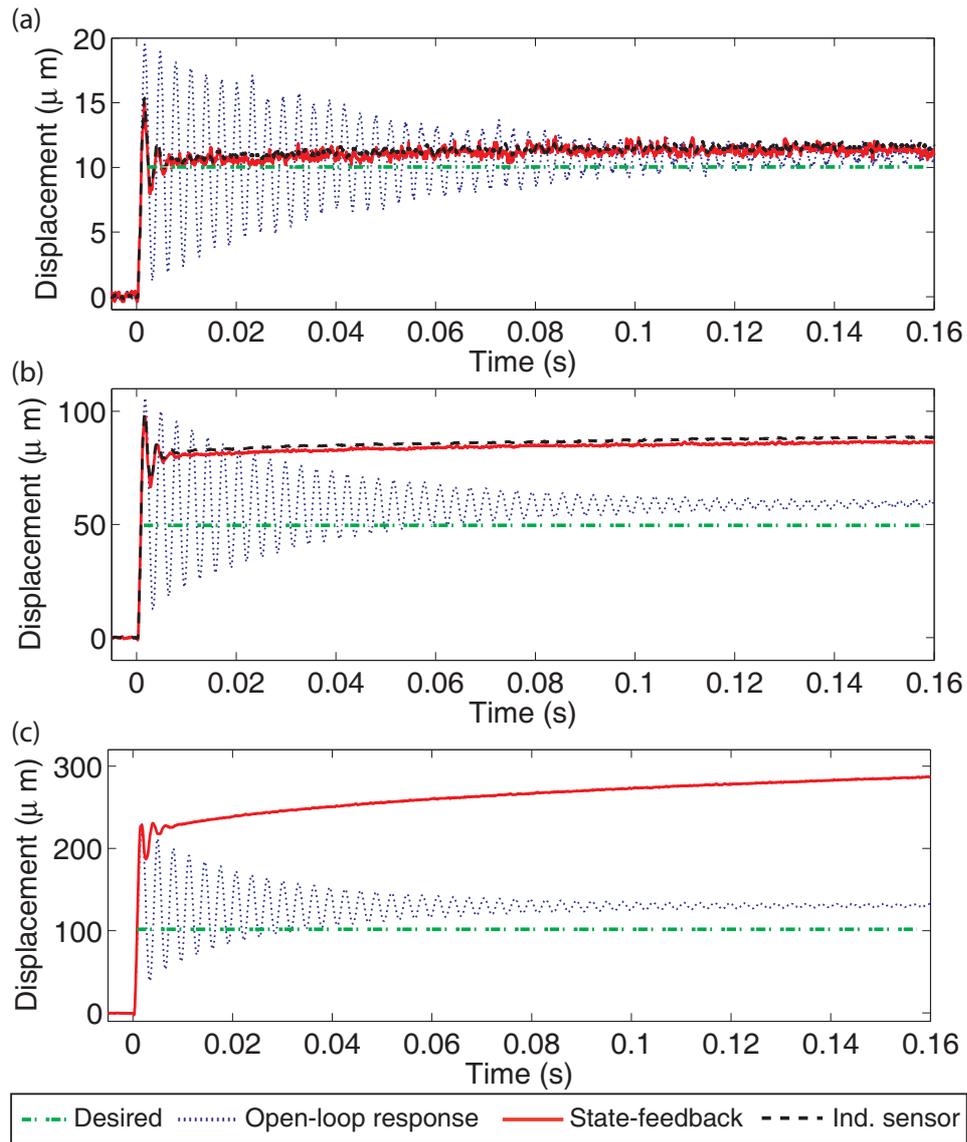


Figure B.12: Tracking results of state-feedback controller.

with a commercially available inductive sensor, but at a fraction of the cost. In particular, the sensor's unfiltered output shows the sensing resolution was within a few

