Pneumatic Muscle Actuator Control

Final Report AFOSR Grant F49620-00-1-0300

John H. Lilly (PI)

Department of Electrical and Computer Engineering
University of Louisville
Louisville, KY

February 2004

AFRL-SR-AR-TR-04-

0113

REPORT DOCUMENTATION PAGE

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	4 Feb 2004	Final 1 May 2001 - 31 Aug 2003

4. TITLE AND SUBTITLE

Pneumatic Muscle Actuator Control

5. FUNDING NUMBERS

F49620-00-1-0300

6. AUTHOR(S)

John H. Lilly (PI)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Louisville

8. PERFORMING ORGANIZATION REPORT NUMBER

G000071FINAL

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

AFOSR
4015 Wilson Blvd. Rm 713
Arlington, VA 22203-1954

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release, distribution unlimited

Approved for public release, distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 Words)*

This research is concerned with investigating methods for the control of McKibben pneumatic actuators, or *pneumatic muscles* (PMs). PMs are a novel type of actuator that closely mimic human skeletal muscles in size and power capabilities. PMs are considered by the Air Force for use in exoskeletons to be worn by humans for strength augmentation and for use as actuators in robotic systems. In this research, we investigate adaptive, sliding mode, and soft computing approaches to control of PMs and robotic systems actuated by PMs. The soft computing approaches include neuro-fuzzy modeling of an actual PM in the Human Effectiveness Lab at Wright Patterson Air Force Base, and evolutionary design of a fuzzy PID controller based on this model. We also investigate a type of MIMO fuzzy model predictive control for a planar arm actuated by four PMs. Some of the controllers are tested on the actual PM at WPAFB while others are proven in simulations. A byproduct of this research is an evolutionary fuzzy training algorithm useful for identification of dynamical systems as well as classification problems.

14. SUBJECT TERMS

Nonlinear control, pneumatic actuators

15. NUMBER OF PAGES

154

16. PRICE CODE

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

Pneumatic Muscle Actuator Control

Final Report AFOSR Grant F49620-00-1-0300

John H. Lilly (PI)

Department of Electrical and Computer Engineering
University of Louisville
Louisville, KY

February 2004

Contents

List of Figures

List of Tables

Summary

This research is concerned with investigating methods for the control of McKibben pneumatic actuators, or *pneumatic muscles* (PMs). PMs are a novel type of actuator that closely mimic human skeletal muscles in size and power capabilities. PMs are considered by the Air Force for use in exoskeletons to be worn by humans for strength augmentation and rehabilitation after injury or illness, and for use as actuators in robotic systems. The control of PMs is a challenging problem due to their highly nonlinear and time-varying nature. In this research, we investigate adaptive, sliding mode, and soft computing approaches to control of PMs and robotic systems actuated by PMs. The soft computing approaches include neuro-fuzzy modeling of an actual PM in the Human Effectiveness Lab at Wright Patterson Air Force Base, and evolutionary design of a fuzzy PID controller based on this model. We also investigate a type of MIMO fuzzy model predictive control for a planar arm actuated by four PMs. Some of the controllers are tested on the actual PM at WPAFB while others are proven in simulations. A byproduct of this research is an evolutionary fuzzy training algorithm useful for identification of dynamical systems as well as classification problems.

1 Objectives

1.1 Introduction

McKibben pneumatic actuators, or pneumatic muscles (PM), are a novel type of pneumatic actuator consisting of a cylindrical, flexible rubber or plastic airtight tube inside a braided plastic sheath (see Figure 2.1). When the tube is inflated it widens and, due to the braided sheath, shortens. When it shortens, the contractile force exerted is quite large in proportion to the PM's weight. Pneumatic muscles have the highest power/weight ratio (1kW/kg [1]) and power/volume ratio (1W/cm^3 [2]) of any actuator. They are roughly the same as human skeletal muscles in size, shape, and power output. A significant advantage of PM actuators is the ability to make them autonomous. They can be energized from a small canister of gas that can rapidly create, from a chemical reaction, large pressures for inflation of the muscle.

PMs have been used for years in robotics to perform precision manipulation tasks ([3]-[11]). They can also be used to actuate an exoskeleton frame worn by humans to enhance strength and/or mobility. Concepts developed from our research can be used to help the disabled obtain enhanced strength and mobility. Such people have suffered from stroke, accidents, or other problems to reduce their physical capabilities.

In order for PMs to be used for precision robotics or for exoskeleton actuation, it is necessary to be able to control them precisely. Since they are highly nonlinear and time varying, their control is a challenging problem. Our research studies the closed-loop control of PM systems for accurate position control. Position control for PMs refers to the control of their length under varying loads. When actuating robots or exoskeletons, this translates into accurate control of joint angles or end-effector spatial positioning.

There have been several investigations into applications of PMs and their properties ([6]-[16]). Pneumatic muscle research is ongoing at the Human Sensory Feedback (HSF) Laboratory at Wright Patterson Air Force Base (WPAFB) ([6], [7]). The HSF Lab contains a PM test station that consists of several PMs, sensors, actuators, and instrumentation to control the PMs' operation. In our research, we have interacted to a great extent with the personnel at WPAFB, testing our controllers on the PMs in their lab. We are indebted to the personnel in this lab for their help in performing our research.

1.2 Objectives and Summary of Research on This Project

In this research, we are concerned with PM position control when used in configurations common to anthropomorphic robotic systems. Since the PM is nonlinear and time-varying, all controllers considered in this research were chosen because they are known to exhibit robustness to parameter uncertainties. The progression of the research on this project is as follows.

Chapter 2: As a first step, we are concerned with obtaining accurate mathematical models for the PM. In this project we have utilized two similar models, both of which were derived by the researchers at WPAFB. The models are second-order differential equations with nonlinear coefficients. These models have aided us immeasurably in our derivation of controllers and in simulating the closed-loop behavior of the PMs.

Chapter 3: We consider nonlinear adaptive control of PMs actuating a planar arm. The PMs act individually in bicep and tricep configurations. The control objective is elbow angle tracking. Simulations of the controlled system are presented.

Chapter 4: We consider single-input sliding mode control of a planar arm actuated by two PMs acting simultaneously in opposing pair configuration, similar to human bicep/tricep interaction. The control objective is elbow angle tracking. Simulation results are presented.

Chapter 5: We consider two-input sliding mode control of a planar arm actuated by four PM groups acting simultaneously in opposing pair configuration. The actuator system consists of two agonist/antagonist pairs – one pair actuating the shoulder, and another pair actuating the elbow. The control objective is spatial end-effector (hand) tracking. Simulation results are presented.

Chapter 6: We derive a recurrent neuro-fuzzy model of a single PM hanging vertically actuating a mass. The model is derived from data taken in the HSF lab at WPAFB. This model is used in Chapter 9 to derive a very effective fuzzy controller for the PM in the HSF lab.

Chapter 7: We present a new method for fuzzy classification that is a byproduct of our research on this project. The method utilizes the VISIT algorithm (pioneered by the PI and coworkers) and proposes an evolutionary method of tuning the parameters for optimal learning.

Chapter 8: We apply the Fuzzy Model Reference Learning Control (FMRLC) methodology to PM control. Results from applying the controller to the PM in the HSF lab at WPAFB are presented.

Chapter 9: We derive a fuzzy P+ID controller for the PM in the HSF lab at WPAFB. The neuro-fuzzy PM model derived in Chapter 6 is utilized, and evolutionary tuning of controller parameters is implemented. Experimental results are presented.

Chapter 10: We derive a fuzzy model predictive controller for a planar arm actuated by four PM groups in opposing pairs, as in Chapter 5. Simulation results are presented.

2 PM Differential Equation Models used in this Research

2.1 Introduction

Attempts have been made to model PMs in various ways, including fuzzy systems and neural networks. In order to design adaptive and sliding mode controllers, a differential equation model of the PM is necessary. Obtaining a DE model of the PM from first principles of physics is difficult, due to the way the PM is constructed (see Figure 2.1).

