FUZZY CONTROL OF THE ELECTROHYDRAULIC ACTUATOR

A Thesis Submitted to the
College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Mechanical Engineering
University of Saskatchewan
Saskatoon

By
Eric B. Sampson

PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Mechanical Engineering
University of Saskatchewan
Saskatoon, Saskatchewan S7N 5A9

ABSTRACT

Industrial applications increasingly require actuators that offer a combination of high force output, large stroke and high accuracy. The ElectroHydraulic Actuator (EHA) was designed by Drs. Habibi and Goldenberg originally as a high-performance actuator for use in robotics. However, it was determined that the EHA had the potential to achieve high positional accuracy. Little research has been performed in the area of high-accuracy hydraulic positioning systems. Therefore, the objective of this study to achieve nano-scale positional accuracy with the EHA while maintaining large stroke and high force output. It was planned to achieve this objective through modification of the prototype EHA and the use of fuzzy control.

During this research project, both hardware and control system modifications to the EHA were performed. A high-precision optical encoder position sensor with a 50 nm resolution was mounted on the inertial load to directly measure the position of the load. A number of device drivers were written to interface the MATLAB real-time control environment with the optical encoder and servo motor amplifier. A Sugeno-inference fuzzy controller was designed and implemented in MATLAB. For comparison purposes, a switched-gain controller and a proportional controller were also implemented in the control environment.

The performance of the fuzzy controller was compared to the switched-gain controller and the proportional controller in a number of tests. First, the regulatory and tracking performance of the EHA with an inertial load of 20 kg was examined. It was determined in the regulatory tests that the positional accuracy of the EHA with the fuzzy controller was excellent, achieving a steady state error of 50 ± 25 nm or less for step inputs in the range 5 cm to 200 nm. The positional accuracy during the tracking tests was found to be reduced compared to the regulatory tests since the actuator did not have sufficient time to settle to final accuracy due to the time-varying input signals. In all cases, it was found that the positional accuracy of the EHA with the fuzzy controller was significantly greater than with the switched-gain and proportional controllers for both regulatory and tracking signals. Testing with the inertial load eliminated or changed was

not performed because the position sensor was mounted to the load, making it unfeasible to alter the load during the time frame of this study.

The regulatory and tracking performance of the EHA with an inertial load of 20 kg plus external resistive loads of 90 to 280 N were investigated. It was found that the positional accuracy of the EHA decreased with the application of an external load to 3.10 ± 0.835 μm for a 1 cm step input (90 N load) and 8.45 ± 0.400 μm for a 3 cm step input (280 N load). Again, the positional accuracy of the EHA decreased during the tracking tests relative to the regulatory tests, for the reason stated above. This implies that the positional accuracy of the EHA with a resistive load is in the micro-scale, rather than the nano-scale as was put forth as the objective of this study. Nevertheless, the positional accuracy of the EHA with the fuzzy controller was found to be significantly greater than with the switched-gain and proportional controllers. It is postulated that the increase in positional error observed during the external load tests was due to an increase in cross-port leakage, relative to the inertial load tests, caused by the pressure differential induced across the actuator by the external load. Methods of reducing the increase in positional error caused by external loads on the EHA remains an area for future study.

ACKNOWLEDGEMENTS

The author would like to express his gratitude to Drs. Richard Burton and Saeid Habibi for their guidance in this project over the past several years. Additional gratitude is due to the individuals who stimulated thought and provided technical aid during this project, especially Doug Bitner, Jeff Dobchuk, Yuvin Chinniah, and Travis Wiens.

Finally, I would also like to acknowledge the unwavering support and love offered throughout my life by my parents, Lyle and Colleen Sampson.

This project was made possible by the financial support of the Natural Sciences and Engineering Research Council of Canada and the Department of Mechanical Engineering.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

List of Abbreviations

AC	Alternating Current
CoA	Center of Area
CoG	Center of Gravity
EHA	ElectroHydraulic Actuator
EMI	ElectroMagnetic Interference
FLC	Fuzzy Logic Controller
LED	Light Emitting Diode
LVDT	Linear Variable Differential Transformer
MoM	Mean of Maxima
NIST	National Institute of Standards and Technology
PC	Proportional Controller
PCI	Peripheral Component Interconnect
PID	Proportional Integral Derivative
PWM	Pulse Width Modulation
RFI	Radio Frequency Interference
SGC	Switched Gain Controller

CHAPTER 1

INTRODUCTION

1.1 Introduction

Many applications exist for an actuator that combines high force output, large stroke and high accuracy. In this context, an actuator is any device which creates mechanical motion through the conversion of various forms of energy into linear or rotary mechanical energy. Typical actuators seen in everyday life include electric motors used to open garage doors, pneumatic cylinders used to power small robots, and hydraulic motors used to power off-highway vehicles.

An important parameter which characterises the performance of an actuator is its force output. The output force of an actuator is defined as the amount of force that a given actuator can exert on an external body along or around its output axis. Hydraulic systems in general can generate a large output force compared to electric or pneumatic systems due to the high pressure of operation and the relative lack of compressibility of the working fluid.

Another parameter which characterises the performance of an actuator is its stroke. Stroke is defined as the maximum amount of extension of a linear actuator or the maximum amount of rotation of a rotary actuator. It is more commonly used to describe linear actuators than rotary actuators because many of the latter can rotate indefinitely and therefore do not have a fixed stroke.

Finally, the most important performance characteristic of a high-accuracy actuator is its accuracy. Accuracy is a general term, and one must ensure that the circumstances under which it is measured is included in the specification. For example, an actuator may claim a certain accuracy without mentioning that the measurement is only valid when the actuator is unloaded. Additionally, accuracy can be specified in both a static and dynamic sense. For example, static accuracy could mean the steady-state error when the actuator is required to move a certain distance, while dynamic accuracy could refer to the accuracy of an actuator when it is tracking a varying input signal. Even something as simple as static accuracy is complicated by concerns such as repeatability and the possible dependence of the accuracy upon the type and magnitude of the demanded input

signal, as well as loading and environmental conditions.

Several examples of applications which require a combination of high force output, large stroke and high accuracy can be given. The first is positioning of test subjects for beam line applications. One such beam line is the Canadian Light Source, which is a synchrotron located at the University of Saskatchewan [1, 2]. Since the beam cannot be easily deflected, it is desirable to move the subject that is being imaged. The subject may have a large mass, requiring an actuator with a high force output in order to position the subject. Since the desired resolution of the image is high, the subject must be positioned precisely so that the beam strikes the desired target in or on the subject.

Another application that requires these characteristics is the positioning of the mirror elements of terrestrial multi-mirror telescopes [3, 4]. As these mirror elements are projected to become larger and heavier as the desired resolution of proposed telescopes increases, the high force output capability of hydraulic actuators may become more valuable. A final type of application that demands extreme accuracy and high force output is micro-machining and high-precision grinding [5, 6]. In this application, the actuator is used to position a cutting or grinding tool against a rotating blank of material. The accuracy of the final part will depend directly on the characteristics of the actuator, causing this to be one of the most demanding applications of any actuator.

These applications demonstrate the need for an actuation system which can position large inertial loads with high accuracy and is capable of significant actuator displacements. This need was the motivation for the research presented in this thesis.

1.2 Literature review

Several solutions exist for high-accuracy actuation. These include piezoelectric actuators, electromagnetic linear motors, voice coil linear actuators, shape memory alloys actuators, leadscrew or ballscrew driven actuators, friction-based ultrasonic piezoelectric actuators, multistage actuators, electrostatic actuators, and magnetostriction actuators.

Each of these solutions has advantages and disadvantages. Piezoelectric actuators are the most common type of high-accuracy actuators. They can be capable of sub-nanometer resolution and high acceleration. However, their stroke is severely limited, often to less than 100 μm. Also, the maximum load of positioning stages such as the "high-load" Physik Instrument P-750 is 10 kg and its maximum output force is 800 N in the push direction and 100 N in the pull direction. The limited stroke of piezoelectric actuators can be improved by employing multiple positioning stages.

For example, a coarse positioning stage such as a electromagnetic linear motor can be combined in series with a fine positioning stage driven by a piezoelectric actuator. This solution tends to be costly and complex as it requires multiple actuators and sensors being controlled simultaneously in order to achieve the desired hybrid characteristics.

Alternately, electromechanical actuators such as electric-motor-driven leadscrews are becoming more popular due to their low energy requirements and low noise creation compared to hydraulics and pneumatics. However, they cannot be used when spark generation is a concern, their power-to-weight ratios are significantly lower than hydraulic actuators, and the control strategies required to allow them to achieve high accuracy can be very complex.

In the field of hydraulics, little emphasis has been placed upon researching the limits of achievable positional accuracy. Kanai et al. investigated the use of force-control rather than flow-control of a hydraulic actuator for use in ultraprecision grinding machines [7]. In this application, high resolution and stiffness is required. The authors achieved a feed resolution of 1 nm/step with a 200 kg load. However, the stroke of their actuator is low at only 20 mm, and the maximum velocity is very low (though unspecified by the authors) due to the nature of the system's operation. Konishi et al. describe a system created by them that uses a single-rod cylinder with fluid delivered from a reciprocating pump powered by a piezoelectric actuator [8]. Experimental accuracies of 10 μm unloaded and 50 μm with a 60 N load are reported by the researchers. In their paper "Nanometer positioning and its micro-dynamics" [9], Futami et al. reference an actuator created by Shintaku [10]. They describe it as consisting of a single hydraulic cylinder, employing the flexibility of an elastic rubber seal for fine positioning of the rod. The actuator is described as having steady state accuracy of 1 nm, but a stroke and maximum velocity too small for practical use.

The ElectroHydraulic Actuator (EHA) system developed by Habibi and Goldenberg in 1995 has shown promise in achieving high positional accuracy at high strokes and loads. The EHA is a closed hydraulic system, where flow returning from the actuator is fed directly into the inlet of the pump rather than into a reservoir. Central to the design of this system is a unique symmetric single rod actuator and the use of inner-loop velocity feedback around the motor, which eliminates deadzone in the motor/pump subsystem. As a result of these design elements, the designers found that the EHA showed great potential with respect to high-accuracy, high-load positioning. As a result, the EHA became the focus of this research project. A detailed description of the operation of the system is deferred to Section 2.2.

In summary, an extensive review of the literature has shown that there are a limited number

of published papers on the topic of high-accuracy hydraulic actuation systems. As such, it was believed that this remains an area in which substantive research needs to be conducted.

1.3 Objectives

Given the limitations inherent in each of the described actuation technologies, there remains a niche for an actuator that combines high accuracy, a large stroke and high force output. The ElectroHydraulic Actuator (EHA) designed by Habibi and Goldenberg was initially developed as a compact, energy-efficient actuator with an accuracy in the micro-scale. It is true that other actuation methods may be able to match or exceed the capabilities of the EHA in specific categories. However, the configuration of the EHA creates a combination of attributes which make it unique in the field of high-accuracy actuation.

The objective of this study is to modify the EHA so that it achieves nano-scale positional accuracy while maintaining its desirable properties of high force output and large stroke. In order to achieve this objective, it is anticipated that both hardware modification and controller redesign is required. This thesis documents the process of completing some of these modifications, the implementation of a fuzzy controller, and the resulting performance of the EHA with both an inertial and a resistive load.

1.4 Thesis outline

This thesis is organized much like the progression of the research process. In Chapter 2, the background of the ElectroHydraulic Actuator (EHA) is given, including a basic review of hydraulic systems, a linear model of the EHA, its parameters and nonlinearities. Chapter 3 discusses the apparatus and the instrumentation used to control and measure the performance of the EHA. In Chapter 4, control of the EHA using traditional methods is investigated. Next, Chapter 5 introduces the topic of fuzzy control and its implementation in the EHA. Chapter 6 presents the comparative results of the experimental tests conducted on the EHA employing both traditional and fuzzy controllers. Finally, Chapter 7 presents the conclusions of this thesis and forwards some recommendations for possible future research.

CHAPTER 2

BACKGROUND OF EHA

2.1 History

The ElectroHydraulic Actuator (EHA) was developed by Dr. A. Goldenberg and Dr. S. Habibi in 1995 [11]. The EHA was intended to address some of the drawbacks of traditional hydraulic systems including high energy use, bulkiness, and the use of expensive servo valves.

Consider Fig. 2.1, which illustrates a typical hydraulic system.

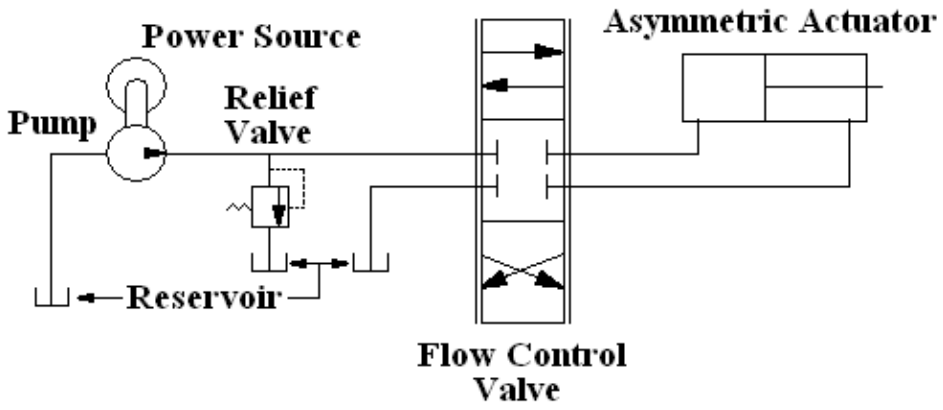

Figure 2.1: Example of a traditional hydraulic system

In order to understand the EHA system, it is useful to first discuss traditional hydraulic systems, which consist of the following components:

- A control system
- A hydraulic power supply consisting of:
  - A power source
  - A pump
  - A relief valve

- A fluid reservoir
- Control valve(s)
- An actuator

The function of the hydraulic power supply is to provide hydraulic fluid at a specified pressure and flow rate. The control valve(s) then modulates and distributes flow to the actuators. The actuators use the energy transported by the fluid in order to produce rotary or linear motion. The reservoir contains the fluid employed in the system and serves to cool and condition the fluid prior to its distribution through the system.

In many traditional hydraulic systems, single-rod asymmetric actuators are used to provide linear actuation. These actuators are asymmetric because the active pressure areas on each side of the piston are unequal, as seen in Fig. 2.2. This causes multiple problems: first, the performance of the actuator is different depending on the direction of motion and second, the use of a "hydrostatic" hydraulic system is complicated by the unequal areas.

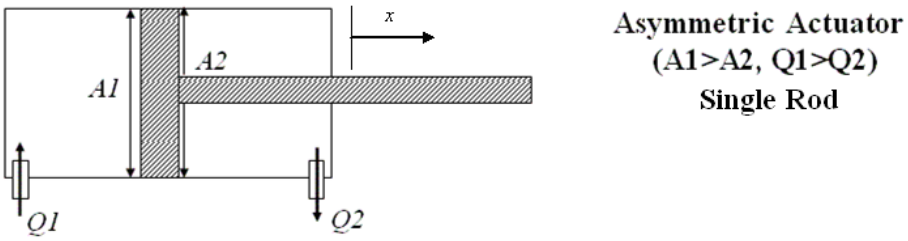

Figure 2.2: Schematic of asymmetric actuator

A hydrostatic hydraulic system is one where the fluid reservoir is eliminated, as shown in Fig. 2.3. Typically each side of the pump is connected directly to the actuator, eliminating the requirement for control valves. Instead, actuator motion is controlled by changing the displacement (amount of fluid pumped per revolution) or shaft speed of the pump. Hydrostatic actuation is desirable for three reasons. First, the removal of the fluid reservoir reduces the volume of space occupied by the actuation system. Second, the energy efficiency of the system can be higher than with traditional valve-controlled hydraulics [12]. This is due to the elimination of energy dissipated as waste heat in the flow control valves. Finally, the removal of the control valves can decrease the overall cost of the system.

It should be mentioned that the previous two points depend on the implementation of the hy-

drostatic system, especially the use of "make-up" pumps. Because even closed hydraulic systems have some fluid leakage, a small amount of additional fluid needs to be introduced into the system to make up for the fluid lost. Typically a small pump known as a make-up pump delivers this small flow, requiring energy from the power source and adding cost to the system. This has the potential of limiting the advantages stated for a hydrostatic system above.

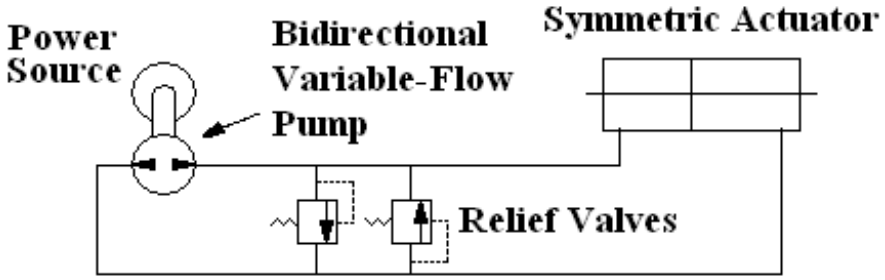

Figure 2.3: Example of a hydrostatic hydraulic system

Using a symmetric double-rod actuator as seen in Fig. 2.4 (or an inherently-symmetric hydraulic motor) is the conventional solution to enable the use of a hydrostatic system, as the active pressure areas on each side of the piston are equal. However, this solution is undesirable in robotics, as the unused second rod creates an area of "dead space" which reduces the workspace of the robot.

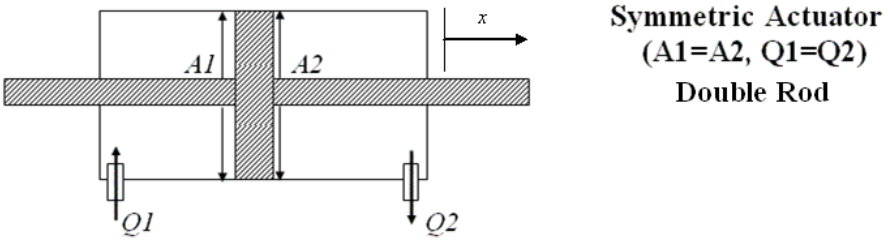

Figure 2.4: Schematic of symmetric actuator

Motivated by this deficiency inherent in double-rod actuators and desiring to create a superior actuator for use in robotics, a novel symmetrical linear actuator was designed by Goldenberg and Habibi [13]. Fig. 2.5 shows the internal construction of the actuator. This new actuator is key to the EHA as it has a single rod yet equal active pressure areas, facilitating its use in a hydrostatic

hydraulic system. The ability to use the actuator in a hydrostatic system reduces the energy usage and size of the system while eliminating the need for servo valves as control elements, as discussed in Section 2.2. The implementation of the symmetrical linear actuator in a compact hydrostatic hydraulic system is the essence of the EHA. The design and operation of the EHA concept is also presented in Section 2.2.

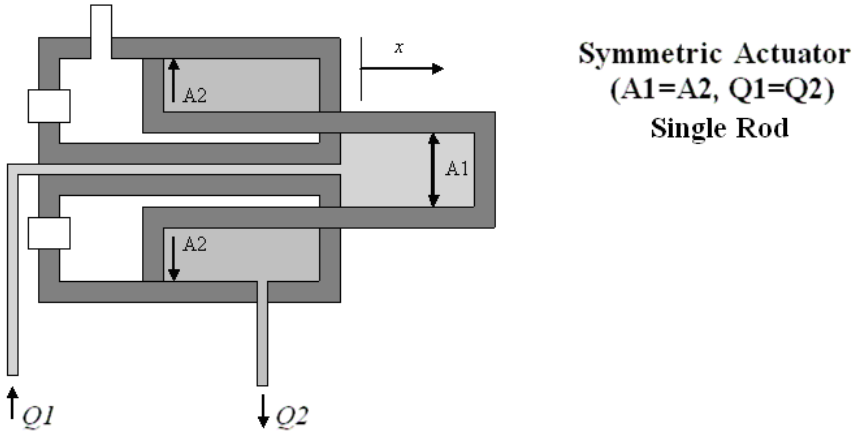

Figure 2.5: Schematic of symmetric linear actuator

2.2 EHA system

Aside from the physical constraints imposed on symmetric actuators employed in robotic systems, the EHA addresses several issues that exist with traditional hydraulic systems as described in Section 2.1. First, the energy efficiency of traditional valve-controlled hydraulic systems is highly dependent on the load, duty cycle and relief valve pressure. This is due to the use of valve-based flow control rather than pump-based flow control.

Consider Fig. 2.1; this circuit consists of a fixed displacement pump, reservoir, relief valve, flow modulating valve and an actuator. For the sake of discussion, assume that the actuator in this system requires X l/m of flow at Y kPa, but the pump delivers 3X l/m of flow. The excess flow must be diverted to the fluid reservoir through the relief valve. In addition, a pressure drop across the flow modulating valve may exist because the pump pressure is at the relief valve setting, which is often higher than the required load pressure. The energy contained in this unused flow going to the reservoir through the relief valve and some of the energy in the flow through the modulating valve is converted to heat. This reduces the efficiency of the system. In contrast, a pump-based

flow control system such as a hydrostatic system delivers only the amount of fluid to the actuator that is required. Also the pump operates at the same pressure as the load, neglecting the typically minor pressure drop that occurs due to the piping between the pump and the load. Because the pump delivers only the required amount of fluid at the required load pressure, the overall efficiency of a hydrostatic system can be higher than with a traditional hydraulic system.

A second issue with traditional hydraulic systems is that the fluid reservoir occupies a significant volume and weight, as it is recommended that the volume of the reservoir be 2-3 times the flow rate in the system. Third, the performance of a system using a traditional single-rod linear actuator depends on the direction of motion. This is due to the unequal areas on each side of the actuator piston, which causes rod velocity and output force to vary with direction of travel for a fixed flow input and pressure differential. This can be corrected through the use of valve control or a variable-speed pump, but as demonstrated by Habibi and Goldenberg [11], the damping ratio of the system will still vary with direction of travel. This is detrimental to the operation of robotic systems, as the system will have to be tuned in order to remain stable in both cases, which artificially limits its performance.

A final issue with traditional hydraulic systems arises when the load is "over-running" or "runaway" rather than inertial or resistive. An over-running load is one where the external force on the actuator acts in the same direction as the motion of the actuator. With traditional non-closed hydraulic systems, additional valves are required to prevent the load from accelerating uncontrollably with an over-running load. In the case of hydrostatic systems, the pump/motor combination itself acts as a brake. This occurs because the force acting to accelerate the load is translated into a shaft torque by the pump, and the motor resists this torque.

Given the drawbacks of traditional hydraulic systems, consider the EHA which consists of the following elements:

- A control System
- A hydraulic power supply consisting of:
  - An electric Motor
  - A bi-directional pump (gear pump in prototype)
  - An accumulator or other make-up fluid system
  - A crossover relief valve
- A Symmetrical linear actuator or rotary motor

The schematic shown in Fig. 2.6 illustrates the construction of the EHA as designed by Habibi and Goldenberg:

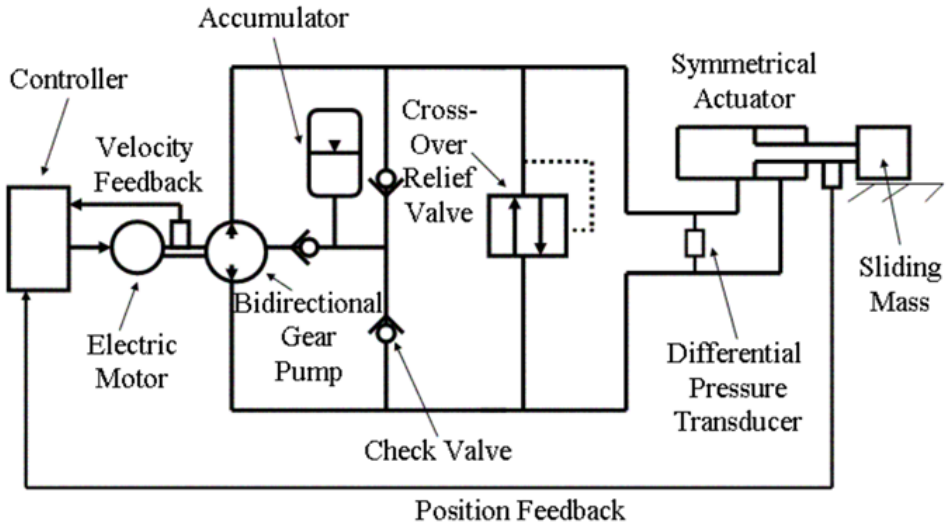

Figure 2.6: Schematic of the EHA, from Chinniah 2004

Significant in the EHA is the absence of a bulky fluid reservoir and expensive servo valve. Instead, the flow of hydraulic fluid to the actuator is controlled by the rotation of the electric motor, which drives the bi-directional gear pump directly. The gear pump has a "fixed displacement", which is defined as volume of fluid pumped per rotation of the pump. Therefore the flow rate of the fluid is directly proportional to the speed of the pump and hence speed of the electric motor.

Since the active pressure areas on each side of the actuator's rod are equal, an equal amount of fluid leaves the symmetrical actuator as enters it. This allows the output of the pump to be directly connected to one chamber of the actuator while the input of the pump is connected to the other chamber of the actuator. In addition, a small accumulator maintains a minimum system pressure to prevent cavitation of the hydraulic fluid, and also makes up for fluid lost due to external leakage. The end result is that controlling the rotational speed of the electric motor produces a proportional actuator rod velocity. The direction of motion of the rod can be controlled by changing the direction of rotation of the electric motor.

The construction of the symmetrical actuator is as shown in Fig. 2.7. Fluid enters the two active chambers $C1$ and $C2$ via ports $O1$ and $O2$. The working area of the two chambers, $A1$ and $A2$, are made equal in area. This is possible because the rod is hollow, creating chamber $C1$ with active

area $A1$ at the end of the rod. Chamber $C2$ is located between the outside of the rod and the inside of the cylinder. Its active area is $A2$, a circular disc attached to the inside end of the rod that seals against the cylinder wall. Also, a third working chamber, $C3$, can be pressurized using port $O3$ in order to provide a bias force. This bias force can be used to counteract a constant external force such as a gravity load. Fluid leakage paths are indicated by the arrows labeled L.

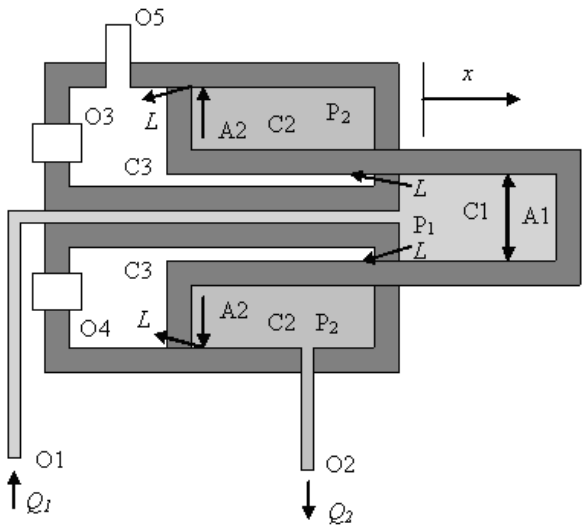

Figure 2.7: Cross-section of the actuator

2.3 Linear models

Habibi and Singh developed a linear model of the EHA to enable simulation of the system and to facilitate the design of controllers for the EHA [14]. This linear model was expanded by the author of this study to allow for the ability to simulate the effect of arbitrary force and flow disturbances on the EHA [15] (for further details see Appendix D). The parameters and nomenclature described in the following section are listed in Table 2.1.

In Fig. 2.8, a simplified block diagram of the EHA is shown. Visible are its constituent elements including the outer-loop controller, the motor/pump subsystem including an inner-loop velocity controller, the hydraulic pump, and the hydraulic subsystem including the hydrostatic circuit and mechanical load. In this case, the load is considered to be primarily inertial, consisting of a fixed mass that slides on horizontal rails. A viscous friction force is included in the model to account for the friction caused by the load rails and actuator seals.

The open-loop output of the EHA in response to a control voltage U is rod velocity, as mentioned in Section 2.2. In order to allow the EHA to function effectively as a positioning system, feedback is required. By measuring the output position using a sensor it is possible to convert the system into a closed-loop positioning system. To do this, the measured rod position is subtracted from the desired position to obtain an error signal. This error signal then becomes the input to the EHA motor controller.

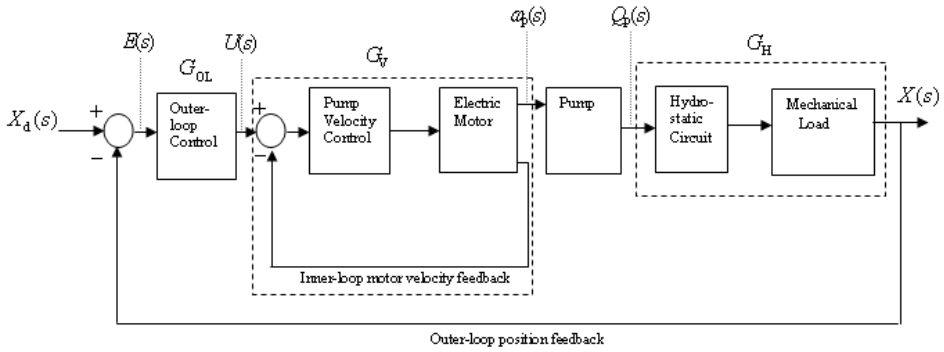

Figure 2.8: Simplified EHA block diagram

Figure 2.9 shows the full block diagram of the EHA including the hydraulic transfer function G_H. Included in this diagram are two disturbance inputs, one for flow disturbances Q_{dis} and one for force disturbances F_{dis}. Flow disturbances are defined as any load flow that is not included in the EHA model, such as leakage of fluid from the system. Similarly force disturbances are defined as any external force applied to the load that is not included in the EHA model. Example of force disturbances could be acceleration forces created by external vibration of the system or application of external forces, such as gravity, to the load.

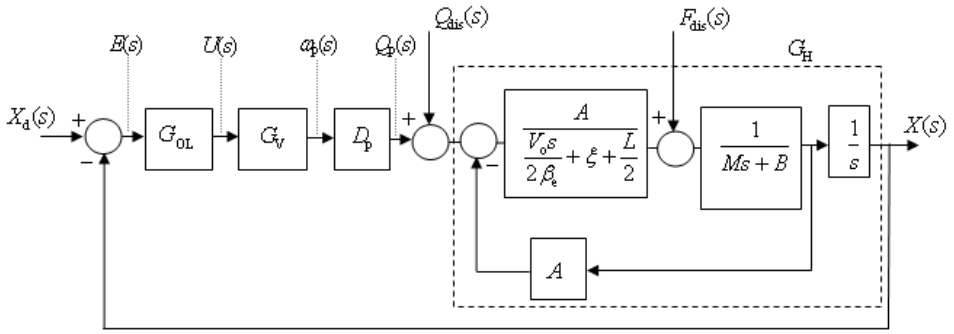

Figure 2.9: EHA block diagram, from Sampson et al. 2004

Sampson et al. presented the closed-loop transfer function between the desired input X_d and the output X [15] (for further details see Appendix D):

$$\frac{X(s)}{X_d(s)} = \frac{K_h\omega_{nh}^2 D_p G_V(s) G_{OL}(s)}{s(s^2+2\zeta_h\omega_{nh}^2)+K_h\omega_{nh}^2 D_p G_V(s) G_{OL}(s)} \tag{2.1}$$

Also developed was the transfer function between a flow disturbance Q_{dis} and the output X:

$$\frac{X(s)}{Q_{dis}(s)} = \frac{K_h\omega_{nh}^2}{s(s^2+2\zeta_h\omega_{nh}^2)+K_h\omega_{nh}^2 D_p G_V(s) G_{OL}(s)} \tag{2.2}$$

Finally, the transfer function between a force disturbance F_{dis} and the output X was developed and found to be as follows:

$$\frac{X(s)}{F_{dis}(s)} = \frac{1/M}{s^2+\frac{B}{M}s+\frac{K_h\omega_{nh}^2(As+D_p G_V(s) G_{OL}(s))}{s+(2\xi+L)\beta_e/V_o}} \tag{2.3}$$

In all cases, the hydraulic gain K_h, damping factor ζ_h and the natural frequency ω_{nh} are defined to be:

$$K_h = \frac{2A}{2A^2+(2\xi+L)B} \tag{2.4}$$

$$\zeta_h = \frac{B}{2M\omega_{nh}} + \frac{(2\xi+L)\beta_e}{2V_o\omega_{nh}} \tag{2.5}$$

$$\omega_{nh} = \sqrt{\frac{2\beta_e A^2+(2\xi+L)\beta_e B}{MV_o}} \tag{2.6}$$

It can be seen by examining the transfer function for the flow and force disturbances that the open-loop controller $G_{ol}(s)$ affects the response of the system to these disturbances. As described

by Sampson et al. [15], the steady state error due to a given flow disturbance $Q_{dis}(s)$ was found to be:

$$E_{ss} = \lim_{s \to 0} \left[\frac{sQ_{dis(s)}}{D_p K_v G_{ol}(s)} \right] \tag{2.7}$$

Similarly, the steady state error due to a force disturbance $F_{dis}(s)$ was found to be:

$$E_{ss} = \lim_{s \to 0} \left[\frac{sF_{dis(s)}(2\xi + L)}{2AD_p K_v G_{ol}(s)} \right] \tag{2.8}$$

Using the preceding equations, the response of the EHA to arbitrary time-varying inputs and disturbances can be simulated. Also, the steady state error in response to a given flow or force disturbance can be calculated. This capability is valuable for the study of the response of the EHA to both desired and undesired inputs, which is important information in the design of controllers for the system.

2.4 Parameters

Many of the parameters of the EHA have been measured since its development approximately ten years ago. However, certain parameters are very difficult to measure in a direct manner. To obtain an accurate estimate of a number of these parameters, Chinniah used Extended Kalman filtering [16, 17].

Kalman filtering is a predictor-corrector method traditionally used to estimate system states. It does so by comparing the predicted output of a simulated system to the output of the real system at a given time with the same input signal. The error between the two outputs is used to correct the estimated states, hence the predictor-corrector label. This technique can be used to predict system parameters by reformulating them as states. In the EHA, two particular parameters that are important to the behaviour of the EHA but are very difficult to measure directly are the bulk modulus and the viscous damping factor.

The bulk modulus of a hydraulic fluid is defined as the fractional change in volume of the fluid for a given change in fluid pressure. Since the mechanical components that are under pressure in the EHA are compressible, they also contribute to the effective bulk modulus of the system. However, since the steel components and pipes used in the EHA are much stiffer than the hydraulic fluid, the fluid dominates the effective bulk modulus. It should be noted that the presence of entrained air in a fluid can reduce the bulk modulus of the fluid substantially. The amount of entrained air at any

time is very operating-condition dependent. Thus it is very important to be able to estimate the bulk modulus in an "online" manner. The technique proposed by Chinniah is able to do this.

The viscous friction factor defines the velocity-dependent force exerted on the actuator by its internal friction and the load. Typically friction in a hydraulic system is modeled as the sum of a direction-dependent static or Coulomb friction force and a velocity-dependent viscous friction force. Using the Kalman filtering technique, Chinniah found that the relationship between actuator velocity and frictional force was more complex, being a roughly parabolic relationship as shown in Fig. 2.10 [16, 18]. In spite of this, an estimate for the viscous damping force was obtained.

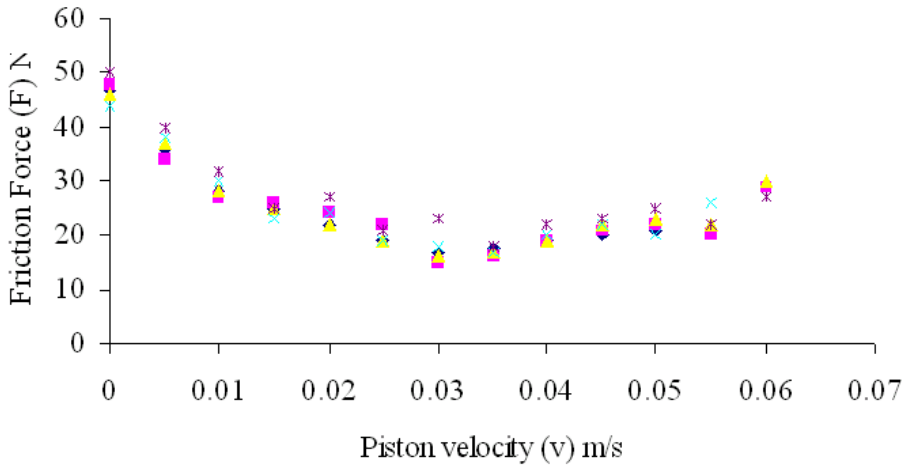

Figure 2.10: Friction as a function of velocity, from Chinniah 2004

Table 2.1 presents the nomenclature and the known parameters of the EHA, obtained using both direct measurement and indirect estimation using Kalman filtering. The latter are indicated with a star beside the value of the parameter.

2.5 Nonlinearities

In order for a system to be classified as being linear, it must satisfy two conditions: homogeneity and superposition. Given a system G, gain α, and arbitrary inputs X and outputs Y these conditions are defined as follows:

- Homogeneity: Given $X \xrightarrow{G} Y$ then $\alpha X \xrightarrow{G} \alpha Y$ for all α
- Superposition: Given $X_1 \xrightarrow{G} Y_1$ and $X_2 \xrightarrow{G} Y_2$ then $X_1 + X_2 \xrightarrow{G} Y_1 + Y_2$

Table 2.1: Parameters and nomenclature

$A_1 = A_2 = A$	Actuator pressure area	$5.05x10^{-4}m^2$
B	Coefficient of friction at load	*760 N/m/s
C_1, C_2, C_3	Actuator chambers	
D_p	Pump volumetric displacement	$1.6925x10^{-7}m^3/rad$
$E(s)$	Error signal	V
E_{ss}	Steady-state positional error	m
F_{dis}	External force displacement	N
$G_{OL}(s)$	Outer loop controller	
$G_V(s)$	Motor/Pump subsystem transfer function	
$G_H(s)$	Hydraulic transfer function	
K_p, K_i	Controller gains	
L	Leakage coefficient	$*2x10^{-15}m^3/s/Pa$
M	Load mass	20 kg
$O1, O2, O3, O4, O5$	Actuator ports	
P_1, P_2	Actuator chamber pressure	Pa
Q_1, Q_2	Actuator chamber flow	m^3/s
$Q_{dis}(s)$	Disturbance flow	m^3/s
$Q_p(s)$	Pump flow	m^3/s
$U(s)$	Motor input voltage	V
V_o	Pipe plus mean actuator chamber volumes	$6.1x10^{-5}m^3$
$X(s)$	Position of actuator	m
$X_d(s)$	Demanded position of actuator	m
β_e	Effective bulk modulus of hydraulic oil	$*2.1x10^8Pa$
K_h	Hydraulic gain	
K_m	Motor gain	40.55 rad/s/V
ω_{nh}	Hydraulic undamped natural frequency	
$\omega_p(s)$	Pump angular velocity rad/s	
ξ	Pump cross-port leakage coefficient	$*1.5x10^{-13}m^3/s/Pa$
ζ_h	Hydraulic damping ratio	
* values obtained from Chinniah (2004)		

In practice, systems are said to be linear if they approach these theoretical ideals within an acceptable tolerance.