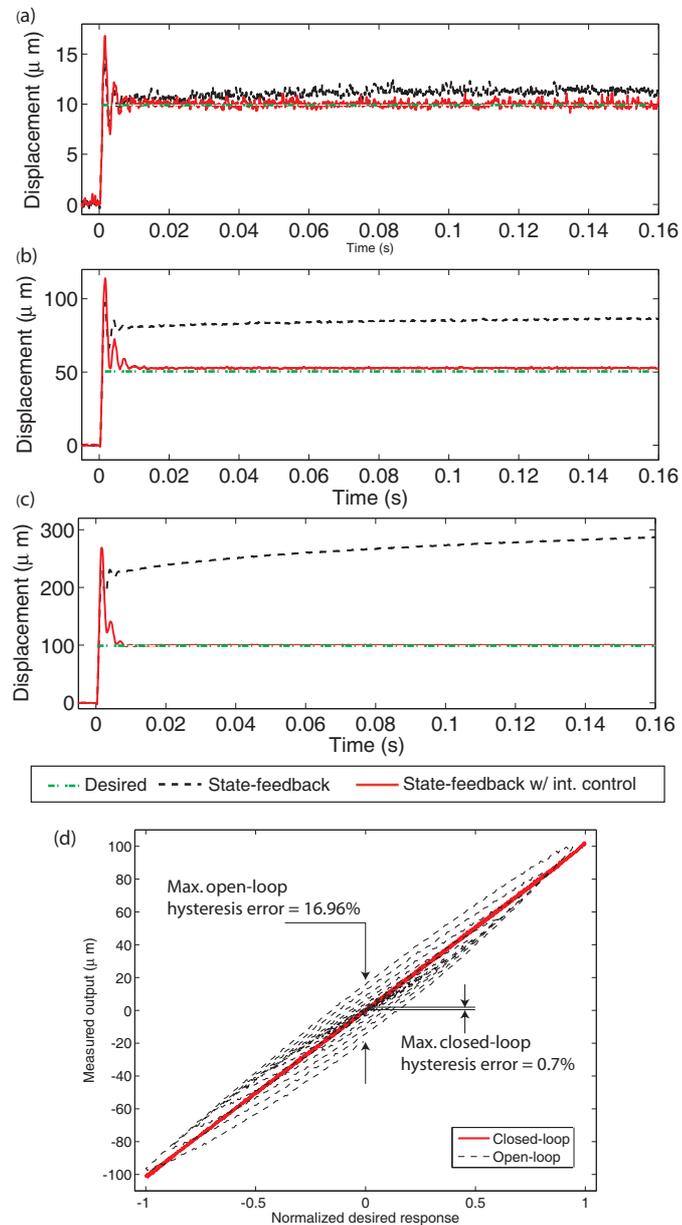


Figure B.13: (a)-(c) State-feedback controller with integral control, displacement and input vs. time. (d) Comparison of hysteresis effect for open-loop (dash line) and closed-loop (solid line).

hundred nanometers. Better performance can be achieved by filtering. It was demonstrated that an IR sensor can be used in feedback to control the hysteresis and creep in a piezoactuator. For example, hysteresis was reduced by over 95% compared to the open-loop case. Some practical considerations include: the IR sensor requires a sufficiently large target; it can be placed further away from the target compared to the inductive sensor; and it has a larger footprint compared to the inductive sensor.

## Appendix C: Matlab Program for Optimal Inversion

This chapter presents the Matlab code for calculating optimal feedforward input  $u_{opt}(t)$  and optimal desired output  $y_{opt}(t)$ . This code is based on the optimal inversion method.

```
%04.08.2008
%optimal input for feedforward control
%=====

clear all
close all

%load desired input trajectory signal-----
load yd01.in;
yd          = yd01(1:29999);
SignalFreq  = 1;           % Hz
SamplingFreq = 500;       % Hz
dt          = 1/SamplingFreq;
t          = dt*[0:length(yd)-1];% time(s)--signal length

% figure(2);clf;
% subplot(211); plot(t,yd);

%=====
%system modeling for (Transfer Function) G
```

```

%=====
w = load('cm.x');           % load frequency
mag = load('cm.txt');       % load mag in dB
ph = load('cp.txt');        % ph in deg.
wn = find(w>130);           % wn = find(w>133);
w = w(1:wn(1));
mag = mag(1:wn(1));
ph = ph(1:wn(1));
return
% figure(1);clf;
% subplot(211); semilogx(w,mag,'k'); hold on
% subplot(212); semilogx(w,ph,'k'); hold on

%complex freq response G (Transfer Function)-----
i = sqrt(-1);
for k =1:length(w)
    a = 10^(mag(k)/20)*cos(ph(k)*pi/180);
    b = 10^(mag(k)/20)*sin(ph(k)*pi/180);
    G(k,1) = a+b*i;          % the Transfer Function of system
end

% make G even and get G_jq (conjugate of G)-----
w = [0;w];
G = [G(1); G];              % Even G
G_jq = conj(G);             % The conjquet of the even TF (G)

ome_g = w*2*pi;             % get the omega of G_ff and G