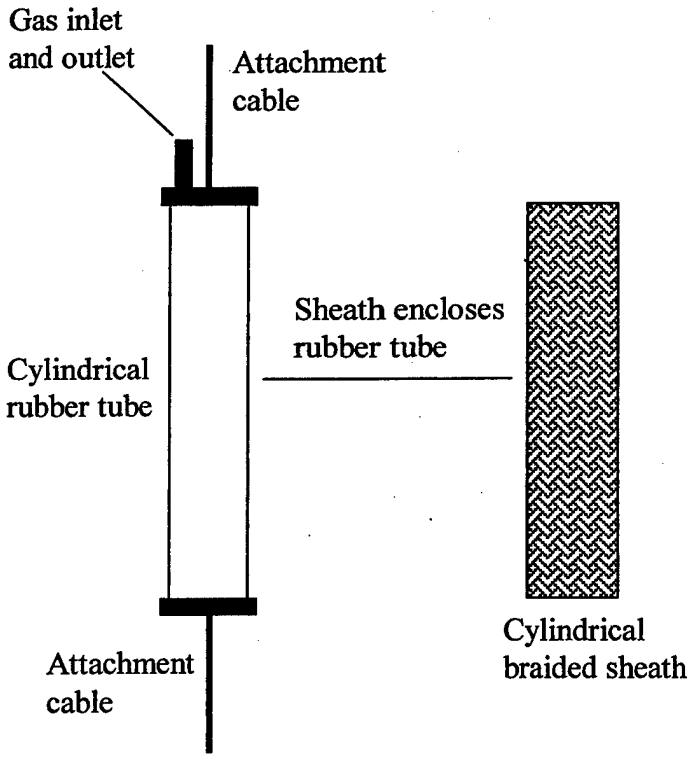


Figure 2.1 - Construction of pneumatic muscle

Alternatively, a DE model can be derived for a particular PM from experimentation in the lab. The Human Sensory Feedback laboratory at WPAFB contains an experimental PM setup that can be used to take sufficient measurements to derive models for particular PMs. This has been done in two separate studies ([7], [17]), both resulting in nonlinear second-order ODE models. Historically, the first of these was done by Repperger et al. [7]. In this model, the coefficients are calculated as nonlinear functions of the PM length. Recently, a similar model, due to Reynolds et al. [17] was derived in which the coefficients depend nonlinearly on the PM internal pressure. In both models,

the coefficients depend on whether the PM is being inflated or deflated. Both models assume an equivalent structure for the PM of a parallel connection of a nonlinear spring, a nonlinear viscous friction, and a contractile element (Figure 2.2).

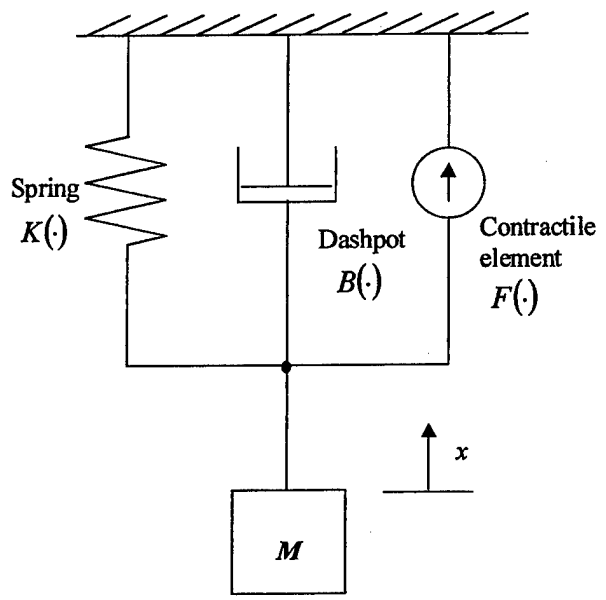


Figure 2.2 – Three-element model for PM

2.2 Repperger Model [7]

The particular PM modeled in [7, 17] has an inner bladder made from a section of 22.2 mm diameter bicycle tubing enclosed in a helically-wound nylon sheath used for supporting electrical cables. The unstretched, uncompressed diameter of the sheath is 31.75 mm. In the Repperger model, the coefficient $K(x)$ is a nonlinear function of the PM length x, and $B(\dot{x})$ is a nonlinear function of the PM rate of change of length. The muscle is inflated (hence shortened) by opening a solenoid that controls the flow of pressurized gas into the rubber bladder. It is deflated by opening another solenoid venting the contents of the bladder to the atmosphere.

Figure 2.3 shows a pneumatic muscle being inflated and lifting a mass. Let the position of the mass when the PM is uninflated be defined as $x=0$. If $x(t)$ is the vertical position of the mass, the differential equation describing the system of Figure 2.3 is

$$M\ddot{x}+B(\dot{x})\dot{x}+K(x)x=F-Mg \qquad (2.2.1)$$

where g is the acceleration of gravity. The coefficients $B(\dot{x})$ and $K(x)$ depend on whether

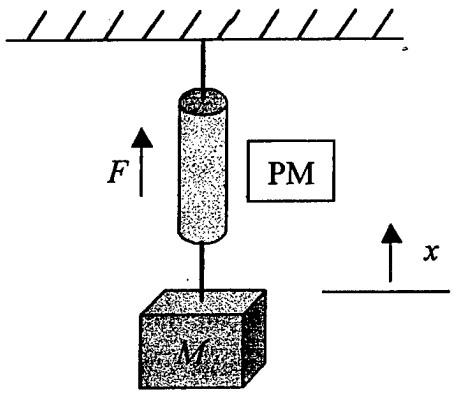


Figure 2.3 – PM hanging vertically lifting a mass

the PM is being inflated or deflated and are defined as [7]

Inflation:
$$\begin{cases} B_i(\dot{x}) = 0.04\dot{x}^2 + 1.3\dot{x} + 12.6 \\ K_i(x) = 1.6x^2 + 10.9x + 27.1 \end{cases} \tag{2.2.2}$$

Deflation:
$$\begin{cases} B_d(\dot{x}) = 0.12\dot{x}^2 + 2.49\dot{x} + 14.48 \\ K_d(x) = 3.6x^2 + 20.7x + 47.23 \end{cases} \tag{2.2.3}$$

In (2.2.1), the system input F is the upward force exerted on the mass by the PM. It is an independent control variable that can be externally commanded by adjusting the PM internal pressure. The force exerted by the viscous friction action of the PM is given by $-B(\dot{x})\dot{x}$, and the force due to the spring action of the PM is given by $-K(x)x$.

2.3 Reynolds Model [17]

The PM is again modeled by the 3-element mechanical model shown in Fig. 2.2, consisting of a contractile element, dashpot, and spring. In the Reynolds model, these three elements all have pressure-dependent coefficients. It is shown in [17] that the system of Figure 2.3 can be modeled as

$$M\ddot{x} + B(P)\dot{x} + K(P)x = F(P) - Mg \tag{2.3.1}$$

where M is the mass lifted by the PM, B is the coefficient of viscous friction, K is the spring coefficient, F is the force exerted by the contractile element, and g is the acceleration of the gravity.

According to [17], B, K and F for the specific PM considered are dependent on the internal pressure of the PM and are given in SI units as:

$$F = 179.2 + 1.39P \tag{2.3.2a}$$

$$K = 5.71 + 0.0307P \tag{2.3.2b}$$

$$B = \begin{cases} 1.01 + 0.00691P \text{ (Inflation)} \\ 0.6 - 0.000803P \text{ (Deflation)} \end{cases} \tag{2.3.2c}$$

where P is the PM supply pressure in kPa. The coefficients specified in (2.3.2) are valid in the range $206.844 < P < 620.532$ kPa ($30 < P < 90$ psi). Note that the coefficients B in deflation are smaller than in inflation. The reason is because in deflation the PM system vents against a constant atmospheric pressure. During inflation, however, the pressure buildup is in a closed volume and the forcing function has to fight against the increasing PM internal pressure as the PM inflates.

Note from (2.3.2) that the contractile force F, viscous damping coefficient B and spring coefficient K are functions of P, which is the control variable, i.e. P is the independent variable that can be commanded by the controller. Therefore, in this model the control variable enters the equations of motion via the coefficients B, K, and F, a situation that is different from most conventional control problems.

3 Adaptive Control of a Planar Arm Actuated by PMs in Bicep and Tricep Configurations

3.1 Introduction

In the adaptive control approach ([18], [19]), we utilize the Repperger model of the PM (2.2.1–2.2.3) [7]. In nonlinear adaptive control, parts of the model are assumed known, and their multiplying coefficients are treated as unknown, with adaptive laws for driving the parameter estimates to their true values. The strengths of the adaptive approach are robustness to parameter uncertainties together with ease of implementation. A weakness is the necessity of knowing the general form of the plant model.

3.2 Dynamic Modeling of Limbs with PM in Bicep and Tricep Positions

The two basic configurations in which the PM can be arranged for use in exoskeletons are the bicep-type (Figure 3.1) and tricep-type (Figure 3.2) configurations. In this chapter, the control problem for both configurations is precise control of the joint angle of a limb which is holding a mass. Specifically, we wish to actuate the PM by inflating and deflating it in such a way that the joint angle follows a reference function of time while the limb holds a mass.