The EHA cannot be classified as a linear system due to the dependence of the output on the magnitude of the input [19, 20]. For example, the steady-state gain of the system varies with the magnitude of the input voltage. Also, the frequency response of the system varies significantly with the magnitude of the input voltage. These qualities can be observed in Fig. 2.11, which was reproduced from Sampson et al. [20] (for further details see Appendix E). This figure shows the experimental response of the system as a function of both frequency and input voltage. It is observed that the high-frequency performance is highly dependent on the input voltage. This demonstrates that the EHA is a nonlinear system that can only be approximated as a linear system for a given operating condition.

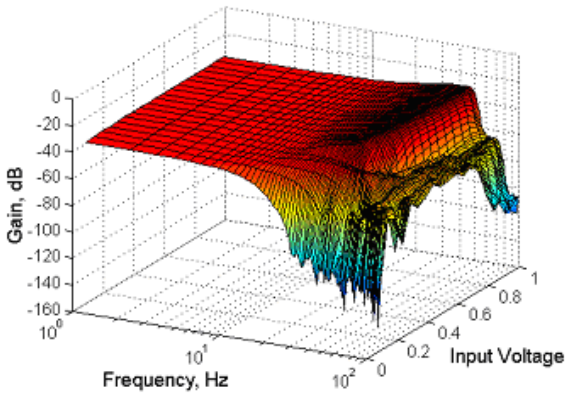

Figure 2.11: Experimental Bode gain plot as function of input voltage, from Sampson et al. 2005

As a result of the nonlinearity of the EHA, the use of a traditional linear controller such as a Proportional Integral Derivative (PID) controller will not result in nano-scale positional accuracy over all operating ranges of the system. This is shown in Chapterřefchap:results. This has led to the investigation of nonlinear controllers for the EHA, such as the switched-gain proportional controller discussed in Chapter 4 and the fuzzy controller discussed in Chapter 5.

CHAPTER 3

APPARATUS AND INSTRUMENTATION

3.1 Overview

Fig. 3.1 shows a picture of the EHA as it currently exists [21]. Visible are the major components of the system such as the actuator, load, motor, pump, accumulator, and relief valve. Not shown are the motor controller and the computer control system. The size of the apparatus can be determined using the 10 cm scale superimposed on the picture.

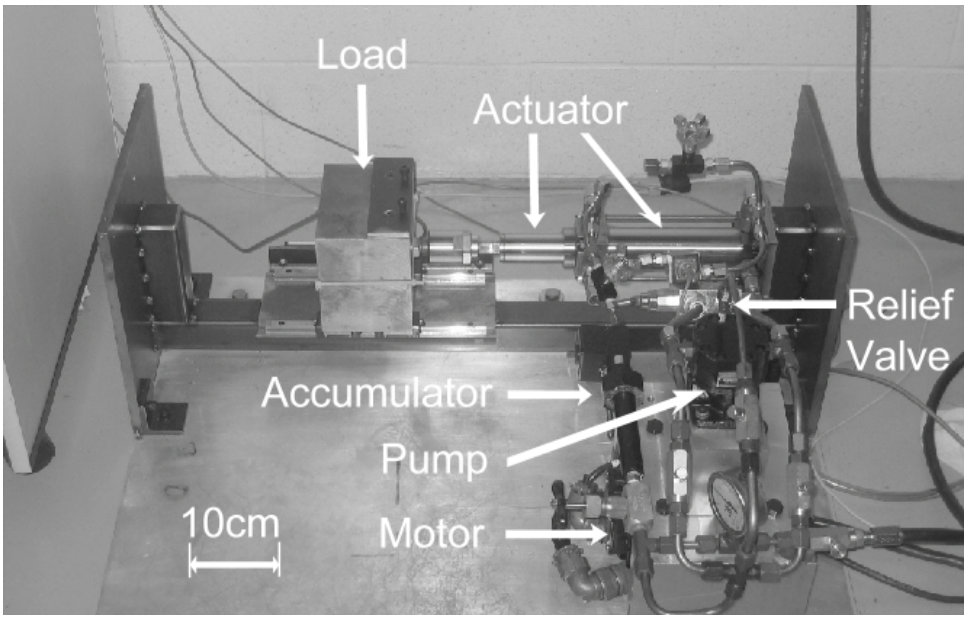

Figure 3.1: Picture of EHA apparatus, modified from Chinniah

In the schematic of the EHA discussed in Chapter 2, the controller was shown as a "black box" that received position and velocity feedback from the system, and calculated a control output for the electric motor. The controller in the actual system is divided into two units, the computer control system and the motor controller. This is shown in Fig. 3.2. The motor control system is an

Infranor SMT-BD1 digital PWM (Pulse Width Modulation) servo amplifier, which provides speed control for sinusoidal brushless AC (Alternating Current) motors. It has two demanded-velocity inputs, an analog input of $\pm$ 10 Volts and a digital RS232 serial input. It also has a motor velocity feedback input, which is used by the amplifier to permit closed-loop motor velocity control. The output of the amplifier is the PWM current required to drive the motor at the demanded velocity. The AC servo motor is a Mavilor MA-30, which has a maximum speed of $\pm$ 4000 rpm, a stall torque of 16 Nm, and a time constant of 10 ms. It also incorporates a magnetic speed resolver, which is used to provide motor velocity feedback to the amplifier.

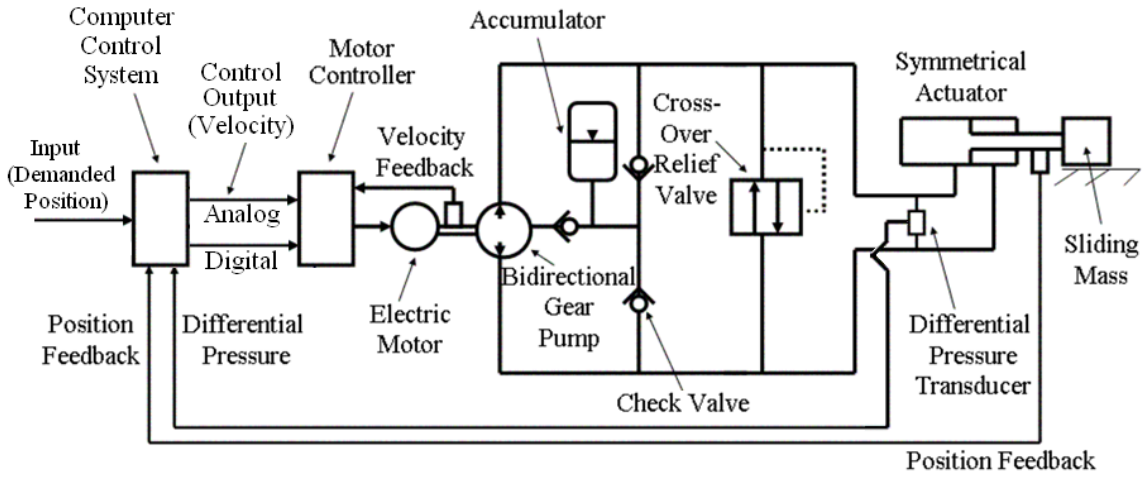

Figure 3.2: Detailed schematic of EHA

The computer control system consists of a personal computer equipped with a digital optical encoder input card, a 16-bit data acquisition system, and the MATLAB Simulink and Real-Time Windows Target control development system operating at a sample rate of 1000 Hz. Fig. 3.3 shows how the real-time control environment appears to the end user. The desired position input is specified in meters in the step input or signal generator blocks as seen in the leftmost part of the control environment. The position of the load is measured by an optical encoder, which is discussed in Section 3.2. The digital signal from the encoder is read by the encoder input card. The block labelled "Gurley ISC3N encoder card" represents the driver which reads the position data from the encoder input card and brings it into the control environment in real time. The position signal is then subtracted from the desired position to create an error signal. This error signal then

passes into the controller subsystem.

In the example shown in Fig. 3.3, the controller is a simple proportional gain of K_p=585. The output of the controller subsystem is the desired motor velocity in Volts. This signal is sent to the motor controller through one of two outputs, either analog or digital. The analog signal is passed to the data acquisition system, which is connected by wires to the analog motor control input. Alternatively, the desired motor velocity can be transmitted to the motor controller over a digital (serial RS232) link.

This is the case in Fig. 3.3, where the block labelled "serial_cstm" translates the desired motor velocity in Volts into the corresponding desired velocity in rpm and transmits the value to the motor controller serially with a transmission rate of approximately 90 Hz. The resolution in terms of the input velocity signal of the motor amplifier is 16 bits. Given a maximum motor velocity of $\pm$ 4000 rpm, the resolution of the motor's velocity is 0.122 rpm/bit. This is theoretically true regardless of which transmission medium is used, but the noise present in the analog transmission of the velocity signal, as discussed below, reduces the effective velocity resolution when using the analog input.

The digital output was implemented because it was found that the noise pickup caused by the analog transmission of the desired motor velocity to the motor controller resulted in errors in the load position. Due to the PWM speed control employed by the motor amplifier, large currents at a high switching rate pass from the motor amplifier to the motor through the motor power cables. This causes the cables, the amplifier and the motor itself to emit significant EMI (ElectroMagnetic Interference) and RFI (Radio Frequency Interference). This interference is coupled into the cables carrying the demanded velocity signal. The interference presents an undesired voltage to analog amplifier inputs, appearing as noise in the demanded velocity signal. This noise causes motor rotation, which leads to undesired load motion thus reducing the positional accuracy of the system.

Finally, Fig. 3.3 shows two analog inputs from the data acquisition system on the right side of the control environment. These inputs represent the system differential pressure and motor velocity. The signals are converted into physical units and then plotted and stored for later analysis.

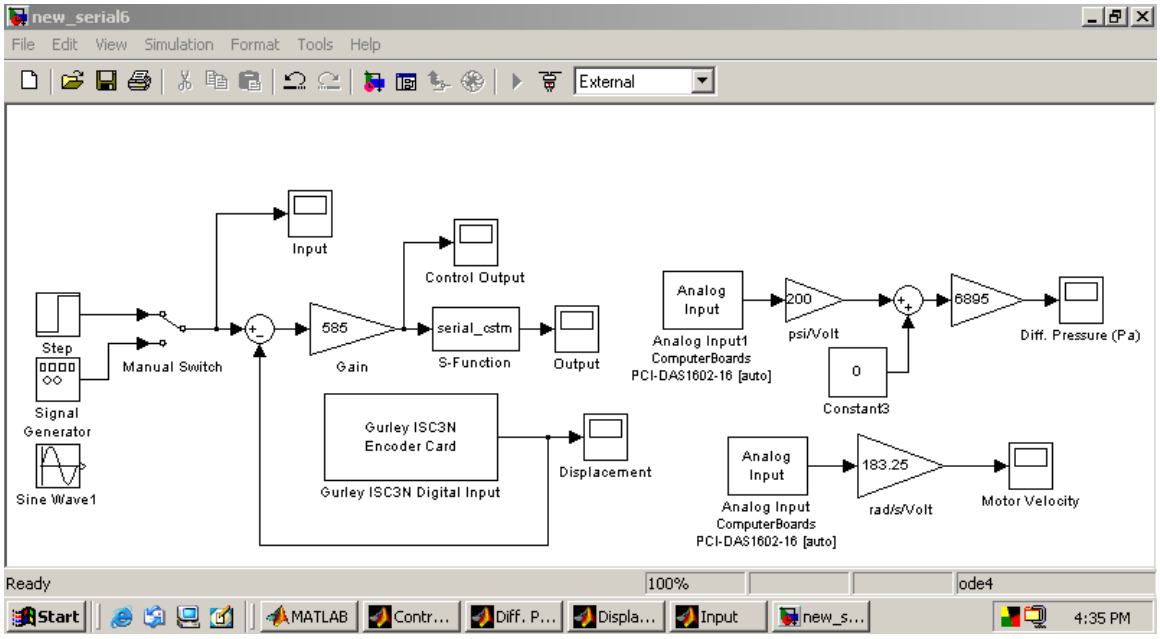

Figure 3.3: MATLAB real-time control environment

In order to use the optical encoder and serial output with the MATLAB real-time control environment, custom device drivers were written in the C language. These drivers act as an interface between the control environment and the computer hardware, reading data from the hardware and then writing data to the hardware to change settings and send out commands. The driver code used to interface with the optical encoder input card and the driver code used to interface with the SMT-BD1 motor controller is reproduced in 7.2.

In order to prepare the motor controller to receive commands from the MATLAB control environment, a number of text files are first sent to the controller using a "terminal communication program" such as HyperTerminal. These files are reproduced in 7.2. The motor controller understands serial communications using the settings of 9600 bps, 8 data bits, (N)o parity, 1 stop bit, and Xon/Xoff flow control.

With respect to Fig. 3.3, to prepare the motor controller to receive serial speed commands from the control environment, the file "initiate.txt" must be transferred to the controller, followed by the file "manual.txt". At this point the load can be moved by the actuator through setting the motor speed by sending the "100rpm.txt","neg100rpm.txt", and "0rpm.txt" files to the motor controller. When the load has been moved to the desired starting position for computer control

using MATLAB, the demanded velocity signals can then be sent to the controller over the serial link or the analog link. In order to use the analog input of the motor controller, the "automatic.txt" file must first be sent. In either case, the motor can be stopped by sending the "stop.txt" file. To prepare the motor controller for shutdown, the "disconnect.txt" file can be sent.

3.2 Position sensor

Since the primary goal of the EHA is high positional accuracy, the position of the load must be measured to at least the desired accuracy of the system. In this case, the desired accuracy is in the order of nanometers (nm), which is 10^{-9}m. Traditional position measurement solutions such as Linear Variable Differential Transformers (LVDT) and linear potentiometers are not capable of this level of accuracy. The best LVDT linearity available is 0.1% of full scale travel. In the case of the 120mm sensor length required by the EHA, the highest accuracy achievable by a LVDT would be $\pm 120\mu$m, which is greater than a tenth of a mm. Therefore, the load position is measured using a linear incremental optical encoder.

Incremental optical encoders operate on the principle of interrupting the transmission of light to a photodector by a masked glass slide. A simplified representation of a rotary optical encoder is shown in Fig. 3.4. The glass slide has two sets of alternating clear and opaque rectangles arranged linearly along the direction of travel of the read head. The "read head" has two sets of LEDs (Light Emitting Diode) and photodetectors on opposite sides of the glass slide, one set for each band of rectangles. The two bands of alternating rectangles are arranged 90° out of phase with each other, which enables the use of quadrature encoding.

Quadrature encoding allows the processing circuitry to determine the direction of motion as well as the incremental change in position of the read head. This is done using the phase difference between the signals coming from the two photodetectors. The photodectors output a high signal when they receive light transmitted through a clear section of the glass slide, and a low signal when the light is block by an opaque section of the glass slide. Circuitry in the read head translates the high/low signals from the photodetectors into a count signal that increments or decrements depending on the direction of movement of the read head. This signal is transmitted in serial format to the PCI (Peripheral Component Interconnect) encoder card connected to the control computer.

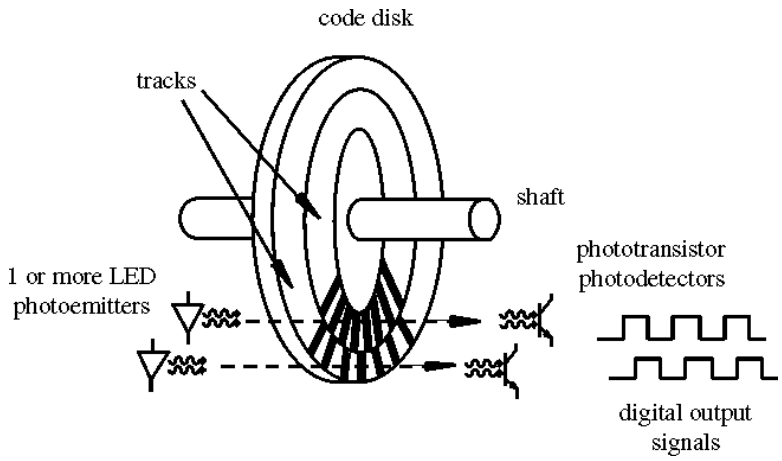

Figure 3.4: Functional diagram of rotary optical encoder, with permission from D. Alciatore and M. Histand

The optical encoder employed in the EHA has a resolution of 50 nm and a guaranteed accuracy of $\pm 3\mu$m/m; the reason for the specification of both resolution and accuracy is discussed in the following section. When the resolution of the encoder is specified as 50 nm, it means that the dark and clear areas on the glass slide are each 50 nm long. This is currently the practical minimum dimension achievable by the company that makes the encoder. It is important to note that for high-resolution sensors such as this, the accuracy and the resolution of the sensor are not equal. The resolution of the sensor is fixed by manufacturing tolerances, and will not change over time except through destructive damage to the printing on the glass slide.

The accuracy of the encoder, however, is not a fixed quantity. It will depend on the strain induced by the mounting of the encoder, the actual length of each individual nominally 50 nm area on the slide, the temperature of the environment, the proper mounting of the read head, and other factors. In addition, the accuracy of a given position measurement depends on the magnitude of displacement. As stated above, the guaranteed accuracy of the encoder is given as $\pm 3\mu$m/m. This means that if the encoder was one meter long and the read head traversed the entire distance, the maximum deviation from a National Institute of Standards and Technology (NIST) traceable measurement standard is 3μm.

Since the travel of the encoder used in this study is 14 cm, and typical displacements of the load range from centimeters to microns, the true accuracy of the measurements presented in this study is more difficult to define. The absolute accuracy of the encoder over the displacements used in the tests will, however, be significantly better than the worst-case accuracy stated by the manufacturer.

Furthermore, the nature of a feedback system is such that if the positional error is within a certain multiple of the resolution of the feedback sensor, the sensor can be replaced with one of equal resolution and increased accuracy in order to achieve increased absolute positional accuracy while achieving the same positional error in terms of resolution.

3.3 Position sensor calibration

Each encoder is tested by the manufacturer to ensure that it meets specifications before it is shipped to the customer. To do so, it is mounted in a temperature-controlled environment and coupled to a HP2258A laser interferometer position measuring system. This system has a resolution of 10 nm. The read head is displaced over its full range of motion while the HP2258A records the position of the read head, and the reading from the encoder itself is recorded. The error of the encoder relative to the HP2258A is obtained by subtracting the position measured by the encoder from the position measured by the HP2258A, which is treated as the reference. In this manner, an error plot which shows the deviation of the position reported by the encoder relative to the reference is produced. The local maximum deviations are compared to the standards of the manufacturer for encoder absolute accuracy.

In this case the guaranteed absolute accuracy is set at $\pm 3\mu$m/m. If the encoder error exceeds this value, it is rejected. Fig. 3.5 shows an error plot for an optical encoder with a stated $\pm 3\mu$m/m absolute accuracy, provided by the manufacturer. The table on the left side of the error plot gives the local maximum deviations in μm with respect to the reference and the displacement where they occurred. It can be seen that in this example, the maximum absolute error is 0.6μm at 4 mm and 24 mm. This is significantly less than that guaranteed by the manufacturer, and shows that the accuracy of these measurement devices is excellent. Finally, the encoder undergoes a series of quality assurance tests before it is sent to the customer.

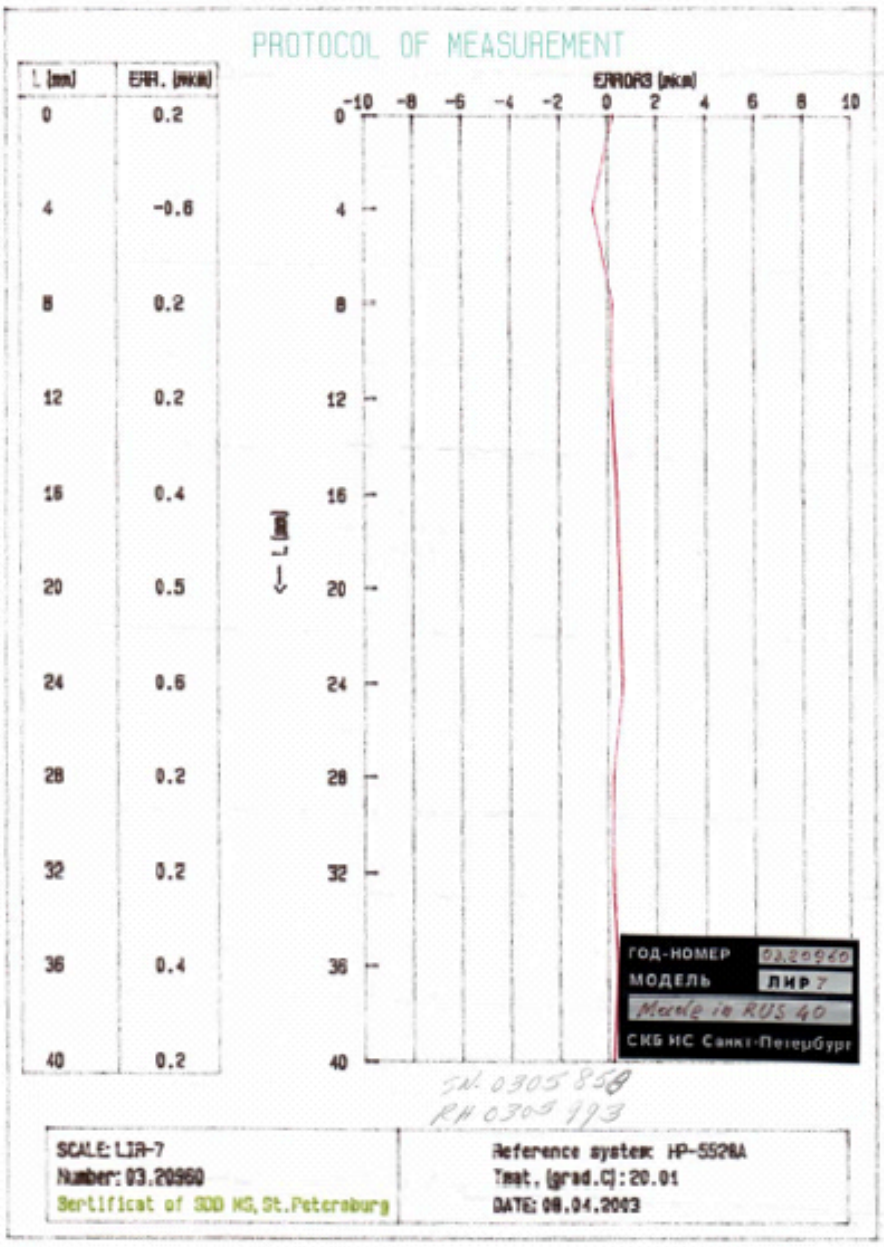

Figure 3.5: Error plot of typical optical encoder

CHAPTER 4

TRADITIONAL CONTROL

While the EHA itself may be physically capable of high-accuracy positioning, its controller has a significant influence on the achieved accuracy. In Section 3.1, the computer control system was introduced. The controller is the most important element of the control system. The controller takes as its input the error signal, which is obtained by subtracting the measured load position from the desired load position. The error signal then is processed by the control algorithm to develop a control signal. The goal of the control signal is to reduce the error signal to as close to zero as possible so that the desired and measured load positions are the same. This is the desired condition for a positioning system such as the EHA. At the beginning of this study, the EHA employed a proportional controller, which is discussed in the following section. The presentation of results of the experimental tests employing this and other controllers is deferred to Chapter 6.

4.1 Proportional control

The traditional solution for controllers employed in a feedback system is a Proportional-Integral-Derivative (PID) controller. In a PID controller, the control signal $u(t)$is calculated by summing three terms; the error signal $e(t)$ times a proportional gain K_P, the integral of the error signal times an integral gain K_I, and the differential of the error signal times a derivative gain K_D. A mathematical representation of the PID control algorithm is illustrated in Eq. 4.1. Each PID gain serves a different purpose; increasing the proportional gain increases the speed of response of the system. Introducing an integral gain tends to reduce the steady-state error of the system, but increases overshoot and decreases stability. Derivative gain tends to reduce the overshoot of a system by introducing additional damping, but also increases the settling time of the system.

$$u(t) = K_P\, e(t) + K_I \int e(t)dt + K_D \frac{de(t)}{dt} \tag{4.1}$$

In the case of the EHA, the designers wanted the system to have a no-overshoot ($\zeta = 1$) re-

sponse. This response implies that the controlled value does not overshoot the set point, in contrast to an underdamped response where it does overshoot. It was desired to have this type response so that the EHA would not overshoot the desired position when employed as an actuator in a robotic system. In this usage, overshoot may cause the actuated robot linkage to strike the work surface, which is undesired behaviour. Habibi and Goldenberg determined that a simple proportional controller gave the desired response, without the need for integral or derivative gain. The proportional gain was chosen by them to be K_P=585, as shown in Fig. 3.3 in Section 3.1. A simulated step response calculated using the transfer function Eq. 2.1 presented in Section 2.3 is shown in Fig. 4.1 [15]. This demonstrates the desirable no-overshoot response for a step input with a proportional controller of gain K_P=585.

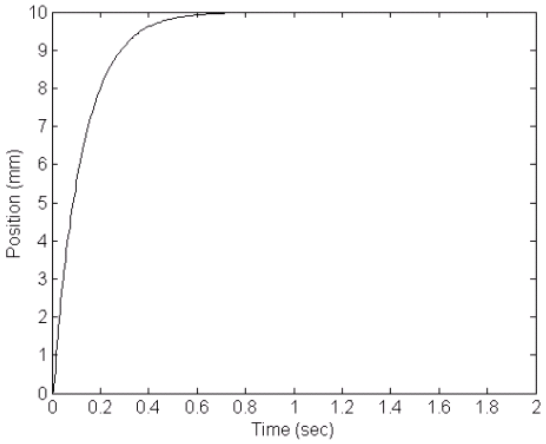

Figure 4.1: Simulated EHA response to 10 mm step input, from Sampson et al. 2004

4.2 Switched-gain control

Experimental tests performed on the EHA determined that the proportional controller was limiting the positional accuracy of the system. In Sampson et al., it was shown that the use of a proportional controller (PC) resulted in a positional accuracy of 2 μm, which was greater than the 1 μm sensor resolution employed during the testing [15] (for further details see Appendix D). Experiments with proportional-integral (PI) controllers found that adding the integral gain could reduce the positional error to less than 1 μm, but took an unacceptably long time to do so. Increasing the integral gain only resulted in limit-cycle oscillation, in that the integral gain built up enough force to overcome

the system static friction, only to overshoot the desired position and cause the cycle to repeat in the opposite direction.

It was shown in Sampson et al. that a switched-gain proportional controller (SGC) could improve the accuracy of the system to less than 1 μm without affecting its desirable large-scale response [15]. A SG controller is one where the parameters of the controller change depending on the operating condition of the system. In the case of the EHA, it was desired to maintain the large-scale response of the proportional controller with K_P=585 while improving the positional accuracy in the micro-scale. To do so, a switch was used in the computer control system to set the control gain K_P to be 585 when the positional error was greater than 5 μm, and set K_P=585+3000=3585 when the positional error was less than 5 μm. Figure 4.2 shows a schematic of the SG controller used in this study.

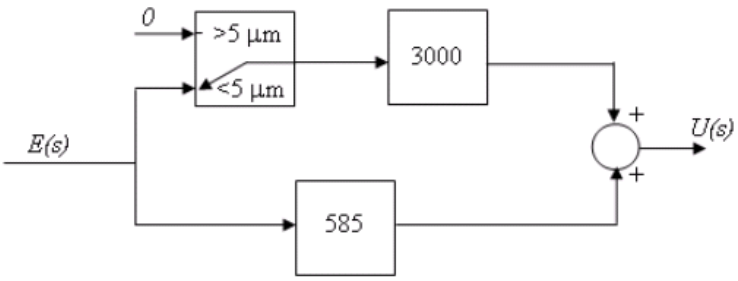

Figure 4.2: Switched-gain proportional controller, as used in Sampson et al. 2004

In Sampson et al. it was shown that a switched-gain controller improved the accuracy of the system compared to a proportional controller; however, the true accuracy of the system could not be measured until the position sensor was upgraded from 1μm resolution to 50 nm resolution. Despite the upgraded sensor, the maximum measurable accuracy of 50 nm could not be achieved using the switched-gain controller. This will be shown by experimental results presented in Chapter 6. In order to achieve this level of accuracy, it was decided a different control strategy should be investigated. Because of the nonlinear properties of the EHA at very small displacements, it was believed that some type of adaptive or fuzzy controller should be considered. After some preliminary studies, it was decided that fuzzy control should be implemented and tested. This controller is described in the following Chapter.

CHAPTER 5

FUZZY CONTROL

5.1 Background

The field of fuzzy control developed out of Lotfi Zadeh's dissatisfaction with the direction that control theory began to head during the 1950s and 1960s. He believed that control theory was becoming increasingly complex and mathematical, while neglecting the importance of practicality and applicability to real-world control problems. He suggested that his idea of fuzzy set theory, developed in 1965 [22], might be useful in addressing his concerns about conventional control theory.

The central idea of fuzzy set theory is that elements can have partial membership in a given set. In contrast, traditional set theory only allows elements to be either complete members of a set or complete non-members of a set. Fuzzy sets therefore allow for the expression of vagueness with respect to set membership, which reflects everyday experience better than traditional set theory. For example, consider the classification of a person as "tall." If it is believed that a person who is 6'2" is tall, where is the limit of being defined as tall? If it is agreed that a person taller than 6' is tall, would it make sense to describe someone who is 5'11.5" as "not tall?" Conventional set theory would say so, but fuzzy set theory would say that this person is "somewhat" or "partially" tall, hence the concept of partial set membership.

Given the advantage of being able to model the vagueness of real world variables, researchers began to investigate the use of fuzzy logic in control systems. The first practical investigations of fuzzy control began with Mamdani et al. In 1975, they applied the first fuzzy logic controller (FLC) to a steam engine [23]. Since that time, substantial research into fuzzy control systems has been performed. As such, fuzzy control is now recognized as a standard, established method of controlling feedback systems, especially those that are nonlinear, complex or ill-defined. In this Chapter, FLCs will be discussed starting with their basis in conventional and fuzzy set theory, and ending in Section 5.8 with a discussion of the FLC employed to control the EHA in this study.

5.2 Conventional sets

In traditional set theory, a set is defined as a collection of "crisp", or discrete, values taken from a universal set. For example, if a universal set X is defined as including all positive integers, or $X = \{1,2,3\ldots\}$, it is possible to define a set $A = \{1,3,6\}$. In a conventional set, each element x of X is either an element of A or not an element of A. This leads to the definition of a characteristic function $\chi_A(x)$ that expresses the membership of a given element x in the set A. If x is a member of A, then the characteristic function evaluated at x is exactly one; conversely, if x is not a member of A then the characteristic function evaluated at x is exactly zero. This is shown graphically in Fig. 5.1. In mathematical terms, the characteristic function is expressed as:

$$\chi(x) = \begin{cases} 1, \text{ if } x \in A \\ 0, \text{ if } x \notin A \end{cases} \tag{5.1}$$

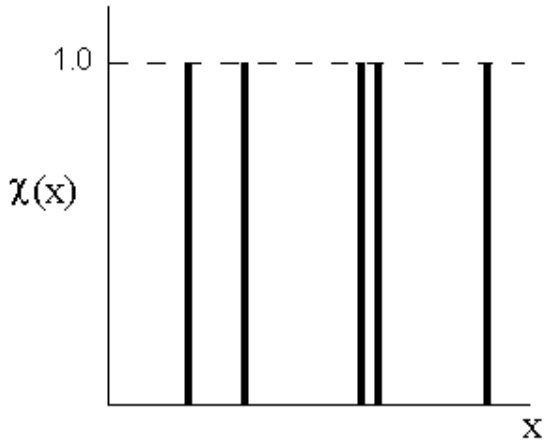

Figure 5.1: Conventional discrete set

5.3 Fuzzy sets

The fundamental difference between conventional set theory and fuzzy set theory is that instead of defining the elements x in the universal set X as being absolutely a member or not a member of set A, fuzzy sets allow the elements to have a degree of membership in fuzzy set A. This is achieved by replacing the characteristic function $\chi_A(x)$ by a membership function, which is allowed to have

values in the continuous range [0,1] inclusive. This membership function for a fuzzy set A is symbolized by $\mu_A(x)$, so that $\mu_A(x_1) = 0$ denotes that x_1 is completely not a member of A, while $\mu_A(x_2) = 1$ denotes that x_2 is completely a member of A. The difference comes when it is expressed that $\mu_A(x_3) = 0.57$, which indicates that x_3 has partial membership in the fuzzy set A. This can be represented mathematically as follows:

$$\mu(x) = \begin{cases} 1, \text{ if } x \in A \\ \{0,1\} \text{ if } x \text{ has partial membership in } A \\ 0, \text{ if } x \notin A \end{cases} \tag{5.2}$$

Given the description of fuzzy sets provided above, one can define a membership function (as used in fuzzy control) as being a fuzzy set that meets certain criteria. The first criterion is that the set must be defined on the real numbers. Second, it must be "normal", so that its values are in the range $0 \geq \mu_A(x) \leq 1$. The final criterion is that the set must be convex (every segment that connects points of the set lies entirely in the set). An example of a membership function is shown in Fig. 5.2. It contains one element corresponding to total membership in the set, where $\mu_A(x) = 1$. The set containing the value of x at this point is known as a fuzzy singleton.

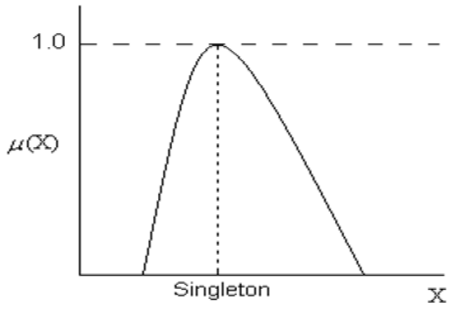

Figure 5.2: Example of a membership function

In order to simplify the implementation of fuzzy sets, the sets can be reduced into piecewise linear functions. Examples include triangular and trapezoidal membership functions, such as those shown in Fig. 5.3. These membership functions are described completely by the values labelled A1, A2, A3 or A4 in the Figure. Note that the maximum value of the membership function is exactly one at x=A2, which is the fuzzy singleton referred to earlier.

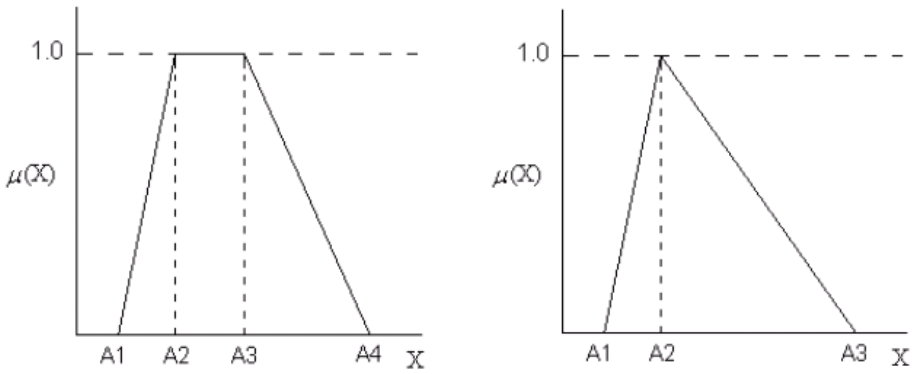

Figure 5.3: Example of membership functions - trapezoidal and triangular

5.4 Fuzzy logic versus probability

Often when first encountering fuzzy logic it is difficult to separate the concepts of fuzzy logic from that of probability. Indeed, the membership function shown in Fig. 5.2 looks similar to a Gaussian distribution as used in probability. However, there is a fundamental difference between the two; probability was developed to express randomness, while fuzzy logic was developed to express vagueness. This is more than a semantic distinction.

Presenting situations typifying randomness and vagueness can highlight this difference. Consider the concept of randomness. If one were to draw four cards from a deck, it is not known which four will be picked. However, once drawn, there is no ambiguity in the result of which four cards have been picked. In other words the outcome is random, but there is no vagueness of concept. On the other hand, if at a restaurant the waiter is tipped three dollars, is this tip considered to be low or high? In this case there is no aspect of randomness - the tip is an exact, known amount. However, there is a large degree of vagueness in the concept of describing the tip as being low or high. For example, it depends on the salary of the tipper, the type of restaurant, the biases of whom is describing the tip, and a virtually infinite number of other factors.

5.5 Fuzzy controllers

The fuzzy sets described in Section 5.3 are essential to the operation of fuzzy controllers as imagined by Zadeh [22] and implemented by Mamdani [23] and others. Though essential, fuzzy sets

alone do not make a fuzzy controller. The controllers consist of four main elements: a fuzzifier, an inference engine, a rule base, and a defuzzifier. These processes are illustrated in Fig. 5.4 and described in the next few sections.

Throughout this section, the operation of fuzzy controllers will be described using an example of an imaginary stadium floodlight controller. In this application, it is assumed that the ambient light level is measured and is to be used to determine the proper power level for the stadium lighting. This is reflected in the specification of the input (fuzzification) and output (defuzzification) elements of the fuzzy controller.

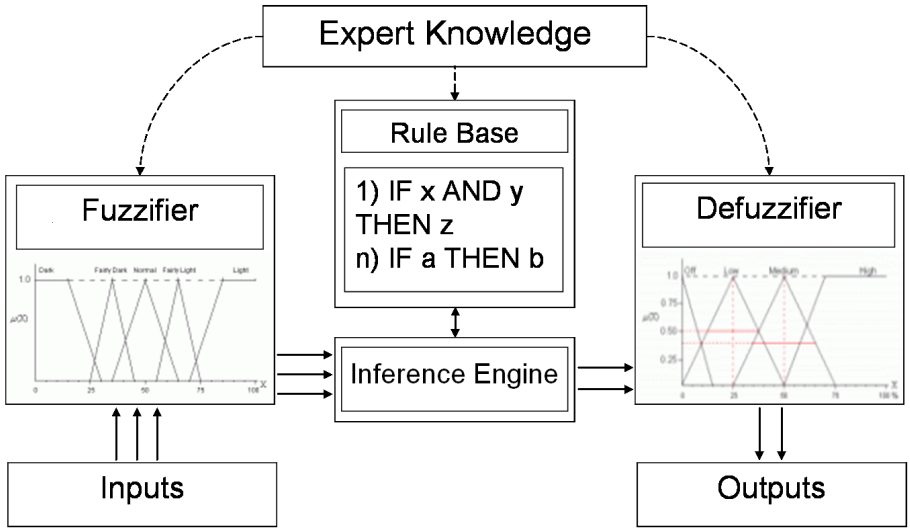

Figure 5.4: Overview of fuzzy controller

5.5.1 Fuzzification

Fuzzification is the process of translating crisp input values into fuzzy linguistic values through the use of membership functions. In other words, determining how much each discrete input value belongs to each input fuzzy set using the corresponding membership function. For example, assume that the input to the stadium floodlight controller is the measured ambient light level. The controller designer might then create the membership function consisting of five triangular membership functions as shown in Fig. 5.5. Each membership function is given a linguistic variable to express its relationship to the state of the input which it describes, such as "dark," "normal," or

"bright."