```



```

% figure(2);
%     subplot(212); stem(w_yd,yd_w)
%     xlim([w_yd(1), SamplingFreq/2])

% information we can get from the FFT of yd ----
ome_yd = w_yd*2*pi;           % half of ome_yd is the omega we
                              % can get, b/c of the FFT property,
                              % we can just get BW of half of
                              % the sampling frequency
ome_yd = ome_yd(1:(end+1)/2);% this is the BW we can really get
                              % (ome is this)--half of the points

%=====
% Resample the Optimal Inversion Transfer Function G_ff
%=====

for k = 1:length(ome_yd),
    if ome_yd(k) <= max(ome_g)
        G_ffn  = interp1(ome_g, G_ff, ome_yd);
        G_n    = interp1(ome_g, G, ome_yd);
        u(k)   = G_ffn(k)*yd_w(k);
        yopt(k) = G_n(k)*u(k);
    else
        yopt(k) = 0;
        u(k)    = 0;
    end
end

end

```

```
yopt2 =conj(yopt(2:end)');  
yopt2 =flipud(yopt2);  
u2 =conj(u(2:end)');  
u2 =flipud(u2);  
  
yopt3 = [yopt'; yopt2];  
u3 = [u'; u2];  
  
yopt      = real(ifft(yopt3));  
yopt      = flipud(yopt);  
uopt      = real(ifft(u3));  
uopt      = flipud(uopt);  
  
figure(3); clf;  
subplot(211); plot(t,uopt); ylabel('Inverse Input');  
subplot(212); plot(t,yd,'b-.',t,yopt,'g--');  
ylabel('x');  
  
return
```

## Appendix D: C Program for Data Acquisition and Control

This chapter presents the *C* code for implementation of controllers, which includes feedback controller implementation, feedforward controller implementation and integrated controller implementation.

### D.1 Feedback controller implementation

```
// =====  
  
#include <stdlib.h>  
#include <stdio.h>  
#include <dos.h>  
#include <conio.h>  
#include <math.h>  
#include <errno.h>  
#include <alloc.h>  
  
#define board1 0x260 /* for board 1 */  
#define board2 0x200 /* for board 2 */  
#define DAC0 0x04 /* Analog output port 0 */  
#define DAC1 0x06 /* Analog output port 1 */  
#define interrupt_loc 0x08  
#define vmax 5  
  
// USER MODIFIABLE VALUES *****
```

```
#define N_CHAN_1 3    /* Number of channels to scan
                      /* on board 1 */
#define N_CHAN_2 1    /* Number of channels to scan
                      /* on board 1 */

#define cycleSize 1600
#define freq 5000     /* Sampling frequency in Hz */

// USER DEFINED ROUTINES *****
void user_init(int,int);    /* Initialization program.
                            /* Sets the frequency */
void user_interface(void); /* Interface task. Executed
                            /* in the foreground */
void user_task(void);      /* Periodic task. Executed
                            /* in the background */
void user_terminate(void); /* Termination task.
                            /* Executed after terminate() */
void user_abort(void);     /* Abort task. Executed
                            /* after ctrl-break */

void newbreak_start(void);
void newtask_init(void);
void terminate(void);
void newbreak_stop(void);
void newtask_start(void);
void out(float,int,int);
void start_AD(int);
void write_data(void);
void read_data(void);
void zero_outputs(void);
```

```

void allocate_memory_for_data(void);
void initialize_y(void);
void cycle_x(int);
int countdown,lobyte,hibyte,delay_count;
float in(int);

// USER VARIABLES HERE *****
char outfile[80]; char cyclefile[]="cycle.in"; char infile[80];
float cycle_data[cycleSize];
float huge *data;          /* pointer to data array
                           /* in memory */
unsigned long DataSize = 120000; /* size of data array */
unsigned long data_i, iStop;
unsigned long d1_offset = 0;
unsigned long d2_offset = 30000;
unsigned long d3_offset = 60000;
unsigned long d4_offset = 90000;
unsigned long int_count;
float InputScale, input, d1, d2, d3, d1_init, d2_init, d3_init;
float ref, y, ek, eint, u, kg, kp, ki, k,td;
int done;

// Main program
void main(void){
    //==== allocate memory for data array
    allocate_memory_for_data();
    //====initialize variables
    done = 0;          /* flag to terminate interrupt routine */