PM in Bicep Configuration

Consider the limb configuration shown in Figure 3.1, which depicts an arm lifting a mass, with the PM in the position of a bicep. The upper arm remains stationary as the PM expands and contracts, moving the forearm. The upper end of the PM and upper arm are attached to a motionless reference point. The mass M is held at the end of the forearm. The forearm, which is considered massless, is attached to the upper arm by a frictionless joint. The PM is attached to the forearm at point A, which is a distance a from the joint. The distance from the center of mass of the load to the joint is L. The forearm is free to rotate through an angle θ, where $\theta = 0^\circ$ corresponds to the arm being fully bent, i.e. the mass in the extreme upward position, and $\theta = 180^\circ$ corresponds to the arm being fully straightened, i.e. the mass in the extreme downward position. For simplicity, we will assume the PM force always acts parallel to the forearm. This assumption is valid so long as θ is not close to either of its extremes.

Since the upward force exerted by the PM on the forearm at point A is $F + B(\dot{x})\dot{x} + K(x)x$, the clockwise torque imparted to the forearm by the PM is $(F + B(\dot{x})\dot{x} + K(x)x)a\sin\theta$. Therefore, the system dynamics are described by:

$$-I\ddot{\theta} = (F + B(\dot{x})\dot{x} + K(x)x)a\sin\theta - MgL\sin\theta \qquad (3.2.1)$$

where $I = ML^2$ is the moment of inertia of the mass about the joint, g is the acceleration

of gravity, and $MgL\sin\theta$ is the counterclockwise torque imparted to the forearm by gravity. Then, using $x = -a(1+\cos\theta)$ and $\dot{x} = a\dot{\theta}\sin\theta$, we can rewrite (3.2.1) entirely in terms of θ as follows:

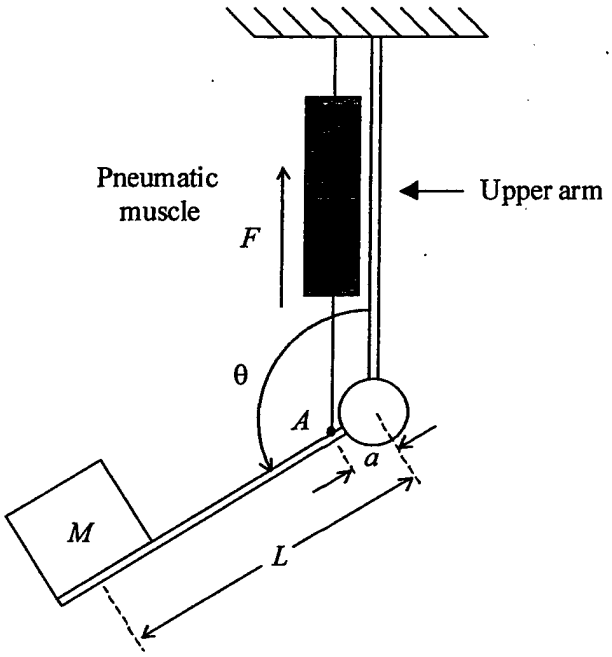


Figure 3.1 – Planar arm with PM in bicep position actuating elbow joint

$$I\ddot{\theta} = -Ba^2\dot{\theta}\sin^2\theta + Ka^2\sin\theta(1+\cos\theta) + MgL\sin\theta - Fa\sin\theta \qquad (3.2.2)$$

where B and K are now expressed in terms of θ and $\dot{\theta}$.

The external input to the system is F, which is determined by how much the PM is inflated. Note that since F is multiplied by $\sin\theta$ in (3.2.2), the system becomes uncontrollable at $\theta = 0$ and at $\theta = 180°$. For this reason, joint angles should not approach these limits. We will see that the tricep configuration does not have this restriction.

PM in Tricep Configuration

Figure 3.2 depicts an arm lifting a mass, with the PM in the position of a tricep. The upper arm remains stationary as the PM expands and contracts, moving the forearm. The lower end of the PM and upper arm are attached to a motionless reference point. The mass M is held at the end of the forearm. The forearm, which is considered massless, is

attached to the upper arm by a frictionless joint. Also at the joint is a frictionless pulley of radius r, over which a cable connecting the PM to the forearm passes. The PM is attached to the forearm at point A, which is a distance c from the joint. The cable makes an angle $\alpha = \sin^{-1}(r/c)$ with the forearm. The distance from the center of mass of the load to the joint is L. The forearm is free to rotate through an angle θ, where $\theta = 0°$ corresponds to the arm being fully straightened, i.e. the mass in the extreme upward position, and $\theta = 180°$ corresponds to the arm being fully bent, i.e. the mass in the extreme downward position.

Since the downward force exerted by the PM is $F + B(\dot{x})\dot{x} + K(x)x$, the clockwise torque imparted to the forearm by the PM is $(F + B(\dot{x})\dot{x} + K(x)x)c\sin\alpha$. Therefore, the system dynamics are described by:

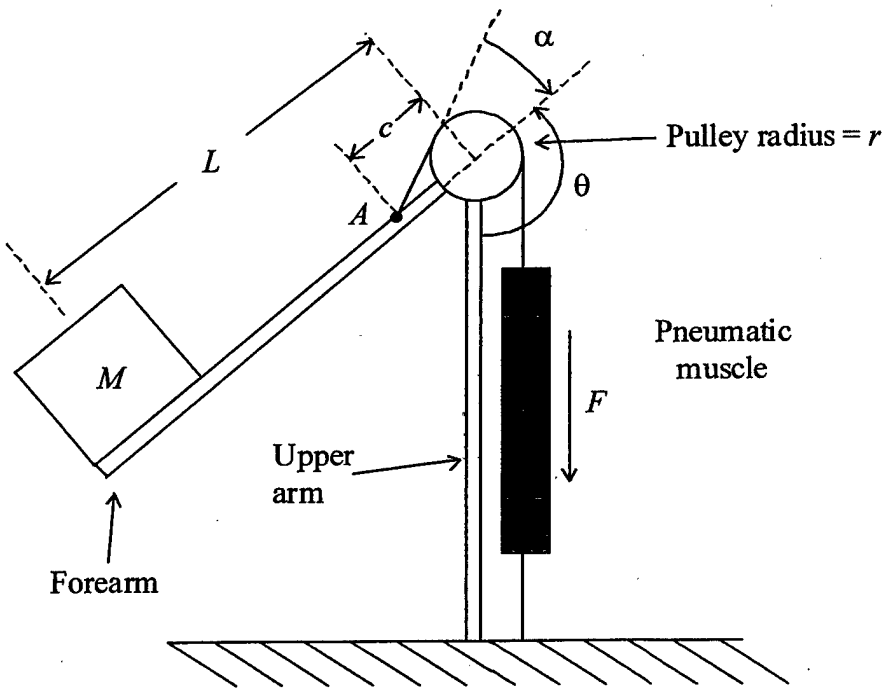


Figure 3.2 – Planar arm with PM in tricep position actuating elbow joint

$$\begin{aligned} -I\ddot{\theta} &= (F + B(\dot{x})\dot{x} + K(x)x)c\sin\alpha - MgL\sin\theta \\ &= (F + B(\dot{x})\dot{x} + K(x)x)r - MgL\sin\theta \end{aligned} \tag{3.2.3}$$

where $I = ML^2$ is the moment of inertia of the mass about the joint, g is the acceleration of gravity, and $MgL\sin\theta$ is the counterclockwise torque imparted to the forearm by gravity. Then, using $x = -\frac{1}{2}\pi r(1+\cos\theta)$ and $\dot{x} = \frac{1}{2}\pi r\dot{\theta}\sin\theta$, we can rewrite (3.2.3)

entirely in terms of θ as follows:

$$I\ddot{\theta} = -\frac{B\pi r^2}{2}\dot{\theta}\sin\theta + \frac{K\pi r^2}{2}(1+\cos\theta) + MgL\sin\theta - rF \qquad (3.2.4)$$

where B and K are now expressed in terms of θ and $\dot{\theta}$. Note that the system with PM in tricep position is controllable for all θ because the force exerted by the PM always acts at an angle α to the forearm regardless of joint angle.

3.3 Adaptive Tracking for Limbs with PM in Bicep and Tricep Positions

The mass M manipulated by the PM can be expected to vary significantly from use to use. Also, the coefficients B and K will vary with PM temperature, and from unit to unit. Also, the physical distances r, a, and L may vary from unit to unit. Therefore, the bicep (3.2.2) and tricep (3.2.4) models are poorly known and time-varying, making nonadaptive control methods vulnerable to failure when used for tracking performance of the PM. Since the nonlinear functions of θ are known in (3.2.2) and (3.2.4) and only their coefficients are uncertain, we utilize a method of nonlinear adaptive tracking based on sliding control ([20], [21]). It uses a well-known result from model reference adaptive control, which we give without proof (see, e.g. [20]).