As seen in Fig. 5.5, a given ambient light level will have a corresponding membership value in at least one set. For instance, a 60% light level has a membership of 0.4 in the Normal set and 0.5 in the Fairly Bright set.

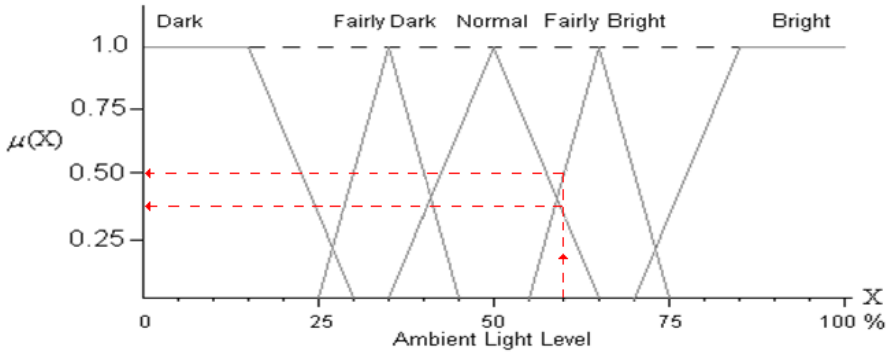

Figure 5.5: Membership function for ambient light level

5.5.2 Fuzzy rule base

The fuzzy rule base is simply a database of the desired control rules for the system. It is most equivalent to the controller of a traditional control system, and formalizes the designer's "expert knowledge" of what control output should result from a given combination of system states, expressed in a linguistic manner. It often takes the form of a truth table consisting of rules constructed in the following form:

- IF <condition1 AND/OR condition2> THEN <consequence>

Using the light level example, we can formulate a rule base such that:

- IF ambient light level is Bright THEN turn the lights Off
- IF ambient light level is Fairly Bright OR Normal THEN turn the lights to Low Power
- IF ambient light level is Normal OR Fairly Dark THEN turn the lights to Medium Power
- IF ambient light level is Dark THEN turn the lights to High Power

In fuzzy logic terminology, the statement following the IF condition is known as the "premise," "antecedent," or "condition." The corresponding statement following THEN is known as the "conclusion" or "consequent." The actual calculation of the consequent using the premises calculated

from the fuzzified inputs is reserved for the inference engine. This aspect of FLC is discussed in the following section.

5.5.3 Inference engine

The inference engine is the heart of a FLC. It acts as the bridge between the fuzzification input stage and defuzzification output stage of the controller, translating the designer's desired control rules from a linguistic representation to a numeric computation. The inference engine can be divided into three elements: aggregation, composition, and accumulation.

The first step of the inference process is known as "aggregation." During this step the premise (IF statement) of each rule in the rule base is calculated using the fuzzified controller inputs. During fuzzification, each condition in the premise is assigned a degree of membership in the corresponding input fuzzy set. In the aggregation process these conditions are aggregated according to the logical statement connecting them, such as AND/OR. It should be noted that the traditional formulation of logical statements such as AND/OR has been modified to accommodate the use of membership functions. In particular, the result of an AND operation is often defined as either the minimum (*min*) of the two fuzzy values compared, or the product (*prod*) of the two values. Similarly, an OR operation is often defined as either the maximum (*max*) of the two fuzzy values compared, or the probabilistic sum (*sum*). The probabilistic sum is defined as the sum of the two values compared, minus their product. Any premise with a value greater than zero means that its corresponding rule is active, or has "fired" in FL terminology.

The second step of the inference process is known as "composition" or "implication." In this step the consequent (THEN statement) of each rule is created using the premises calculated in the first step. The output of the composition step is not a single value for each rule in the rule base, but rather one modified output fuzzy set for each rule. These modified output sets are known as "implied" fuzzy sets. The modification is controlled by the premise calculated in the aggregation step. There are two fundamental methods of creating the fuzzy sets that are the result of composition. They are the same function described in the AND operation; *min* and *prod*. The *min* operation truncates the output fuzzy set at the value of the premise while the *prod* operation scales the output fuzzy set according to the premise.

The third and final step of the inference process is known as "accumulation," "composition," or "results aggregation." In this step the implied fuzzy sets that are the output of the composition process are combined into an accumulated fuzzy set, which is the input to the defuzzification

process. The sets are combined by calculating the union of the implied membership functions. It should be noted that there is an overlap in the terminology that different authors use to describe the elements of the inference engine. This can lead to difficulties when comparing texts.

As an example of the inference process, consider the ambient light level membership function introduced earlier. The 60% ambient light level has a membership of 0.4 in the Normal set, 0.5 in the Fairly Bright set, and 0 in all other sets. The aggregation process is applied to each rule in the rule base. In this example, consider the fuzzy OR operation to be the maximum of the two compared values. The premise for the "Low Power" set contains an OR connective, which means that the value of the premise for that particular rule is equal to the maximum of 0.4 (Normal set) and 0.5 (Fairly Bright set). Therefore, the premise for the Low Power set is 0.5. The premise for the "Medium Power" set also contain an OR connective. In this case, the two degrees of membership being compared are 0.4 (Normal set) and 0 (Fairly Dark set), as shown in Fig. 5.5. The maximum of 0.4 and 0 is 0.4, so this will be the value of the premise for the Medium Power set. This completes the aggregation process, since the premise of all other rules is zero.

The second element of the inference engine is the "composition" process. In this step, the output membership functions are modified by the AND operation. In the above example the *min* operator has been chosen for the AND operation, so that the output membership functions are truncated at the premise values calculated during the previous aggregation process. This can be seen in Fig. 5.6.

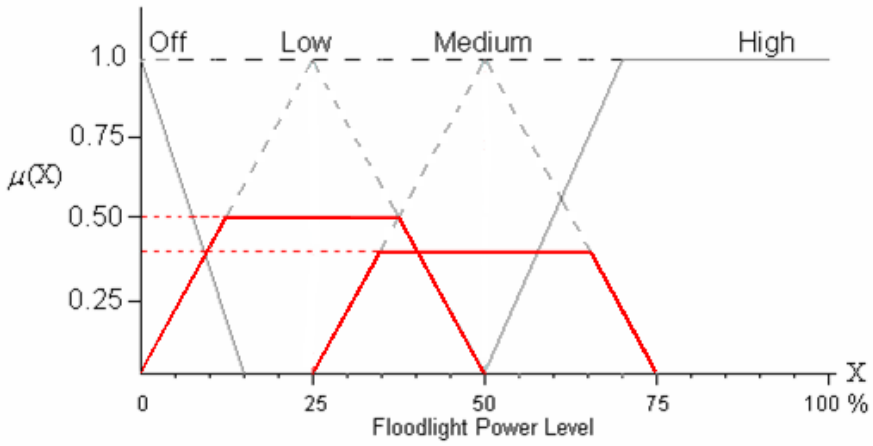

Figure 5.6: Membership function for floodlight power level, truncated at premise values

The third and final element of the inference engine is the "accumulation" process. In the literature, this step is sometimes considered a part of the defuzzification process. The accumulation process involves combining the implied membership functions that are the output of the composition process into a single output membership function that is suitable for defuzzification. This is achieved by calculating the union of the implied membership functions, as is shown in Fig. 5.7.

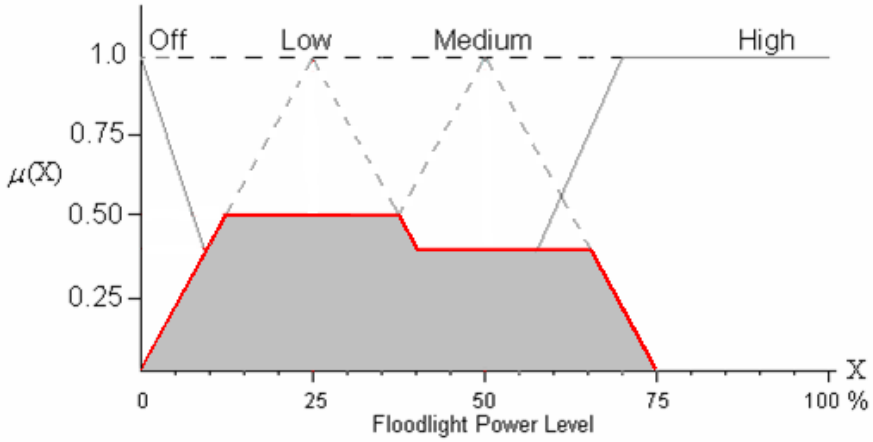

Figure 5.7: Results aggregation of implied membership functions

5.5.4 Defuzzification

In general, the system to be controlled using a FLC requires a crisp or discrete input, rather than a membership function such as is produced by the inference engine. Defuzzification is the process of converting the fuzzy output set which is a result of the inference process into a discrete number suitable for input to the plant.

There are many different methods of defuzzification described in the literature, with varying levels of complexity. Two fundamental methods are known as the Mean of Maxima (MoM) method and the Center of Gravity (CoG) method.

The MoM method can be interpreted as calculating the mean of the local maxima of the accumulated fuzzy set. The MoM method can be described as follows:

- Sum the positions x of the maxima of the fuzzy set.
- Divide the value from the first step by the number of maxima m.

$$u = \frac{\sum\{x|(\mu(x) = max(\mu(x)))\}}{m} \tag{5.3}$$

The CoG method, also known as the Center of Area (CoA) method, can be interpreted as determining a weighted average of the elements in the set. The CoG method involves two steps:

- Multiply the membership degree $\mu(x_i)$ of each element i by the singleton value x_i of the membership function.
- Sum the values obtained in the first step and divide the total by the sum of the output membership degrees.

$$u = \frac{\sum_i \mu(x_i) x_i}{\sum_i \mu(x_i)} \tag{5.4}$$

Continuing with the aforementioned example, using the Center of Gravity defuzzification method the centroid of the area can be calculated as shown in Fig. 5.7. The area is divided into two parts to facilitate the calculation; the trapezoid A_1 and the parallelogram A_2. The centroids of each of these areas are: $\bar{y}_1 = 0.2083$, $\bar{x}_1 = 25\%$, $\bar{y}_2 = 0.2$, $\bar{x}_2 = 57.5\%$. The areas of A_1 and A_2 are 25 and 10 respectively. The centroid of the combined area can be determined from the following formula:

$$\bar{x} = \frac{\bar{x}_1 A_1 + \bar{x}_2 A_2}{A_1 + A_2} \tag{5.5}$$

$$\bar{y} = \frac{\bar{y}_1 A_1 + \bar{y}_2 A_2}{A_1 + A_2} \tag{5.6}$$

Substituting the values into the equations, the centroid of the overall area is determined to be $\bar{x} = 34.3\%$ and $\bar{y} = 0.206\%$. This is shown in Fig. 5.8. Therefore the floodlight power level should be set to 34% when the ambient light level is 60%. Similar calculations would give the power level for any given ambient light level.

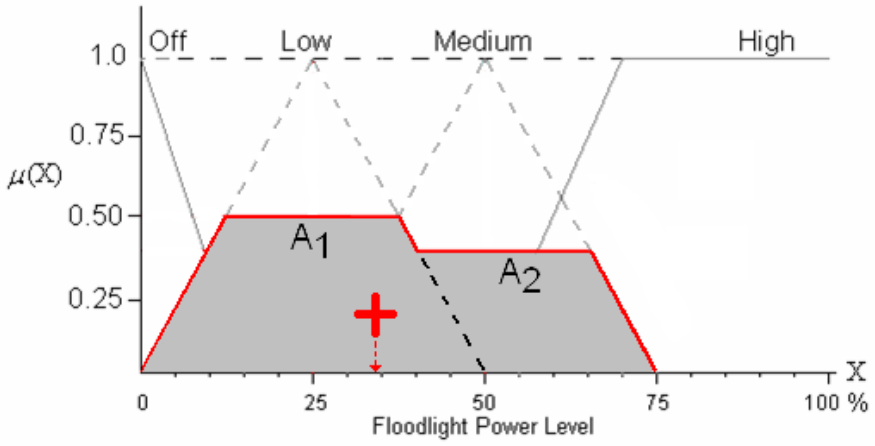

Figure 5.8: Center of Gravity defuzzification

5.6 Sugeno fuzzy inference

The fuzzy controllers described so far in this chapter have been of the Mamdani-inference type. However, there exists an alternate type of inference engine that is very well suited for interpolating between a number of linear controllers. This type is known as a Sugeno-inference controller, after its inventor [24].

In the EHA switched-gain controller, two discrete proportional gains were switched depending on the magnitude of the positional error. For the fuzzy controller, it was desired to smoothly alter the control gain between a number of discrete control gains depending on the magnitude of the positional error. Therefore, a Sugeno-inference controller was implemented.

The Mamdani-type and Sugeno-type fuzzy controllers are similar in many respects. The first two parts of the inference process are the same, the fuzzification of the input and the aggregation/composition process. However, for the Sugeno-inference controller, there is no accumulation process, and the output membership functions are functions of the input premises, typically either a linear function or a constant (fuzzy singleton). Therefore the defuzzification process may be computationally simpler than with a Mamdani-inference controller, where defuzzification often involves the calculation of centroids of areas with complex shapes as in Fig. 5.8.

In a Sugeno controller, the rule base has the form IF <$input_1 = x_1$ AND $input_1 = x_2$> THEN <Output $u = ax_1 + bx_2 + c$>. If the output membership functions are constant as employed in this study, then $a = b = 0$. The defuzzification process consists simply of a weighted average of all the

rule outputs, such that:

$$u = \frac{\sum_i c_i x_i}{\sum_i c_i} \tag{5.7}$$

The application of Sugeno-inference can be illustrated using the floodlight power level example previously developed. As mentioned above, the fuzzification, aggregation and composition processes are the same as with the Mamdani-inference. However, the output membership functions are replaced by fuzzy singletons instead of membership functions. These singletons are modified during the composition process to create "implied" fuzzy sets by being truncated at the premise values calculated during the aggregation process, similar to the earlier Mamdani-inference example. This is shown in Fig. 5.9.

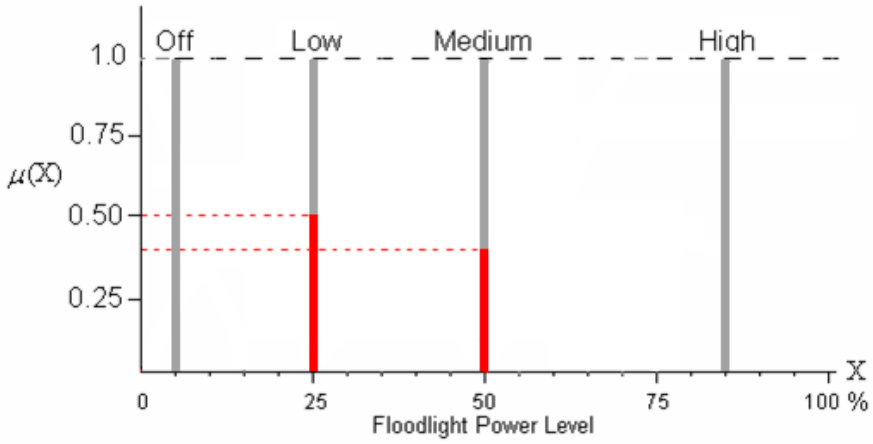

Figure 5.9: Sugeno inference example

In order to calculate the crisp output from the implied output membership functions, Eq. 5.7 is used. The resulting control output of 36% using the Sugeno-inference method is very close to the 34% obtained using the Mamdani-inference method.

$$u = \frac{\sum_i c_i x_i}{\sum_i c_i} = \frac{0.50 * 25\% + 0.40 * 50\%}{0.50 + 0.40} = 36.1\% \tag{5.8}$$

It was decided to use a Sugeno-inference controller for the EHA due to its simplicity and its suitability for interpolating between multiple linear controllers. The following section describes the design of the Sugeno-inference fuzzy controller as applied to the EHA system.

5.7 Design of the EHA fuzzy controller

The first step of fuzzy controller design is to select the number of controller inputs and corresponding number of input membership functions required for the fuzzification process. In the case of the EHA controller, it was believed that the positional error alone would be sufficient input to the fuzzy controller, as the initial concept for the controller was to vary the proportional control gain as a function of the positional error. Traditional fuzzy controller design suggests that 3-7 input membership functions per input are sufficient [25]. It was thus decided to employ four input membership functions.

It was known that the input to the controller would fall into three main categories: "large error," "medium error," and "small error." The "large error" category was deemed to be any error greater than one mm. The "small error" category was deemed to be any error less than roughly 150 μm. The "medium error" category would fall between the other two. Since there were four input membership functions and three categories of error, it was decided to assign one membership function to the "large error" category, one to the "medium error" category, and two to the "small error" category.

Given the desired specifications described above, four input membership functions were created. They were given the linguistic labels "large," "medium," "small," and "verysmall" according to the magnitude of positional error that they represented. These input membership functions are shown below in Fig. 5.10.

It was decided to use Gaussian membership functions rather than triangular or trapezoidal membership functions in order to achieve the desired smooth interpolation of the output proportional gain K_P. Triangular and trapezoidal membership functions contain discontinuities in their derivatives, which can result in abrupt changes in the output of the controller. This is undesirable as it would cause the actuator to exert unnecessary, large, transient forces as it tried to accelerate to match the abrupt changes in controller output.

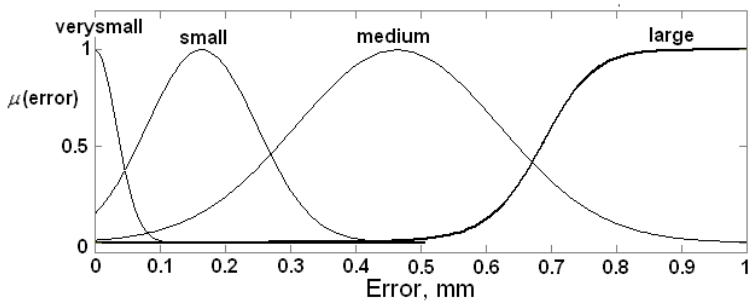

Figure 5.10: Input membership functions

The second step of the fuzzy controller design process was to define the output membership functions. It was decided to use four membership functions. It was known that the output, which is the controller proportional gain, should range from K_P = 585 when the positional error was large to a maximum of K_P = 1250 when the error was very small. This maximum gain was less than that employed with the switched-gain controller, since it was determined experimentally that higher gains increased the amount of overshoot and increased the settling time of the system. Since it was decided to use a Sugeno-inference fuzzy controller, the output membership functions are singletons. This means that each output membership function consists of a single value, the desired proportional gain K_P. This can be observed in Fig. 5.11. The four output membership functions are labelled "low," "average," "high," and "veryhigh," according to the proportional gain that they represent. The respective proportional gain was chosen to be K_P = 585, 750, 1000, and 1250. The magnitude of these gains was determined experimentally using step input tests to give the desired response and positional accuracy.

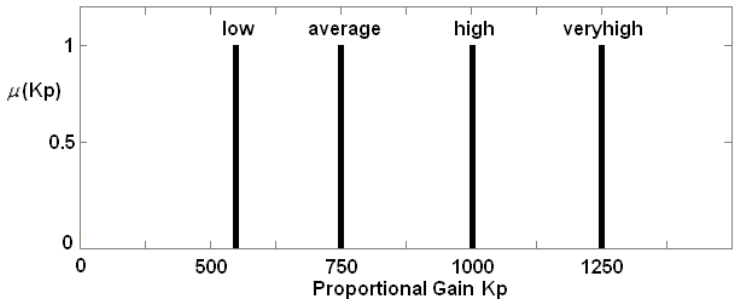

Figure 5.11: Output membership functions

The third step of the fuzzy controller design was the specification of the rule base, which links the fuzzifier and the defuzzifier. There are four rules, which correspond to the same number of input

and output membership functions. These rules are implied by the desired output of the controller. For example, the system performance is desired to be similar to the original proportional controller in the large-scale. Therefore, the first rule is "If error is large, then gain is low." This means that K_P will be approximately 585 when the positional error is greater than roughly one mm. Therefore, the system with the FLC will behave similar to the system with the proportional controller when the error is large, as is desired. On the other hand, it is desired that the proportional gain become higher as the positional error decreases in order to reduce the error as much as possible. Therefore the last rule is "If error is verysmall, then gain is veryhigh."

The following is the rule base that was used in the EHA fuzzy controller.

- IF error is large THEN gain is low
- IF error is medium THEN gain is average
- IF error is small THEN gain is high
- IF error is verysmall THEN gain is veryhigh

Once the input membership functions, the output membership functions, and the rule base have been designed, all that remained was to select the operations used in the inference process and the defuzzification method. It was decided to select the most widely-used operations for simplicity. Therefore, the *min* operator was chosen for the AND operation , and the *max* operator was chosen for the OR operation employed. The former is used to truncate the output fuzzy singletons at the value of the premise corresponding to each of the four rules. The latter was not used in this controller, as the rule base did not contain any OR conditions to be evaluated.

The final step of the controller design process was the selection of a defuzzification method. Since a Sugeno inference engine with singleton output membership functions was employed, the equation used to calculate the crisp output (proportional gain K_P) is as presented in Eq. 5.7.

5.8 Implementation of the EHA fuzzy controller

The following section describes the implementation of the Sugeno fuzzy controller. It was desired to maintain the large-scale response of the existing proportional controller, but increase the positional accuracy of the system in the nano-scale. Fig. 5.12 shows the Matlab real-time control environment used implement the various controllers, the results of which are presented in Chapter 6. The fuzzy controller was developed using the Matlab Fuzzy Logic Toolbox, which contains tools designed to aid in creating fuzzy controllers.

The control environment was configured so that one of the three controllers (proportional, switched-gain, and fuzzy) could be selected before beginning a given test. The input to each of the controllers is the positional error of the EHA, while the output of the controllers is the proportional gain K_P. This gain is multiplied by the positional error to obtain the control signal which is output to the EHA over the digital serial cable. It should be noted that the fuzzy controller includes a "motor-speed threshold" function. It was found during preliminary experimental studies that the operation of the system was improved by implementing a threshold when using the fuzzy controller, so that any demanded motor speed below one rpm was set to be exactly one rpm. Motor speeds lower than this resulted in actuator motion too slow to correct positional errors in an acceptable amount time.

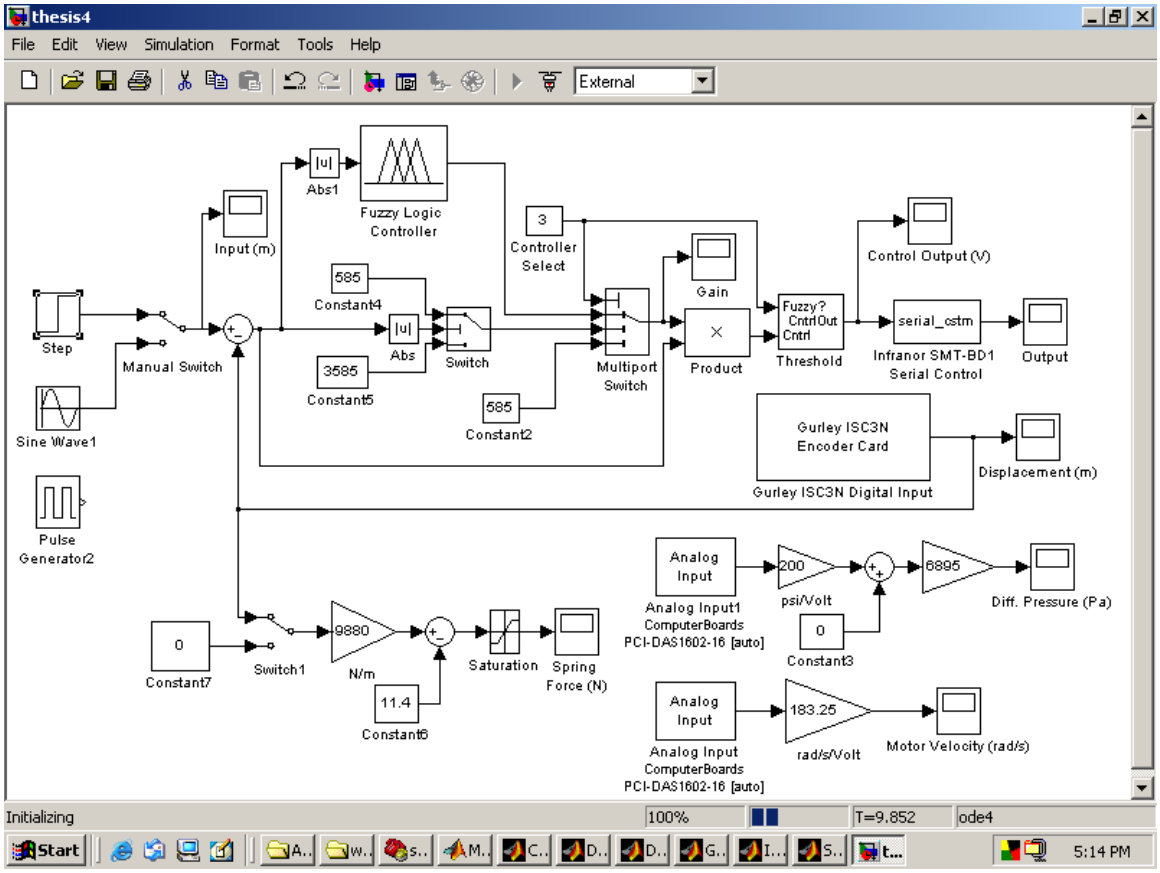

Figure 5.12: Matlab control environment

The Matlab Fuzzy Logic Toolbox contains two tools intended to aid in the visualization of the operation of the FLC. The first is the rule viewer. It allows the control designer to specify an arbitrary input to the controller, as illustrated by the vertical line overlaid on the input membership

functions on the left side of Fig. 5.13. The degree of activation of the input membership functions is illustrated by the highlighted area under each input function. The corresponding controller output is indicated by the short vertical line overlaid on the output membership functions on the right side of the figure. The degree of activation of the output membership functions is illustrated by the percentage of each output singleton that is highlighted. In the example, it is shown that for a positional error of 57.4μm, the output proportional gain will be K_P=1060.

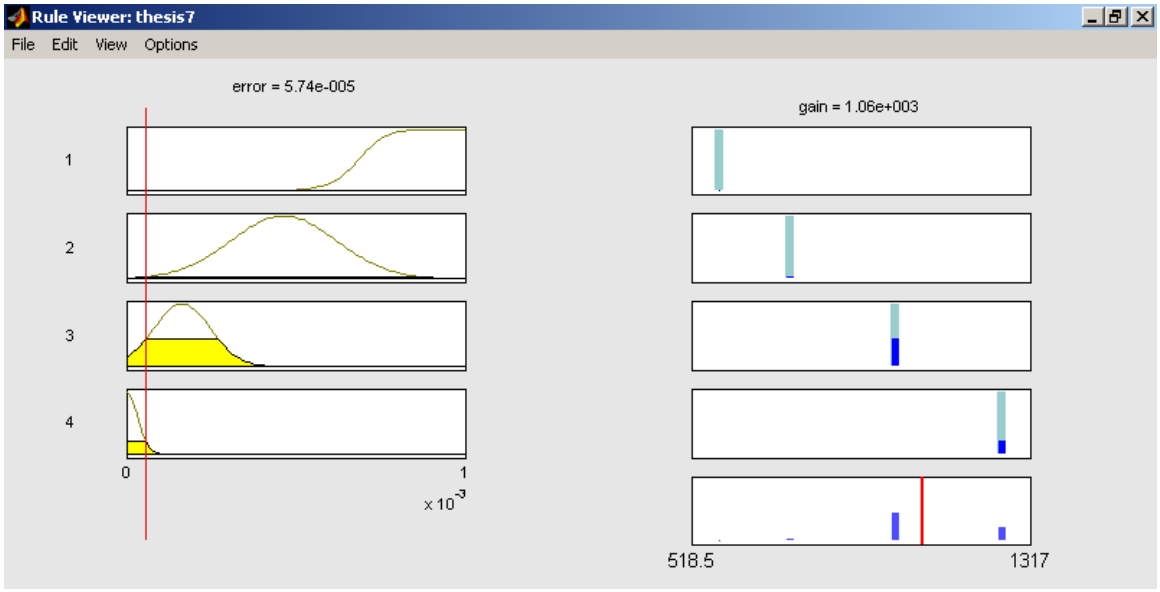

Figure 5.13: FLC rule viewer

The second tool provided to visualize the operation of the FLC is the surface viewer, which is shown in Fig. 5.14. This tool plots the output of the controller over the entire range of inputs. In the EHA FLC, it can be seen that the output of the controller ranges from K_P=585 to 1250 when the input error goes from $\geq$1 mm to 0 mm. It should be noted that the gain varies quite smoothly between the two extremes, as is desired.

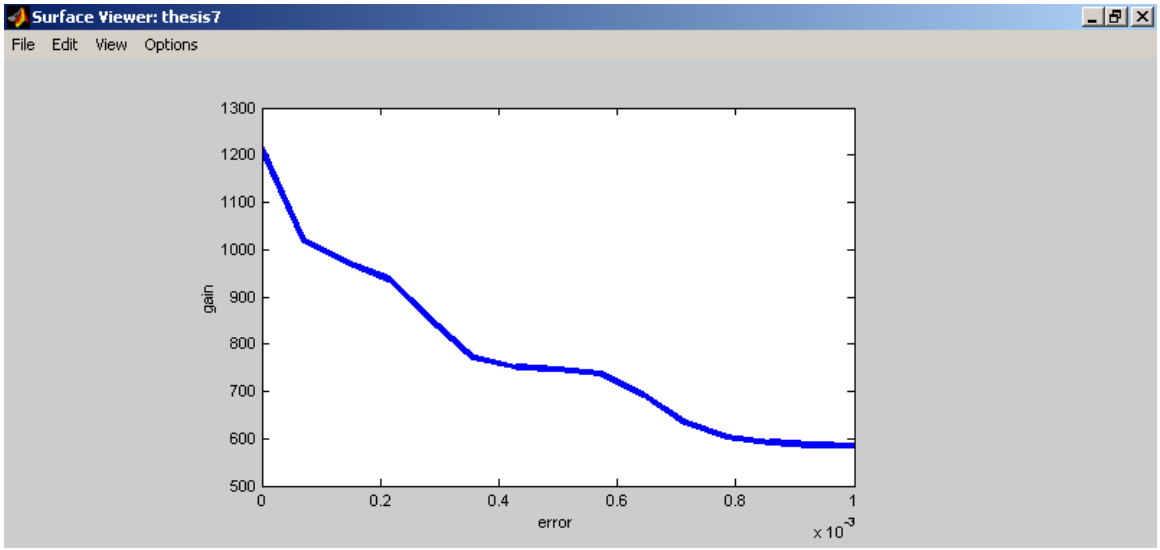

Figure 5.14: FLC surface viewer

The three different controllers (proportional, switched-gain, and fuzzy) presented in Chapters 4 and 5 will have a significant impact on the performance of the EHA. This impact is investigated quantitatively and qualitatively in Chapter 6.

CHAPTER 6

EXPERIMENTAL RESULTS

This chapter presents experimental results which were obtained from tests performed on the EHA using three different controllers. The proportional controller (PC) and switched-gain controller (SGC) were described in Chapter 4. The fuzzy logic controller (FLC) was described in Chapter 5.

In all cases presented in this Chapter, the communication between the MATLAB real-time control environment and the motor amplifier took place over a digital serial format, as discussed in Chapter 3. The advantage of digital communication compared to the previously-employed analog communication was the significant reduction in noise, which resulted in unwanted movement of the actuator. This is illustrated in Fig. 6.1, in which the displacement of the actuator under conditions of zero input to the motor amplifier and no positional feedback is shown. Noise in the analog line was interpreted by the motor amplifier as an input signal, resulting in an erratic movement of the actuator about 1 μm from the zero position (6.1, solid line). In contrast, when the digital line was used, movement could not be detected within the 50 nm resolution of the transducer (dashed line). Given that the desired positional accuracy of the EHA was 50 nm, digital communication was used for all the tests presented in this chapter.

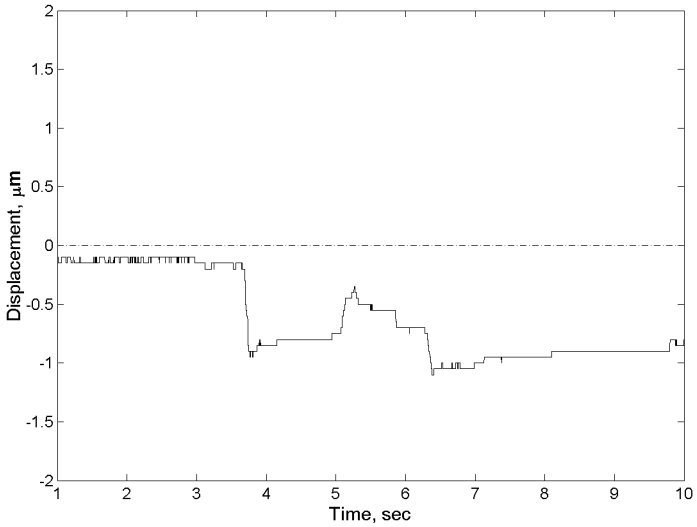

Figure 6.1: Open loop output with zero input, analog (solid line) versus digital (dashed line) amplifier communication

For all results presented in this chapter, the data is presented as the mean of the measured quantity plus/minus one standard deviation. The standard deviation is a measure which quantifies the variability of the measurement. It implies that there is a 68% likelihood that an individual measurement will fall within the range specified by one standard deviation.

6.1 Inertial load

The objective of the first series of tests performed on the EHA was to characterize its performance when employing the three different controllers (fuzzy, switched-gain, and proportional) with a dominantly-inertial load consisting of a 20 kg mass mounted on horizontal linear rails. However, it should be noted that these rails did contribute frictional forces to the system. The behaviour of the friction characteristics of this actuator were established in an earlier study by Chinniah [16]. The same inertial load was used in all experiments since the position sensor was mounted to the load, making it unfeasible to alter the load during the time frame of this study.

The first test with the inertial load consisted of a desired one cm step input, delayed by one second to ensure any initial transients had died down. This test was repeated 10 times with each controller at the same operating and initial conditions. Because of the quantity of tests performed, only a representative transient test is presented for each controller. A typical full-scale plot of the

EHA's response to the 1 cm step when using the FLC is shown in Fig. 6.2. The displacement of the actuator is shown by the solid line. The proportional gain as a function of time is shown by the dashed line.

The response of the EHA to this input appears identical regardless of the controller employed, for the scale used on the displacement axis in Fig. 6.2. Therefore, only one response at this scale is presented (it should be noted that as the resolution of the displacement scale increases, then significant differences can be observed among the three controllers). What is notable for this step response is that for all controllers, the settling time was approximately 0.7 seconds, and that the response approaches critical damping as was desired.

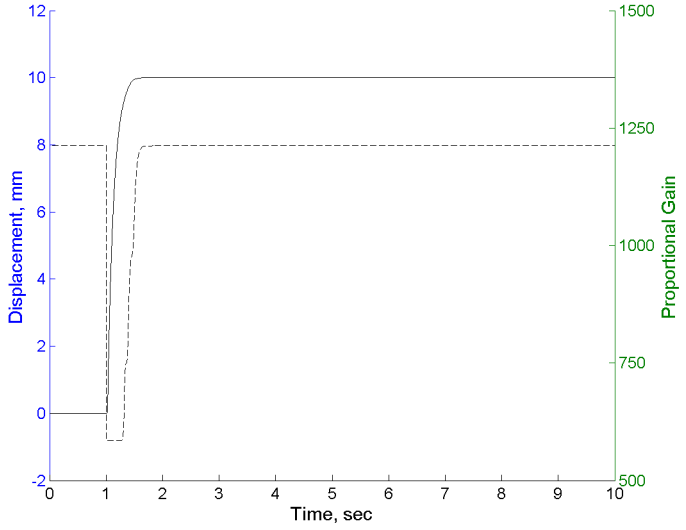

Figure 6.2: Response to 1 cm step input with FLC, solid line = displacement, dashed line = proportional gain

As mentioned, when the resolution of the displacement scale in Fig. 6.2 is enhanced, the difference between the various controllers in their positional accuracy as the system approached steady state becomes significant. In the following Figures, only these enhanced "micro scale" responses are shown.

Figures 6.3, 6.4, and 6.5 show the steady state response of the system when employing the proportional, switched-gain, and fuzzy controllers respectively. It can be observed that there is an overshoot of roughly 0.5 μm with the FLC versus roughly 2 μm with the SGC. The PC does not exhibit any overshoot, but has an increased level of steady-state error relative to the other

controllers.

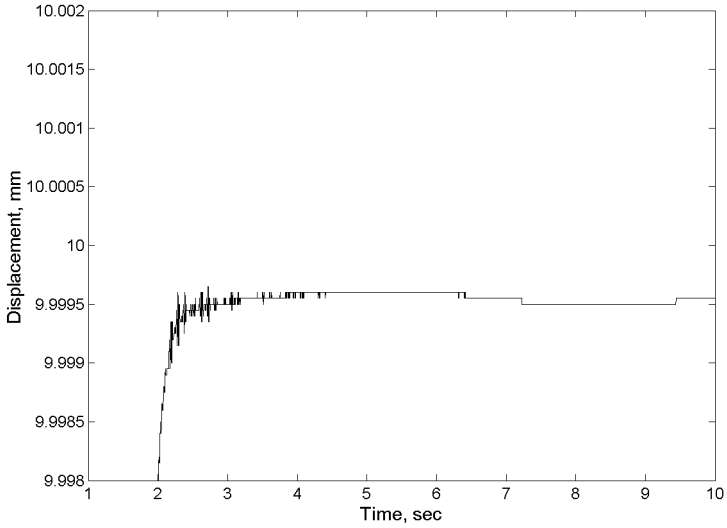

Figure 6.3: Steady state response with PC, 1 cm step input

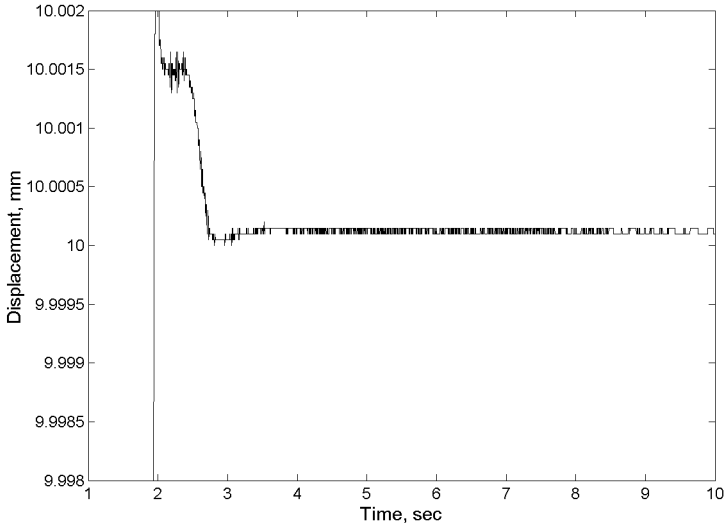

Figure 6.4: Steady state response with SGC, 1 cm step input

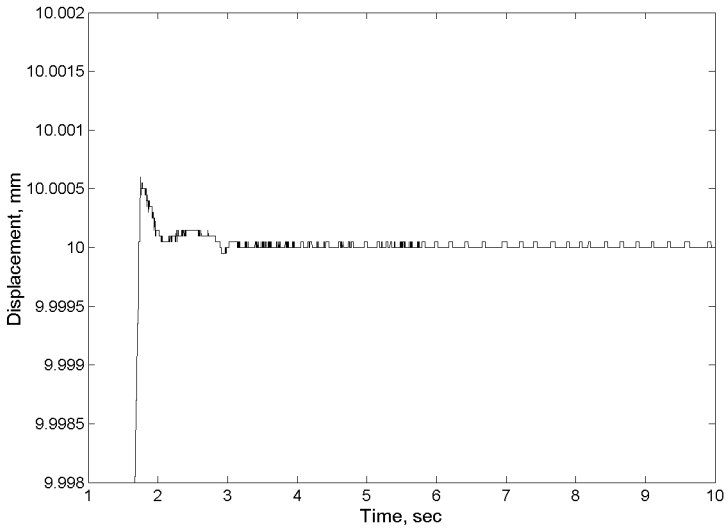

Figure 6.5: Steady state response with FLC, 1 cm step input

A summary of the performance of the three controllers when the EHA was given a 1 cm step input is tabulated in Table 6.1. Because the position sensor is digital, the calculated mean error is quantized at 50 nm steps to obtain the discrete mean error. To calculate the discrete mean error, any mean error that is less than 25 nm from the nearest increment of 50 nm is rounded up to that increment. The FLC performs significantly better than the SGC and the PC in all respects. It has a faster and more consistent settling time, which was defined as the time it took for the position measurement to reach $\pm$ 50 nm of its steady state value. The percent overshoot with the FLC is also lower and more consistent than with the SGC. Finally, the discrete mean steady state error with the FLC is 0 nm,$\pm$ the 25 nm discretization increment of the encoder. This is much better than the 150 nm steady state error with the SGC and 400 nm error with the PC.