```

```
data_i = 0;          /* initialize data index counter */
int_count = 0;      /* initialize interrupt counter */
eint = 0;
k = 0;
td = 2e-4;

//==== load data from file, store in 'data' vector
printf("Enter INPUT file: ");  gets(infile);
printf("Enter OUTPUT file: "); gets(outfile);
printf("Sampling freq: %d Hz\n",freq);
printf("Input Scale: "); scanf("%f",&InputScale);
//printf("Stop index: "); scanf("%ld",&iStop);
iStop = 30000;
printf("kg: "); scanf("%f",&kg);
printf("kp: "); scanf("%f",&kp);
printf("ki: "); scanf("%f",&ki);
read_data();

//====initialize board(s) and zero outputs
user_init(board1,N_CHAN_1);
//user_init(board2,N_CHAN_2);
zero_outputs();
//printf("Press any key to CYCLE system\n"); getch();
//printf("Cycling...."); cycle_x(1);
//delay(500); cycle_y(1);
//printf("Cycling complete!\n");
printf("Press any key to BEGIN\n"); getch();
zero_outputs();
```

```

newtask_init();newbreak_start(); newtask_start();
while(!done){
    //user_interface();
}
terminate();
zero_outputs();
write_data();
zero_outputs();
}

void user_task(void) {
    // Send and collect =====

    start_AD(board1);                /* read PIN 3 */
    d3 = in(board1) - d3_init;
    //data[int_count + d4_offset] = d3;

    start_AD(board1);                /* read PIN 2 */
    d2 = in(board1) - d2_init;
    //data[int_count + d3_offset] = d2;

    start_AD(board1);                /* read PIN 1 */
    y = in(board1) - d1_init;    // output

    ref = kg*data[int_count + d1_offset];
    ek = ref - y;
    eint = ek*td + eint;
    //u = (kp*ek + ki*eint) + data[int_count + d2_offset];

```

```
u = kp*ek + ki*eint;
if(u>4.0){u = 4.0;}
if(u<-4.0){u = -4.0;}
out(u,board1,DAC0); /* output A0 0 */

data[int_count + d4_offset] = u;
data[int_count + d3_offset] = y;

k++;
if(k>=10){k = 0;
    int_count++;
}

if (int_count >= iStop){
    done = 1;
    int_count = iStop;
}

}

void user_interface(void){
    //printf("%ld, u: %3.2f, d1: %3.2f, d2: %3.2f,
    //d3: %3.2f\n",int_count,input,d1,d2,d3);
    printf("ref = %f y = %f u = %f\n",ref,d1,u);
}

void allocate_memory_for_data(void){
    /* allocate memory for data array */
    unsigned long k;
    data = (float huge *)farcalloc(DataSize,sizeof(float));
```

```

    if (data == NULL){ printf("Not enough memory to store data\n");
        exit(1);}
    for(k=0;k<DataSize;k++){data[k] = 0.0;}
}

void zero_outputs(void){
    out(0.0,board1,DAC0);
    out(0.0,board1,DAC1);
    start_AD(board1);                /* read */
    d3_init = in(board1);
    start_AD(board1);                /* read */
    d2_init = in(board1);
    start_AD(board1);                /* read */
    d1_init = in(board1);
}

void read_data(void){
    FILE *fp1;
    float temp1,temp2,temp3;
    unsigned long i;
    if((fp1=fopen(infile,"r"))==NULL){
        perror("Error opening input file!!");
        exit(EXIT_FAILURE);
    }
    for(i=0;i<(DataSize);i++){
        fscanf(fp1,"%f\n",&temp1);
        data[i]=InputScale*temp1;
    }
    fclose(fp1);
    printf("Input data acquired from file!\n");
}