Lemma: Consider two signals e and ϕ related by the following dynamic equation

$$e(t) = H(p)\left[k\phi^T(t)v(t)\right] \qquad (3.3.1)$$

where $e(t)$ is a scalar output signal, $H(p)$ is a strictly positive real transfer function, k is an unknown constant with a known sign, $\phi(t)$ is a $m\times 1$ vector function of time, and $v(t)$ is a measurable $m\times 1$ vector. If the vector ϕ varies according to

$$\dot{\phi}(t) = -\mathrm{sgn}(k)\gamma e v(t) \qquad (3.3.2)$$

with γ being a positive constant, then $e(t)$ and $\phi(t)$ are globally bounded. Furthermore, if v is bounded, then $e(t)\to 0$ as $t\to\infty$.

Bicep Adaptive Tracking

Consider the problem of the arm lifting a mass with PM in bicep position as in Figure 3.1. If we substitute $x = -a(1+\cos\theta)$ and $\dot{x} = a\dot{\theta}\sin\theta$ in (3.2.2), we get an equation in the form

$$h\ddot{\theta}+a_1\dot{\theta}^3\sin^4\theta+a_2\dot{\theta}^2\sin^3\theta+a_3\dot{\theta}\sin^2\theta+a_4\sin\theta(1+\cos\theta)^3$$
$$+a_5\sin\theta(1+\cos\theta)^2+a_6\sin\theta(1+\cos\theta)+a_7\sin\theta=-F\sin\theta$$

or

$$h\ddot{\theta}+\sum_{i=1}^{7}a_i f_i(\theta,\dot{\theta})=-F\sin\theta \tag{3.3.3}$$

where $h=I/a$, $a_1,\ldots,a_7$ are parameters which depend on the physical properties of the system (i.e. M, a, L, the coefficients in (2.2.2) and (2.2.3), etc.), and $f_1,\ldots,f_7$ are known functions of θ and $\dot{\theta}$.

Assume that h, $a_1,\ldots,a_7$ are unknown and it is desired that the PM angle $\theta(t)$ track a known angle $\theta_d(t)$. Define the error $e(t)=\theta(t)-\theta_d(t)$. Also define the auxiliary signal

$$y_r(t)=y_d(t)-\lambda_0 e(t) \tag{3.3.4}$$

where λ_0 is a positive constant and the combined error

$$s=\dot{e}+\lambda_0 e \tag{3.3.5}$$

Consider a control F such that

$$-F\sin\theta=\hat{h}\ddot{y}_r-ks+\sum_{i=1}^{7}\hat{a}_i f_i \tag{3.3.6}$$

where k is a positive constant and $\hat{h},\hat{a}_1,\ldots,\hat{a}_7$ are estimates of the unknown parameters $h, a_1,\ldots,a_7$. With this control law, we have the following result concerning the stability of asymptotic tracking of the arm with PM in bicep position:

Theorem 1: Consider the PM in bicep position moving a mass (Figure 3.1). Assume the PM spring and viscous friction coefficients are as in (2.2.2). If the force F delivered by the PM satisfies (3.3.6), then all signals of the adaptive system are bounded with $\lim_{t\to\infty}e(t)=0$ provided the parameter estimates are adjusted according to

$$\dot{\hat{h}} = -\gamma s \ddot{y}_r \tag{3.3.7a}$$

$$\dot{\hat{a}}_i = -\gamma s f_i, \quad i = 1,\ldots 7 \tag{3.3.7b}$$

where γ is a positive constant.

Proof: It can be shown that the tracking error from control law (3.3.6) is

$$h\dot{s} + ks = \tilde{h}\ddot{y}_r + \sum_{i=1}^{7} \tilde{a}_i f_i \tag{3.3.8}$$

where $\tilde{h} = \hat{h} - h$ and $\tilde{a}_i = \hat{a}_i - a_i$. This can be rewritten as

$$s = \frac{1/h}{p + k/h}\left[\tilde{h}\ddot{y}_r + \sum_{i=1}^{7} \tilde{a}_i f_i\right] \tag{3.3.9}$$

This is in the form of (3.3.1) with the transfer function obviously being strictly positive real. Therefore, we have from the lemma that all signals of the adaptive system are bounded.

Consider the Lyapunov function candidate

$$V = hs^2 + \gamma^{-1}\left[\tilde{h}^2 + \sum_{i=1}^{7} \tilde{a}_i^2\right] \tag{3.3.10}$$

where $\tilde{h} = \hat{h} - h$ and $\tilde{a}_i = \hat{a}_i - a_i$. It is straightforward to show that the derivative of V along the trajectories of the closed-loop system is given by

$$\dot{V} = -2ks^2 \tag{3.3.11}$$

Therefore, we have $s \to 0$ as $t \to \infty$. It follows that $\lim_{t\to\infty} e(t) = 0$. ◊

Tricep Adaptive Tracking

Consider the problem of the arm moving a mass with PM in tricep position as in Figure 3.2. If we substitute $x = -\frac{1}{2}\pi r(1+\cos\theta)$ and $\dot{x} = \frac{1}{2}\pi r\dot{\theta}\sin\theta$ in (3.2.4), we get an equation in the form

$$h\ddot{\theta} + a_1(\dot{\theta}\sin\theta)^3 + a_2(\dot{\theta}\sin\theta)^2 + a_3\dot{\theta}\sin\theta + a_4(1+\cos\theta)^3$$
$$+ a_5(1+\cos\theta)^2 + a_6(1+\cos\theta) + a_7\sin\theta = -F$$

or

$$h\ddot{\theta} + \sum_{i=1}^{7} a_i f_i(\theta, \dot{\theta}) = -F \tag{3.3.12}$$

where $h = I/r$, $a_1,\ldots,a_7$ are parameters which depend on the physical properties of the system (i.e. $M, a, L,$ the coefficients in (2.2.2) and (2.2.3), etc.), and $f_1,\ldots,f_7$ are known functions of θ and $\dot{\theta}$.

Assume that $h, a_1,\ldots,a_7$ are unknown and it is desired that the PM angle $\theta(t)$ track a known angle $\theta_d(t)$. Define the quantities $e(t) = \theta(t) - \theta_d(t)$, $y_r(t)$, and s as above.

Consider a control F such that

$$-F = \hat{h}\ddot{y}_r - ks + \sum_{i=1}^{7} \hat{a}_i f_i \tag{3.3.13}$$

where k is a positive constant and $\hat{h}, \hat{a}_1,\ldots,\hat{a}_7$ are estimates of the unknown parameters $h, a_1,\ldots a_7$. Then we have the following result concerning the stability of asymptotic tracking of the arm with PM in tricep position using the control (3.3.13):

Theorem 2: Consider the PM in tricep position lifting a mass (Figure 3.2). Assume the PM spring and viscous friction coefficients are as in (2.2.2). If the force F delivered by the PM satisfies (3.3.13), then all signals of the closed-loop system are bounded with $\lim_{t\to\infty} e(t) = 0$ provided the parameter estimates are adjusted according to

$$\dot{\hat{h}} = -\gamma s\ddot{y}_r \tag{3.3.14a}$$
$$\dot{\hat{a}}_i = -\gamma s f_i, \quad i = 1,\ldots,7 \tag{3.3.14b}$$

where γ is a positive constant.

Proof: The proof is similar to that of Theorem 1.

Comment 1: To implement the control laws (3.3.6) and (3.3.13) it is necessary to measure θ and $\dot{\theta}$. This should be no problem in PM applications, because these are the joint angle and its rate of change, respectively and are easily measured.

Comment 2: As stated above, the bicep control F is multiplied by $\sin\theta$, which vanishes as the arm approaches either the vertical-up or vertical-down position. Thus, the arm cannot be controlled in the vicinity of these positions. For this reason, care should be taken to avoid arm angles close to vertical-up or vertical-down for bicep control. Tricep configuration does not have this limitation due to the fact that the force is always applied at an angle α to the forearm, regardless of the joint angle.