Table 6.1: Comparison of controller performance, 1 cm step

Controller	Mean settle time seconds	Mean overshoot percent	Mean error nm	Discrete mean error nm
Fuzzy	3.08 ± 0.275	0.00879 ± 0.00206	20.8 ± 25.7	0 ± 25
Switched	3.26 ± 0.760	0.0176 ± 0.00652	133 ± 53.7	150 ± 50
Proportional	3.40 ± 0.969	-0.00265 ± 0.00315	400 ± 418	400 ± 400

The second test of the EHA with an inertial load was a 5 cm step response. Figures 6.6 to 6.8 show the steady state response of the system when employing the three controllers (again using the enhanced scale). It can be seen that the FLC results in smaller overshoot of the desired position than the SGC. Also, the poor steady-state positional error of the PC relative to the other controllers can be observed. Table 6.2 summarizes the performance parameters of the EHA with the three controllers. The superior response of the system with the FLC controller is again evident, with a steady state error of only 50 ± 25 nm. To put this in context, this is equal to 0.0001% error.

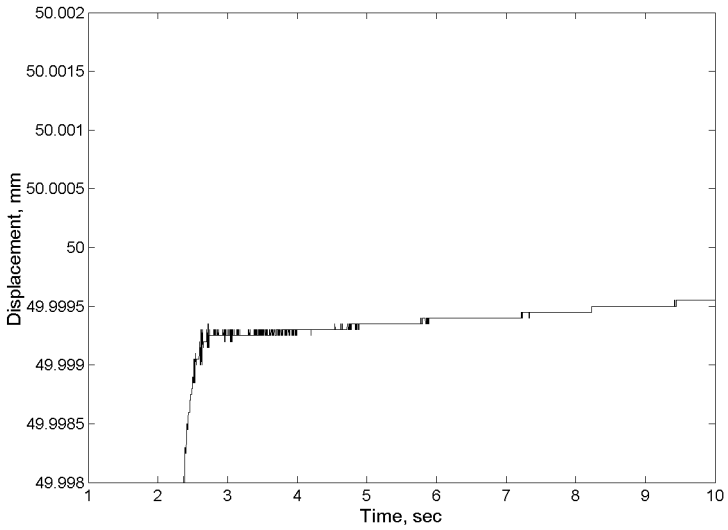

Figure 6.6: Steady state response with PC, 5 cm step input

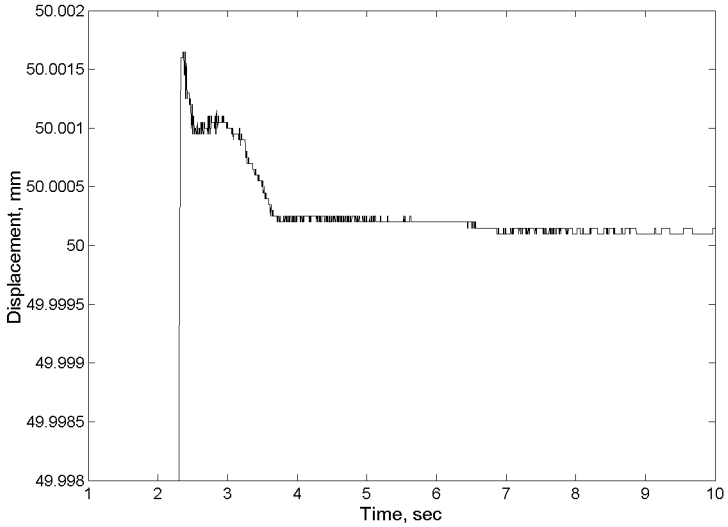

Figure 6.7: Steady state response with SGC, 5 cm step input

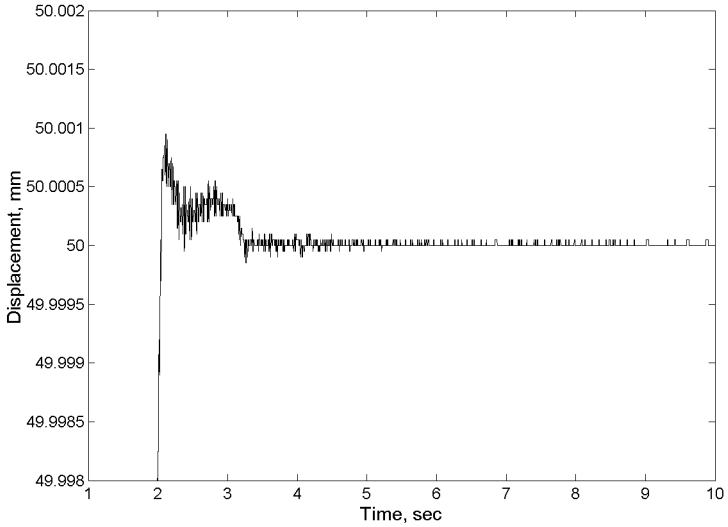

Figure 6.8: Steady state response with FLC, 5 cm step input

Table 6.2: Comparison of controller performance, 5 cm step

Controller	Mean settle time seconds	Mean overshoot percent	Mean error nm	Discrete mean error nm
Fuzzy	3.26 ± 0.128	0.00203 ± 0.000153	50 ± 0	50 ± 25
Switched	4.07 ± 0.456	0.00330 ± 0.0004	216 ± 57.7	200 ± 50
Proportional	3.14 ± 0.393	-0.0005 ± 0.000608	367 ± 284	350 ± 300

Figure 6.9 to 6.11 show the steady state response of the system to a 100 μm step input when employing the three controllers. The initial position of the actuator was 0 cm, as in the previous two tests. It is observed in Table6.3 that the settling time and percent overshoot of the FLC is superior to the SGC. As well, the steady state error with the FLC is below the resolution of the sensor, compared to the 100 nm and 1650 nm positional error with the SGC and PC respectively.

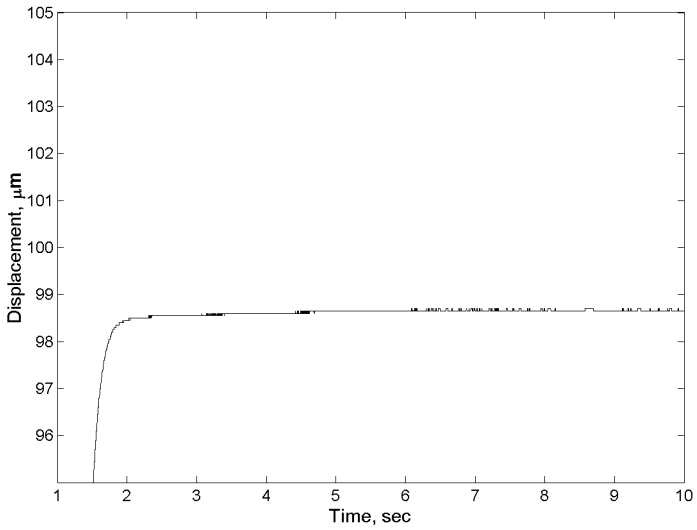

Figure 6.9: Steady state response with PC, 100 μm step input

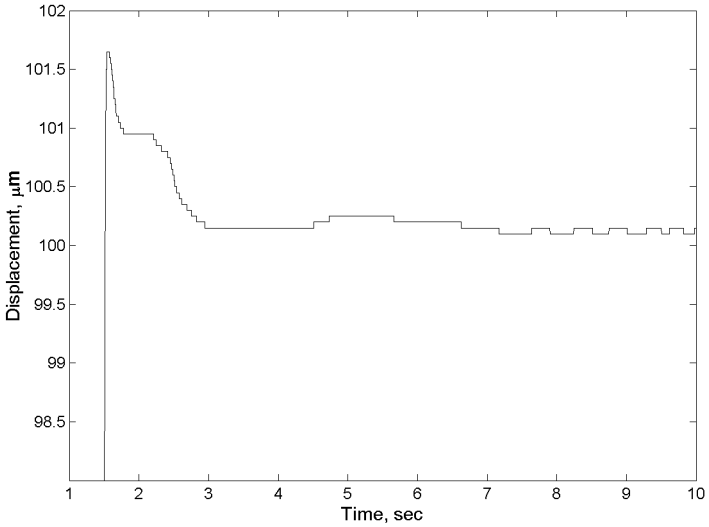

Figure 6.10: Steady state response with SGC, 100 μm step input

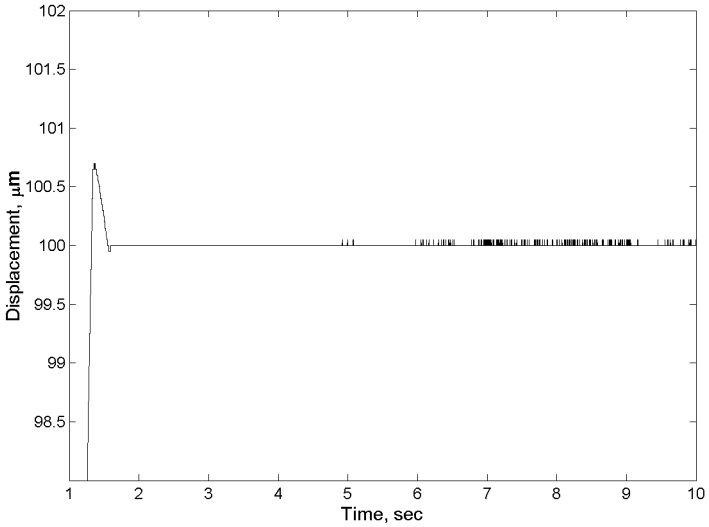

Figure 6.11: Steady state response with FLC, 100 μm step input

Table 6.3: Comparison of controller performance, 100 μm step

Controller	Mean settle time seconds	Mean overshoot percent	Mean error nm	Discrete mean error nm
Fuzzy	1.54 ± 0.0776	0.567 ± 0.126	0 ± 0	0 ± 25
Switched	2.68 ± 0.301	1.75 ± 0.361	83.3 ± 57.7	100 ± 50
Proportional	2.06 ± 0.169	-1.48 ± 0.318	1650 ± 436	1650 ± 450

Tests were also performed using the three controllers with step inputs of 10 μm, 1 μm, and 200 nm. Because the response plots for these tests are very similar to the plots presented in Figs.6.3 to 6.11, their presentation is deferred to Appendix C. Instead, a summary of the results of these tests are given in Tables 6.4 to 6.6 respectively.

It can be seen that the positional error with the FLC controller is 50 ± 25 nm for step inputs in the range of 5 cm to 200 nm with an inertial load of 20 kg. This demonstrates the very high level of accuracy that the EHA system is capable of when employing a high-performance controller. In contrast, for the same inputs the error with the SGC ranges from 200 ± 50 nm to 50 ± 25 nm while the error with the PC ranges from 1650 ± 400 nm to 200 ± 25 nm.

Table 6.4: Comparison of controller performance, 10 μm step

Controller	Mean settle time seconds	Mean overshoot percent	Mean error nm	Discrete mean error nm
Fuzzy	1.92 ± 0.0413	9.17 ± 0.289	0 ± 0	0 ± 25
Switched	3.93 ± 0.0802	13.5 ± 0.50	50 ± 0	50 ± 25
Proportional	1.96 ± 0.0355	-11.0 ± 2.0	1167 ± 161	1150 ± 150

Table 6.5: Comparison of controller performance, 1 μm step

Controller	Mean error nm	Discrete mean error nm
Fuzzy	50 ± 0	50 ± 25
Switched	217 ± 28.9	200 ± 50
Proportional	750 ± 0	750 ± 0

Table 6.6: Comparison of controller performance, 200 nm step

Controller	Mean error nm	Discrete mean error nm
Fuzzy	0 ± 0	0 ± 25
Switched	83.3 ± 28.9	100 ± 50
Proportional	200 ± 0	200 ± 25

An additional set of tests was conducted to determine the ability of the EHA to track time-varying input positions. To do so, a low frequency (0.2 Hz) sine wave with a 1 cm peak amplitude was set as the demanded position. This test replicates tasks that the EHA might be applied to, such as following a contour in a machining operation.

Because the full scale response of the EHA to the sinusoidal input was the same for all controllers, only a typical response for the FLC is presented. Figure 6.12 shows the full scale response of the EHA to the sine wave input with the FLC. The response is shown by the solid line, with the varying proportional gain indicated by the dashed line. The operation of the FLC can be seen in the smoothly-varying value of the proportional gain. As the system approaches its maximum displacement and the positional error decreases, the proportional gain increases from the initial value of 585 to the maximum value of nearly 1250. This reduces the steady state error when the actuator's velocity approaches zero at the peaks of the sine wave.

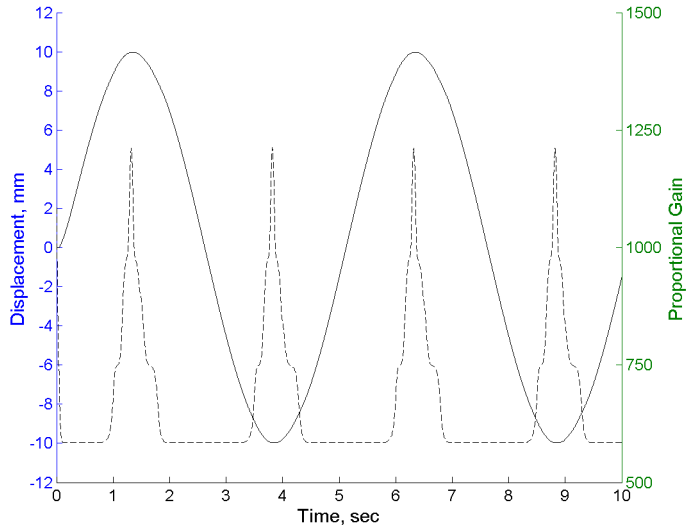

Figure 6.12: Response to 1 cm 0.2 Hz sine wave input with FLC„ solid line = displacement, dashed line = proportional gain

As was the case for the step inputs, as the scale of the position becomes enhanced, the differences between the three controllers becomes very evident. The steady state errors at the peaks of the sine wave with the three controllers are shown in Figs. 6.13 to 6.15.

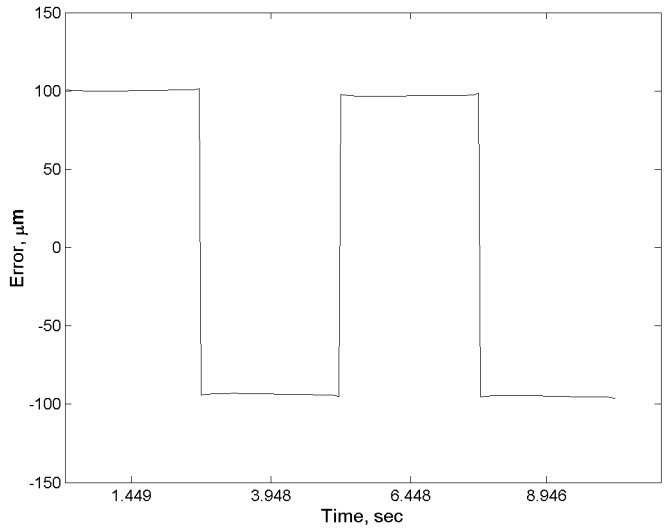

Figure 6.13: Steady state error with PC, 1 cm 0.2 Hz sine wave input

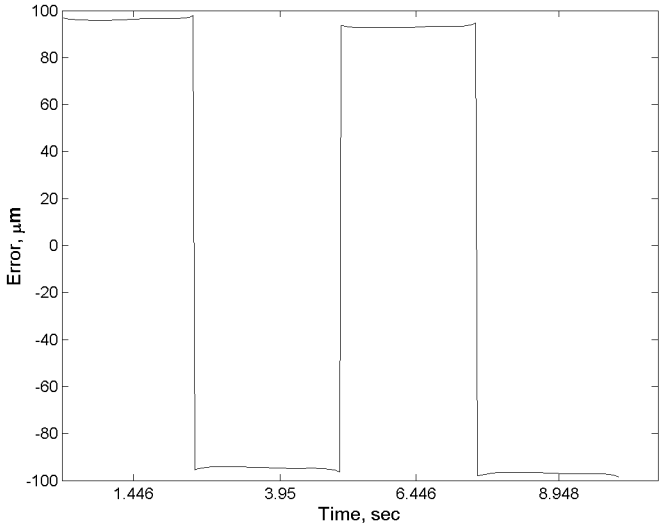

Figure 6.14: Steady state error with SGC, 1 cm 0.2 Hz sine wave input

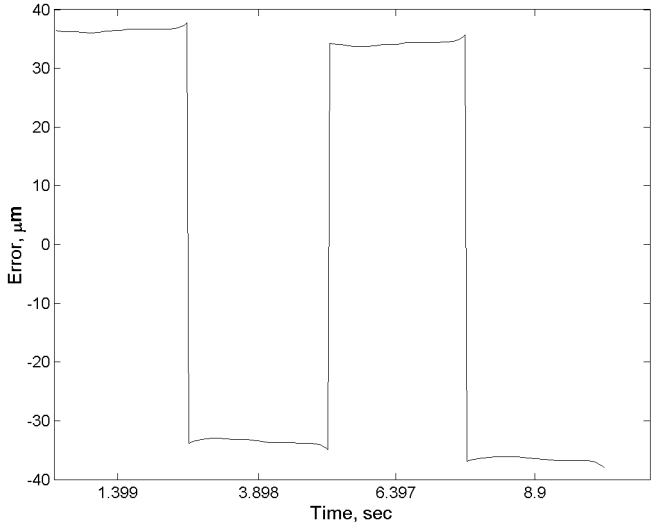

Figure 6.15: Steady state error with FLC, 1 cm 0.2 Hz sine wave input

Table 6.7 presents the mean steady state error and discretized mean error at the peaks of the 0.2 Hz 1 cm sine wave input with the three controllers. The FLC controller outperforms the SGC and the PC by nearly a factor of three, with a mean steady state error of 34.8 μm. This shows that the EHA is capable of following contour inputs within 50 μm steady state accuracy at this particular amplitude and frequency.

It is noted that the use of a time-varying signal such as the sine wave input described above reduced the steady state positional accuracy significantly compared to the non time-varying set point change represented by the step input tests. However, it should be kept in mind that in the step input tests, there exists an approximately two second time period between when the actuator reaches 95% of its steady state value and when it finally reaches steady state. Because the steady state period at the peaks of the sine wave is roughly half a second, the actuator never reaches its ultimate steady state accuracy when tracking the sine wave signal, leading to the reduced positional accuracy demonstrated in Table 6.7.

Table 6.7: Comparison of controller performance, 1 cm 0.2 Hz sine wave (errors evaluated at the sinusoidal peaks)

Controller	Mean error μm	Discrete mean error μm
Fuzzy	34.8 ± 0.486	34.8 ± 0.486
Switched	96.0 ± 0.808	96.0 ± 0.808
Proportional	97.0 ± 0.50	97.0 ± 0.50

A series of tests were performed with a sine wave input of 0.2 Hz with a 100 μm peak amplitude. The objective of this test was to investigate the ability of the EHA to track low amplitude time-varying signals. This was important because as the magnitude of a time-varying signal decreases, the required acceleration of the EHA decreases as well. This leads to the friction force becoming dominant rather than the force required to accelerate the inertial load. To illustrate the effect of friction on the response of the EHA, consider Fig. 6.16 in which the overall response of the system to the sine wave when using the FLC is presented.

It should be noted that the effect of static friction can be seen clearly in this figure. When the load is at a rest at the peaks of the sine wave, it takes more force to break it loose than to keep it in motion. This is because the static friction coefficient is generally higher than the dynamic friction coefficient. Therefore, the pressure in the actuator builds up to a level high enough to break the load free, causing it to accelerate faster than the hydraulic flow can keep up with. The actuator then slows for a small amount of time until the flow catches up and maintains the load at the desired velocity.

This situation can also occur with the step inputs presented earlier. However, the effect is exacerbated by the time-varying nature of the sine wave. In the case of the step input, when the actuator "unsticks," it overshoots the desired position so that the error signal becomes large enough to tend to drive the system back. For the sine wave, when the system "unsticks," it tends to stop again because the actuator arrives near the current position of the sine wave, which has changed in value since the actuator was stuck. This means that the error is relatively small, so that it takes some time for the error to become large enough to overcome the stiction.

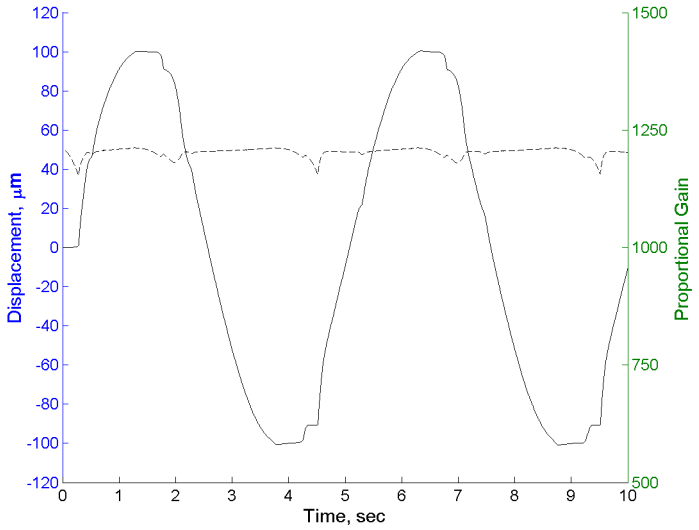

Figure 6.16: Response to 100 μm 0.2 Hz sine wave input with FLC, solid line = displacement, dashed line = proportional gain

Figures 6.17 to 6.19 show the steady state error at the peaks of the sine wave with the three controllers. It can be seen that the appearance of the error at the peaks varies significantly with the type of controller used.

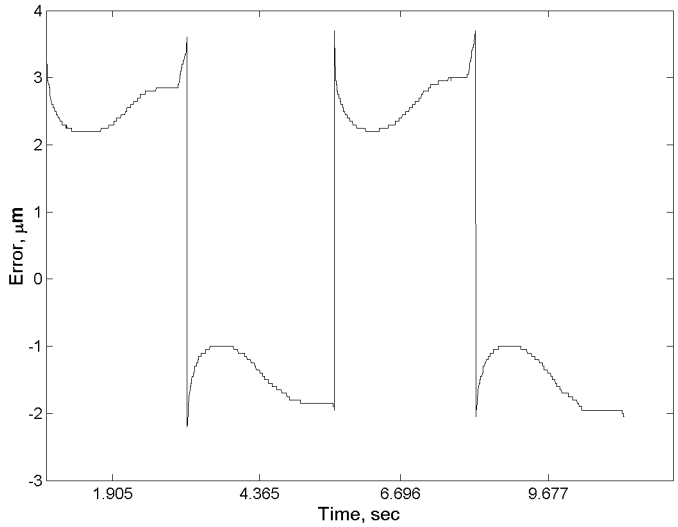

Figure 6.17: Steady state error with PC, 100 μm 0.2 Hz sine wave input

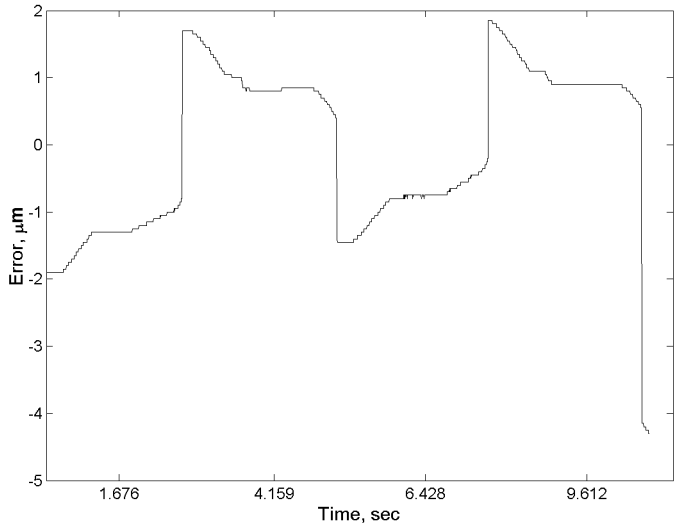

Figure 6.18: Steady state error with SGC, 100 μm 0.2 Hz sine wave input

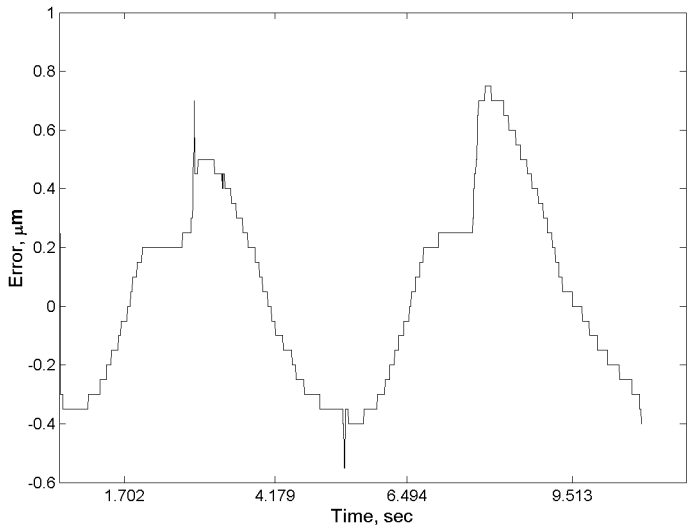

Figure 6.19: Steady state error with FLC, 100 μm 0.2 Hz sine wave input

Table 6.8 summarizes the mean steady state error at the peaks of the 100 μm amplitude sine wave for the various controllers. Again, the FLC controller performs significantly better than the other two, its 0.350 μm error being roughly three times better than that of the SGC and over six times better than that of the PC. The actuator dwells at the zero-velocity peaks of the sine wave for a longer period of time for the 100 μm sine wave than the 1 cm sine wave. This results in increased positional accuracy for the former compared to the latter, since the actuator has a greater amount of time to settle towards its ultimate accuracy.

Table 6.8: Comparison of controller performance, 100 μm 0.2 Hz sine wave

Controller	Mean error μm	Discrete mean error μm
Fuzzy	0.350 ± 0.050	0.350 ± 0.050
Switched	1.02 ± 0.161	1.0 ± 0.150
Proportional	2.30 ± 0.180	2.30 ± 0.20

In summary, tests comparing the performance of the EHA with a dominantly-inertial load using fuzzy, switched-gain and proportional controllers were presented. In all cases, the system

performed significantly better when using the fuzzy controller than the other two controllers. In fact, the positional error of the system was 50 ± 25 nm or less for step inputs ranging from 5 cm to 200 nm in magnitude. However, it was found that the EHA demonstrated significantly reduced steady state positional accuracy at the peaks of a time-varying sine wave input. It is believed that this is due to the fact that the actuator never reaches its ultimate steady state due to the sine wave reversing its direction before the actuator completely settles.

In the following section, the performance of the EHA under the influence of an external load acting on the actuator with the three controllers will be investigated.

6.2 Inertial and resistive external load

The tests performed on the EHA thus far have all used a dominantly-inertial load of 20 kg. In order to determine the performance of the EHA under an external resistive load, a compressive coil spring was anchored between the EHA's frame and one face of the load mass. The steel spring had a free length of 10 cm, a compressed length of 4 cm, a diameter of 4.3 cm, and a wire diameter of 0.47 mm.

It was desired to be able to calculate the force exerted by the spring for a given amount of compression, so that the displacement of the actuator could be used to determine the force exerted on the inertial load by the spring. In order to measure the force exerted by the spring throughout its compression, a load-deflection test was performed using an Instron ATM. The spring was compressed to a maximum compression of 5 cm, with the compressive force and spring compression being measured at 542 points. The result of the load-deflection test is shown in Fig. 6.20. It can be seen that the load-deflection curve is linear.

A linear regression line was fit to the data using the least-squares method. The coefficient of determination, which represents the proportion of variation accounted for by the regression line, was $R^2 = .9996$. This indicates that the fit of the regression line to the data was excellent, with one being the best possible fit. The equation of the regression line was $Force = 9.88X_d - 11.4$ with the displacement X_d in mm and the force exerted in N. This equation was used to calculate the force the spring exerts on the load given the displacement of the actuator.

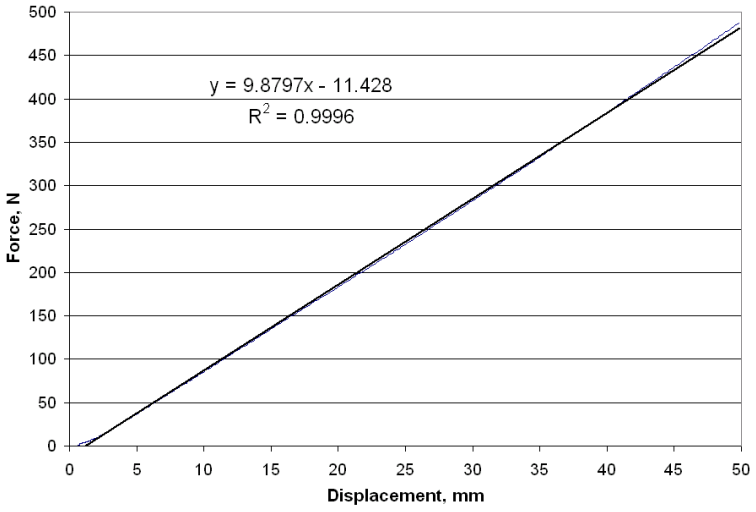

Figure 6.20: Load-deflection test of external load spring

The first test performed on the EHA with this resistive external load was a 1 cm step input. The compression of the coil spring produced a maximum load of approximately 90 N at 1 cm of displacement. Figure 6.21 shows the full scale response of the actuator to the 1 cm step input. As in earlier results the full scale response of the EHA to this input looks identical regardless of the controller used; hence only a representative step response with the FLC is shown. This figure demonstrates that the presence of an external resistive load does not alter the full scale response of the EHA to a step input appreciably.

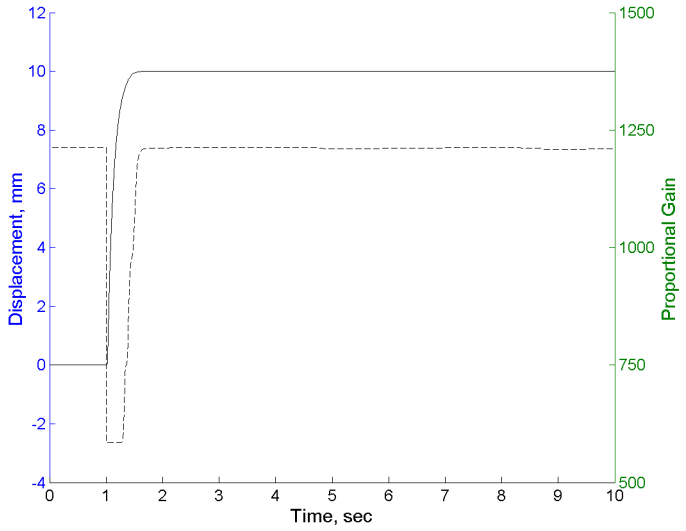

Figure 6.21: Response to 1cm step input under load with FLC, solid line = displacement, dashed line = proportional gain

The steady state response of the system to the 1 cm step input with the three controllers is shown at an enhanced scale in Figs. 6.22 to 6.24. It can be seen that there are significant differences between these figures and the ones presented in the previous section, when the actuator was not under the influence of an external load. First, the positional error has increased from 0 ± 0.025 μm to 3.1 ± 0.85 μm. Second, the system does not achieve a true steady state; there are "dips" in the position of the load of roughly 0.5 μm.

It is postulated that both these changes are caused by pump and actuator cross-port fluid leakage. In order to maintain the actuator in a fixed position under the influence of an external force, there must exist a force differential on the rod of the actuator. In turn, there must exist a pressure differential across the piston of the actuator to cause this force. This pressure differential drives fluid flow from one side of the actuator to the other, resulting in actuator cross-port fluid leakage. The leakage acts as a disturbance input to the system as discussed Sampson et al., reducing the positional accuracy [15] (for further details see Appendix D).

In order to maintain the actuator in a fixed position given cross-port leakage, the pump must provide enough flow to compensate for the leakage. Because the system uses a gear pump, each time a gear tooth opens to the pump output chamber to deliver a unit of fluid, the volume of the fluid decreases as it is pressurized to equalize with the pressure in the chamber. This reduction in

fluid volume causes the position of the actuator to slip back a few microns each time a gear tooth opens to the output chamber.

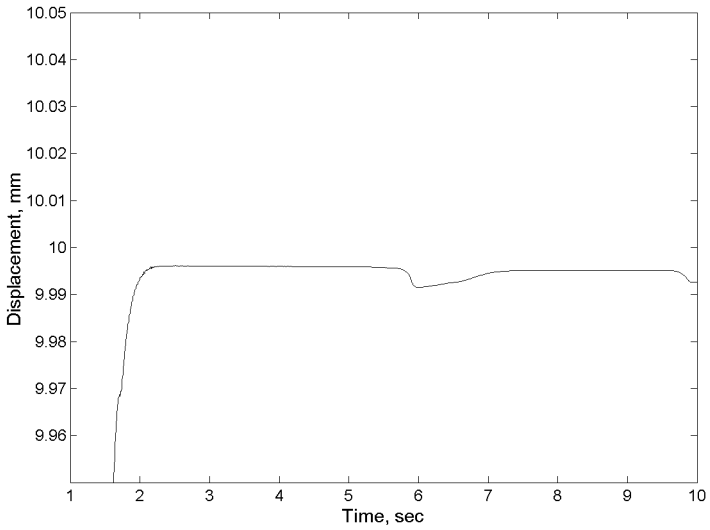

Figure 6.22: Steady state response under load with PC, 1cm step input

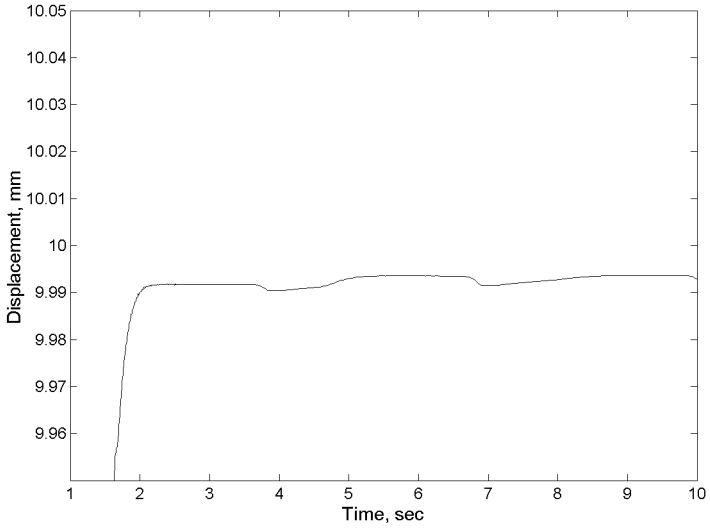

Figure 6.23: Steady state response under load with SGC, 1cm step input

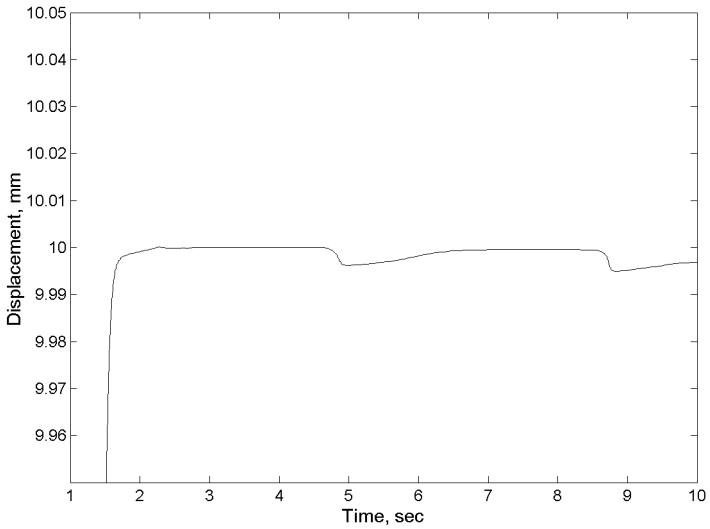

Figure 6.24: Steady state response under load with FLC, 1cm step input

Regardless of the increase in positional error as a result of the external load, the accuracy of the EHA during this test is within 5 μm. Table 6.9 presents the error of the EHA with the three controllers with a 1 cm step while under the influence of a 90 N (maximum) external resistive load. The FLC performs better than either other controller, achieving a mean 3.1 μm error with the 90 N (maximum) load.

Table 6.9: Comparison of controller performance under 90 N load, 1cm step

Controller	Mean error μm	Discrete mean error μm
Fuzzy	3.10 ± 0.835	3.10 ± 0.850
Switched	4.68 ± 3.54	4.70 ± 3.55
Proportional	5.48 ± 1.01	5.50 ± 1.00

A similar test was performed with the three controllers, except that the magnitude of the step input was increased to 3 cm. This resulted in a maximum external load of approximately 280 N (maximum) being applied to the actuator. Figures 6.25 to 6.27 show the steady state response of the system to the step input. It can be seen that both the steady state error and the frequency/magnitude

of the "dips" have increased. This is consistent with what would be expected due to the increased fluid leakage resulting from the larger 280 N external load.