```

```
}  
// Cycles x uN times  
void cycle_x(int uN){  
    int i,j;  
    float temp1;  
    FILE *fp3;  
    if((fp3=fopen(cyclefile,"r"))==NULL){  
        perror("Error opening Cycle Data File!"); exit(1);  
    }  
    else{  
        for(i=0;i<cycleSize;i++){  
            fscanf(fp3,"%f\n",&temp1);  
            cycle_data[i]=temp1*.3;  
        }  
    }  
    fclose(fp3);  
    for(i=0;i<uN;i++){  
        for(j=0;j<cycleSize;j++){  
            out(cycle_data[j],board1,DAC0);  
            delay(10);  
            printf("Cycle x [%d, %d]\n",i,j);  
        }  
    }  
}  
void write_data(void){  
    unsigned long i;  
    FILE *fp2;  
    if((fp2=fopen(outfile,"w"))==NULL){
```

```

    printf("Error opening %s!! No data saved!\n",outfile);
}
else{
    for (i=0;i<DataSize;i++){
        fprintf(fp2,"%f\n",data[i]);
    }
    printf("Done writing OUTPUT DATA to %s\n",outfile);
    fclose(fp2);
}
}
}

```

## D.2 Feedforward controller implementation

```

// =====

#include <stdlib.h>
#include <stdio.h>
#include <dos.h>
#include <conio.h>
#include <math.h>
#include <errno.h>
#include <alloc.h>

#define board1 0x260          /* for board 1 */
#define board2 0x200          /* for board 2 */
#define DAC0 0x04            /* Analog output port 0 */
#define DAC1 0x06            /* Analog output port 1 */
#define interrupt_loc 0x08

```

```
#define vmax 5

// USER MODIFIABLE VALUES *****
#define N_CHAN_1 3          /* Number of channels
                           /* to scan on board 1 */
#define N_CHAN_2 1          /* Number of channels to
                           /* scan on board 1 */
#define cycleSize 1600
#define freq 5000          /* Sampling frequency
                           /* in Hz */

// USER DEFINED ROUTINES *****
void user_init(int,int);   /* Initialization program.
                           /* Sets the frequency */
void user_interface(void); /* Interface task. Executed
                           /* in the foreground */
void user_task(void);     /* Periodic task. Executed
                           /* in the background */
void user_terminate(void); /* Termination task.
                           /* Executed after terminate() */
void user_abort(void);    /* Abort task. Executed
                           /* after ctrl-break */

void newbreak_start(void);
void newtask_init(void);
void terminate(void);
void newbreak_stop(void);
void newtask_start(void);
void out(float,int,int);
```

```

void start_AD(int);
void write_data(void);
void read_data(void);
void zero_outputs(void);
void allocate_memory_for_data(void);
void initialize_y(void);
void cycle_x(int);
int countdown,lobyte,hibyte,delay_count;
float in(int);

// USER VARIABLES HERE *****
char outfile[80]; char cyclefile []="cycle.in"; char infile[80];
float cycle_data[cycleSize];
float huge *data;          /* pointer to data array in memory */
unsigned long DataSize = 120000; /* size of data array */
unsigned long data_i, iStop;
unsigned long d1_offset = 0;
unsigned long d2_offset = 30000;
unsigned long d3_offset = 60000;
unsigned long d4_offset = 90000;
unsigned long int_count;
float InputScale, input, d1, d2, d3, d1_init, d2_init, d3_init;
float ref, y, ek, eint, u, kg, kp, ki, k,td;
int done;

// Main program
void main(void){
    //==== allocate memory for data array