Comment 3: The assumption of coefficients (2.2.2) are necessary so that the plant parameters are constants. This assumption is equivalent to stipulating that the PM is not allowed to deflate. This may be the case if e.g. the task is to lift a mass. If the PM were inflating and deflating, the PM spring and viscous friction coefficients would switch between (2.2.2) and (2.2.3). Therefore, the plant parameters would be time varying and the proofs to Theorems 1 and 2 would be considerably more difficult. The assumptions are technically convenient to prove the theorems, but do not appear to be necessary for asymptotic tracking in simulations in which both inflation and deflation are involved. Of course, the theorems also apply to the case where only coefficients (2.2.3) are assumed. In this case the PM is not allowed to inflate but only deflate, which might be the case when the task is to lower the mass.

3.4 Simulation Results

The systems of (3.2.2) and (3.2.4) were simulated using a 4th-order Runge-Kutta algorithm with a step size of 0.01 seconds. The control for the bicep configuration is given by (3.3.6) and (3.3.7). The control for the tricep configuration is given by (3.3.13) and (3.3.14). The results of these simulations are given below.

Bicep Simulation

Assume a configuration as in Figure 3.1. Let $L = 0.5\,\text{m}$, $a = 0.025\,\text{m}$, $M = 50\,\text{kg}$, and $g = 9.807\text{m/sec}^2$. Since $a = 0.025\,\text{m}$, the full travel of the forearm from $\theta = 180^\circ$ (arm fully straightened) to $\theta = 0^\circ$ (arm fully bent) corresponds to a maximum change in length of the PM of 0.05 m. This corresponds approximately to the actual capability of the PM considered above.

The desired trajectory for the joint angle between 0 and 15 seconds is

$$\theta_d(t) = \left|60^\circ + 62.5^\circ(\sin(2\pi f_1 t) + \sin(2\pi f_2 t))\right| \tag{3.4.1}$$

with $f_1 = 0.01\,\text{Hz}$ and $f_2 = 0.1\,\text{Hz}$. Therefore, $\theta_d(t)$ is a sum of two sinusoids with initial condition $\theta_d(0) = 60^\circ$. This trajectory spans monotonically increasing joint angles from 60° to approximately 110° and corresponds to the arm lifting a along the prescribed trajectory.

The input to the PM is given by (3.3.6) and (3.3.7) with $\gamma = 3\times10^4$, $\lambda_0 = 1$, and $k = 1$. The initial parameter guesses are zero, and the initial joint angle is 42°. The desired and actual joint angles are shown in Figure 3.3. It is seen that the arm asymptotically tracks $\theta_d(t)$ after the initial adaptation stage (approximately 4 seconds).

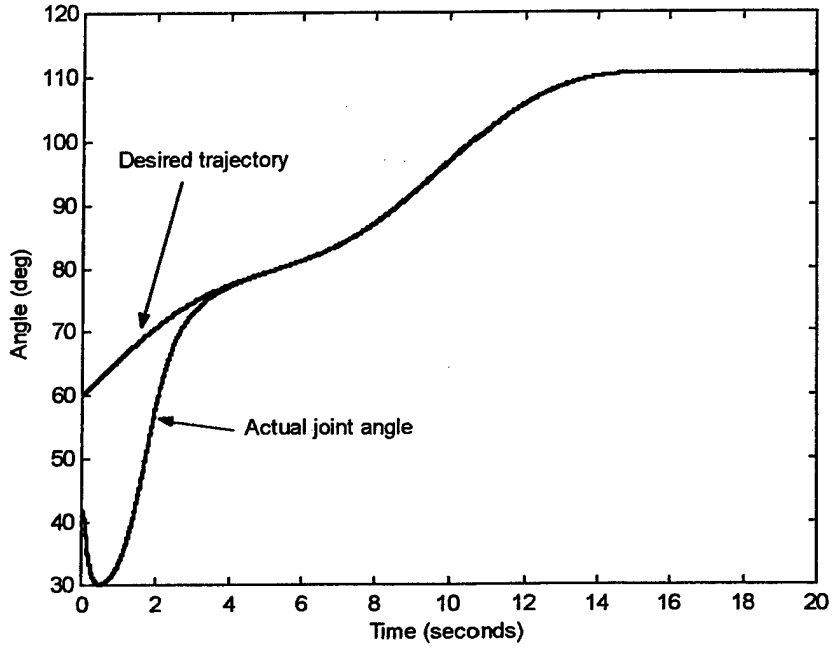


Figure 3.3 - Bicep adaptive tracking, $\gamma = 3\times10^4$, $\lambda_0 = 1$, $k = 1$.

Tricep Simulation

Assume a configuration as in Figure 3.2. Let $L = 0.5\,\text{m}$, $r = (0.05/\pi)\,\text{m}$, $M = 50\,\text{kg}$, and $g = 9.807\text{m/sec}^2$. Since $r = (0.05/\pi)\text{m}$, the full travel of the forearm from $\theta = 180^\circ$ (arm fully bent) to $\theta = 0^\circ$ (arm fully straightened) corresponds to a maximum change in length of the PM of 0.05 m.

The desired trajectory for the joint angle is again as in (3.4.1), which corresponds to lowering the mass along the prescribed trajectory. The input to the PM is given by (3.3.13) and (3.3.14) with $\gamma = 1\times10^5$, $\lambda_0 = 1$, and $k = 1$. The initial parameter guesses are zero, and the initial joint angle is 42°. The desired and actual joint angles are shown in Figure 3.4. Again we have asymptotic tracking except in the initial adaptation stage.

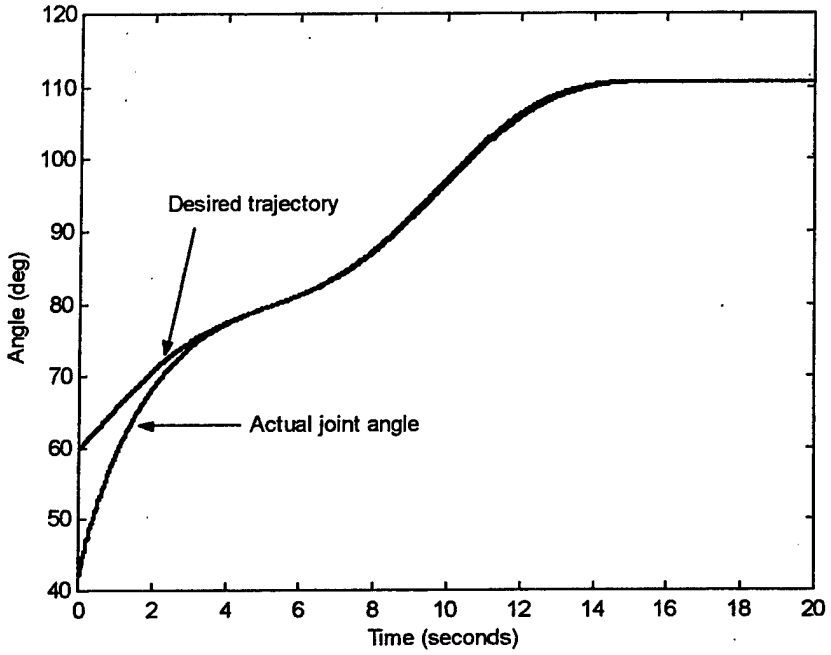


Figure 3.4 - Tricep adaptive tracking, $\gamma = 1 \times 10^5$, $\lambda_0 = 1$, $k = 1$.

3.5 Discussion

The simulations of PM in bicep and tricep configurations have been designed to closely conform to use in PM-actuated exoskeletons. In an exoskeleton, there are no rigid rods for forearm and upper arm, but the exoskeleton may possess some form of rigidity, i.e. a rigid enclosure for a human limb. Exoskeleton PMs are arranged in configurations very similar to human skeletal muscles, i.e. agonist/antagonist or bicep/tricep pairs. The bicep and tricep results in this paper apply to PMs used anywhere in an exoskeleton (arms, legs, etc.), as long as they are arranged in bicep or tricep configurations. An exoskeleton PM in the tricep configuration must have a path over which the PM cable passes to attach to the limb past the joint. This path has been modeled as a frictionless pulley in this study.

In typical exoskeleton applications, the mass actuated by the PM, or the moment of inertia of the moving joint, will vary significantly due to changing joint angles. For instance, when moving a mass from one point to another, the arm may bend, changing the load to the PM, which nevertheless must actuate the limb to follow a desired reference trajectory. This situation arises in robotics as well. Also, the nonlinear spring and nonlinear viscous damping coefficients are poorly known and change with time. This is because with use, the PM heats up, changing these coefficients. In addition, physical properties of the exoskeleton, i.e. arm lengths, distances to attach points, etc. may be poorly known.