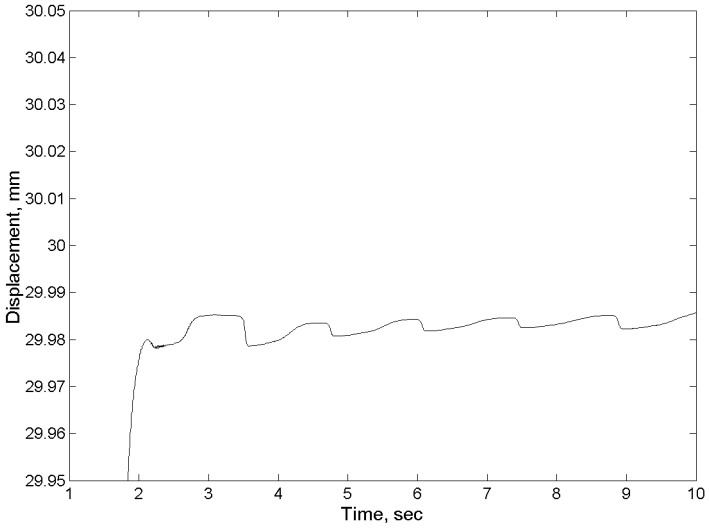

Figure 6.25: Steady state response under load with PC, 3cm step input

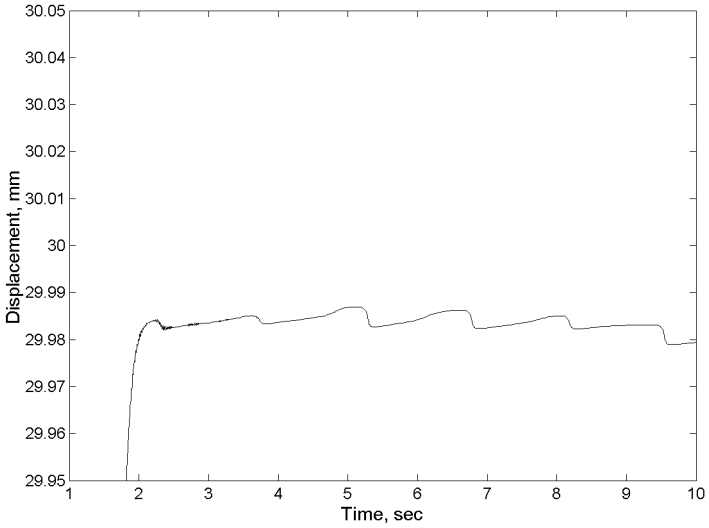

Figure 6.26: Steady state response under load with SGC, 3cm step input

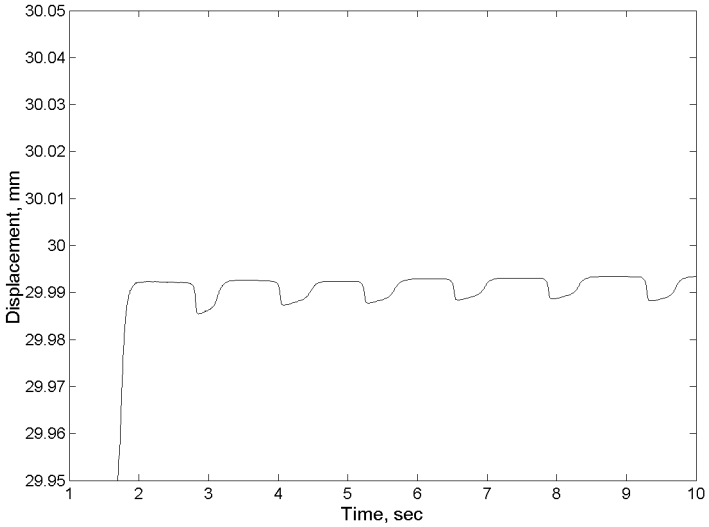

Figure 6.27: Steady state response under load with FLC, 3cm step input

Table 6.10 contains the mean steady state error of the EHA for a 3cm step input under a 280 N (maximum) external load. The positional error with the fuzzy controller is less than half the magnitude of the error with the other two controllers, at 8.47 $\pm$ 0.375 μm. However, this error is an increase of 2.9 times the positional error of 3.1 μm observed under a 90 N external load. This suggests that the positional error when then actuator is under the influence of an external load may be linearly related to the magnitude of the load.

Table 6.10: Comparison of controller performance under 280 N load, 3cm step

Controller	Mean error μm	Discrete mean error μm
Fuzzy	8.47 $\pm$ 0.375	8.45 $\pm$ 0.400
Switched	18.23 $\pm$ 1.83	18.25 $\pm$ 1.85
Proportional	18.1 $\pm$ 1.50	18.1 $\pm$ 1.50

Tests were also performed using the three controllers with step inputs of 100 μm and 10 μm under the influence of a 280 N external load resulting from a 3 cm positional bias. The plots of the responses from these tests are presented in Appendix C. A summary of the results of these tests are

given in Table 6.11 and 6.12 respectively.

It can be seen that the steady state positional error of 9.25 $\pm$ 1.15 μm for the 100 μm step input with the fuzzy controller is roughly half the error measured with the two other controllers. Similarly, for the 10 μm step input, the FLC had a 2.65 $\pm$ 0.750 μm steady state error compared to 5.60 $\pm$ 3.55 μm with the proportional controller.

Table 6.11: Comparison of controller performance under 280 N load, 100 μm step (3cm bias)

Controller	Mean error μm	Discrete mean error μm
Fuzzy	9.25 $\pm$ 1.14	9.25 $\pm$ 1.15
Switched	18.2 $\pm$ 1.73	18.2 $\pm$ 1.725
Proportional	18.1 $\pm$ 1.20	18.1 $\pm$ 1.20

Table 6.12: Comparison of controller performance under 280 N load, 10 μm step (3cm bias)

Controller	Mean relative error μm	Discrete mean error μm
Fuzzy	2.67 $\pm$ 0.725	2.65 $\pm$ 0.750
Switched	3.98 $\pm$ 3.09	4.00 $\pm$ 3.10
Proportional	5.58 $\pm$ 3.56	5.60 $\pm$ 3.55

The final two tests performed on the EHA system under the influence of an external load were tracking a low-frequency sine wave. In each case, the sine wave input was biased so that it was centered around 2cm of displacement. This corresponds to an external load of approximately 180 N.

The input of the first test was a 1 cm peak magnitude, 0.3 Hz sine wave biased by 2 cm. The full scale response of the EHA to this input with the FLC is shown in Fig. 6.28 (as before, the responses for all the controllers at full scale were non-distinguishable). The increased gain of the FLC as the positional error approaches zero at the peaks of the sine wave can be observed.

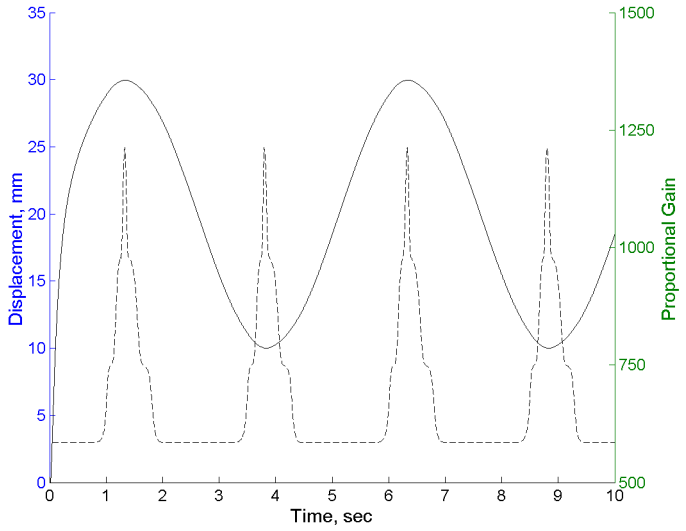

Figure 6.28: Response to 1cm 0.2 Hz sine wave input under load with FLC, 2cm bias, solid line = displacement, dashed line = proportional gain

The steady state positional error plots for the three controllers are very similar to those shown in the section without the external load and are presented in Appendix C for completeness. The steady state positional mean errors are tabulated in Table 6.13. It is interesting to note that the values of the errors presented in the table for the loaded system are very similar to the values presented in Table 6.7 for the unloaded system. This is likely due to the fact that with a sine wave input of this magnitude, the actuator does not dwell long enough at steady state for leakage due to the external load to have a large effect on the positional accuracy.

Table 6.13: Comparison of controller performance under 180 $\pm$90 N load, 1cm sine wave (2cm bias)

Controller	Mean error μm	Discrete mean error μm
Fuzzy	34.6 $\pm$ 1.39	34.6 $\pm$ 1.40
Switched	95.1 $\pm$ 5.06	95.1 $\pm$ 5.05
Proportional	94.1 $\pm$ 2.88	94.1 $\pm$ 2.90

The input of the second test was a 100 μm peak magnitude, 0.3 Hz sine wave biased by 2 cm.

The response of the EHA to this input with the FLC is shown in Fig. 6.29. The same "dips" that were visible in the loaded step input plots can also be observed in this figure.

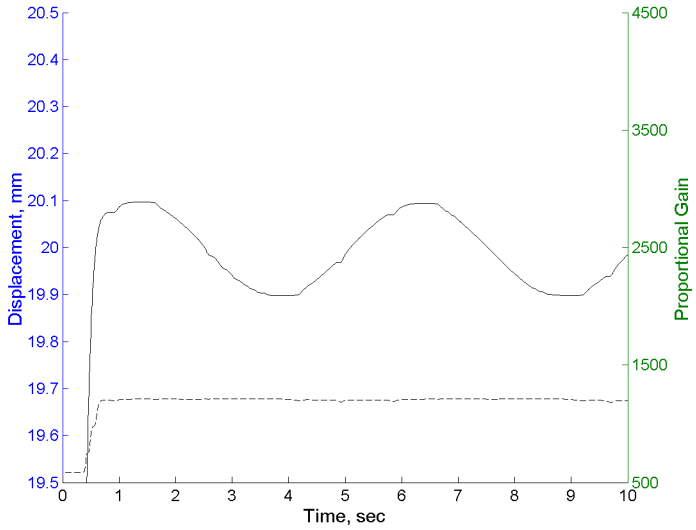

Figure 6.29: Response to 100um 0.2 Hz sine wave input under load with FLC, 2cm bias, solid line = displacement, dashed line = proportional gain

Table 6.14 presents the mean steady state positional error of the EHA in response to a 100 μm 0.2 Hz sine under an external load of 180 N. Again the FLC outperforms the other two controllers, giving a positional error of 19.1 μm compared to roughly 30 μm for the SGC and PC. This amount of error is over an order of magnitude worse than the error measured with a similar input but no external load in the previous section. This leads to the conclusion that the effect of cross-port leakage due to external loads has a significant impact on the positional accuracy of the EHA when the system remains at steady state for lengths of time greater than a few tenths of a second.

Table 6.14: Comparison of controller performance under 180 N load, 100um sine wave (2cm bias)

Controller	Mean error μm	Discrete mean error μm
Fuzzy	19.1 ± 2.83	19.1 ± 2.85
Switched	29.2 ± 13.7	29.2 ± 13.7
Proportional	30.1 ± 3.34	30.1 ± 3.35

6.3 Summary and discussion

The tests performed in this chapter suggest that the influence of an external load on the EHA system has a large impact on its steady state positional accuracy. This is a result of the pressure differential across the actuator caused by the external force leading to cross-port fluid leakage. This leakage acts as a disturbance input to the system, reducing its accuracy. The influence of an external load on the accuracy of the EHA could be reduced in several ways. If the external force is resistive (acting to retract the rod), then the third "bias" chamber of the actuator could be pressurized to give a force equal to the external force but opposite in direction. This would greatly reduce the pressure differential across the actuator working chambers, reducing the leakage that leads to reduce accuracy. This was not verified experimentally at this point because the force in the chamber would have to vary in an equal but opposite manner to that of the external spring. Although such as system could be built using a pressure control valve, it was considered beyond the scope of this initial study (see Recommendations).

Other alternatives include approaches such as reducing the leakage through physical modifications of the system including different seals, machining tolerances, or types of fluids. Finally, the use of a zero flow-ripple pump such as a screw pump would reduce the "dips" seen in the actuator's position which are a result of the discrete nature of flow delivery of the gear pump currently used in the EHA.

CHAPTER 7

CONCLUDING COMMENTS

7.1 Conclusions

As presented in Section 1.3, an objective of this study was the modification of the EHA so that the system achieves nano-scale positional accuracy while maintaining its desirable properties of high force output and large stroke. It was anticipated that both hardware modification and controller redesign would be required in order to achieve this objective.

This objective was met for one type of load, in that the final EHA configuration using the fuzzy logic controller described in Section 5.8 and a dominantly-inertial load of 20 kg achieved a positional accuracy of 50 $\pm$ 25 nm for a range of step inputs from 5 cm to 200 nm. In fact, the steady state positional error of the EHA for the majority of the step inputs in this range was less than the 50 nm resolution of the position sensor. This level of accuracy was achieved without altering the desired qualities of the EHA, such as a critically damped response with a 0.7 second rise time, high force output, and 12 cm stroke. In comparison, for the same inputs the error with the SGC ranged from 200 $\pm$ 50 nm to 50 $\pm$ 25 nm while the error with the PC ranged from 1650 $\pm$ 400 nm to 200 $\pm$ 25 nm.

Additional tests were performed to determine the ability of the EHA to track time-varying signals. These signals replicate those typical of some possible applications of the EHA, including the contouring operations used in machining. A sine wave input of 1 cm peak magnitude and 0.2 Hz frequency was selected to test the large-scale tracking response of the EHA. The steady state error at the peaks of the sine wave was found to be 34.8 $\pm$ 0.486 μm when employing the fuzzy controller. A second test was performed with a sine wave input of 100 μm peak magnitude and 0.2 Hz frequency, to test the micro-scale tracking ability of the EHA. The steady state error was found to be 0.350 $\pm$ 0.050 μm when using the fuzzy controller. These tests demonstrate that the EHA is able to track time-varying signals while maintaining accuracy in the μm scale, even though there is an increase in the positional error compared to input set-point changes such as step inputs.

In addition to the tests described above using a dominantly-inertial load of 20 kg, a series of tests with an external resistive load was performed. These tests were designed to investigate the change in performance of the EHA when exposed to external forces varying in magnitude from 90 to 280 N, in addition to the inertial load of 20 kg. The external load was applied using a coil spring that was compressed between the frame of the EHA and the actuator. A spring load represents a "worst case scenario," since the load changes with the displacement of the actuator. Nevertheless, it does serve to demonstrate the effects of pressure differentials (as a result of frictional or other forces) on leakage and hence accuracy. The test signals were selected to be as similar to the ones used in the inertial load tests as possible, to permit direct comparison of the effect of an external load on the positional accuracy of the EHA.

The first tests performed with the external loads involved step inputs of 1 cm and 3 cm with a steady state load of roughly 90 N and 280 N respectively. The steady state positional error in each case was 3.10 ± 0.835 μm and 8.45 ± 0.400 μm respectively when employing the fuzzy controller. The error when using the switched-gain and proportional controllers was two to three times larger than with the fuzzy controller. From these two tests, it appears that the steady state positional error varies linearly with the external load, as the error tripled when the force increased by the same amount. The second tests performed with the external loads were step inputs of 100 μm and 10 μm, both with an external load of 280 N. In all cases the positional error was significantly less with the fuzzy controller than the other two controllers, further demonstrating the superiority of the FLC with respect to high-accuracy positioning.

Finally, tracking tests under the influence of an external load were performed. In this case, the input was a sine wave with a bias of 2 cm, 0.2 Hz frequency and amplitudes of 1 cm and 100 μm peak. The 2 cm bias resulted in an external force of 180N being applied to the actuator. In the case of the 1 cm magnitude sine wave, the steady state positional error was 34.6 ± 1.40 μm when employing the fuzzy controller, roughly a third of the 95 μm error exhibited when using the switched-gain and proportional controllers. For the 100 μm magnitude sine wave, the error was 19.1 ± 2.85 μm with the fuzzy controller, significantly less than the error when employing the other two controllers.

The conclusions reached by the study are as follows. First, the EHA as it existed at the end of the study is able to position a dominantly-inertial load of 20 kg with a positional accuracy of 50 ± 25 nm over its 12 cm range of travel. This was due to the implementation of the fuzzy controller presented in this study. Second, the system is able to track time-varying input signals,

but with a reduced positional accuracy since the actuator did not have sufficient time to settle to its final accuracy due to the varying input signal. Third, the accuracy of the EHA is degraded by the application of a resistive external load, in this study by a factor of 50 with a 90 N (maximum) load and a factor of 150 with a 280 N (maximum) load. The final conclusion drawn was that performance of the EHA using the fuzzy controller was superior than the performance of the EHA when using the switched-gain or proportional controllers for all cases investigated during this study.

7.2 Future work

Although the EHA achieved excellent positional accuracy with a dominantly-inertial load of 20 kg, it was determined that the accuracy decreased significantly when an external load of 90 to 280 N was added. For example, the positional error with the fuzzy controller when performing a step displacement of 1 cm was 0 ± 25 nm with the inertial load compared to 3.10 ± 0.835 μm with the combined inertial load of 20 kg and resistive load of 90 N. While this level of accuracy is still excellent, it is desirable to modify the EHA further so that the accuracy of the system is as independent of the load as possible.

It is believed that the primary source of error when an external load is applied to the actuator is due to cross-port leakage in the actuator and the pump. This leakage occurs because the pump needs to create a pressure differential across the actuator to counteract the external force. The pressure differential drives leakage flow across the actuator and pump chambers. This leakage flow acts as a disturbance to the system, decreasing its positional accuracy. In addition, the gear pump that is currently employed in the system leads to "dips" in the steady state position of the actuator with an external load, as is seen in Section 6.2.

In order to decrease the positional error caused by leakage due to external loads, several areas could be investigated. First, better seals and machining tolerances in the construction of the actuator and pump could reduce the magnitude of the leakage. Second, the third chamber of the actuator could be pressurized to reduce the pressure differential across the actuator working chambers, which would reduce the actuator cross-port leakage. Third, the type of pump employed in the system could be changed to reduce the magnitude of the "dips" in the steady state response, which are caused by the discrete fluid chambers of the gear pump currently used in the system. Finally, alternate control algorithms could be devised to reduce the error caused by external loads, possibly using the signal measured by the differential pressure transducer to determine the magnitude of the

external force and take measures to counteract it.

REFERENCES

[1] G Bancroft. The Canadian Light Source - history and scientific prospects. *Canadian Journal of Chemistry*, 82(6), 2004.

[2] L. Dallin, I. Blomqvist, M. De Jong, D. Lowe, and M. Silzer. The Canadian Light Source. *Proceedings of the IEEE Particle Accelerator Conference*, 1, 2003.

[3] J. Meng, J. Franck, G. Gabor, R. Jared, R. Minor, and B. Schaefer. Position actuators for the primary mirror of the W.M. Keck telescope. *Proceedings of SPIE - The International Society for Optical Engineering*, 1236, 1990.

[4] K. Lorell, J. Aubrun, R. Clappier, B. Shelef, and G. Shelef. Design and preliminary test of a precision segment positioning actuator for the California Extremely Large Telescope primary mirror. *Proceedings of SPIE - The International Society for Optical Engineering*, 4840, 2002.

[5] A. Woronko, J. Huang, and Y. Altintas. Piezoelectric tool actuator for precision machining on conventional cnc turning centers. *Precision Engineering*, 27(4), 2003.

[6] X. Gang, D. Mei, and Z Chen. A micro-feed actuator for ultra-precision grinding of brittle materials. *Key Engineering Materials*, 258-259, 2004.

[7] A. Kanai, H. Sano, J. Yoshioka, and M. Miyashita. Positioning of a 200 kg carriage on plain bearing guideways to nanometer accuracy with a force-operated linear actuator. *Nanotechnology*, 2, 1991.

[8] K. Konishi, K. Hashimoto, T. Miyamoto, and T. Tamura. Hydraulic actuators driven by piezoelectric elements. *Transactions of the Japan Society of Mechanical Engineers, Part C*, 61(591), 1995.

[9] S. Futami, A. Furutani, and S. Yoshida. Nanometer positioning and its micro-dynamics. *Nanotechnology*, 1, 1990.

[10] K. Shintaku and A. Shimizu. Precise long displacement control in a teflon sealed linear actuator. *Nippon Kikai Gakkai Ronbunshu, C Hen*, 53(492), 1987.

[11] S. Habibi and A. Goldenberg. Design and analysis of a symmetrical linear actuator for hydraulic systems. *Transactions of the CSME*, 23:377–397, 1999.

[12] R. Lambeck. *Hydraulic pumps and motors*. Marcel Dekker, Inc., 270 Madison Ave., New York, New York, 1983.

[13] S. Habibi and A. Goldenberg. Design of a new high performance electrohydraulic actuator. *IEEE/ASME Transactions on Mechatronics*, 5(2), 2000.

[14] S. Habibi and G. Singh. Derivation of design requirements for optimization of a high performance hydrostatic actuation system. *SAE paper 2000-01-2619, 2000 SAE International Off-Highway and Powerplant Congress and Exposition*, 2000.

[15] E. Sampson, S. Habibi, R. Burton, and Y. Chinniah. Effect of controller in reducing steady-state error due to flow and force disturbances in the electrohydraulic actuator system. *International Journal of Fluid Power*, 5(2), August 2004.

[16] Y. Chinniah. *Fault Detection in the Electrohydraulic Actuator Using Extended Kalman Filter*. PhD thesis, University of Saskatchewan, May 2004.

[17] Y. Chinniah, R. Burton, S. Habibi, and E. Sampson. La maintenance prédictive d'une machine hydraulique en utilisant le filtre de Kalman (predictive maintenance of a hydrostatic system using the Kalman filter). *Proceedings of the Canadian Congress of Applied Mechanics*, 2005.

[18] Y. Chinniah, S. Habibi, R. Burton, and E. Sampson. Modeling and estimation of the nonlinear friction characteristics in a hydraulic actuator. *Proceedings of the Canadian Congress of Applied Mechanics*, 2005.

[19] S. Habibi, V. Pastrakuljic, and A. Goldenburg. Model identification and analysis of a high performance hydrostatic actuation system. *SAE paper 2000-01-2619, 2000 SAE International Off-Highway and Powerplant Congress*, 2000.

[20] E. Sampson, S. Habibi, R. Burton, and Y. Chinniah. Model identification of the electrohydraulic actuator for small signal inputs. *Proceedings of the Bath Workshop on Power Transmission and Motion Control*, September 2005.

[21] S. Habibi, R. Burton, and E. Sampson. High precision hydrostatic actuation systems for micro and nano manipulation of heavy loads. *ASME Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, 2005 (In press).

[22] L. Zadeh. Fuzzy sets. *Information and Control*, 8, 1965.

[23] E. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7(1), 1975.

[24] M. Sugeno and T. Takagi. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15, 1985.

[25] K. Passino and S. Yurkovich. *Fuzzy Control*. Addison-Wesley Longman, Inc., 2725 Sand Hill Road, Menlo Park, California, 1998.

APPENDIX A: COMPUTER CODE

This section contains the computer code written to enable the MATLAB real-time control environment to read the displacement signal measured by the optical encoder and send a desired motor velocity to the motor controller over a serial RS232 link. The code presented here was adapted from example code provided by MATLAB in default installations. However, the modifications were substantial and the bulk of the code presented here was written specifically for the purposes of this study. It should be noted that the code was tested using MATLAB versions 5.3 and 6.1 only, and that modifications to the code may be necessary when using other versions of MATLAB.

```
/* $Date: 06/11/2003 $
 * $Revision: 1.0 $
 * $Author: sampson $
 *
 * File: isc3n.c
 *
 * Abstract:
 *      S-Function device driver for the digital input section of
 *      the Gurley Precision Instrument ISC3N I/O board.
 *
 * Adapted from the Mathworks das16di driver, 1998/04/28 04:19:32
 * Copyright (c) 1994-1998 by The MathWorks, Inc. All Rights Reserved.
 */

#define S_FUNCTION_NAME isc3n
#define S_FUNCTION_LEVEL 2

#include  <stdlib.h>     /* malloc(), free(), strtoul() */
#include "simstruc.h"    /* the simstruct access macros */
#include "stdio.h"



/*====================================================================*
 * Number of S-function Parameters and macros to access from the SimStruct *
 *====================================================================*/

#define NUM_PARAMS                  (7)
#define BASE_ADDRESS_PARAM          (ssGetSFcnParam(S,0))
```

```
#define CHANNEL_PARAM    (ssGetSFcnParam(S,1))
#define SAMPLE_TIME_PARAM          (ssGetSFcnParam(S,2))
#define CONV_FACTOR_PARAM          (ssGetSFcnParam(S,3))
#define PRESET_PARAM               (ssGetSFcnParam(S,4))
#define MODE_PARAM                 (ssGetSFcnParam(S,5))
#define ACCESS_HW_PARAM            (ssGetSFcnParam(S,6))


/*================================================*
 * Macros to access the S-function parameter values *
 *================================================*/

#define CHANNEL                    ((uint_T) mxGetPr(CHANNEL_PARAM)[0])
#define SAMPLE_TIME                ((real_T) mxGetPr(SAMPLE_TIME_PARAM)[0])
#define PRESET_RAW                 ((float) mxGetPr(PRESET_PARAM)[0])
#define PRESET    PRESET_RAW*CONV_FACTOR
#define CONV_FACTOR                ((float) mxGetPr(CONV_FACTOR_PARAM)[0])
#define MODE                       ((uint_T) mxGetPr(MODE_PARAM)[0])

/*=========================================================*
 * (Hardware Specific) Macros pertaining to the I/O board *
 *=========================================================*/

#include "isc3n.h"

#ifndef ACCESS_HW
# define ACCESS_HW                 (mxGetPr(ACCESS_HW_PARAM)[0] != 0.0)
#endif


/*======================*
 * Miscellaneous macros *
 *======================*/

#define BASE_ADDR_PARAM_STRLEN     (128)


/*======================*
 * S-function User Data *
 *======================*/

typedef struct {
    uint_T    baseAddr;
} DIInfo;



/*====================*
```

```
 * S-function methods *
 *====================*/



/* Function: mdlInitializeSizes ===============================================
 *
 */
static void mdlInitializeSizes(SimStruct *S)
{
ssSetNumSFcnParams(S, NUM_PARAMS);
    if (ssGetNumSFcnParams(S) == ssGetSFcnParamsCount(S)) {

        if (ssGetErrorStatus(S) != NULL) {
            return;       //Error reported in mdlCheckParameters
        }
    } else {
        return; // Parameter mismatch will be reported by Simulink
    }

    /* None of this s-functions's parameters are tunable during simulation */
    {
        int_T i;
        for (i=0; i < NUM_PARAMS; i++) {
            ssSetSFcnParamNotTunable(S, i);
        }
    }

    ssSetNumSampleTimes( S, 1);
    ssSetNumInputPorts(  S, 0);
    ssSetNumOutputPorts( S, 1);
    ssSetOutputPortWidth(S, 0, 1);
}



/* Function: mdlInitializeSampleTimes =========================================
 *
 */
static void mdlInitializeSampleTimes(SimStruct *S)
{
    ssSetSampleTime(S, 0, SAMPLE_TIME);
    ssSetOffsetTime(S, 0, 0.0);
}

/*static void InitializeCard(uint_T CHANNEL)
{
diSelectChannel(CHANNEL);
```

```
}*/

#define MDL_START
#if defined(MDL_START)
/* Function: mdlStart =======================================================
 *
 */
static void mdlStart(SimStruct *S)
{



if (ACCESS_HW) {
        DIInfo *diInfo = ssGetUserData(S);
        char_T baseAddrStr[BASE_ADDR_PARAM_STRLEN];
        uint_T baseAddr;


        /* Initialize diInfo (pointer saved in the user data) */
        mxGetString(BASE_ADDRESS_PARAM, baseAddrStr, BASE_ADDR_PARAM_STRLEN);
        baseAddr = (uint_T) strtoul(baseAddrStr, NULL, 0);


        /* Selects the channel (connector) to which the encoder is attached*/
diSelectChannel(CHANNEL);

/* Sets the Mode of operation - 0=continuous count, 1=reference initialization*/
diSetMode(baseAddr,MODE);

/* Resets the Trigger Latch so that the Reference bit can be read*/
diResetLatch(baseAddr);

/* Resets onboard counter to preset value before reading values from the encoder*/
diResetCounter(baseAddr,PRESET);

/*Troubleshooting code - prints contents of mode of operation registers in hex
printf("Offset 0x03 is:%x\n",ReadByteFromHwPort(baseAddr + 0x03));
printf("Offset 0x04 is:%x\n",ReadByteFromHwPort(baseAddr + 0x04));
*/

if (diInfo != NULL) {
            free(diInfo);
        }
        if ((diInfo = malloc(sizeof(DIInfo))) == NULL) {
            ssSetErrorStatus(S,"Memory Allocation Error\n");
            return;
        }
        diInfo->baseAddr   = baseAddr;
```

```
        ssSetUserData(S, (void*) diInfo);

        printf("\nisc3n: Hardware Access Enabled\n");
    } else if (ssGetSimMode(S) == SS_SIMMODE_NORMAL) {
        printf("\nisc3n: Hardware Access Disabled\n");
    }
}
#endif /* MDL_START */



/* Function: mdlOutputs =======================================================
 *
 */
static void mdlOutputs(SimStruct *S, int_T tid)
{
    real_T *y = ssGetOutputPortRealSignal(S,0);
uint_T  i=0;


if (ACCESS_HW) {

DIInfo *diInfo  = ssGetUserData(S);
    uint_T baseAddr = diInfo->baseAddr;


/* Raw output of encoder is microns - this converts output to desired units
 * Note that the conversion factor must be terminated with .0 */
    y[i] = (diGetValue(baseAddr))/CONV_FACTOR;

}
else {
        y[i] = 0.0;
    }
}



/* Function: mdlTerminate =====================================================
 *
 */
static void mdlTerminate(SimStruct *S)
{
    DIInfo *diInfo = ssGetUserData(S);

    free(diInfo);
    ssSetUserData(S,NULL);
}
```

```
#ifdef  MATLAB_MEX_FILE    /* Is this file being compiled as a MEX-file? */
#include "simulink.c"      /* MEX-file interface mechanism */
#else
#include "cg_sfun.h"       /* Code generation registration function */
#endif

/* EOF: isc3n.c */

/* $Date: 06/11/2003 $
 * $Revision: 1.0 $
 * $Author: sampson $
 *
 * File: isc3n.h
 *
 * Abstract:
 *      S-Function device driver for the digital input section of
 *      the Gurley Precision Instrument ISC3N I/O board.
 *
 * Adapted from the Mathworks das16di driver, 1998/04/28 04:19:32
 * Copyright (c) 1994-1998 by The MathWorks, Inc. All Rights Reserved.
 */

#ifndef __ISC3N2__
#define __ISC3N2__

#include "tmwtypes.h"
#include "drt_comp.h"


int Port1,Port2;



/* $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
 * $Accessing the Digital Input section of the I/O board$
 * $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
 */




/* Selects the Channel that the encoder is connected to*/
void diSelectChannel (uint_T Channel)
```

```
{
if (Channel == 0)
{
printf("\nChannel=0\n");
Port1=0x59;
Port2=0x58;
}

else if (Channel == 1)
{
printf("\nChannel=1\n");
Port1=0x5B;
Port2=0x5A;
}

else if (Channel == 2)
{
printf("\nChannel=2\n");
Port1=0x5D;
Port2=0x5C;
}
}

/* Initializes offset 0x02 of the register with the relevent port data to select
 * the user-defined channel that the encoder is connected to*/
#define diInitialize1(bA,Port1)   WriteByteToHwPort(bA + 0x02, Port1)
#define diInitialize2(bA,Port2)   WriteByteToHwPort(bA + 0x02, Port2)

#define diResetLatch(bA) WriteByteToHwPort(bA + 0x05, 0x00)

//#define diSetMode(bA,Mode)   WriteByteToHwPort(bA + 0x04, Mode)

/* Splits the 32-bit preset into one of 4 bytes, i determines which byte is returned*/
#define ByteSplitter(i,v)  ((v >> ((i-1)*8)) & 0xff)

/* Presets the bytes in the counter to the value supplied byt the user*/
#define diResetMSB1(bA,Byte1)     WriteByteToHwPort(bA + 0x01, Byte1)
#define diResetLSB1(bA,Byte2)     WriteByteToHwPort(bA, Byte2)
#define diResetMSB2(bA,Byte3)     WriteByteToHwPort(bA + 0x01, Byte3)
#define diResetLSB2(bA,Byte4)     WriteByteToHwPort(bA, Byte4)

/* Reads the 4 bytes that make up the 32-bit counter*/
#define diReadMSB1(bA)      ReadByteFromHwPort(bA + 0x01)
#define diReadLSB1(bA)      ReadByteFromHwPort(bA)
#define diReadMSB2(bA)      ReadByteFromHwPort(bA + 0x01)
#define diReadLSB2(bA) ReadByteFromHwPort(bA)

/* Combines two bytes to make the Most Significant Word or MSW*/
```

```
#define diReadMSW(bA)     ((diReadLSB1(bA)) | (diReadMSB1(bA)<<8))

/* Combines two bytes to make the Least Significant Word or LSW*/
#define diReadLSW(bA)     ((diReadLSB2(bA)) | (diReadMSB2(bA)<<8))

/* Combines the MSW and the LSW to get the 32-bit final counter value*/
#define diGetValue(bA)      ((diGetLSW(bA)) | (diGetMSW(bA)<<16))

void diSetMode(uint_T bA, uint_T mode)
{
if (mode==0) {
WriteByteToHwPort(bA + 0x03, 0x00);
WriteByteToHwPort(bA + 0x04, 0x00);
}
else if (mode==1) {
WriteByteToHwPort(bA + 0x03, 0x10);
WriteByteToHwPort(bA + 0x04, 0x2A);
}

}

/* Calls the reset macros in the correct order to set counter to user preset*/
void diResetCounter(uint_T bA, int Preset)
{
char Byte1=0,Byte2=0,Byte3=0,Byte4=0;

/* Splits the 32-bit preset into 4 bytes*/
Byte1=ByteSplitter(4,Preset);
Byte2=ByteSplitter(3,Preset);
Byte3=ByteSplitter(2,Preset);
Byte4=ByteSplitter(1,Preset);


diInitialize1(bA,Port1);
diResetMSB1(bA,Byte1);
diResetLSB1(bA,Byte2);

diInitialize2(bA,Port2);
diResetMSB2(bA,Byte3);
diResetLSB2(bA,Byte4);
}


/* Initializes the register and reads the MSW*/
int32_T diGetMSW(uint_T bA)
{
    diInitialize1(bA,Port1);
return diReadMSW(bA);
```

```
}


/* Initializes the register and reads the LSW*/
int32_T diGetLSW(uint_T bA)
{
diInitialize2(bA,Port2);
    return diReadLSW(bA);
}



#endif /* __ISC3N__ */

/* EOF: isc3n.h */

/* $Date: 01/11/2005 $
 * $Revision: 1.0 $
 * $Author: sampson $
 *
 * File: serial_cstm.c
 *
 * Abstract:
 *      S-Function device driver for serial communication with the
 *      Infranor SMT-BD1 servo amplifier.
 *
 * Adapted from the Mathworks das16di driver, 1998/04/28 04:19:32
 * Copyright (c) 1994-1998 by The MathWorks, Inc. All Rights Reserved.
 */

#define S_FUNCTION_NAME serial_cstm
#define S_FUNCTION_LEVEL 2

#include  <stdlib.h>     /* malloc(), free(), strtoul() */
#include "simstruc.h"    /* the simstruct access macros */
#include "stdio.h"
#include <dos.h>
#include <stdio.h>
#include <conio.h>




/*=====================================================================*
 * Number of S-function Parameters and macros to access from the SimStruct *
 *=====================================================================*/

#define NUM_PARAMS                (4)
```

```
#define BASE_ADDRESS_PARAM         (ssGetSFcnParam(S,0))
#define BAUD_RATE_PARAM        (ssGetSFcnParam(S,1))
#define SAMPLE_TIME_PARAM          (ssGetSFcnParam(S,2))
#define ACCESS_HW_PARAM            (ssGetSFcnParam(S,3))


/*=================================================*
 * Macros to access the S-function parameter values *
 *=================================================*/

#define BAUD_RATE                  ((uint_T) mxGetPr(CHANNEL_PARAM)[0])
#define SAMPLE_TIME                ((real_T) mxGetPr(SAMPLE_TIME_PARAM)[0])

/*=========================================================*
 * (Hardware Specific) Macros pertaining to the I/O board *
 *=========================================================*/

#include "serial_cstm.h"

#ifndef ACCESS_HW
# define ACCESS_HW                 (mxGetPr(ACCESS_HW_PARAM)[0] != 0.0)
#endif


/*======================*
 * Miscellaneous macros *
 *======================*/

#define BASE_ADDR_PARAM_STRLEN     (128)


/*======================*
 * S-function User Data *
 *======================*/

typedef struct {
    uint_T    baseAddr;
} DIInfo;



/*====================*
 * S-function methods *
 *====================*/



/* Function: mdlInitializeSizes ===============================================
```

```
 *
 */
static void mdlInitializeSizes(SimStruct *S)
{
ssSetNumSFcnParams(S, NUM_PARAMS);
    if (ssGetNumSFcnParams(S) == ssGetSFcnParamsCount(S)) {

        if (ssGetErrorStatus(S) != NULL) {
            return;       //Error reported in mdlCheckParameters
        }
    } else {
        return; // Parameter mismatch will be reported by Simulink
    }

    /* None of this s-functions's parameters are tunable during simulation */
    {
        int_T i;
        for (i=0; i < NUM_PARAMS; i++) {
            ssSetSFcnParamNotTunable(S, i);
        }
    }

    ssSetNumSampleTimes( S, 1);
    ssSetNumInputPorts( S, 1);
    ssSetNumOutputPorts( S, 1);
    ssSetInputPortWidth(S, 0, 1);
    ssSetOutputPortWidth(S, 0, 1);
    ssSetInputPortDirectFeedThrough(S, 0, 1);
    ssSetInputPortRequiredContiguous(S,0,1);
    ssSetNumRWork(S, 2);
}



/* Function: mdlInitializeSampleTimes =========================================
 *
 */
static void mdlInitializeSampleTimes(SimStruct *S)
{
    ssSetSampleTime(S, 0, SAMPLE_TIME);
    ssSetOffsetTime(S, 0, 0.0);
}

/*static void InitializeCard(uint_T CHANNEL)
{
diSelectChannel(CHANNEL);
}*/
```

```
#define MDL_START
#if defined(MDL_START)
/* Function: mdlStart =======================================================
 *
 */
static void mdlStart(SimStruct *S)
{
    /*printf("Access Hardware is:?\n",ACCESS_HW);*/


DIInfo *diInfo = ssGetUserData(S);
char_T baseAddrStr[BASE_ADDR_PARAM_STRLEN];
uint_T baseAddr;

real_T *rwork = ssGetRWork(S);


         rwork[0] = 0.0;
rwork[1] = 1.0;




        /*Initialize diInfo (pointer saved in the user data) */
        mxGetString(BASE_ADDRESS_PARAM, baseAddrStr, BASE_ADDR_PARAM_STRLEN);
        baseAddr = (uint_T) strtoul(baseAddrStr, NULL, 0);



/* Initializes serial port */
InitializeComm(baseAddr);



if (diInfo != NULL) {
            free(diInfo);
        }
        if ((diInfo = malloc(sizeof(DIInfo))) == NULL) {
            ssSetErrorStatus(S,"Memory Allocation Error\n");
            return;
        }
        diInfo->baseAddr   = baseAddr;
        ssSetUserData(S, (void*) diInfo);

        printf("\nisc3n: Hardware Access Enabled\n");

}
#endif /* MDL_START */
```

```
/* Function: mdlOutputs =======================================================
 *
 */
static void mdlOutputs(SimStruct *S, int_T tid)
{

real_T *u = ssGetInputPortRealSignal(S,0);
real_T *y = ssGetOutputPortRealSignal(S,0);
uint_T  i=0;
DIInfo *diInfo  = ssGetUserData(S);
     uint_T baseAddr = diInfo->baseAddr;
real_T *rwork = ssGetRWork(S);
real_T eps=0.0;
real_T delta=0.0;
real_T FlowCtrl;
real_T RTS = rwork[1];




if ((u[0] - rwork[0]) > 0.0) delta = (u[0] - rwork[0]);
else delta = -(u[0] - rwork[0]);

if (u[0] > 10.0) u[0] = 10.0;
if (u[0] < -10.0) u[0] = -10.0;

if (delta >= eps){

if (RTS == 1.0){

SendComm(u[0],baseAddr);

y[0] = 1.0; //limited by nothing
}
else y[0] = 2.0; //limited by transmission medium
}

else y[0] = 0; //limited by similarity to previous

printf("testing outputs");

FlowCtrl = GetComm(baseAddr);

rwork[1] = FlowCtrl;
```

```
}

#define MDL_UPDATE
/* Function: mdlUpdate ======================================================
*/
static void mdlUpdate(SimStruct *S, int_T tid)
{
    real_T *u = ssGetInputPortRealSignal(S,0);
    real_T *rwork = ssGetRWork(S);
    int_T  i=0;



    rwork[0] = u[0];

}


/* Function: mdlTerminate =====================================================
 *
 */
static void mdlTerminate(SimStruct *S)
{
    DIInfo *diInfo = ssGetUserData(S);
uint_T baseAddr = diInfo->baseAddr;
SendComm(0,baseAddr);

    free(diInfo);
    ssSetUserData(S,NULL);

}




#ifdef  MATLAB_MEX_FILE    /* Is this file being compiled as a MEX-file? */
#include "simulink.c"      /* MEX-file interface mechanism */
#else
#include "cg_sfun.h"       /* Code generation registration function */
#endif

/* EOF: isc3n.c */


/* $Date: 01/11/2005 $
 * $Revision: 1.0 $
 * $Author: sampson $
```

```
 *
 * File: serial_cstm.h
 *
 * Abstract:
 *      S-Function device driver for serial communication with the
 *      Infranor SMT-BD1 servo amplifier.
 *
 * Adapted from the Mathworks das16di driver, 1998/04/28 04:19:32
 * Copyright (c) 1994-1998 by The MathWorks, Inc. All Rights Reserved.
 */

#ifndef __SERIAL__
#define __SERIAL__

#include "tmwtypes.h"
#include "drt_comp.h"




void InitializeComm(uint_T bA)
{

 WriteByteToHwPort(bA + 1 , 0);   /* Turn off interrupts - bA */

/*         PORT 1 - Communication Settings         */

 WriteByteToHwPort(bA + 3 , 0x80);  /* SET DLAB ON */
 WriteByteToHwPort(bA + 0 , 0x0C);  /* Set Baud rate - Divisor Latch Low Byte */
  /* Default 0x03 =  38,400 BPS */
  /*         0x01 = 115,200 BPS */
  /*         0x02 =  57,600 BPS */
  /*         0x06 =  19,200 BPS */
  /*         0x0C =   9,600 BPS */
  /*         0x18 =   4,800 BPS */
  /*         0x30 =   2,400 BPS */
 WriteByteToHwPort(bA + 1 , 0x00);  /* Set Baud rate - Divisor Latch High Byte */
 WriteByteToHwPort(bA + 3 , 0x03);  /* 8 Bits, No Parity, 1 Stop Bit */
 WriteByteToHwPort(bA + 2 , 0xC7);  /* FIFO Control Register */
 WriteByteToHwPort(bA + 4 , 0x0B);  /* Turn on DTR, RTS, and OUT2 */

printf("\nTest - comm port initialized\n");
}
```

```
void SendComm(real_T In_Volts,uint_T bA)
{

float float_val;

int int_val;
int nib3;
int nib2;
int nib1;
int nib0;

float_val=In_Volts*400*32767/4000;


/*printf("The float value is %f\n", float_val);*/


if (float_val >= 0)
{
int_val= (int) (float_val +.5);
/*printf("The hex value is %x\n", int_val);*/
}
else
{
int_val = (int) (-float_val +.5);
int_val = ~int_val;
int_val = int_val & 0xffff;
/*printf("The hex value is %x\n", int_val);*/
}

nib3 = int_val & 0xf000;
nib3 = nib3 >> 12;
if (nib3 > 9) nib3 = nib3 + 55;
else   nib3 = nib3 + 48;
/*y3[i] = nib3;
printf("The nib3 value is %i\n", nib3);*/

nib2 = int_val & 0x0f00;
nib2 = nib2 >> 8;
if (nib2 > 9) nib2 = nib2 + 55;
else   nib2 = nib2 + 48;
/*y2[i] = nib2;
printf("The nib2 value is %i\n", nib2);*/

nib1 = int_val & 0x00f0;
nib1 = nib1 >> 4;
if (nib1 > 9) nib1 = nib1 + 55;
```

```
else   nib1 = nib1 + 48;
/*y1[i] = nib1;
printf("The nib1 value is %i\n", nib1);*/

nib0 = int_val & 0x000f;
if (nib0 > 9) nib0 = nib0 + 55;
else   nib0 = nib0 + 48;
/*y0[i] = nib0;
printf("The nib0 value is %i\n", nib0);*/



WriteByteToHwPort(bA, 67);
WriteByteToHwPort(bA, 86);
WriteByteToHwPort(bA, nib3);
WriteByteToHwPort(bA, nib2);
WriteByteToHwPort(bA, nib1);
WriteByteToHwPort(bA, nib0);
WriteByteToHwPort(bA, 13);

printf("\nByte written\n");
}

real_T GetComm(uint_T bA)
{
real_T FlowCtrl=0;
real_T Flag=0;
uint_T Buffer=0;
uint_T i=1;

for (i; i<10; i++) {

Buffer = ReadByteFromHwPort(bA);

if (Buffer == 58) Flag = 1.0;
}

if (Flag == 1.0) FlowCtrl = 1.0;
else FlowCtrl = 0.0;


return (FlowCtrl);

}

#endif /* __serial_cstm__ */

/* EOF: serial_cstm.h */
```