```

```
allocate_memory_for_data();
//====initialize variables
done = 0;          /* flag to terminate interrupt routine */
data_i = 0;       /* initialize data index counter */
int_count = 0;   /* initialize interrupt counter */
eint = 0;
k = 0;
td = 2e-4;

//==== load data from file, store in 'data' vector
printf("Enter INPUT file: ");  gets(infile);
printf("Enter OUTPUT file: "); gets(outfile);
printf("Sampling freq: %d Hz\n",freq);
printf("Input Scale: "); scanf("%f",&InputScale);
//printf("Stop index: "); scanf("%ld",&iStop);
iStop = 30000;
printf("kg: "); scanf("%f",&kg);
printf("kp: "); scanf("%f",&kp);
printf("ki: "); scanf("%f",&ki);
read_data();

//====initialize board(s) and zero outputs
user_init(board1,N_CHAN_1);
//user_init(board2,N_CHAN_2);
zero_outputs();
//printf("Press any key to CYCLE system\n"); getch();
//printf("Cycling..."); cycle_x(1);
//delay(500); cycle_y(1);
```

```

//printf("Cycling complete!\n");
printf("Press any key to BEGIN\n"); getch();
zero_outputs();
newtask_init();newbreak_start(); newtask_start();
while(!done){
    //user_interface();
}
terminate();
zero_outputs();
write_data();
zero_outputs();
}

void user_task(void) {
    // Send and collect =====

    start_AD(board1);                /* read PIN 3 */
    d3 = in(board1) - d3_init;
    //data[int_count + d4_offset] = d3;

    start_AD(board1);                /* read PIN 2 */
    d2 = in(board1) - d2_init;
    //data[int_count + d3_offset] = d2;

    start_AD(board1);                /* read PIN 1 */
    y = in(board1) - d1_init;    // output

    ref = kg*data[int_count + d1_offset];

```

```
ek = ref - y;
eint = ek*td + eint;
//u = (kp*ek + ki*eint) + data[int_count + d2_offset];
u =0.87*data[int_count + d2_offset];
if(u>4.0){u = 4.0;}
if(u<-4.0){u = -4.0;}
out(u,board1,DAC0); /* output A0 0 */

data[int_count + d4_offset] = u;
data[int_count + d3_offset] = y;

k++;
if(k>=10){k = 0;
    int_count++;
}

if (int_count >= iStop){
    done = 1;
    int_count = iStop;
}

}

void user_interface(void){
    printf("ref = %f y = %f u = %f\n",ref,d1,u);
}

void allocate_memory_for_data(void){
    /* allocate memory for data array */
    unsigned long k;
```

```

data = (float huge *)farcalloc(DataSize,sizeof(float));
if (data == NULL){ printf("Not enough memory to store data\n");
    exit(1);}
for(k=0;k<DataSize;k++){data[k] = 0.0;}
}
void zero_outputs(void){
    out(0.0,board1,DAC0);
    out(0.0,board1,DAC1);
    start_AD(board1);                /* read */
    d3_init = in(board1);
    start_AD(board1);                /* read */
    d2_init = in(board1);
    start_AD(board1);                /* read */
    d1_init = in(board1);
}
void read_data(void){
    FILE *fp1;
    float temp1,temp2,temp3;
    unsigned long i;
    if((fp1=fopen(infile,"r"))==NULL){
        perror("Error opening input file!!");
        exit(EXIT_FAILURE);
    }
    for(i=0;i<(DataSize);i++){
        fscanf(fp1,"%f\n",&temp1);
        data[i]=InputScale*temp1;
    }
    fclose(fp1);
}