Therefore, adaptive control methods have been applied to this problem, since fixed

controllers are less robust to parameter changes than adaptive ones. The simulations in Section 3.4 were also carried out with a PID controller designed to give good performance with $M = 50\,\text{kg}$. If the mass remains in the vicinity of this value, the PID gives good results. However, if the mass changes significantly, the fixed PID cannot stabilize the system. With the adaptive controllers given in Section 3.3, M can undergo a threefold change while retaining adequate tracking. However, the fixed PID is much less tolerant to changes in M, failing to stabilize the system for $M > 80\,\text{kg}$.

Finally, we note that in real applications of PMs, they will most probably be arranged in agonist/antagonist pairs, as in [4]. Therefore, there will be a bicep/tricep pair rather than a single bicep or tricep acting alone. This would increase joint impedance and result in a more stable joint angle control problem. The present paper is intended to study the action of individual muscles only, without introducing agonist/antagonist interaction.

3.6 Conclusions

Dynamic models for pneumatic muscles in bicep and tricep configurations actuating a mass have been derived. These configurations are very similar to exoskeleton applications in which PMs are used to increase strength and mobility in humans. The models are second-order and nonlinear in the joint angle. Their form makes them amenable to nonlinear adaptive control techniques, since the nonlinear functions of the joint angle are known, with only physical constants of the system being unknown. Simulations of closed-loop adaptive tracking of limbs moving masses show that adaptive control techniques are superior to fixed methods, i.e. fixed PID controllers, for this application.

4 Sliding Mode Control of Planar Arm with Two PMs

4.1 Introduction

Sliding control is a very promising method of PM control. It has the advantage that it can provide accurate tracking with bounded error in the presence of model uncertainties. A disadvantage is that it can produce a high amount of control effort due to chattering. This is because the control law is discontinuous across the sliding surface. For this reason, a boundary layer is usually designed into the control law in which the control is linear in the vicinity of the sliding surface. Sliding mode control is ideal for PM control because the PM model is usually poorly known, nonlinear, and time-varying, necessitating some type of robust control strategy. This accounts for the success of adaptive, variable-structure, and soft computing approaches also.

In this chapter, the control problem is elbow angle tracking for a planar arm. The elbow is actuated by a pair of opposing PMs, one in bicep and one in tricep position. We first derive a mathematical model of the arm with antagonistic PM actuators, then formulate the sliding mode controller to produce accurate tracking of the elbow angle. We address the important problem of bicep/tricep static internal pressures producing stable arm motion when control is absent. Finally, we present the results of computer simulations of the arm under the conditions of different actual arms (hence different truth models), and temperature variations.

4.2 Planar Arm Dynamic Model

In this chapter, the PM is modeled with the Reynolds model (2.3.1), (2.3.2). From (2.3.1), the total force exerted by the PM on the mass is $F - B\dot{x} - Kx$. The internal pressure P of the PM is an independent control variable that can be externally commanded by adjusting the inflation and deflation solenoids. If several PMs are present, each one has its own F, K, and B coefficients, its own internal pressure P, and its own inflation or deflation status. In this chapter, we assume all PMs are identical so their coefficients are the same.

Consider the planar manipulator configuration shown in Figure 4.1, which depicts an arm actuating a mass with PMs in bicep and tricep positions. The upper arm remains stationary as the PMs expand and contract, moving the forearm. The upper ends of the PMs and upper arm are attached to a stationary reference point. A mass M is held at the end of the forearm. The forearm, which is considered massless without loss of generality, is attached to the upper arm by a frictionless revolute joint, or "elbow." The PMs are attached to the forearm at point A, which is a

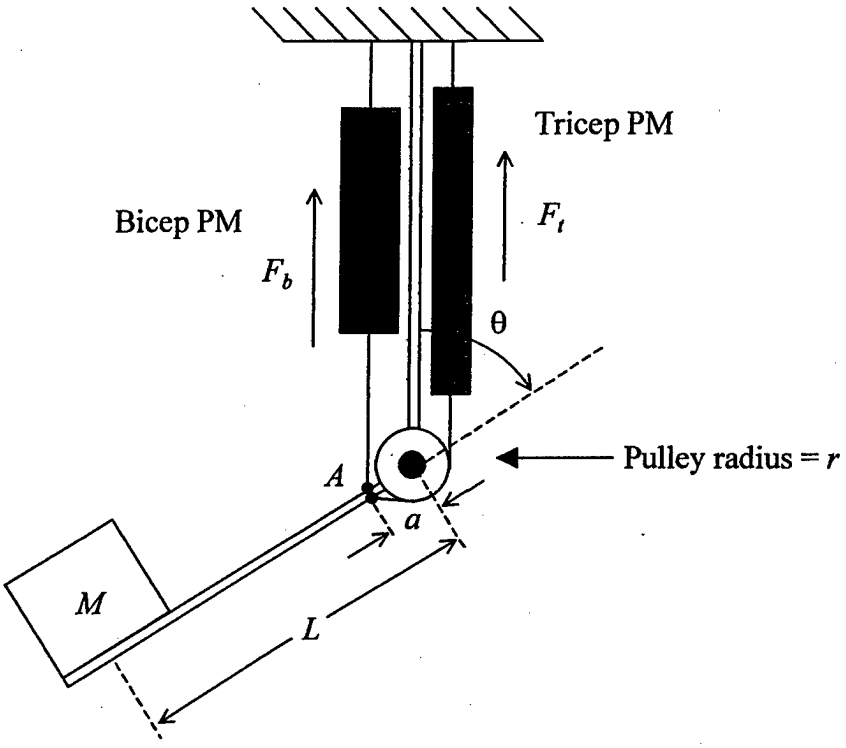


Figure 4.1 - Planar arm with PMs in bicep/tricep positions

distance a from the joint's axis of rotation. The distance from the center of mass of the load to the joint is L. Also at the joint is a frictionless pulley of radius r, over which a cable connecting the tricep PM with the forearm passes. The tricep cable makes an angle $\alpha = \sin^{-1}(r/a)$ with the forearm. The forearm is free to rotate through an angle θ, where $\theta = 0°$ corresponds to the arm being fully straightened, i.e. the mass in the extreme downward position, and $\theta = 180°$ corresponds to the arm being fully bent, i.e. the mass in the extreme upward position. If the bicep PM contraction is x_b and the tricep PM contraction is x_t, we have $x_b = a(1 - \cos\theta)$ and $x_t = a(1 + \cos\theta)$.

Let the bicep PM internal pressure be P_b and the tricep PM internal pressure be P_t. Also let the bicep PM coefficients be F_b, B_b, K_b and the tricep PM coefficients be F_t, B_t, K_t. Then, from (2.3.1) the clockwise torque exerted by the bicep PM on the forearm is $(F_b(P_b) - K_b(P_b)x_b - B_b(P_b)\dot{x}_b)a\sin\theta$ and the counterclockwise torque exerted by the tricep PM is $(F_t(P_t) - K_t(P_t)x_t - B_t(P_t)\dot{x}_t)r$. The equations of motion of the system of Figure 4.1 can be derived by summing torques about the elbow [22]:

$$I\ddot{\theta} = (F_b - K_b x_b - B_b\dot{x}_b)a\sin\theta - (F_t - K_t x_t - B_t\dot{x}_t)r - MgL\sin\theta \quad (4.2.1)$$

where $I = ML^2$ is the moment of inertia of the mass about the elbow, g is the acceleration of gravity, and $MgL\sin\theta$ is the counterclockwise torque imparted to the forearm by gravity acting on the mass. Note that since the bicep force is multiplied by $a\sin\theta$, the bicep loses controllability at $\theta = 0°$ and $\theta = 180°$. For this reason, the arm angle should be kept away from these extremes. This situation could be avoided, e.g. by wrapping the bicep cable around the elbow pulley also. The tricep does not suffer this drawback since its cable always makes an angle of $\alpha = \sin^{-1}(r/a)$ with the forearm regardless of θ.

From (2.3.2), let $F_b = F_0 + F_1 P_b$, $F_t = F_0 + F_1 P_t$, $K_b = K_0 + K_1 P_b$, $K_t = K_0 + K_1 P_t$, $B_b = B_0 + B_1 P_b$, and $B_t = B_0 + B_1 P_t$ where $F_0 = 179.2$, $F_1 = 1.39$, $K_0 = 5.71$, $K_1 = 0.0307$, and B_0, B_1 depend on whether the PM in question is being inflated or deflated, as follows:

$$B_0 = \begin{cases} 1.01, & \text{inflation} \\ 0.6, & \text{deflation} \end{cases} \tag{4.2.2a}$$

$$B_1 = \begin{cases} 0.00691, & \text{inflation} \\ -0.000803, & \text{deflation} \end{cases} \tag{4.2.2b}$$

The internal bicep and tricep pressures P_b and P_t are the control variables that can be independently commanded by the controller as inputs to the system. Note that the PM dynamics depend on whether the PM is being inflated or deflated.