APPENDIX B: COMMUNICATION FILES

This section contains the text files necessary to initialize the motor controller using the serial link, as discussed in Section 3.1.

initiate.txt (read columnwise starting at "AD0000" and ending at "SS")

AD0000	KV	MS
AD	IN	SS
ID	IM	VL
ME	TL	TL
SS	VL	SS
NP	RC	IM
PM	ZN	IN
CL	ZP	RC
AF	ZW	SS
FC	VE	SS
BP	HW	ER
KP	SS	SS
KI	MS	AR
FI	MA	SS
KE	MS1	

manual.txt

```
SS
MS
DI
MA
SS
```

stop.txt

```
SS
AR
SS
```

automatic.txt

```
SS
MS
AN
MA
SS
```

disconnect.txt

```
MS
SS
MS0
```

10rpm.txt

```
CV0052
```

100rpm.txt

```
CV0333
```

neg100rpm.txt

```
CVFEB1
```

0rpm.txt

```
CV0000
```

APPENDIX C: ADDITIONAL RESULTS

This section contains figures detailing the response of the EHA with the proportional, switched-gain and fuzzy controllers. The results of the tests are summarized in the tables presented in Chapter 6, but examination of the behaviour of the EHA as a function of time as the magnitude and type of the input as well as the loading conditions are changed.

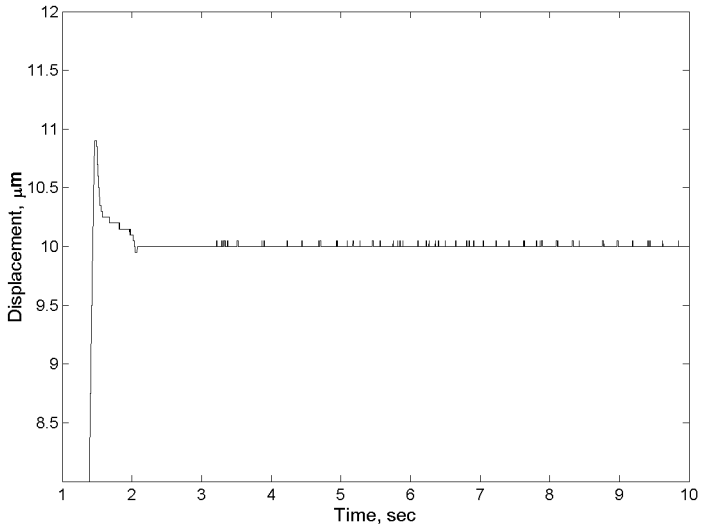

Figure C.1: Steady state response with FLC, 10 μm step input

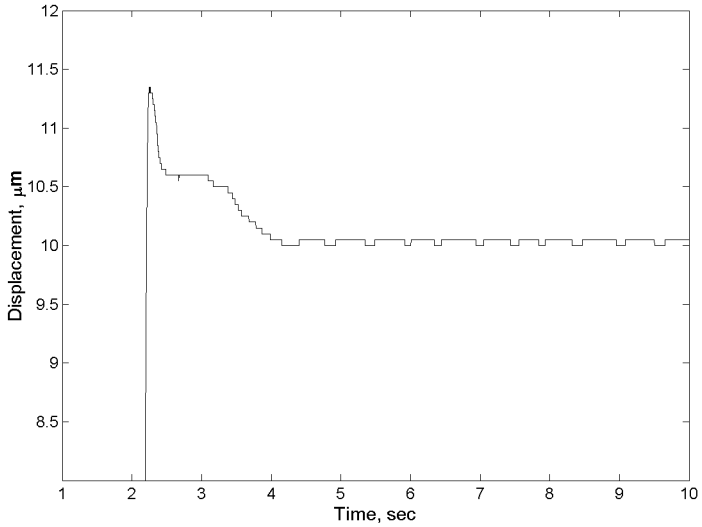

Figure C.2: Steady state response with SGC, 10 μm step input

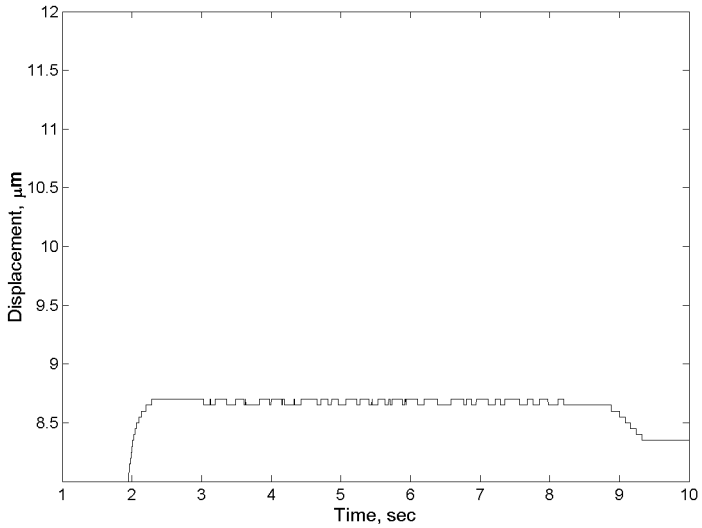

Figure C.3: Steady state response with PC, 10 μm step input

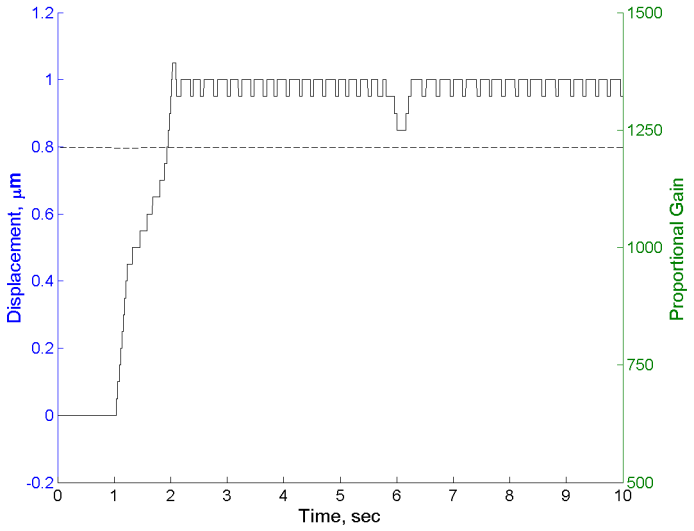

Figure C.4: Steady state response with FLC, 1 μm step input

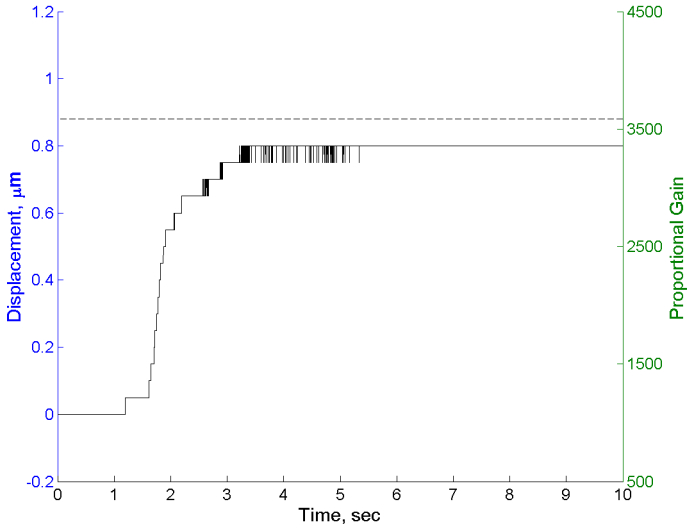

Figure C.5: Steady state response with SGC, 1 μm step input

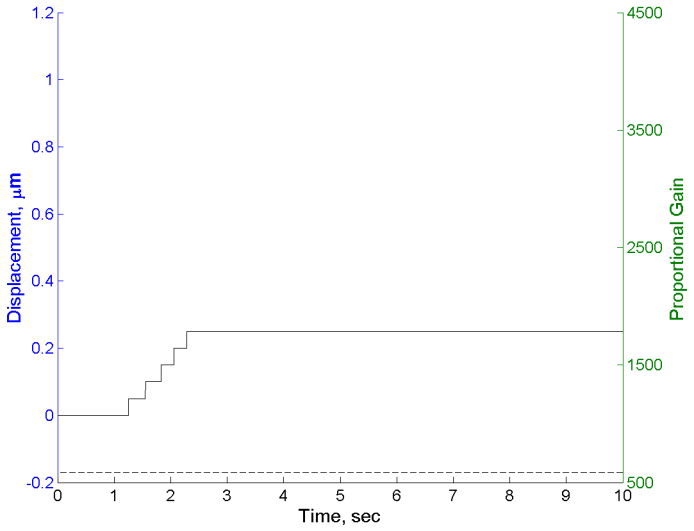

Figure C.6: Steady state response with PC, 1 μm step input

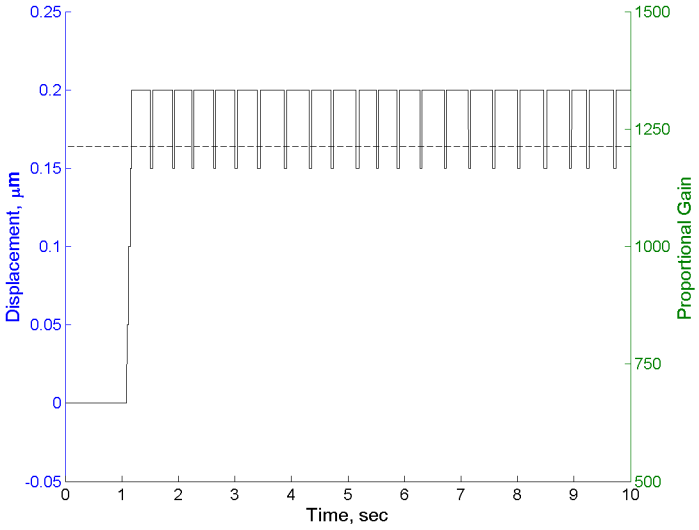

Figure C.7: Steady state response with FLC, 200 nm step input

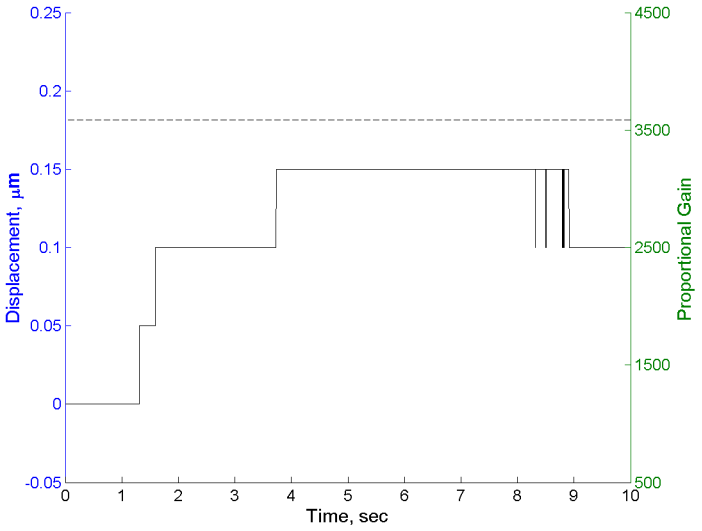

Figure C.8: Steady state response with SGC, 200 nm step input

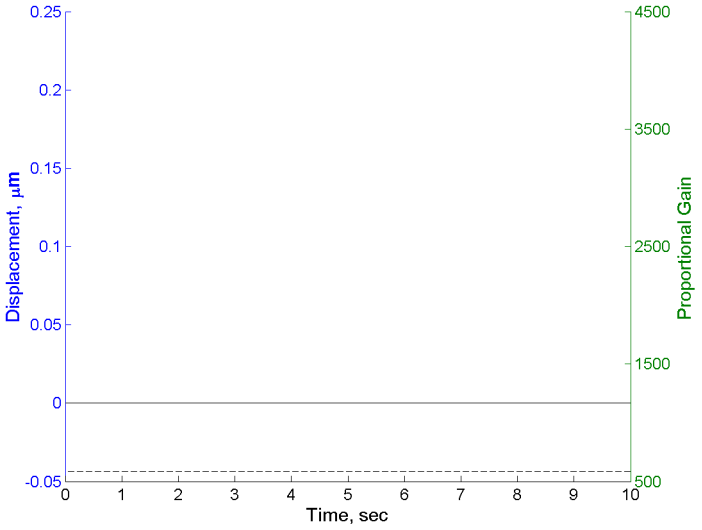

Figure C.9: Steady state response with PC, 200 nm step input

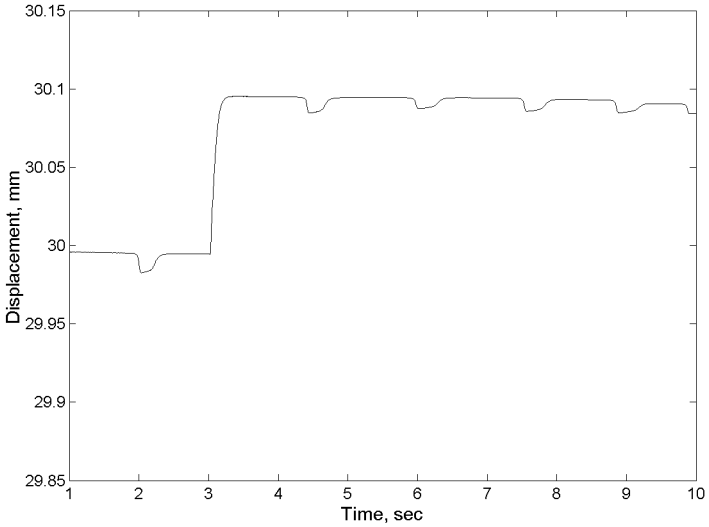

Figure C.10: Steady state response under load with FLC, 100 μm step input (3cm bias)

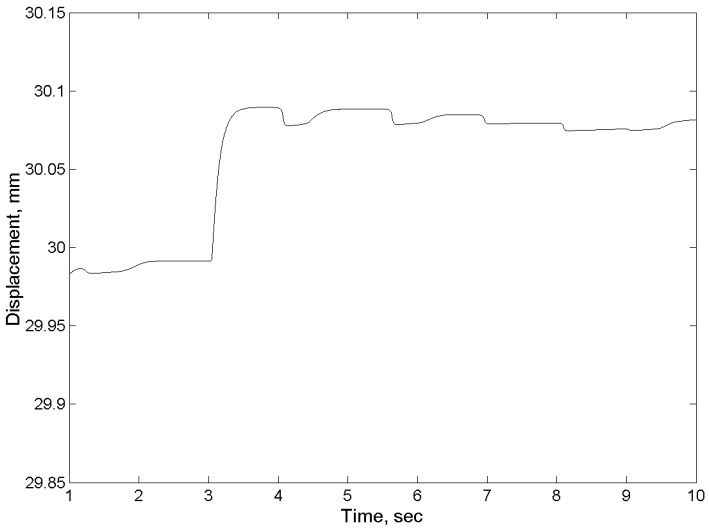

Figure C.11: Steady state response under load with SGC, 100 μm step input (3cm bias)

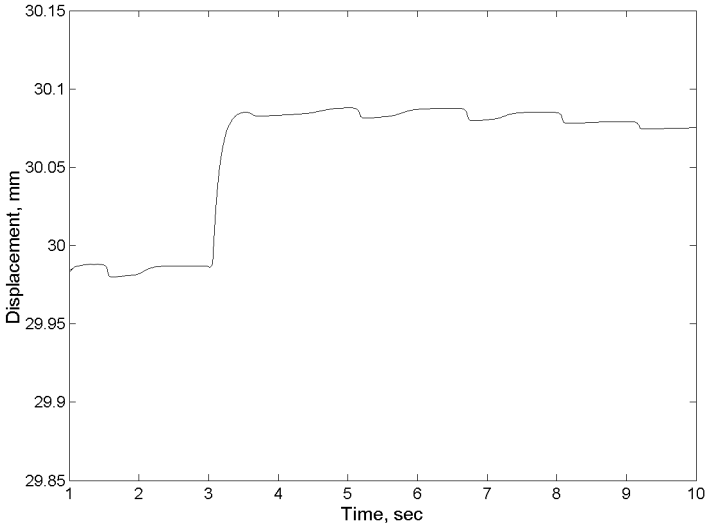

Figure C.12: Steady state response under load with PC, 100 μm step input (3cm bias)

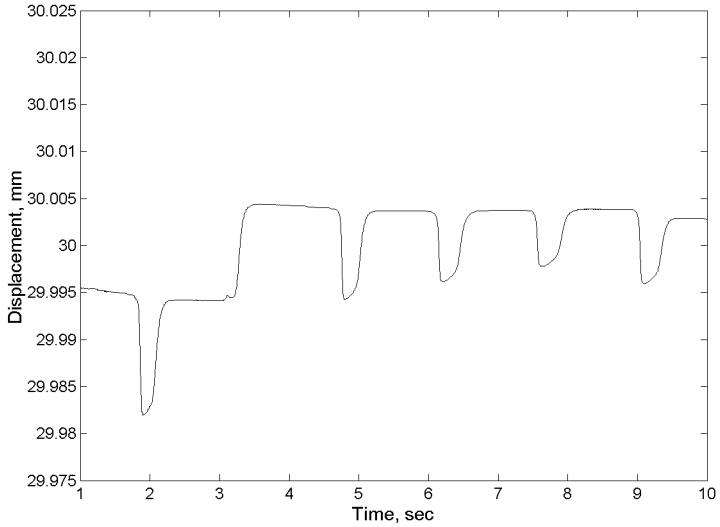

Figure C.13: Steady state response under load with FLC, 10 μm step input (3cm bias)

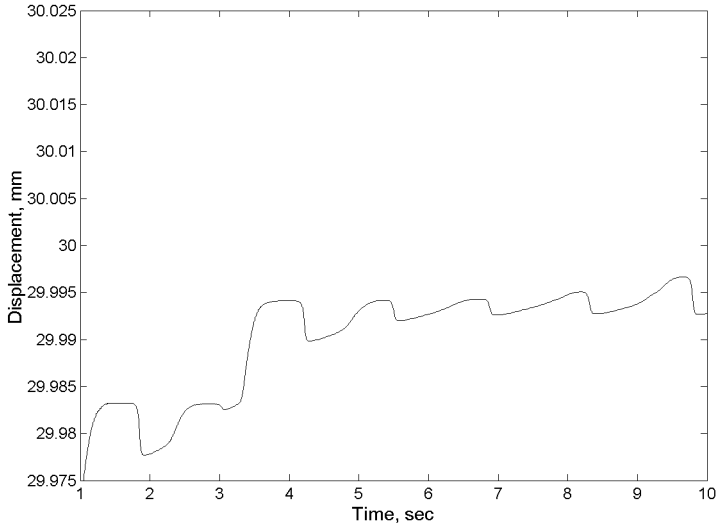

Figure C.14: Steady state response under load with SGC, 10 μm step input (3cm bias)

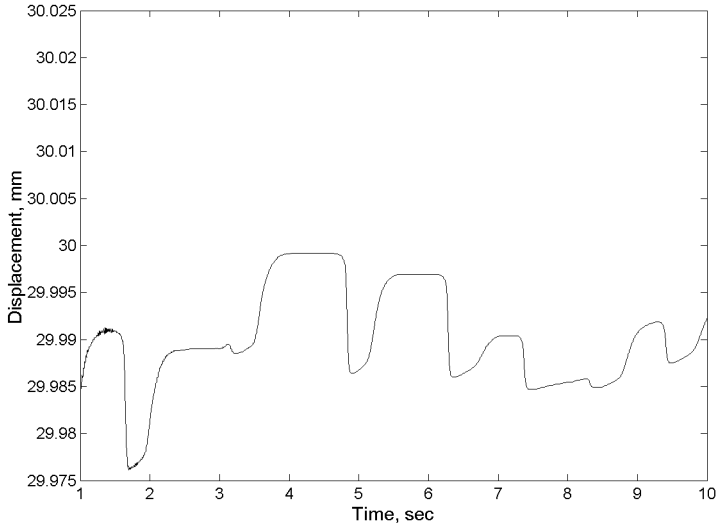

Figure C.15: Steady state response under load with PC, 10 μm step input (3cm bias)

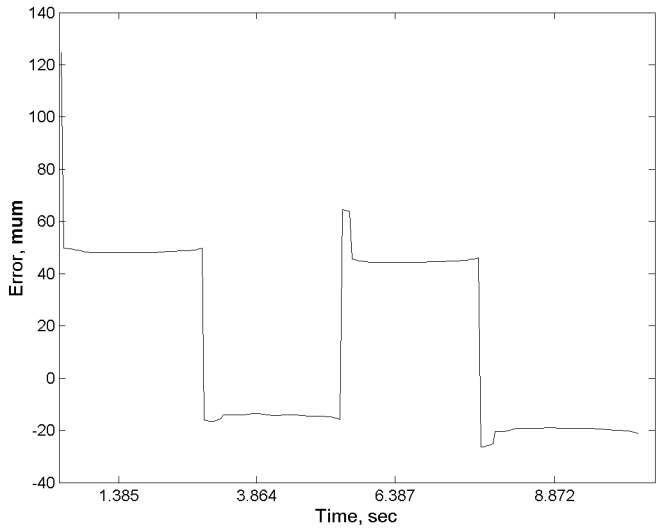

Figure C.16: Steady state error under load with FLC, 1cm sine wave input

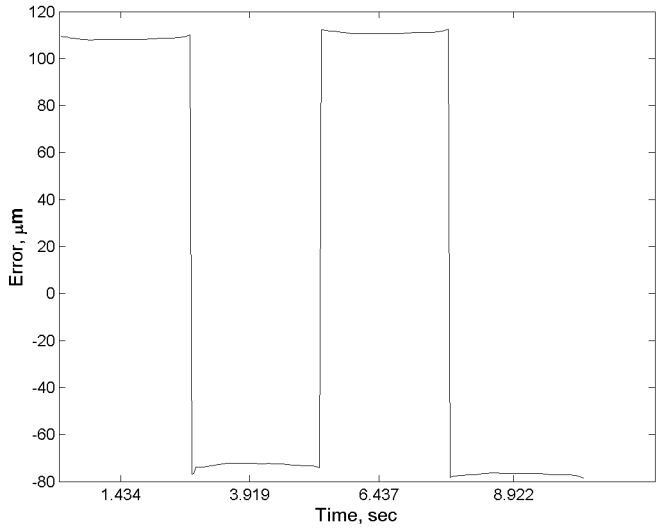

Figure C.17: Steady state error under load with SGC, 1cm sine wave input

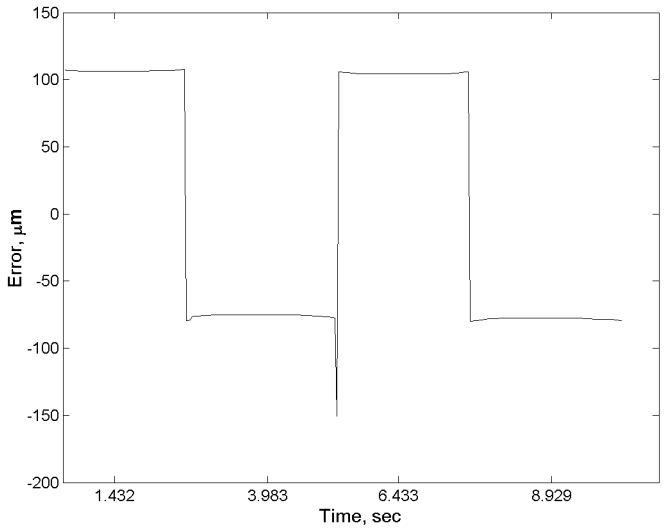

Figure C.18: Steady state error under load with PC, 1cm sine wave input

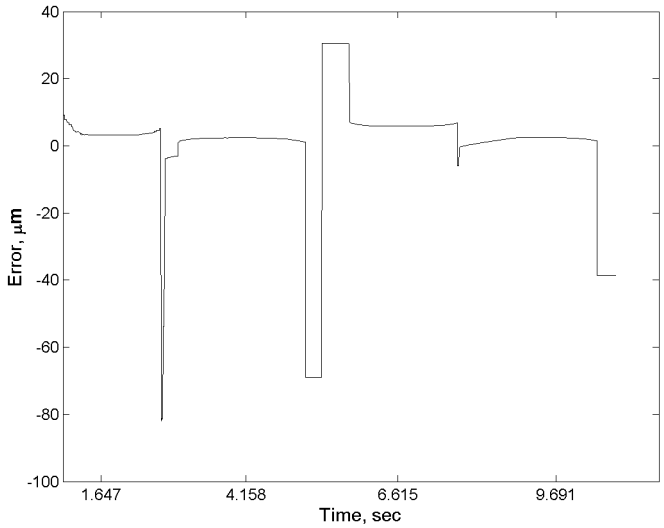

Figure C.19: Steady state error under load with FLC, 100um sine wave input

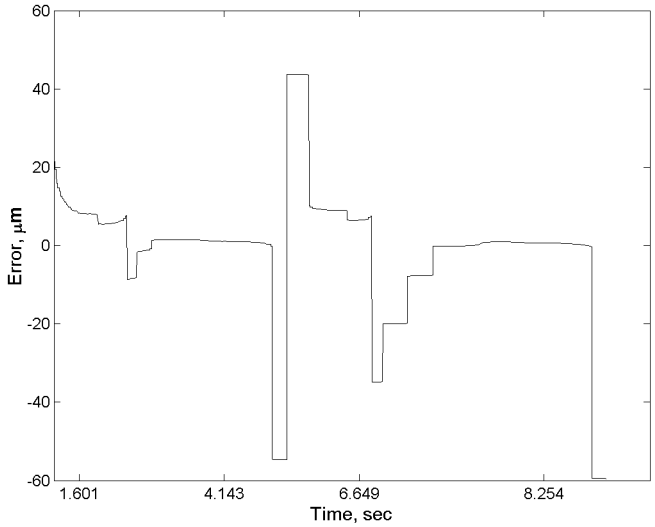

Figure C.20: Steady state error under load with SGC, 100um sine wave input

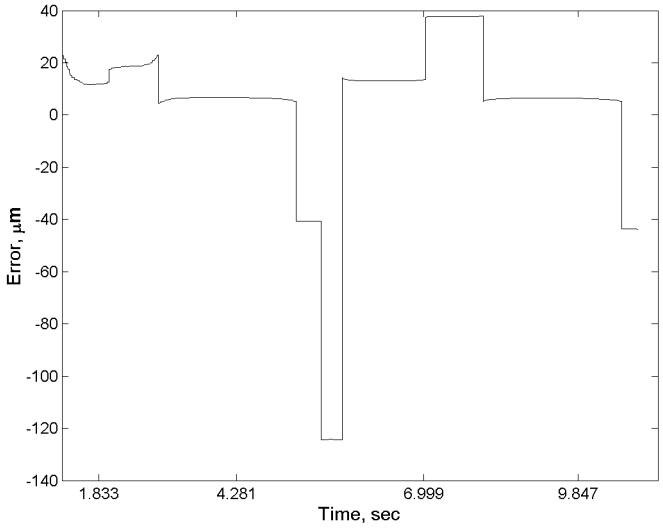

Figure C.21: Steady state error under load with PC, 100um sine wave input

APPENDIX D: INTERNATIONAL JOURNAL OF FLUID POWER (IJFP) PAPER

This section includes the paper which was published by the International Journal of Fluid Power in August, 2004. In the paper the linear model of the EHA previously developed by Habibi and Singh is expanded to include the ability to simulate the effect of arbitrary time-varying force and flow disturbances on the EHA. This paper is included with the express permission of the journal's publishers.

EFFECT OF CONTROLLER IN REDUCING STEADY-STATE ERROR DUE TO FLOW AND FORCE DISTURBANCES IN THE ELECTROHYDRAULIC ACTUATOR SYSTEM

Eric Sampson, Saeid Habibi, Richard Burton and Yuvin Chinniah

Department of Mechanical Engineering, University of Saskatchewan, 57 Campus Drive, Saskatoon, Saskatchewan, Canada S7N 5A9
eric.s@usask.ca, saeid_habibi@engr.usask.ca, richard_burton@engr.usask.ca, yac494@mail.usask.ca

Abstract

This paper pertains to the nonlinear control of a high-precision hydrostatic actuation system known as the ElectroHydraulic Actuator (EHA). It describes the action of the controller in reducing the steady state error resulting from flow and force disturbances. The EHA uses inner-loop pump velocity feedback to achieve an unprecedented level of accuracy for a hydrostatic system. A published mathematical model of the EHA is reviewed and expanded to produce an equation that predicts the response of the EHA to both desired inputs as well as flow and force disturbances. This equation suggests that the use of a proportional outer-loop controller should result in steady-state error as a result of these disturbances, but that a PI outer-loop controller should eliminate the steady-state error. Experimental results from a prototype of the EHA demonstrate that due to the nonlinear friction present in the actuator, the use of a conventional proportional or PI controller is not sufficient to effectively deal with these disturbances. However, a nonlinear proportional outer-loop controller does result in a substantial performance improvement in regards to disturbance rejection for positional accuracy. Experiments conducted on the prototype using the nonlinear controller reveal that it is capable of a positional accuracy of 1 μm for a load of 20 kg.

Keywords: actuator, electrohydraulic, hydrostatic, micro-precision, disturbance rejection

1 Introduction

Traditional hydraulic systems have several drawbacks with regards to their use in high-performance applications. These include:

- High energy usage due to the pumps being driven continuously by constant-speed motors even when the actuator is not moving,
- requirement of a large central fluid distribution system including a reservoir, and
- requirement of expensive servo valves for precision control.

A novel hydraulic actuation system referred to as the ElectroHydraulic Actuator (EHA) has been developed to overcome these drawbacks while providing high positional accuracy, as presented by Habibi and Goldenberg (1999). The EHA uses a bi-directional fixed-displacement pump powered by a variable-speed servomotor, Watton (1989), Arnautovic (1993), Desai and Bobrow (1989), Manring and Lueke (1989), Merrit (1967). The actuator is directly connected to the pump as shown in Fig. 1. The pump's speed and direction determine the flow to and from the actuator and hence its displacement. An integral contributor to the high performance of the EHA is inner-loop velocity control of the pump velocity. This reduces the dead-band common in hydrostatic actuation systems that limit their performance and positional accuracy. A prototype of the EHA has demonstrated an excellent level of performance with a nonlinear control approach, being capable of moving a 20 kg load with an accuracy of 1 μm and a critically damped rise time of 0.3 s.

In sections 1 to 6 of the paper, a method is developed to theoretically predict the steady-state positional error of the EHA as a result of both flow and force disturbances. A flow disturbance can be defined as any load flow that is not included in the EHA model, such as changes in actuator seal performance causing

additional external leakage. A force disturbance can similarly be defined as any external force applied to the load that is not included in the EHA model. Examples of force disturbances could be nonlinear friction between the rails and the load, or sudden application of an external force.

The development of a way to model these disturbances is important because it allows one the opportunity to simulate the effects of arbitrary time-varying flow and force disturbances on the EHA system using a simulation package on a computer. For this paper, the purpose of developing an EHA model that includes these disturbances was to predict the steady-state error as a result of flow and force disturbances when using a proportional versus a proportional-integral (PI) controller. In section 7 of the paper, experiments that were conducted to verify the accuracy of the predictions are presented; in addition a controller is developed to minimize the steady-state error of the EHA.

2 EHA System

The EHA system consists of the following components:

- Controller
- Electric Motor
- Bi-Directional Gear Pump
- Accumulator
- Pressure, Position and Speed Sensors
- Crossover Relief Valve
- Symmetrical Actuator
- Load

A simplified schematic of the system is shown as Fig. 1.

The three-phase brushless AC electric motor directly drives the pump, which controls the flow of hydraulic oil to the two active chambers of the actuator. Depending on the load, the resulting pressure differential between the actuator chambers then applies a net force on the external load. In this case, the load is a 20 kg steel block mounted on two linear rails which is displaced horizontally by the force exerted on it by the actuator.

The symmetrical linear actuator used in the prototype has a single rod and has been designed expressly for the system, as described in Habibi and Goldenberg (1999). The actuator has two working chambers C1 and C2, illustrated in Fig. 2. The rod is hollow and has a circular disc at the end of the rod inside the actuator. Hydraulic fluid enters the two chambers of the actuator through ports O1and O2.