```

```
    printf("Input data acquired from file!\n");
}
// Cycles x uN times
void cycle_x(int uN){
    int i,j;
    float temp1;
    FILE *fp3;
    if((fp3=fopen(cyclefile,"r"))==NULL){
        perror("Error opening Cycle Data File!"); exit(1);
    }
    else{
        for(i=0;i<cycleSize;i++){
            fscanf(fp3,"%f\n",&temp1);
            cycle_data[i]=temp1*.3;
        }
    }
    fclose(fp3);
    for(i=0;i<uN;i++){
        for(j=0;j<cycleSize;j++){
            out(cycle_data[j],board1,DAC0);
            delay(10);
            printf("Cycle x [%d, %d]\n",i,j);
        }
    }
}
void write_data(void){
    unsigned long i;
    FILE *fp2;
```

```

if((fp2=fopen(outfile,"w"))==NULL){
    printf("Error opening %s!! No data saved!\n",outfile);
}
else{
    for (i=0;i<DataSize;i++){
        fprintf(fp2,"%f\n",data[i]);
    }
    printf("Done writing OUTPUT DATA to %s\n",outfile);
    fclose(fp2);
}
}

```

### D.3 Integrated feedforward and feedback controller implementation

```

// =====

#include <stdlib.h>
#include <stdio.h>
#include <dos.h>
#include <conio.h>
#include <math.h>
#include <errno.h>
#include <alloc.h>

#define board1 0x260          /* for board 1 */
#define board2 0x200          /* for board 2 */
#define DAC0 0x04            /* Analog output port 0 */
#define DAC1 0x06            /* Analog output port 1 */

```

```

#define interrupt_loc 0x08
#define vmax 5

// USER MODIFIABLE VALUES *****
#define N_CHAN_1 3      /* Number of channels to scan on board 1 */
#define N_CHAN_2 1      /* Number of channels to scan on board 1 */
#define cycleSize 1600
#define freq 5000 /* Samp freq in Hz */
// USER DEFINED ROUTINES *****
void user_init(int,int); /* Initialization program.
                        /* Sets the frequency */
void user_interface(void); /* Interface task. Executed
                        /* in the foreground */
void user_task(void); /* Periodic task. Executed
                        /* in the background */
void user_terminate(void); /* Termination task. Executed
                        /* after terminate() */
void user_abort(void); /* Abort task. Executed
                        /* after ctrl-break */

void newbreak_start(void);
void newtask_init(void);
void terminate(void);
void newbreak_stop(void);
void newtask_start(void);
void out(float,int,int);
void start_AD(int);
void write_data(void);
void read_data(void);

```

```

void zero_outputs(void);
void allocate_memory_for_data(void);
void initialize_y(void);
void cycle_x(int);
int countdown,lobyte,hibyte,delay_count;
float in(int);

// USER VARIABLES HERE *****
char outfile[80]; char cyclefile []="cycle.in"; char infile[80];
float cycle_data[cycleSize];
float huge *data;          /* pointer to data array in memory */
unsigned long DataSize = 120000; /* size of data array */
unsigned long data_i, iStop;
unsigned long d1_offset = 0;
unsigned long d2_offset = 30000;
unsigned long d3_offset = 60000;
unsigned long d4_offset = 90000;
unsigned long int_count;
float InputScale, input, d1, d2, d3, d1_init, d2_init, d3_init;
float ref, y, ek, eint, u, kg, kp, ki, k,td;
int done;

// Main program
void main(void){
    //==== allocate memory for data array
    allocate_memory_for_data();
    //====initialize variables
    done = 0;          /* flag to terminate interrupt routine */