Let us assume that the bicep and tricep internal pressures are given by

$$P_b = P_{b0} + \Delta p \tag{4.2.3a}$$

$$P_t = P_{t0} - \Delta p \tag{4.2.3b}$$

where P_{b0} is a nominal constant internal bicep PM pressure, P_{t0} is a nominal constant internal tricep PM pressure, and Δp is an arbitrary function of time. With the definitions in (4.2.3), (4.2.1) can be written as a single-input single-output system, with input Δp and output θ.

According to (4.2.3), the tricep deflates when the bicep inflates and vice versa. Therefore, one set of B parameters will apply to one of the PMs while the other set applies to the other PM at a given time. When the inflation/deflation status of the PMs changes, they trade B parameters. We denote the bicep B coefficients as B_{0b} and B_{1b}, and the tricep B coefficients as B_{0t} and B_{1t}.

4.3 Bicep and Tricep Nominal Pressures for Desired Equilibrium Angle

In order that the arm has a stable equilibrium at an angle θ_e, we seek to balance the steady-state clockwise and counterclockwise torques about the elbow by choice of bicep and tricep nominal pressures P_{b0} and P_{t0}. From (4.2.1), the total steady-state clockwise torque is

$$\tau_{cw} = [F_1 P_{b0} - (K_0 + K_1 P_{b0})a(1 - \cos\theta)]a\sin\theta \tag{4.3.1}$$

and the total steady-state counterclockwise torque is

$$\tau_{cc} = [F_1 P_{t0} - (K_0 + K_1 P_{t0})a(1 + \cos\theta)]r + MgL\sin\theta \tag{4.3.2}$$

for a constant angle θ. Equating these torques at a desired equilibrium angle θ_e results in the following relationship between P_{b0} and P_{t0}:

$$P_{b0} = mP_{t0} + c \tag{4.3.3}$$

where

$$m = [F_1 - K_1 a(1 + \cos\theta_e)]r/D \tag{4.3.4a}$$

$$c = (MgL\sin\theta_e - K_0 ar(1 + \cos\theta_e) + K_0 a^2(1 - \cos\theta_e)\sin\theta_e)/D \tag{4.3.4b}$$

and

$$D = [F_1 - K_1 a(1 - \cos\theta_e)]a\sin\theta_e \tag{4.3.5}$$

Note that the denominator D is nonzero for all arm angles in the open interval $(0, \pi)$.

Thus we have a relationship between the nominal bicep and tricep pressures that depends on the system coefficients, the mass M, and the desired equilibrium angle. Therefore, one of them (say P_{t0}) may be chosen arbitrarily, and the other (P_{b0}) is determined by this choice. These nominal pressures will be used in equations (4.2.3) to determine the total bicep and tricep pressures. The significance of these nominal pressures is that if the control input $\Delta p = 0$, the elbow angle reverts to the equilibrium angle θ_e, thus it is well-behaved even if the control loop is opened. In practice, these pressures could be easily determined experimentally, e.g. by setting P_{t0} to some nominal value, then

adjusting P_{b0} until the desired equilibrium arm angle is obtained. In our simulations below, we arbitrarily choose $\theta_e = 90°$.

4.4 Sliding Mode Control for Planar Arm with PMs in Bicep/Tricep Positions

Combining (4.2.1) with the above relationships for F, B, and K, we obtain the following 2nd order equation describing the system of Figure 4.1:

$$\ddot{\theta} = f(\theta, \dot{\theta}) + b(\theta, \dot{\theta})\Delta p \tag{4.4.1}$$

where

$$f(\theta, \dot{\theta}) = \sum_{i=1}^{6} f_i \xi_i(\theta, \dot{\theta}) \tag{4.4.2a}$$

$$b(\theta, \dot{\theta}) = \sum_{i=1}^{6} b_i \xi_i(\theta, \dot{\theta}) \tag{4.4.2b}$$

with $\xi_1 = \dot{\theta}\sin^2\theta$, $\xi_2 = \sin\theta(\cos\theta - 1)$, $\xi_3 = \dot{\theta}\sin\theta$, $\xi_4 = 1 + \cos\theta$, $\xi_5 = \sin\theta$, $\xi_6 = 1$, $f_1 = (-B_{b0} - B_{b1}P_{b0})a^2/I$, $f_2 = (K_0 + K_1 P_{b0})a^2/I$, $f_3 = (-B_{t0} - B_{t1}P_{t0})ar/I$, $f_4 = (K_0 + K_1 P_{t0})ar/I$, $f_5 = (aF_1 P_{b0} - MgL)/I$, $f_6 = -F_1 P_{t0} r/I$, $b_1 = -B_{1b}a^2/I$, $b_2 = K_1 a^2/I$, $b_3 = B_{1t}ar/I$, $b_4 = -K_1 ar/I$, $b_5 = F_1 a/I$, and $b_6 = F_1 r/I$.

Because of our imperfect knowledge of coefficients F, K, and B, we must assume that $f(\theta, \dot{\theta})$ and $b(\theta, \dot{\theta})$ in (4.4.1) are imprecise. Assume the extent of the imprecision on f can be bounded by a known continuous function of θ and $\dot{\theta}$. Similarly, we assume that the extent of the imprecision on b can be bounded by a known, continuous function of θ and $\dot{\theta}$. The control problem is to get the elbow angle $\theta(t)$ to track a desired trajectory $\theta^*(t)$ in the presence of model imprecision on $f$and b.

Assume we have an estimate $\hat{f}(\theta, \dot{\theta})$ of f and let $F(\theta, \dot{\theta})$ be a positive function such that

$$|\hat{f} - f| \leq F \tag{4.4.3}$$

Further assume the control gain $b(\theta, \dot{\theta})$ is unknown but that we have upper and lower bounds for it, i.e. we have quantities $b_{\min}$ and $b_{\max}$ such that $0 < b_{\min} \leq b \leq b_{\max}$ where $b_{\min}$ and $b_{\max}$ may depend on θ and $\dot{\theta}$. Define

$$\gamma = \sqrt{b_{max}/b_{min}} \tag{4.4.4}$$

Let the estimation of b be $\hat{b} = \sqrt{b_{min} b_{max}}$.

Let $\theta^*(t)$ be a smooth function of time that represents the desired angular trajectory for the elbow angle. Consider the sliding surface $\sigma = 0$, with

$$\sigma = \dot{\tilde{\theta}} + \mu\tilde{\theta} \tag{4.4.5}$$

where $\tilde{\theta} = \theta - \theta^*$ is the tracking error and μ is a scalar design parameter. Then the sliding mode control law is given by

$$\Delta p = \frac{(\hat{\Delta} p - q\text{sat}(\sigma/T))}{\hat{b}} \tag{4.4.6}$$

where

$$\hat{\Delta} p = (-\hat{f} + \theta^* - \mu\dot{\tilde{\theta}}) \tag{4.4.7}$$

$$q \geq \gamma(F + \epsilon) + (\gamma - 1)|\hat{\Delta} p| \tag{4.4.8}$$

ϵ is a positive constant, and T is the thickness of a "boundary layer," which is a neighborhood of the sliding surface introduced to reduce control chattering. Thus the boundary layer is defined by $B = \{(\theta, \dot{\theta}) : |\sigma(\theta, \dot{\theta})| \leq T\}$. We have the following result concerning the tracking accuracy of the above closed-loop system.