The working areas of the chambers A1 and A2 are made equal in area, which results in symmetrical flow if leakage is neglected. There also exists a third working chamber C3, which can be pressurized via port O5 to provide a bias to counteract a constant external force. Finally, ports O3 and O4 are provided to drain any actuator fluid leakage.

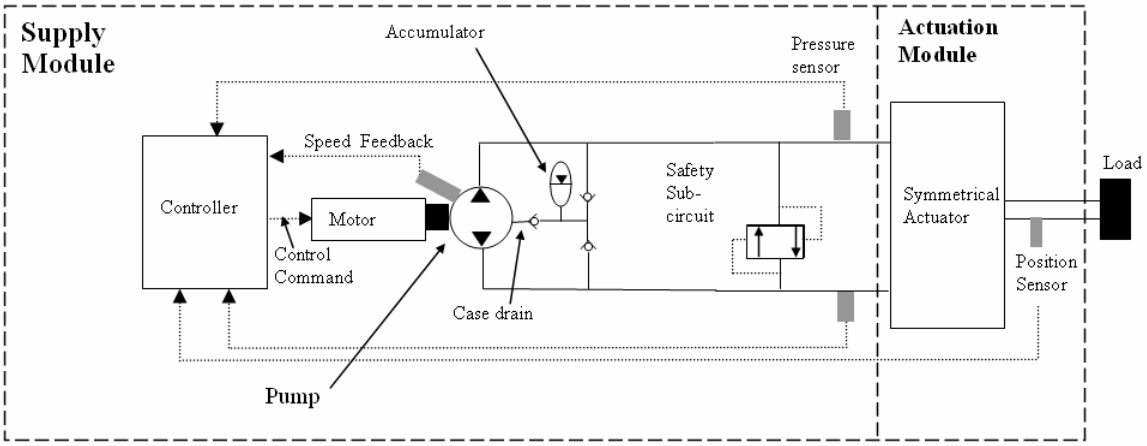

Fig. 1: *Schematic of EHA*

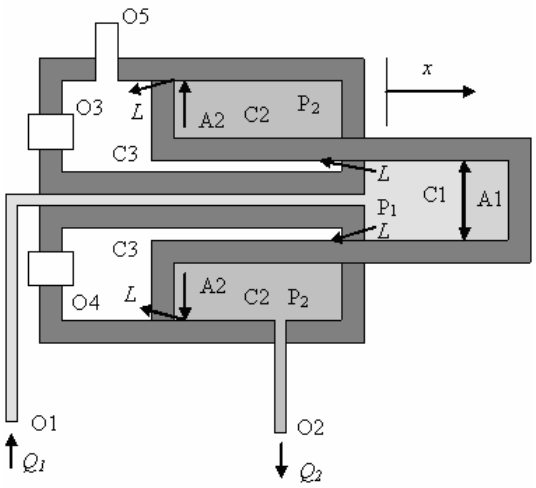

Fig. 2: *Cross-Section of Actuator*

An accumulator is connected to the low-pressure case drain of the gear pump. It prevents cavitation, and replaces fluid lost due to leakage. The accumulator sets the minimum system pressure and can be adjusted from 0.28-0.69 MPa (40-100 psi). The upper limit is set by the pump seals, which would be damaged if the case drain pressure exceeds 0.69 MPa.

The crossover relief valves increase the safety of the system by preventing excessive pressure build-up if the actuator reaches the end of its travel, or if a fault were to occur. Recently a new position sensor has been added to the system to measure the displacement of the load. The sensor, an optical linear encoder, has a resolution of 1 μm. This sensor has enabled more exact positional data to be obtained, increasing the accuracy of the system.

3 EHA Model

A mathematical model of the EHA system was developed in Habibi and Singh (2000). In this paper, an analysis of the effect of flow disturbances and force disturbances on the system is considered by using an expanded EHA model. Figure 3 shows a simplified block diagram for the system showing the outer-loop position controller $G_{OL}(s)$, the inner-loop pump velocity loop $G_V(s)$, and the hydraulic transfer function $G_H(s)$.

The EHA system as described in Habibi and Singh (2000) used a proportional outer-loop controller such that:

$$G_{OL}(s) = \frac{U(s)}{E(s)} = 585 \qquad (1)$$

The motor/pump subsystem consists of a bi-directional pump driven directly by an AC electrical motor controlled by a velocity controller. An identified model of the overall motor/pump subsystem transfer function was described in Habibi, Pastrakuljic and Goldenberg (2000). As such, the motor/pump subsystem with a PI inner-loop controller can be described by Eq. 2, an identified transfer function:

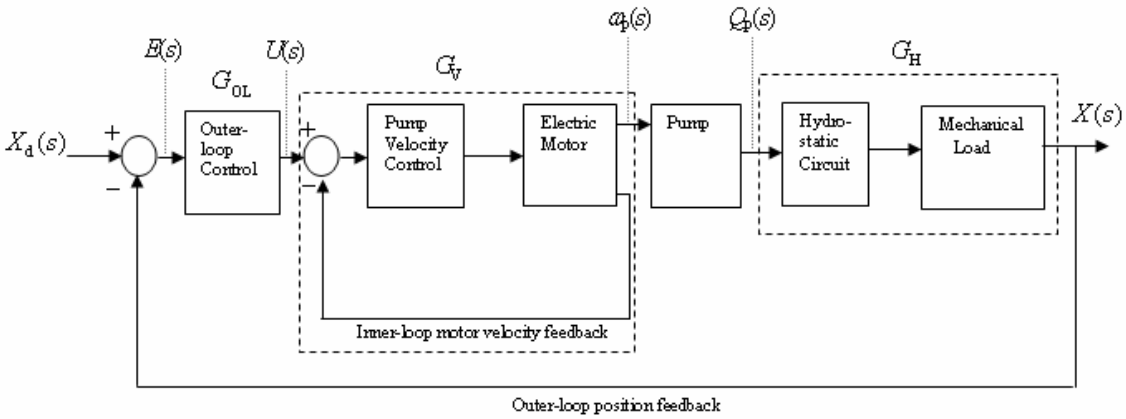

Fig. 3: *Simplified EHA block diagram*

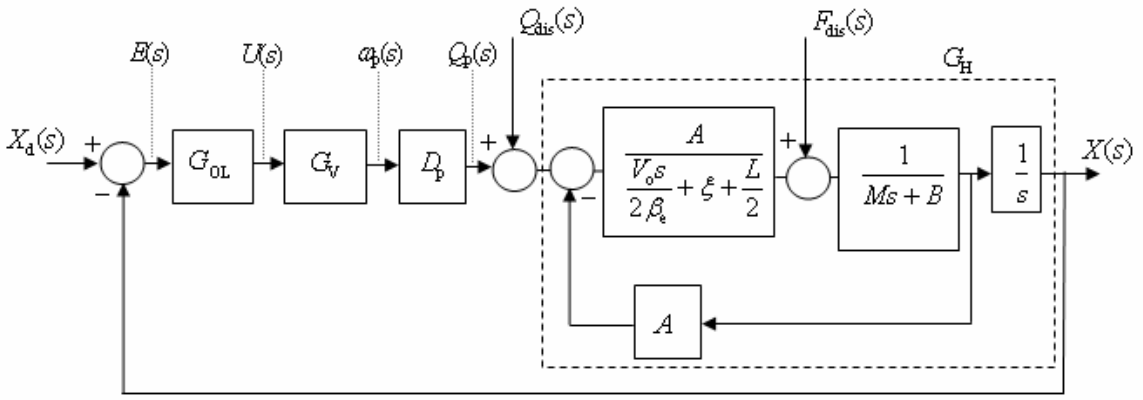

Fig. 4: *EHA block diagram showing hydraulic transfer function* $G_H(s)$

$$G_V(s)=\frac{\omega_p(s)}{U(s)}=\frac{0.2779s+40.55}{5.780\times10^{-5}s^2+1.016\times10^{-2}s+1} \quad (2)$$

let $\kappa_v = 40.55$, the motor gain.

Further to the hydraulic model of the EHA as described in Habibi and Singh (2000) and neglecting the pressure drop across the lines while introducing a disturbance flow Q_{dis}, a simplified linear relationship is obtained as:

$$D_p\omega_p+Q_{dis}=A\dot{x}+\frac{V_o}{2\beta_e}\left(\frac{dP_1}{dt}-\frac{dP_2}{dt}\right)+\xi(P_1-P_2)+\frac{L}{2}(P_1-P_2) \quad (3)$$

From Eq. 3, the resulting transfer function between the actuator pressure differential and the fluid flow is obtained as:

$$\frac{P_1(s)-P_2(s)}{D_p\omega_p(s)+Q_{dis}(s)-AX(s)s}=\frac{1}{\frac{V_o s}{2\beta_e}+\xi+\frac{L}{2}} \quad (4)$$

The EHA prototype has a load that consists of a mass M that slides on horizontal rails that contribute a viscous damping term B, and experiences a disturbance force F_{dis}. This results in the following load model:

$$(P_1-P_2)A+F_{dis}=M\ddot{x}+B\dot{x} \text{ or} \quad (5)$$

$$\frac{X(s)s}{(P_1(s)-P_2(s))A+F_{dis}(s)}=\frac{1}{Ms+B} \quad (6)$$

Combining Eq. 1, 2, 4 and 6 results in the system block diagram shown in Fig. 4. Included in dashed lines in Fig. 4 is the hydraulic transfer function $G_H(s)$, which relates the output $X(s)$ to the input flow $D_p\omega_p(s)+Q_{dis}(s)$.

4 Effect of Flow Disturbances

The inclusion of a term for flow disturbances, $Q_{dis}(s)$, in the EHA model allows for the investigation of external pump leakage and other sources of flow disturbance, including changes in actuator seal performance. It will now be shown that the EHA system using a proportional controller will theoretically exhibit a steady-state error if there is a constant flow disturbance $Q_{dis}(s)$.

In order to demonstrate this, it is desirable to obtain the transfer function between the flow disturbance $Q_{dis}(s)$ and the output $X(s)$. To do so, the inputs $X_d(s)$ and $F_{dis}(s)$ shown in Fig. 4 are set to zero. Since $F_{dis}(s) = 0$, the hydraulic transfer function $G_H(s)$ shown in dashed lines in Fig. 4 can be expressed in the standard form as:

$$G_H(s)=\frac{X(s)}{D_p\omega_p(s)+Q_{dis}(s)}=\frac{\kappa_h\omega_{nh}{}^2}{s(s^2+2\zeta_h\omega_{nh}s+\omega_{nh}{}^2)} \quad (7)$$

where

$$\kappa_h=\frac{2A}{2A^2+(2\xi+L)B}$$

$$\zeta_h=\left(\frac{B}{2M\omega_{nh}}+\frac{(2\xi+L)\beta_e}{2V_o\omega_{nh}}\right) \text{ and}$$

$$\omega_{\text{nh}} = \sqrt{\left(\frac{2\beta_{\text{e}} A^2 + (2\xi + L)\beta_{\text{e}} B}{MV_{\text{o}}} \right)}$$

Rearranging Fig. 4 allows the relationship between the flow disturbance $Q_{\text{dis}}(s)$ and the output $X(s)$ to be illustrated in a conventional manner, as shown in Fig. 5.

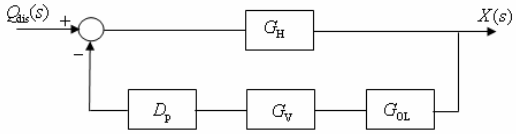

Fig. 5: *Block diagram for flow disturbances*

From Fig. 5, the transfer function between $Q_{\text{dis}}(s)$ and $X(s)$ can be derived as:

$$\frac{X(s)}{Q_{\text{dis}}(s)} = \frac{G_{\text{H}}(s)}{1 + G_{\text{H}}(s) D_{\text{p}} G_{\text{V}}(s) G_{\text{OL}}(s)} \tag{8}$$

or

$$\frac{X(s)}{Q_{\text{dis}}(s)} = \frac{\kappa_{\text{h}} {\omega_{\text{nh}}}^2}{\left[\begin{array}{l} \{ s(s^2 + 2\zeta_{\text{h}} \omega_{\text{nh}} s + {\omega_{\text{nh}}}^2) \ldots \\ \quad + \kappa_{\text{h}} {\omega_{\text{nh}}}^2 D_{\text{p}} G_{\text{V}}(s) G_{\text{OL}}(s) \} \end{array} \right]} \tag{9}$$

To investigate the effect of flow disturbances on the position of the actuator, the final value theorem is used to determine the steady-state error E_{ss} in the system output, given a step flow disturbance $Q_{\text{dis}}(s) = 1\times10^{-7}/s$ m^3/s. This disturbance flow was selected because it is approximately 10% of the maximum pump flow that occurs when the system is moved 0.01 m in 1 sec.

$$\begin{aligned} E_{\text{ss}} &= \lim_{s\to 0} \left[s Q_{\text{dis}}(s) \frac{X(s)}{Q_{\text{dis}}(s)} \right] \\ &= \left[\frac{s Q_{\text{dis}}(s)}{D_{\text{p}} \kappa_{\text{v}} G_{\text{OL}}(s)} \right]_{s\to 0} \end{aligned} \tag{10}$$

It should be noted that even though the model of the hydraulic system is Type 1 (as indicated by Eq. 7), the steady-state error is not alleviated in the presence of flow disturbances. This can be intuitively verified by considering the integral action of the hydraulic transfer function. Since the integral action is downstream of the error source, as shown in Fig. 4, the system will integrate the error causing the actuator to move. This will continue until the error becomes of the same magnitude as the flow disturbance but of opposite sign. When that occurs, the input of the integrator becomes zero since the integrated error signal and the flow disturbance will cancel each other out. The result is that the output position will then remain fixed at a non-zero steady-state error.

Since the steady-state error predicted by Eq. 10 includes the transfer function of the outer-loop controller $G_{\text{OL}}(s)$, the response of the EHA to flow disturbances depends on the type of controller adopted. This is now considered.

4.1 Proportional Controller

The initial design of the EHA system as presented by Habibi and Singh (2000) employed a proportional outer-loop controller with $G_{\text{OL}}(s) = K_{\text{p}} = 585$. Using this controller, Eq. 10 predicts the steady-state error as a result of a 1×10^{-7} m^3/s step flow disturbance to be:

$$E_{\text{ss}} = \left[\frac{1\times10^{-7}}{6.863\times10^{-6}(585)} \right]_{s\to 0} = 24.9\mu\text{m} \tag{11}$$

This analysis indicates that a step flow disturbance of 1×10^{-7}m^3/s would theoretically result in a steady-state position error of 25 μm. This implies that, when using a proportional controller with the EHA, significant actuator displacements may occur as a result of flow disturbances that are less than ten percent of the pump flow $Q_{\text{P}}(s)$.

4.2. Proportional-Integral Controller

The steady-state error in response to a step flow disturbance is theoretically eliminated if an integrator is introduced upstream of the location where the disturbance enters the system model. This can be achieved if the outer-loop controller is changed from proportional to proportional-integral,where $G_{\text{OL}}(s) = K_{\text{p}} + \frac{K_{\text{i}}}{s} = 585 + \frac{K_{\text{i}}}{s}$ such that:

$$E_{\text{ss}} = \left[\frac{1\times10^{-7}}{6.863\times10^{-6}\left(585 + \frac{K_{\text{i}}}{\text{s}} \right)} \right]_{s\to 0} = 0\text{m} \tag{12}$$

This suggests that adding integral action to the outer-loop controller before the disturbance input theoretically eliminates the

steady-state position error as a result of the flow disturbance. However, as will be shown in the experimental data, the practical aspect of using integral action to reduce steady state error at the micrometer level is not without its problems.

Specifically, the presence of nonlinear friction in the actuator results in limit cycle oscillation. Due to the small magnitude of the error present near steady state, it takes a significant amount of time for the integrator to accumulate enough control action in order to overcome the static friction of the motionless actuator. When it does, the actuator will accelerate rapidly since the dynamic friction is smaller in magnitude than the static friction, resulting in an overshoot of the desired position. The instantaneous error then drives the error accumulated in the integrator towards zero thus holding the actuator at a position with non-zero steady-state error. The entire cycle will then repeat, resulting in the limit cycle oscillation seen in Fig. 12.

5 Effect of Force Disturbances

The effect of force disturbances on the actuator may be modelled in a similar manner to the flow disturbances discussed above. If the inputs $X_d(s)$ and $Q_{dis}(s)$ are set to zero, the relationship between $X(s)$ and $F_{dis}(s)$ may be illustrated by rearranging the system block diagram of Fig. 4 to produce Fig. 6.

From Fig. 6 the transfer function between $X(s)$and $F_{dis}(s)$ may be determined by simplifying the block diagram to obtain the following:

$$\frac{X(s)}{F_{dis}(s)} = \frac{1/M}{s^2 + \frac{B}{M}s + \frac{\kappa_h \omega_{nh}{}^2 (As + D_p G_V(s) G_{OL}(s))}{s + (2\xi + L)\beta_e / V_o}} \tag{13}$$

The final value theorem can be used as in Eq. 14 to determine the actuator system output when the input is a step force disturbance $F_{dis}(s) = 1000/s$ N. This value of force disturbance was selected as it represents the force applied by gravity to a vertical actuator by a nominal human mass of 100 kg. This situation could occur if a person were to sit down on a positioning table controlled in the vertical axis by an EHA.

$$\begin{aligned} E_{ss} &= \lim_{s\to 0}\left[sF_{dis}(s)\frac{X(s)}{F_{dis}(s)} \right] \\ &= \left[\frac{sF_{dis}(s)(2\xi + L)}{2AD_p\kappa_v G_{OL}(s)} \right]_{s\to 0} \end{aligned} \tag{14}$$

Equation 14 implies that the response of the EHA to force disturbances depends on the type of outer-loop controller adopted, similar to the response of the EHA to flow disturbances. This dependency is now considered.

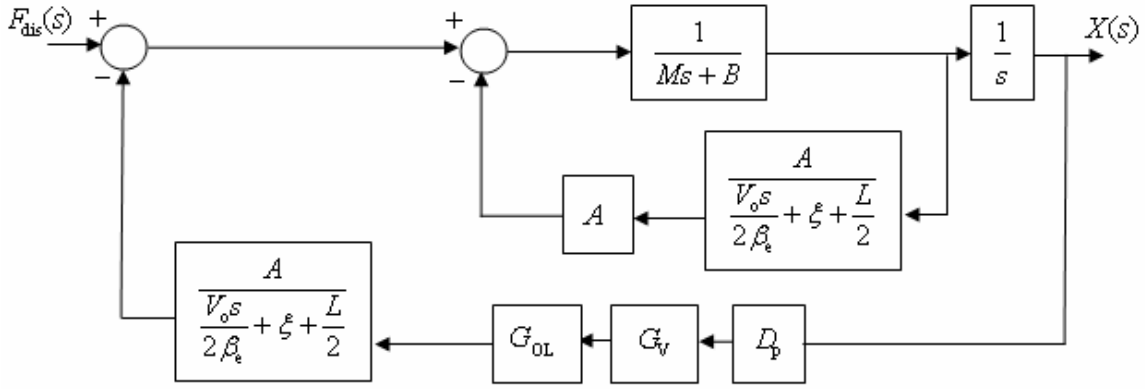

Fig. 6: *Block diagram for force disturbances*

5.1. Proportional Controller

Using Eq. 14, the response to a 1000 N force disturbance can be predicted for the original proportional outer-loop controller $G_{OL}(s) = K_p = 585$ as:

$$E_{ss} = \left[\frac{1000}{22950(585)} \right]_{s \to 0} = 74 \mu \text{m} \tag{15}$$

This analysis indicates that a force disturbance of 1000 N would theoretically result in a significant steady-state positional error of 74 μm.

5.2. Proportional-Integral Controller

The steady-state error in response to a step force disturbance is theoretically eliminated if the outer-loop controller is changed from proportional to proportional-integral, where $G_{OL}(s) = K_p + \frac{K_i}{s} = 585 + \frac{K_i}{s}$ such that:

$$E_{ss} = \left[\frac{1000}{22950\left(585 + \frac{K_i}{s} \right)} \right]_{s \to 0} = 0 \text{m} \tag{16}$$

6 Complete EHA Transfer Function

From the block diagram in Fig. 4 and setting the disturbance inputs $Q_{dis}(s)$ and $F_{dis}(s)$ to be zero, a transfer function may be determined between the actuator position $X(s)$ and the desired position $X_d(s)$:

$$\frac{X(s)}{X_d(s)} = \frac{\kappa_h \omega_{nh}{}^2 D_p G_V(s) G_{OL}(s)}{\left[\begin{array}{l} \{ s(s^2 + 2\zeta_h \omega_{nh} s + \omega_{nh}{}^2) \ldots \\ \quad + \kappa_h \omega_{nh}{}^2 D_p G_V(s) G_{OL}(s) \} \end{array} \right]} \tag{17}$$

Substituting the relevant parameters with $G_{OL}(s) = K_p = 585$ into Eq. 17 in MATLAB® Simulink produced the theoretical closed-loop response to a 10 mm step input shown in Fig. 7.

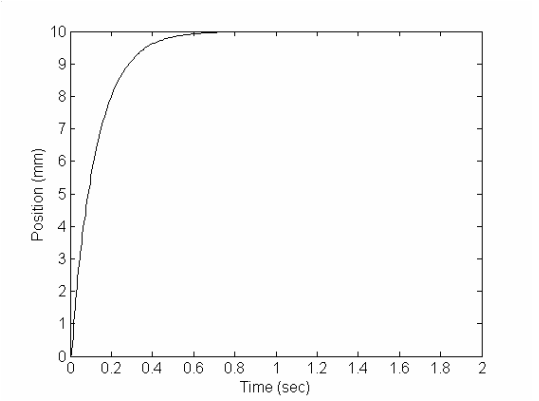

Fig. 7: *Theoretical response to 10 mm step input*

Using the principal of superposition for a linear system, Eq. 9, 13 and 17 can be combined to produce Eq. 18. This equation predicts the output $X(s)$ of the EHA given any combination of the three time-varying inputs: desired position $X_d(s)$, flow disturbance $Q_{dis}(s)$, and force disturbance $F_{dis}(s)$.

$$X(s) = \frac{X(s)}{X_d(s)} X_d(s) + \frac{X(s)}{Q_{dis}(s)} Q_{dis}(s) \ldots + \frac{X(s)}{F_{dis}(s)} F_{dis}(s) \tag{18}$$

Equation 18 is useful because it allows the response of the system to arbitrary disturbances to be simulated that would be complicated using mathematical analysis. For instance, one could predict the system response if the load was travelling with constant velocity and the actuator experienced additional load resistance as a result of increased friction due to local damage to the load rails. This could be accomplished by setting the desired position input $X_d(s)$ to be a linearly increasing function, and then adding a step friction force to the force disturbance input $F_{dis}(s)$. Similarly, one could predict the effect of a change in external actuator leakage while the actuator is maintaining a constant position. This could be modelled by applying the desired position signal to the input $X_d(s)$, and then adding a small negative flow to the flow disturbance input $Q_{dis}(s)$ after the actuator position has settled.

In summary, the theoretical analysis presented here leads to an equation that can be used for predicting the response of the EHA with respect to the desired input as well as arbitrary time-varying flow and force disturbances. This equation demonstrates that constant flow and force disturbances will result in steady-state positional errors if a proportional outer-loop controller is used. However, a proportional-

integral controller theoretically eliminates steady-state error as a result of these disturbances. Experimental results however prove otherwise due to the nonlinear effects that are only dynamically significant when dealing with extreme positional accuracy in this system as discussed in the following section.

7 Effect of Controller on Experimental Performance

In sections 1 to 6, the steady-state error of the EHA as a result of "noise" due to flow and force disturbances was predicted. In the following section, these predictions are tested experimentally to verify their accuracy.

Figure 8 shows the experimental system response to a 10 mm step input with the EHA connected to a 20 kg horizontally-sliding load. This response is obtained using the proportional controller $G_{OL}(s) = K_p = 585$ as employed in Habibi and Singh (2000). From this graph the 0.3 s rise time and the 0.6 s settling time of the system can be observed. Figure 9 shows a magnified version of Fig. 8, demonstrating the steady-state error of 2 μm as theoretically predicted in Habibi and Goldenberg (2000). The position sensor employed is an optical encoder with 0.17 m travel, 1 μm resolution and an accuracy of ±3 μm/m. The sampling time for all the experiments was 0.001 s.

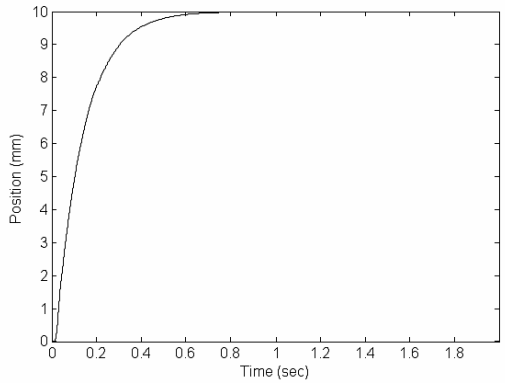

Fig. 8: *Experimental response to 10 mm step input*

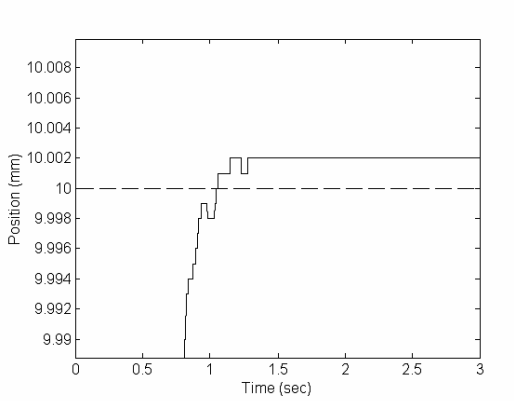

Fig. 9: *Response to step input near steady state for proportional controller*

7.1 Gain Scheduling

Since the overall time response of the system with the original proportional controller was excellent, it was desirable to maintain this overall response while reducing the steady-state error of 2 μm. Gain scheduling is a method that can be used to provide two control strategies in order to achieve this goal. When the error signal is large, the original proportional controller is employed. However, when the error signal becomes small as the actuator approaches the steady-state position, an additional control element is added. Because the error signal is very small at the time that the additional controller is added, its gain can be very high without causing saturation of the system or excessive overshoot.

Due to the fact that gain scheduling is employed for the nonlinear integral and nonlinear proportional controllers, the overall system response as shown in Fig. 8 will not change appreciably. However, the response near their steady-state accuracy level, as shown in Fig. 9 with the proportional controller, will vary.

7.2 Nonlinear Integral Controller

As predicted by Eq. 12 and 16, an integral outer-loop controller would theoretically eliminate the steady-state error present in Fig. 9. However, experience and simulations indicated that adding simple integral action to the existing controller would increase the percentage overshoot significantly. This was not acceptable, as it was desired to maintain the overall system response as shown in Fig. 8.

In order to add integral action to the controller without affecting the large-signal response, a nonlinear outer-loop integral

controller as shown in Fig. 10 was employed. This sets the input of the integrator to zero when the error signal is more than ±5 μm, and switches on the integrator when the error is less than 5 μm. This error threshold was chosen because it is slightly larger than the typical steady-state error of 2-3 μm.

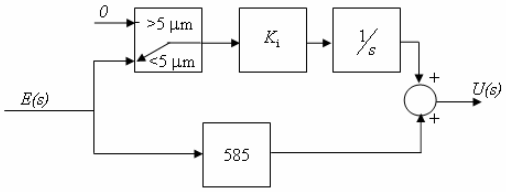

Fig. 10: *Nonlinear outer-loop integral controller*

Step input tests with a magnitude of 10 mm were performed using the nonlinear integral controller with K_i = 0, 500, 1000 and 1500. The results are shown below in Fig. 11, which has been magnified to show detail. Until approximately 3 seconds, the response of the system is similar regardless of the integral gain K_i. After this time, the response differs depending on the integral gain employed.

It can be seen that with a low integral gain of K_i = 500, the steady-state error is reduced to less than 1 μm. However, the 3.5 seconds required to achieve this state is unacceptably long. Higher integral gains reduced the amount of time required for the system to overcome static friction, but resulted in limit cycle oscillation. This is due to the nonlinear friction present in the actuator, which causes the actuator to overshoot the desired position, stick and repeat as shown in Fig. 12.

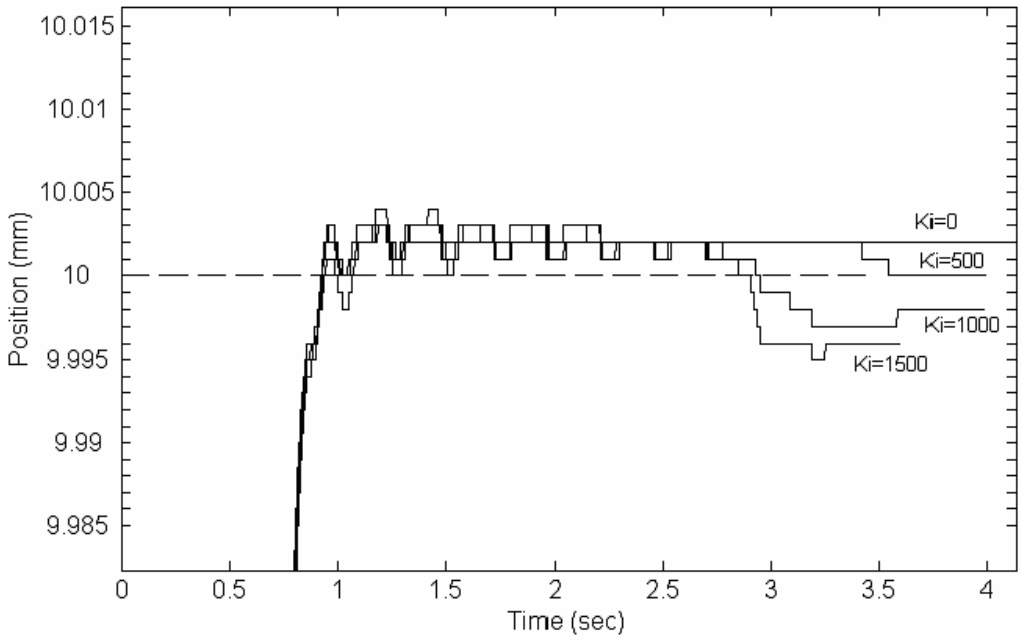

Fig. 11: *Experimental step response with varying integral gains*

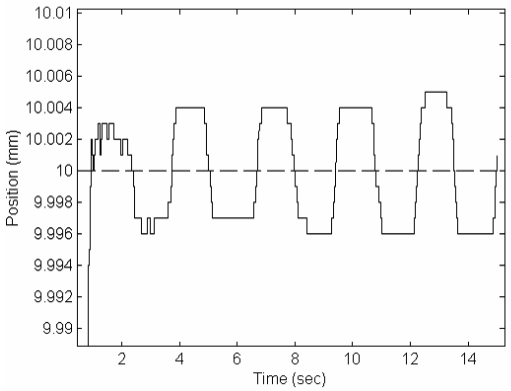

Fig. 12: *Experimental step response with integral gain $K_i = 500$*

For this reason, nonlinear integral control was found to be an ineffective method of reducing steady-state error in the EHA system. Nonlinear proportional control was the next approach to be examined.

7.3 Nonlinear Proportional Controller

A gain and phase margin analysis of the EHA model with a proportional controller was performed using MATLAB®. The original system with controller $G_{OL}(s) = K_p = 585$ has 26 dB of gain margin and 88 degrees of phase margin. It was found that K_p could be increased to 3585 while maintaining an acceptable 10 dB of gain margin and 76 degrees of phase margin, Ogata (2002).

This gain level was chosen because it was experimentally determined to be effective in decreasing the steady-state error to less than the resolution of the position sensor. To avoid altering the original response of the system, the proportional gain was split up into two parts; a fixed gain of 585 and a switched gain of 3000 that was added when the error signal was less than the ±5 μm error threshold that was discussed earlier. This nonlinear proportional controller can be seen in Fig. 13.

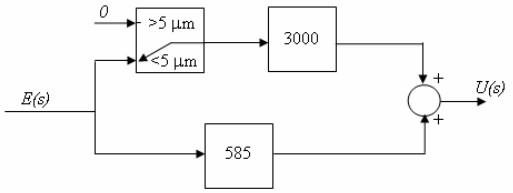

Fig. 13: *Nonlinear outer-loop proportional controller*

This nonlinear proportional control strategy of greatly increasing the stiffness of the controller when the position error becomes small was very effective with the EHA. Figure 14 shows a magnified graph of the response of the system to a 10 mm step input. It can be seen that the error becomes less than the 1 μm resolution of the optical encoder in approximately 1.5 seconds.

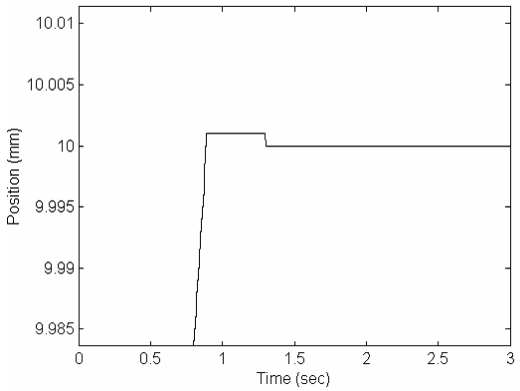

Fig. 14: *Experimental step response with nonlinear proportional controller*

The same control strategy also enables repeatable steps as small as 1 μm to be performed, as shown in Fig. 15 in response to a 1 μm step input. The output of the position sensor is discrete with a 1 μm resolution ±0.5 quantum. Therefore, the response shown in the Fig. 15 indicates only that the steady-state position is within ±0.5 μm of the desired position and does not show the details of the actuator's movement.

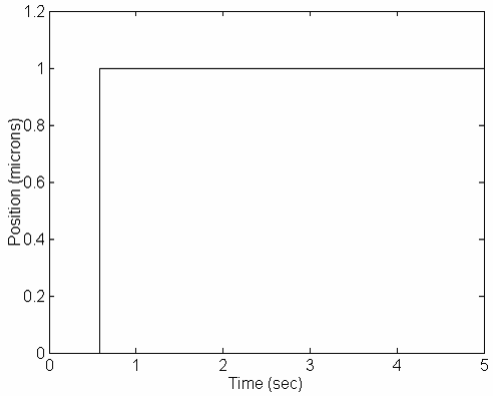

Fig. 15: *Experimental 1 μm step response*

Finally, a test was conducted to verify the nonlinear proportional controller's ability to reject force disturbances while maintaining a

constant position. The desired input $X_d(s)$ was set to be zero, and the load was struck moderately with a hammer in the axial direction three times. The impulse generated by the hammer was calculated to be approximately 40 N from knowledge of the active area of the actuator and the differential pressure across it during the impulse. This test was intended to simulate impulsive force disturbances, such as a mass being dropped onto a positioning table controlled in the vertical axis by an EHA.

Figure 16 shows the experimental response to the impulse force disturbances. It can be seen that the EHA returns to its original position, within 1 μm, in approximately 1 second. This demonstrates that the nonlinear proportional controller is indeed capable of rejecting force disturbances while maintaining the desired transient response and providing 1 μm positional accuracy.

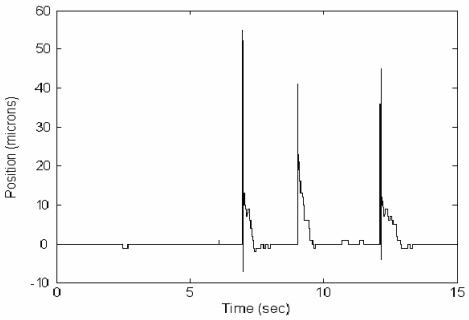

Fig. 16: *Experimental force impulse rejection response*

8 Conclusions

The ElectroHydraulic Actuator (EHA) model presented in Habibi and Singh (2000) has been expanded to include inputs for flow and force disturbances, producing a single equation that predicts the response of the EHA to both desired inputs and disturbance inputs. An analysis using this equation suggests that using the proportional outer-loop controller will result in steady-state error with constant force and flow disturbances. It also suggests that a proportional-integral outer-loop controller will eliminate the steady-state error as a result of the disturbances.

Tests were conducted to verify these predictions. Nonlinear controllers were used in an effort to retain the desirable overall response of the proportional controller while reducing its 2 μm steady-state error. The nonlinear integral controller was found to be ineffective, as low integral gains took too long to reduce the steady-state error and higher integral gains resulted in limit cycle oscillation as a result of nonlinear load friction. In contrast, a nonlinear proportional outer-loop control strategy reduced the steady-state error as a result of these disturbances to the limit of the optical encoder accuracy. Furthermore, this control strategy demonstrated the ability to reject force impulse disturbances, returning to the desired position within 1 μm while preserving the system's transient response and positional accuracy.

Using the nonlinear proportional controller, the EHA demonstrated a high level of performance. With a 20 kg load, the encoder positional accuracy of 1 μm was achieved while preserving critical damping and very satisfactory settling and rise times. Repeatable steps of 1 μm were exhibited despite the presence of nonlinear friction in the system.

With this level of performance, the EHA has applications in areas requiring precise positioning unprecedented in conventional fluid power and geared electrical actuation systems. Furthermore, the EHA is compact, modular, energy efficient, and capable of a high force output.

Nomenclature

Table 1: Nomenclature and Values

$A1 = A2 = A$	Actuator pressure area	5.05×10^{-4} m^2
B	Coefficient of friction at load	*760 N/m/s
$C1, C2, C3$	Actuator chambers	
D_p	Pump volumetric displacement	1.6925×10^{-7} m^3/rad
$E(s)$	Error signal	V
E_{ss}	Steady-state positional error	m
$F_{dis}(s)$	External force displacement	N
$G_{OL}(s)$	Outer loop controller	
$G_V(s)$	Motor/Pump subsystem transfer function	
$G_H(s)$	Hydraulic transfer function	

K_p, K_i	Controller gains	
L	Leakage coefficient	*2×10^{-15} $m^3/s/Pa$
M	Load mass	20 kg
$O1, O2, O3, O4, O5$	Actuator ports	
P_1, P_2	Actuator chamber pressure	Pa
Q_1, Q_2	Actuator chamber flow	m^3/s
$Q_{dis}(s)$	Disturbance flow	m^3/s
$Q_p(s)$	Pump flow	m^3/s
$U(s)$	Motor input voltage	V
V_o	Pipe plus mean actuator chamber volumes	6.1×10^{-5} m^3
$X(s)$	Position of actuator	m
$X_d(s)$	Demanded position of actuator	m
β_e	Effective bulk modulus of hydraulic oil	*2.1×10^8 Pa
κ_h	Hydraulic gain	
κ_v	Motor gain	40.55 rad/s/V
ω_{nh}	Hydraulic undamped natural frequency	
$\omega_p(s)$	Pump angular velocity	rad/s
ξ	Pump cross-port leakage coefficient	*1.5×10^{-13} $m^3/s/Pa$
ζ_h	Hydraulic damping ratio	
* values obtained from Chinniah (2004)		

References

Arnautovic, S. 1993. *Electrohydraulic Actuator*. Technical Report, University of Toronto.

Chinniah, Y. 2004. *Fault Detection in the Electrohydraulic Actuator Using Extended Kalman Filter*. Phd thesis, University of Saskatchewan, Canada.