```

```
data_i = 0;          /* initialize data index counter */
int_count = 0;      /* initialize interrupt counter */
eint = 0;
k = 0;
td = 2e-4;

//==== load data from file, store in 'data' vector
printf("Enter INPUT file: ");  gets(infile);
printf("Enter OUTPUT file: "); gets(outfile);
printf("Sampling freq: %d Hz\n",freq);
printf("Input Scale: "); scanf("%f",&InputScale);
//printf("Stop index: "); scanf("%ld",&iStop);
iStop = 30000;
printf("kg: "); scanf("%f",&kg);
printf("kp: "); scanf("%f",&kp);
printf("ki: "); scanf("%f",&ki);
read_data();

//====initialize board(s) and zero outputs
user_init(board1,N_CHAN_1);
//user_init(board2,N_CHAN_2);
zero_outputs();
//printf("Press any key to CYCLE system\n"); getch();
//printf("Cycling...."); cycle_x(1);
//delay(500); cycle_y(1);
//printf("Cycling complete!\n");
printf("Press any key to BEGIN\n"); getch();
zero_outputs();
```

```

newtask_init();newbreak_start(); newtask_start();
while(!done){
    //user_interface();
}
terminate();
zero_outputs();
write_data();
zero_outputs();
}

void user_task(void) {
    // Send and collect =====

    start_AD(board1);                /* read PIN 3 */
    d3 = in(board1) - d3_init;
    //data[int_count + d4_offset] = d3;

    start_AD(board1);                /* read PIN 2 */
    d2 = in(board1) - d2_init;
    //data[int_count + d3_offset] = d2;

    start_AD(board1);                /* read PIN 1 */
    y = in(board1) - d1_init;    // output

    ref = kg*data[int_count + d1_offset];
    ek = ref - y;
    eint = ek*td + eint;
    u = (kp*ek + ki*eint) +0.87*data[int_count + d2_offset];

```

```
if(u>4.0){u = 4.0;}
if(u<-4.0){u = -4.0;}
out(u,board1,DAC0); /* output A0 0 */

data[int_count + d4_offset] = u;
data[int_count + d3_offset] = y;

k++;
if(k>=10){k = 0;
    int_count++;
}

if (int_count >= iStop){
    done = 1;
    int_count = iStop;
}

}

void user_interface(void){
    printf("ref = %f y = %f u = %f\n",ref,d1,u);
}

void allocate_memory_for_data(void){
    /* allocate memory for data array */
    unsigned long k;
    data = (float huge *)farcalloc(DataSize,sizeof(float));
    if (data == NULL){ printf("Not enough memory to store data\n");
        exit(1);}
    for(k=0;k<DataSize;k++){data[k] = 0.0;}
```

```
}  
void zero_outputs(void){  
    out(0.0,board1,DAC0);  
    out(0.0,board1,DAC1);  
    start_AD(board1);           /* read */  
    d3_init = in(board1);  
    start_AD(board1);           /* read */  
    d2_init = in(board1);  
    start_AD(board1);           /* read */  
    d1_init = in(board1);  
}  
void read_data(void){  
    FILE *fp1;  
    float temp1,temp2,temp3;  
    unsigned long i;  
    if((fp1=fopen(infile,"r"))==NULL){  
        perror("Error opening input file!!");  
        exit(EXIT_FAILURE);  
    }  
    for(i=0;i<(DataSize);i++){  
        fscanf(fp1,"%f\n",&temp1);  
        data[i]=InputScale*temp1;  
    }  
    fclose(fp1);  
    printf("Input data acquired from file!\n");  
}  
// Cycles x uN times  
void cycle_x(int uN){
```

```
int i,j;
float temp1;
FILE *fp3;
if((fp3=fopen(cyclefile,"r"))==NULL){
    perror("Error opening Cycle Data File!"); exit(1);
}
else{
    for(i=0;i<cycleSize;i++){
        fscanf(fp3,"%f\n",&temp1);
        cycle_data[i]=temp1*.3;
    }
}
fclose(fp3);
for(i=0;i<uN;i++){
    for(j=0;j<cycleSize;j++){
        out(cycle_data[j],board1,DAC0);
        delay(10);
        printf("Cycle x [%d, %d]\n",i,j);
    }
}
}
void write_data(void){
    unsigned long i;
    FILE *fp2;
    if((fp2=fopen(outfile,"w"))==NULL){
        printf("Error opening %s!! No data saved!\n",outfile);
    }
    else{
```

```
for (i=0;i<DataSize;i++){
    fprintf(fp2,"%f\n",data[i]);
}
printf("Done writing OUTPUT DATA to %s\n",outfile);
fclose(fp2);
}
}
```