Desai, J. and **Bobrow, J**. 1989. Modelling and Analysis of a High Torque Hydrostatic Actuator for Robotic Applications. *Exp. Robotics*.

Habibi, S. and **Goldenberg, A**. 1999. Design and Analysis of a Symmetrical Linear Actuator for Hydraulic Systems. *Transactions of the CSME*, Vol. 23, No. 3 & 4, pp. 377-397.

Habibi, S. and **Goldenburg, A.** 2000. Design of a New High Performance Electrohydraulic Actuator. *IEEE/ASME Transactions on Mechatronics,* Vol. 5, No. 2.

Habibi, S., Pastrakuljic, V. and **Goldenburg, A.** 2000. Model Identification and Analysis of a High Performance Hydrostatic Actuation System. *SAE paper 2000-01-2619, 2000 SAE International Off-Highway & Powerplant Congress and Exposition*.

Habibi, S. and **Singh, G.** 2000. Derivation of Design Requirements for Optimization of a High Performance Hydrostatic Actuation System. *International Journal of Fluid Power,* Vol. 1, No. 2.

Manring, N. and **Lueke, G**. 1998. Modelling and Designing a Hydrostatic Transmission with a Fixed-Displacement Motor. *Journal of Dyn. Sys. Meas. & Cont,* Vol 120, pp. 45-50.

Merritt, H. 1967. *Hydraulic Control Systems*. John Wiley & Sons.

Ogata, K. 2002. *Modern Control Engineering, 4th Ed*. Prentice Hall.

Watton, J. 1989. *Fluid Power Systems*. Prentice Hall.

APPENDIX E: BATH 2005 POWER TRANSMISSION AND MOTION CONTROL (PTMC) CONFERENCE PAPER

This section includes the paper which will be presented at the 2005 Power Transmission and Motion Control (PTMC) conference in Bath, England during September 2005. It has been accepted for publication in the conference proceedings. It contains an experimental determination of the transfer function of the EHA over a range of small-signal inputs from 0.10 to 1.0 Volts. This paper is included with the express permission of the conference proceeding's publishers.

Model Identification of the Electrohydraulic Actuator for small signal inputs

Eric Sampson †, Saeid Habibi ‡, Richard Burton ‡, and Yuvin Chinniah §
† The MathWorks, Inc.
‡ Department of Mechanical Engineering, University of Saskatchewan
§ Research Institute for Occupational Health and Safety (IRSST)

ABSTRACT

A prototype of a high-precision hydrostatic actuation system, referred to as the ElectroHydraulic Actuator (EHA), has demonstrated an unprecedented level of accuracy for large load manipulation. This prototype has been able to move an inertial load of 20 Kg with an accuracy of 1 micron, and is currently being modified for sub-micron precision operation. The aim of this paper is to characterize the EHA using an empirical model for operation in the micron and sub-micron range. Previous experimental studies on the EHA have indicated that the system is nonlinear, but can be characterized as piecewise linear. The natural frequency and damping ratio of the EHA varies in a piecewise manner depending on its operating range, which is largely determined by the magnitude of the input signal. System characterization at input levels corresponding to micro-precision movements of the load has not been previously reported and is investigated in this paper. Piecewise linear empirical models are identified in view of implementing a nonlinear fuzzy control strategy.

NOMENCLATURE

$A1 = A2 = A$	Actuator pressure area	5.05×10^{-4} m^2
B	Coefficient of friction	*760 N/m/s
$C1, C2, C3$	Actuator chambers	
D_p	Pump volumetric displacement	1.6925×10^{-7} m^3/rad
$E(s)$	Error signal	V
E_{ss}	Steady-state positional error	m
$F_{dis}(s)$	External force displacement	N
$G_{OL}(s)$	Outer loop controller	
$G_V(s)$	Motor/Pump subsystem transfer function	
$G_H(s)$	Hydraulic transfer function	
K_p, K_i	Controller gains	
L	Leakage coefficient	$*2 \times 10^{-15}$ m^3/s/Pa
M	Load mass	20 kg
$O1, O2, O3$ $O4, O5$	Actuator ports	
P_1, P_2	Actuator chamber pressure	Pa

Q_1, Q_2	Actuator chamber flow	m^3/s
$Q_{dis}(s)$	Disturbance flow	m^3/s
$Q_p(s)$	Pump flow	m^3/s
$U(s)$	Motor input voltage	V
V_o	Pipe plus mean actuator chamber volumes	6.1×10^{-5} m^3
$X(s)$	Position of actuator	m
$X_d(s)$	Demanded position of actuator	m
β_e	Effective bulk modulus of hydraulic oil	$*2.1 \times 10^8$ Pa
κ_h	Hydraulic gain	
κ_v	Motor gain	40.55 rad/s/V
ω_{nh}	Hydraulic undamped natural freq.	
$\omega_p(s)$	Pump angular velocity	rad/s
ξ	Pump cross-port leakage coefficient	$*1.5 \times 10^{-13}$ m^3/s/Pa
ζ_h	Hydraulic damping ratio	
* Values obtained from (1)		

1. INTRODUCTION

A novel hydraulic actuation system referred to as the ElectroHydraulic Actuator (EHA) has been developed to overcome the drawbacks of conventional hydrostatic systems while providing high positional accuracy, as presented by Habibi and Goldenberg (2,3). The EHA uses a bi-directional fixed-displacement pump powered by a variable-speed servomotor (4,5,6,7,8). The actuator is directly connected to the pump as shown in Figure 1. The pump's speed and direction determine the fluid flow to and from the actuator and its resulting displacement. A prototype of the EHA has demonstrated an excellent level of performance. Using a nonlinear control approach, it has proved capable of moving a 20 kg load with an accuracy of 1 micron and a critically damped rise time of 0.3 s. Earlier work has demonstrated that the behavior of the EHA is both nonlinear and dependant on its input voltage. In this paper the EHA system will be modeled for input voltages less than 1.0V, a region where nonlinearities dominate and where precision operational control is most influenced. Previous work (9) has modeled the system for input voltages greater than 1.0V.

2. EHA SYSTEM

A brief introduction to the EHA system is beneficial at this point. The EHA system consists of the following components: Controller, Electric Motor, Bi-Directional Gear Pump, Accumulator, Sensors (Pressure, Position and Speed), Crossover Relief Valve, Symmetrical Actuator, and Load. A simplified schematic of the system is shown in Figure 1.

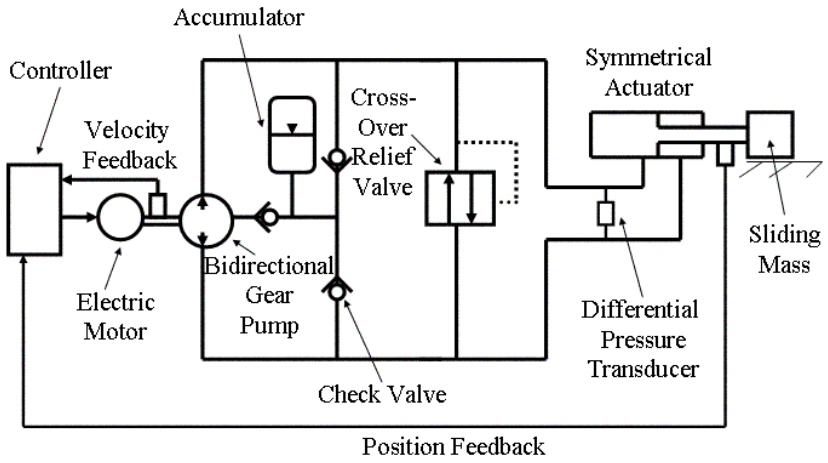

Figure 1: Schematic of ElectroHydraulic Actuator system

The three-phase brushless AC electric motor directly drives the pump. The pump in turn controls the flow of hydraulic oil to the two active chambers of the actuator. The pressure differential between the actuator chambers, which is a result of the resistance of the external load to motion, applies a net force on the load. In this case, the load is a 20 kg steel block mounted on two linear rails which is displaced horizontally by the force exerted on it by the actuator.

The symmetrical linear actuator used in the prototype has a single rod and has been designed expressly for the system, as described in (2,3). The actuator has two working chambers $C1$ and $C2$, illustrated in Figure 2. Chamber $C1$ is inside the hollow rod while chamber $C2$ is between the outside of the rod and the inside of the cylinder. Hydraulic fluid enters the two chambers of the actuator through ports $O1$ and $O2$. The working areas of the chambers $A1$ and $A2$ are made equal in area. This results in symmetrical flow, if leakage L is neglected. There also exists a third working chamber $C3$, which can be pressurized via port $O5$ to provide a bias to counteract a constant external force. Finally, ports $O3$ and $O4$ are provided to drain any actuator fluid leakage.

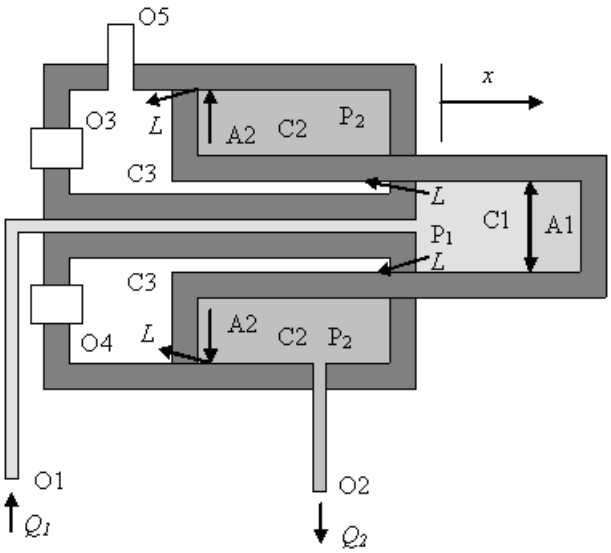

Figure 2: Cross-section of the actuator

An accumulator is connected to the low-pressure case drain of the gear pump. It prevents cavitation, and replaces fluid lost due to external leakage. The accumulator sets the minimum system pressure and can be adjusted from 2.76-6.9 Bar (40-100 psi). The crossover relief valve increases the safety of the system by preventing excessive pressure build-up if the actuator reaches the end of its travel, or if a fault were to occur. Finally, a position sensor measures the displacement of the load. The sensor, an optical linear encoder, has a resolution of 1 micron.

3. EHA MODEL

A mathematical model of the EHA system was developed in (10). This model was expanded by the author (11) to include the ability to simulate the effects of arbitrary time-varying force disturbances and flow disturbances on the EHA system. Figure 3 shows a simplified block diagram for the system showing the outer-loop position controller $G_{\mathrm{OL}}(s)$, the inner-loop electrical subsystem $G_{\mathrm{V}}(s)$, and the hydraulic subsystem $G_{\mathrm{H}}(s)$.

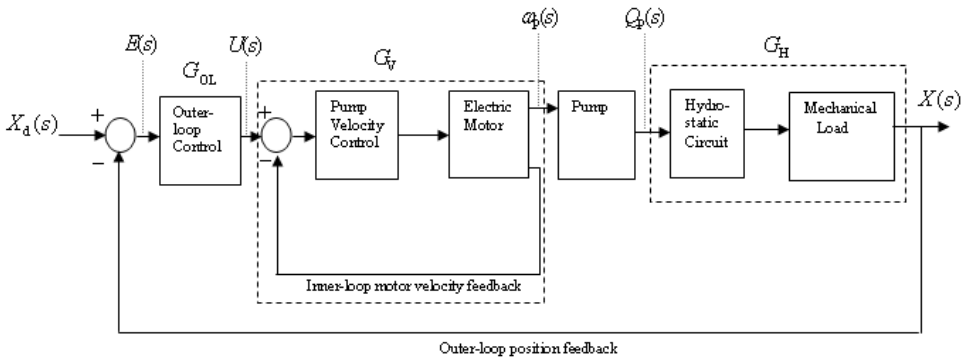

Figure 3: Simplified EHA block diagram

In (9), the transfer function of the electrical subsystem, $G_V(s)$, was identified using a constrained quadratic optimization technique. For a 5V sinusoidal input, the subsystem was found to be second-order with a zero:

$$\frac{\omega_p(s)}{U(s)} = \frac{2.779\times10^{-1}s + 40.55}{5.780\times10^{-5}s^2 + 1.061\times10^{-2}s + 1} \tag{1}$$

The analytical transfer function of the hydraulic subsystem, $G_H(s)$, presented in (10) is second-order Type 1:

$$\frac{X(s)}{\omega_p(s)} = \frac{\frac{2\beta_e A}{MV_o}D_p}{s^3 + s^2\left(\frac{B}{M} + \frac{(2\xi + L)\beta_e}{V_o}\right) + s\left(\frac{2\beta_e A^2}{MV_o} + \frac{(2\xi + L)\beta_e B}{MV_o}\right)} \tag{2}$$

Substituting known system parameters into this theoretical transfer function results in the following hydraulic transfer function:

$$\frac{X(s)}{\omega_p(s)} = \frac{29.42}{s^3 + 39.04s^2 + 8.783\times10^4 s} \tag{3}$$

Combining Equations 1 and 3, the theoretical open-loop EHA transfer function for a 5V input is as follows, in pole-zero form:

$$\frac{X(s)}{U(s)} = \frac{141400\,(s + 145.9)}{s\,(s^2 + 183.6s + 1.73\times10^4)\,(s^2 + 39.04s + 8.783\times10^4)} \tag{4}$$

4. EXPERIMENT SPECIFICATION

In order to obtain the dynamic characteristics of the system for inputs less than 1.0V, a PC running MATLAB® Simulink and containing a 12-bit DAQ system was employed. The PC was used to both generate the input signal to the external controller and record the position of the load mass. The sampling time of the system was 0.001s.

The input signal was chosen to be a constant-magnitude discretely-swept sine wave. The frequency content was 1Hz to 500Hz in steps of 1 Hz. Each frequency was repeated for two full cycles to allow for settling time. A total of 18 inputs with magnitudes of 1.0V RMS to 0.1V RMS in steps of 0.05V were used as inputs for the open-loop system identification process.

The position signal was measured using a digital-output optical encoder with 1 micron resolution. Recorded signals from a typical test with a 0.45V RMS input signal and corresponding output position are shown in Figure 4. To eliminate the drift present in the output position data, the recorded position was numerically differentiated to obtain the output velocity, as shown in Figure 5. The output velocity was then used as the system output for the identification process. Due to the resolution of the position sensor and the sample time, the resolution of the velocity signal was 0.001m/s. Identification for input voltages less than 0.1V was not feasible using the current position sensor, since the differentiated velocity signal began to approach the resolution of the sensor.

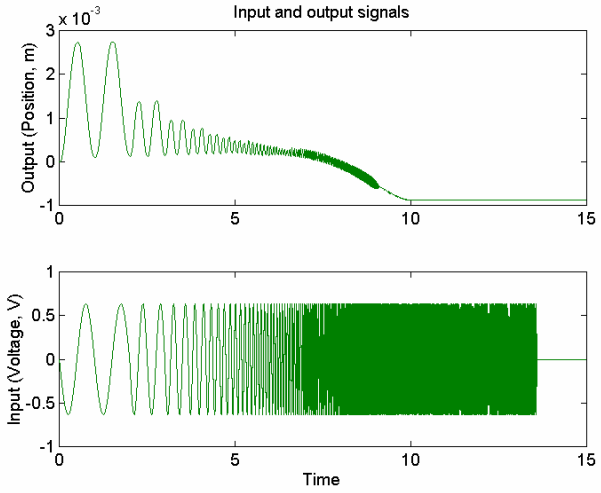

Figure 4: Output position and input voltage

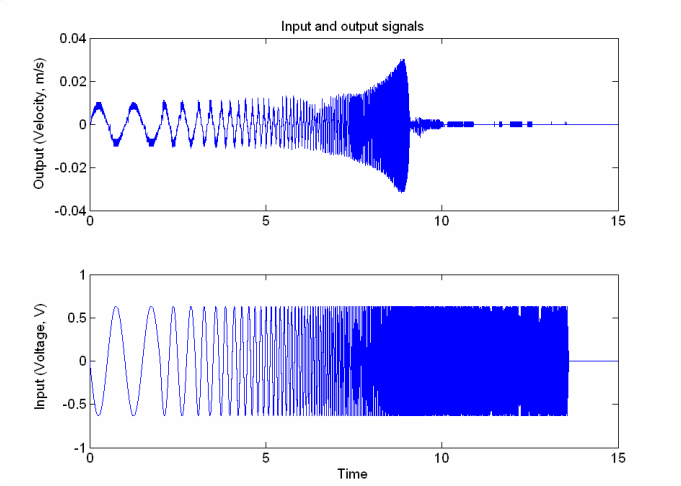

Figure 5: Output velocity and input voltage

5. INITIAL TESTS

The open-loop DC gain of the EHA system was determined for the input range -1.0V to 1.0V by setting the input to be a constant voltage, calculating the resulting load velocity and then determining the ratio of velocity/voltage. It can be seen in Figure 6 that the gain remains fixed at 0.0136m/s/V for the majority of the range tested, only changing as the input approached 0V. This demonstrates that the gain of the system was not significantly operating-point dependant over the range of interest.

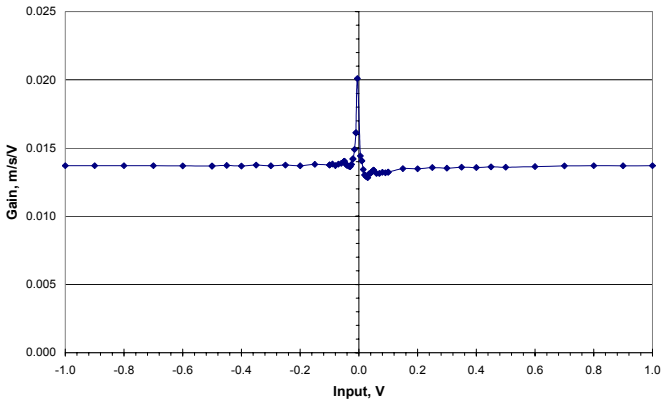

Figure 6: DC gain as a function of input voltage

To obtain the impulse response from the velocity data, a 100th order Finite Impulse Response (FIR) model was estimated using the Auto-Regression (AR) model structure (12). This produced the impulse response shown in Figure 7:

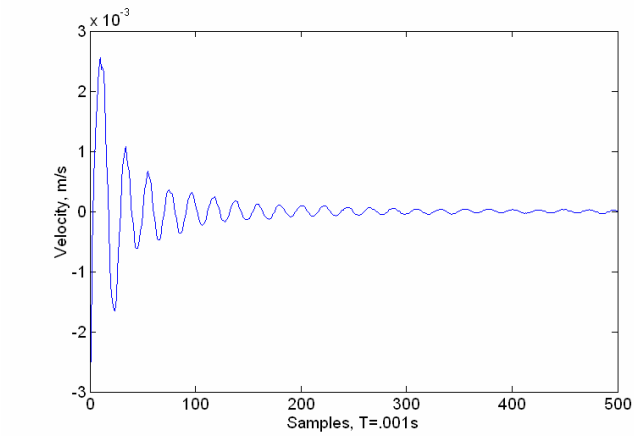

Figure 7: Computed velocity impulse response (from AR model)

The first forty coefficients of the Auto-Regression (AR) model "B" polynomial were used to construct a 20x20 Hankel matrix of Markov parameters. The singular values of the Markov parameters were then determined and plotted, as shown below in Figure 8. From inspection of this figure, it appears that the system is predominantly second order due to the prominence of the first two singular values. However, the system also appears to have some fourth order characteristics, as shown by the next two singular values being within a factor of 100 of the dominant singular values.

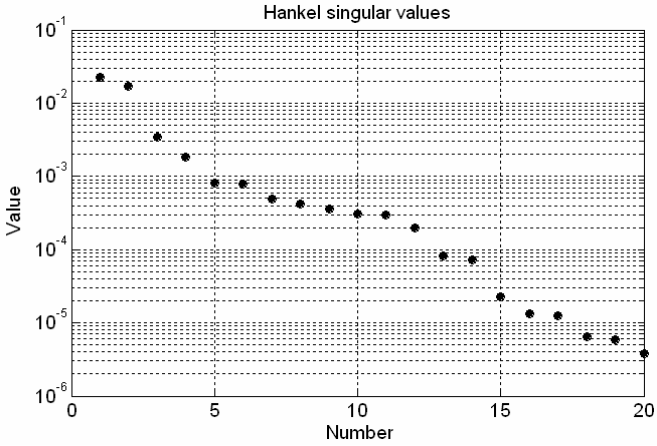

Figure 8: Hankel singular values of Markov parameters

6. DATA & PROCESSING

Once the form of the input signals was chosen and the initial tests completed, the working and validation data was obtained from the system. The working data was that used in the estimation process, while the validation data was reserved for verification of the performance of the models. The input and output signals were recorded with the input

voltage ranging from 0.1V RMS to 1.0V RMS in steps of 0.05V. The output velocity response was obtained by differentiating the recorded position data.

Figure 10 shows the working data set in a frequency-domain form. The transfer function magnitude as a function of frequency data was obtained using the Empirical Transfer Function Estimation (ETFE) technique at 512 points from 0-500Hz with a Hamming window of lag size 100 (12). Figure 10 shows the resulting Bode gain plot as a function of input voltage. Noticeable is the consistent underdamped response in the region 0.35V to 1.0V, with a bandwidth of approximately 55 Hz. Below this voltage, the system becomes overdamped and the bandwidth decreases significantly to approximately 8 Hz at 0.1V. The gain of the system is consistent at -37 dB for the majority of the input range, decreasing to -40 dB for the 0.1V input.

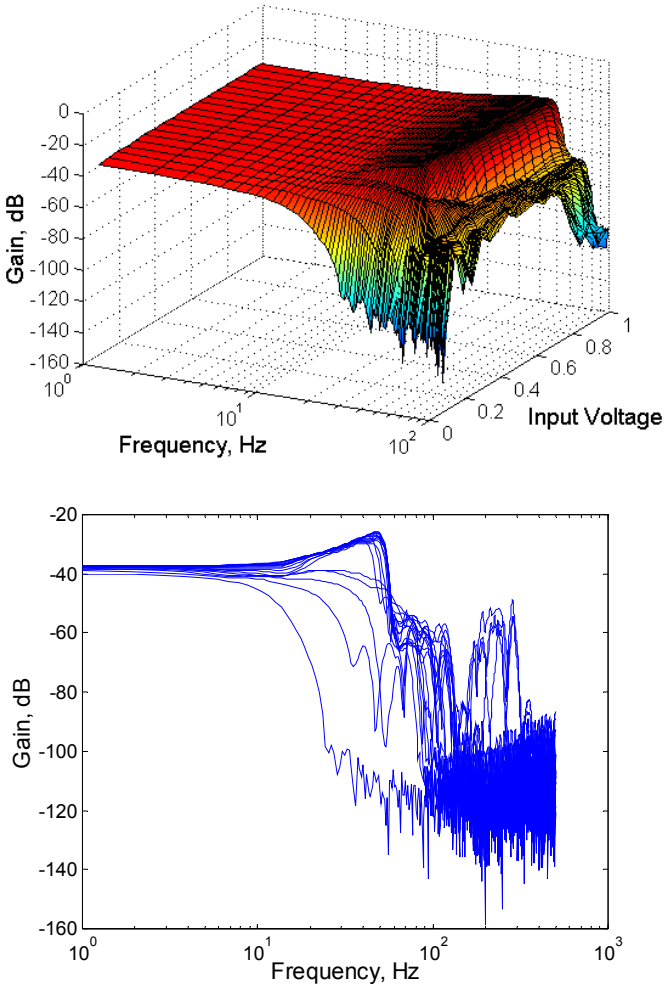

Figure 10: Experimental Bode gain plot as function of input voltage

7. MODEL IDENTIFICATION

The model identification process was performed using a frequency-domain identification methodology employing the working data shown in Figure 10. Specifically, the MATLAB® FREQID toolbox developed by de Callafon (13) was used to fit a linear time-invariant model to the frequency data recorded for the input voltage range 0.1V to 1.0V. This toolbox employs a least-squares curve fit attempting to minimize an arbitrarily weighted 2-norm between the data and the frequency response of the model. The weighting function for each of the inputs was specified in the frequency domain using the GUI tools available in the toolbox.

The model order was chosen to be fourth-order with one zero, to match the order of the theoretical system given in Eq. 4. After a model was predicted for each voltage input, dynamically insignificant poles and zeroes were manually removed. This gave the transfer functions shown in Table 1, listed in pole-zero form.

Table 1: Identified transfer functions

Input Voltage	Transfer Function $\frac{\dot{X}(s)}{U(s)}$
1.0-0.35	$\frac{187200(s+262.3)}{(s^2+192.2s+3.815\times10^4)(s^2+70.04s+9.532\times10^4)}$
0.30	$\frac{242.0(s+87.55)}{(s+51.77)(s^2+117.6s+3.364\times10^4)}$
0.25	$93760/[(s+141.3)(s^2+136.5s+5.815\times10^4)]$
0.20	$\frac{240.1(s+55.22)}{(s+38.61)(s^2+207.7s+3.009\times10^4)}$
0.15	$0.3948(s+374)/[(s+177.4)(s+78.55)]$
0.10	$0.4064(s+7)/[(s+37.91)(s+9.237)]$

Figure 11 shows the computed Bode gain plot for the identified models over the range of interest. In comparison with the experimental plot in Figure 10, it can be seen that the 0.35V to 1.0V range is matched quite well. Inspecting the experimental Bode plot shown in Figure 10, it appears that the system's output decreases, or rolls off, at approximately 80 dB/decade. As the input voltage decreases below the 0.35V level, the ability of the identification method to match the roll off characteristic of the system becomes impaired. This is likely due to an increase in the noise content of the data associated with the lower input levels.

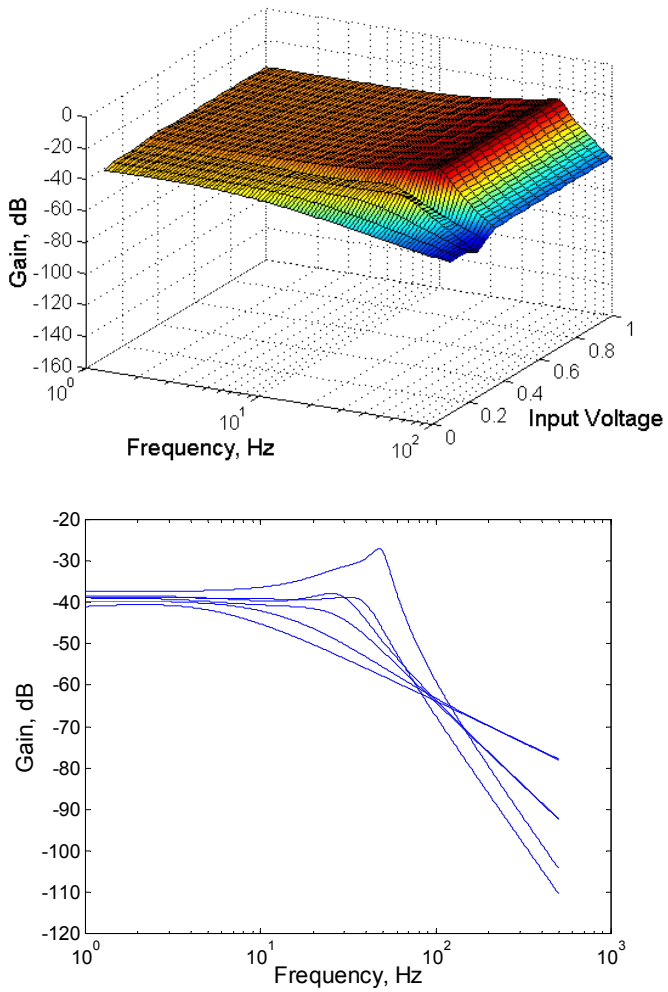

Figure 11: Identified Bode gain plot as a function of input voltage

8. VALIDATION

Once models were identified for each input voltage, the response of the model to the input data was simulated for validation purposes. The validation data set mentioned in the Data & Processing section was plotted on the same graphs as the simulated response in order to visually verify the validity of the models. Figures 12-15 show the simulated and experimental response to the discrete swept sine wave from 1-500Hz. The 'envelope' of the simulated response is overlaid on the experimental response as a black line in order to facilitate comparison between the simulation and the experimental responses. The agreement is generally very good up to a minimum of 10 Hz (in the 0.10V case) and to a maximum of 50 Hz (in the 0.60V case). It can be seen in all cases that the fit is not exact at

low frequencies. This is due to the optimization method employed in the FREQID toolbox, which gives an equal weight to each data point. Alternatively, decreasing the weights logarithmically as the frequency increases would compensate for the increasing number of data points per decade, as is discussed in (12). Furthermore, the accuracy of the identified models decreases noticeably as the input voltage decreases towards 0.1V. This may be due to the resolution of the velocity signal, which is obtained by differentiating the position signal. For example, the peak magnitude of the velocity signal for the 0.1V input is approximately 0.002 m/s, which is comparable to the sensor resolution of 0.001 m/s.

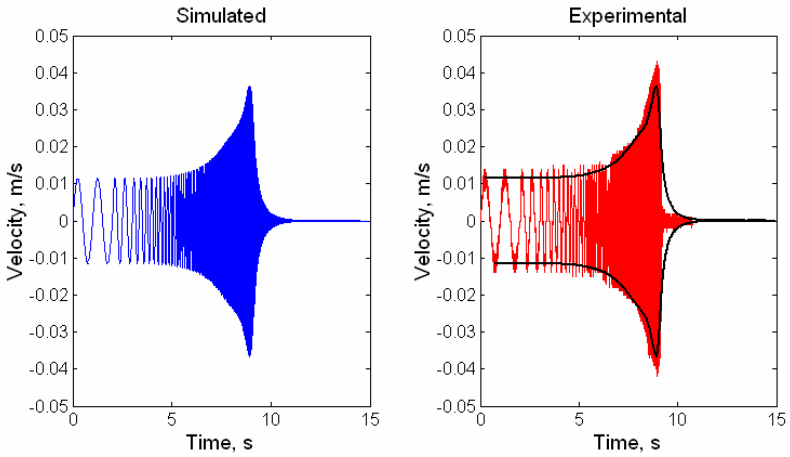

Figure 12: **Comparison of simulated and experimental response for 0.60V input (typical of 1.0V-0.35V range)**

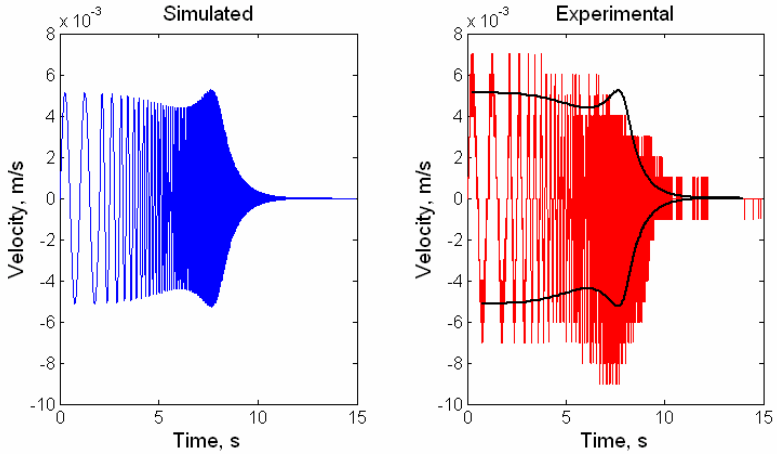

Figure 13: Comparison of simulated and experimental response for 0.30V input

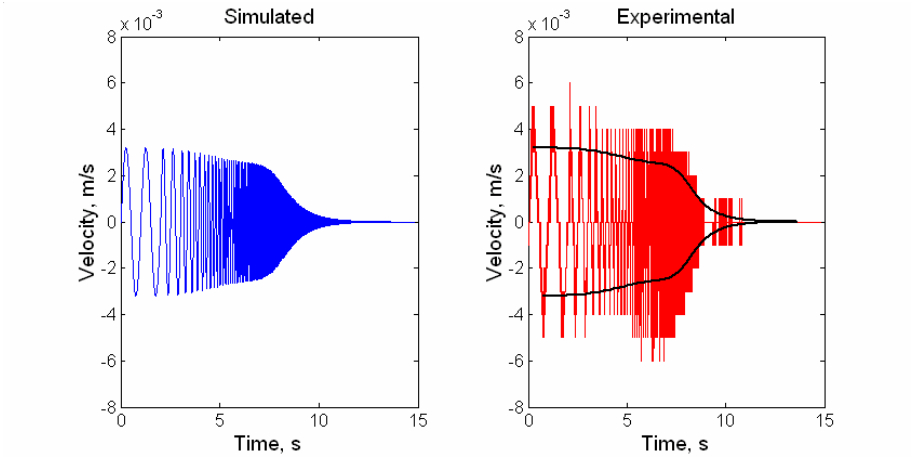

Figure 14: Comparison of simulated and experimental response for 0.20V input

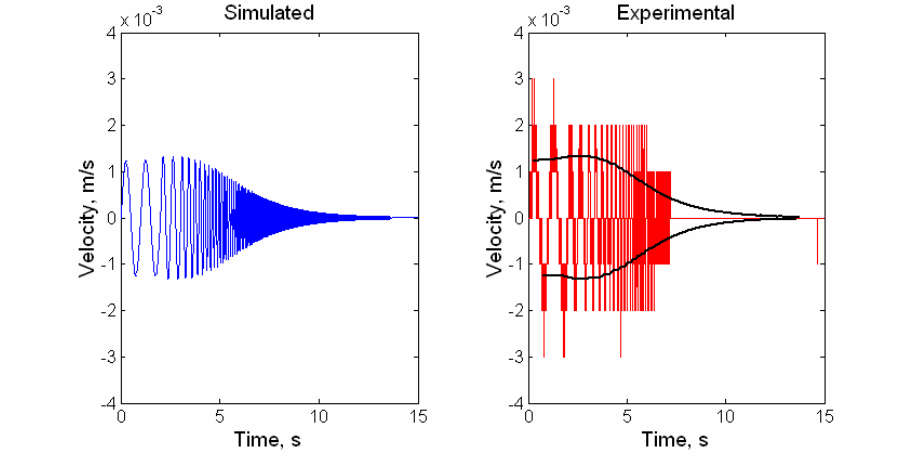

Figure 15: Comparison of simulated and experimental response for 0.10V input

Finally, Figure 16 shows the closed-loop experimental and simulated response to a 0.01m step input. The model simulated in this case was the one identified for the 0.35V to 1.0V range. In both cases, a proportional controller $G_{ol} = K_p = 585$ as employed in (10) was used. Excellent agreement between the simulated and experimental response is demonstrated, neglecting the time delay of 0.04 seconds that was not included in the system models.

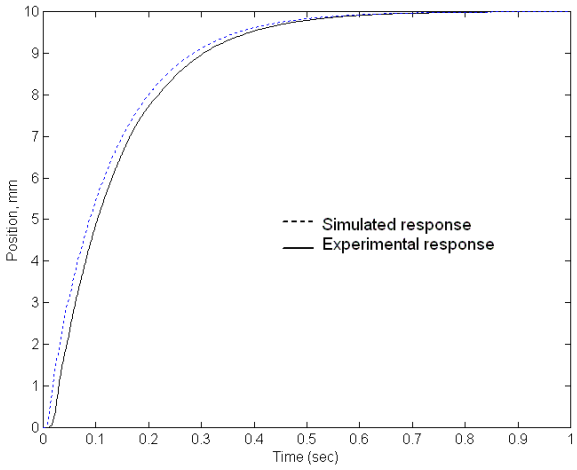

Figure 16: Simulated and experimental closed-loop step response

9. CONCLUDING STATEMENTS

The ElectroHydraulic Actuator (EHA) is a high-performance positioning system that employs a novel symmetrical linear actuator. As such, it has unique characteristics which were investigated in this paper for the input voltage range of 0.1V to 1.0V RMS. It has been found that the system exhibits consistent performance in the 0.35V to 1.0V RMS input range, with an underdamped response having a bandwidth of 55 Hz and a gain of -37 dB. As the input voltage decreases, the bandwidth becomes significantly lower and system becomes overdamped. At a 0.1V RMS input, the bandwidth has decreased to 8 Hz and the gain to -40dB. As such, a linear controller will not be able to achieve optimal control over the tested range, since the performance characteristics vary appreciably with the input voltage. A control scheme which can reflect the nonlinear damping ratio, gain, and natural frequency in the region of less than 1V input and can allow a smooth transition between the identified regions is necessary for nanometer precision control. For this reason, nonlinear control of the EHA system using a Fuzzy controller and other nonlinear controllers are now being investigated.

REFERENCES

(1) Chinniah, Y. 2004. *Fault Detection in the Electrohydraulic Actuator Using Extended Kalman Filter*. PhD thesis. University of Saskatchewan.

(2) Habibi, S. and Goldenberg, A. 1999. Design and Analysis of a Symmetrical Linear Actuator for Hydraulic Systems. *Transactions of the CSME*, Vol. 23, No. 3 & 4, pp. 377-397.

(3) Habibi, S. and Goldenburg, A. 2000. Design of a New High Performance Electrohydraulic Actuator. *IEEE/ASME Transactions on Mechatronics,* Vol. 5, No. 2.

(4) Arnautovic, S. 1993. *Electrohydraulic Actuator*. Technical Report. University of Toronto

(5) Desai, J. and Bobrow, J. 1989. Modelling and Analysis of a High Torque Hydrostatic Actuator for Robotic Applications, *Exp. Robotics*

(6) Manring, N. and Lueke, G. 1998. Modelling and Designing a Hydrostatic Transmission with a Fixed-Displacement Motor, *Journal of Dyn. Sys. Meas. & Cont.* Vol 120, pp. 45-50.

(7) Merritt, H. 1967. *Hydraulic Control Systems*. John Wiley & Sons.

(8) Watton, J. 1989. *Fluid Power Systems.* Prentice Hall.

(9) Habibi, S., Pastrakuljic, V. and Goldenburg, A. 2000. Model Identification and Analysis of a High Performance Hydrostatic Actuation System, *SAE paper 2000-01-2619, 2000 SAE International Off-Highway & Powerplant Congress and Exposition*.

(10) Habibi, S. and Singh,G. 2000. Derivation of Design Requirements for Optimization of a High Performance Hydrostatic Actuation System, *International Journal of Fluid Power,* Vol.1, No. 2.

(11) Sampson, E., Habibi, S., Burton, R. and Chinniah, Y. 2004. Effect of Controller in Reducing Steady-State Error due to Flow and Force Disturbances in the ElectroHydraulic Actuator System, *International Journal of Fluid Power*, Vol. 5, No. 2

(12) Ljung, L. 1999. *System Identification – Theory for the User, 2nd Ed.* Prentice Hall PTR.

(13) de Callafon, R. A. and Van de Hof, P. M. J. "Freqid -- frequency domain identification toolbox for use with Matlab*," Sel. Top. Identification and. Modeling. Control*., Vol. 9, pp. 129--134, 1996.

ACKNOWLEDGMENTS

The authors acknowledge the financial support of the National Science and Engineering Research Council of Canada and the Department of Mechanical Engineering, University of Saskatchewan during this study.