Theorem: Consider the planar arm with PMs in bicep/tricep configuration (Figure 4.1). Let the bicep and tricep pressures be as in (4.2.3) and (4.4.6) where $\theta^*(t)$ is a smooth function of time such that $\theta^*(t) \in (0, \pi)$ $\forall t$. Then the elbow angle $\theta(t)$ satisfies

$$\lim_{t_0 \to \infty} \sup_{t \geq t_0} |\theta(t) - \theta^*(t)| \leq \frac{T}{\mu} \tag{4.4.9}$$

Proof: It is straightforward to show that the surface $\sigma = 0$ is a moving straight line of slope $-\mu$ in $(\theta, \dot{\theta})$ space passing through the moving point $(\theta^*, \dot{\theta}^*)$. Consider the trajectory for an arbitrary initial condition $\theta(0)$ outside the boundary layer, i.e. such that $|\sigma(0)| \geq T$. Differentiating (4.4.5) and using (4.4.6) with $|\sigma| \geq T$, we have

$$\dot{\sigma} = f + \frac{[-b\hat{f} + (b-\hat{b})(\ddot{\theta}^* - \mu\dot{\tilde{\theta}}) - bq\text{sgn}(\sigma)]}{\hat{b}} \tag{4.4.10}$$

Evaluating the derivative of $V = \frac{1}{2}\sigma^2$ in the region $|\sigma| \geq T$, we have

$$\dot{V} = \sigma\left\{f + \frac{[-b\hat{f} + (b-\hat{b})(\ddot{\theta}^* - \mu\dot{\tilde{\theta}}) - bq\text{sgn}(\sigma)]}{\hat{b}}\right\}$$

$$= \sigma(f - b\hat{b}^{-1}\hat{f}) + \sigma(1 - b\hat{b}^{-1})(-\ddot{\theta}^* + \mu\dot{\tilde{\theta}}) - b\hat{b}^{-1}q|\sigma| \tag{4.4.11}$$

Then we have

$$\dot{V} \leq -\epsilon|\sigma| \tag{4.4.12}$$

provided

$$q \geq \epsilon b^{-1}\hat{b} + |b^{-1}\hat{b}f - \hat{f} + (b^{-1}\hat{b} - 1)(-\ddot{\theta}^* + \mu\dot{\tilde{\theta}})| \tag{4.4.13}$$

Using (4.4.3), we have that (4.4.13) is satisfied if

$$q \geq \epsilon b^{-1}\hat{b} + b^{-1}\hat{b}F + |b^{-1}\hat{b} - 1||\hat{f} - \ddot{\theta}^* + \mu\dot{\tilde{\theta}}| \tag{4.4.14}$$

Since $\gamma \geq b^{-1}\hat{b}$, we have that (4.4.14) is satisfied if q is as in (4.4.8). Thus the state trajectory approaches the boundary layer for all initial conditions outside it.

Now consider an arbitrary point $(\theta(0), \dot{\theta}(0))$ such that $|\sigma(\theta(0), \dot{\theta}(0))| > T$, and let t_T be the time taken for the system trajectory to reach the edge of the boundary layer from this point. Integrating (4.4.12) from $t = 0$ to t_T and considering initial points $(\theta, \dot{\theta})$ on both sides of the boundary layer results in

$$t_T \leq \frac{1}{\epsilon}|\sigma(0)| \tag{4.4.15}$$

Therefore, from any initial state $(\theta, \dot{\theta})$ outside the boundary layer, the system trajectory reaches the boundary layer in a finite time, and (4.4.12) guarantees that the boundary layer is invariant, i.e. $|\sigma| \leq T, \forall t \geq t_T$.

For $t \geq t_T$, we can use (4.4.5) to find the Laplace transform of $\tilde{\theta}(t)$:

$$\tilde{\theta}(s) = \frac{1}{s+\mu}\sigma(s) \tag{4.4.16}$$

which, together with the fact that $|\sigma| \leq T, \forall t \geq t_T$, easily yields (4.4.9). □

Therefore, we have that the sliding mode control law (4.4.6) guarantees that the state trajectory reaches the boundary layer in a finite time whatever the initial state, and inside the boundary layer constrains trajectories to stay inside it for all later time and approach a neighborhood of $\tilde{\theta} = 0$ asymptotically as $t \to \infty$ (4.4.9). Asymptotic tracking of the elbow angle to within a guaranteed accuracy is therefore obtained in spite of modeling errors which may be present in PM coefficients, masses, distances, etc.

4.5 Simulation Results

The planar arm of Figure 4.1 with PMs in bicep/tricep pair configuration is simulated using a 4th-order Runge-Kutta algorithm with a step size of 0.01 seconds. Let $L = 0.4572\,\text{m}$, $a = 0.1016\,\text{m}$, and $r = 0.0762$ m.

We investigate closed-loop tracking performance for two reference trajectories. One of these is a sum of sinusoids and the other is a square wave-type function with smooth transitions between constant values rather than discontinuous jumps. We begin with a nominal plant model and assume the true system is within ±50% of this nominal model. We observe the tracking performance for three different actual plants within this range, with the three plants defined by choosing all B, K, and F coefficients randomly within this range. We also simulate temperature variation effects by letting the coefficients B and K decrease slowly over the time of the simulation. This simulates the effect of the PM's temperature increasing with use over time. For all simulations, we choose $q = 1.1[\gamma(F+\epsilon) + (\gamma - 1)|\,\widehat{\Delta p}|]$, satisfying (4.4.8).

Modeling Errors

Case 1: The desired trajectory for the joint is

$$\theta^*(t) = \frac{\pi}{2} + 0.5(\sin(2\pi f_1 t) + \sin(2\pi f_2 t) + \sin(2\pi f_3 t)) \qquad (4.5.1)$$

with $f_1 = 0.02$ Hz, $f_2 = 0.05$ Hz, and $f_3 = 0.09$ Hz.

Let $M = 21.89$ kg, $\mu = 10$ and $\epsilon = 10$ (μ and ϵ chosen by trial and error to yield good performance). We choose the boundary layer thickness as $T = 1$. From (4.4.9), we have that the guaranteed tracking precision is $T/\mu = 0.1$ radians. Also let $P_{t0} = 275.79$ kPa (40 psi), $P_{b0} = 371.49$ kPa (53.88 psi) for a θ_e of $\pi/2$, satisfying (4.3.3).

Assume that the true values of $f(\theta, \dot{\theta})$ and $b(\theta, \dot{\theta})$ in (4.4.1) are known to fall within $\pm$ 50% of the best estimates we have of them, which are $\widehat{f}(\theta, \dot{\theta})$ and $\widehat{b}(\theta, \dot{\theta})$. Then we have $F = 0.5|\widehat{f}|$, $b_{\max} = 1.5\widehat{b}$, $b_{\min} = 0.5\widehat{b}$, and γ is determined as 1.7321 by (4.4.4).

The sliding control input to the PM is given in (4.4.6) with parameters defined as above. The tracking performance for a typical actual arm within $\pm$50% of the model $\widehat{f}$, $\widehat{b}$ is shown in Figure 4.2. Figure 4.3 shows the tracking errors for three different actual arms f, b within this range, using the same controller. It is seen that for all systems the tracking error is within predicted bounds.

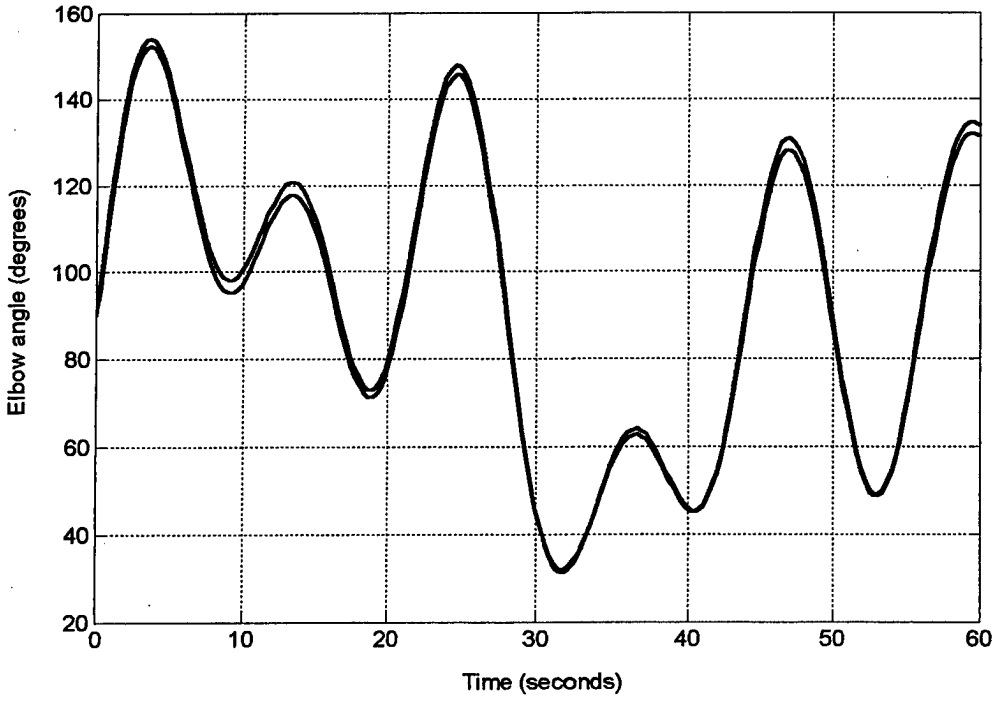


Figure 4.2 - Tracking performance, M = 21.89 kg.

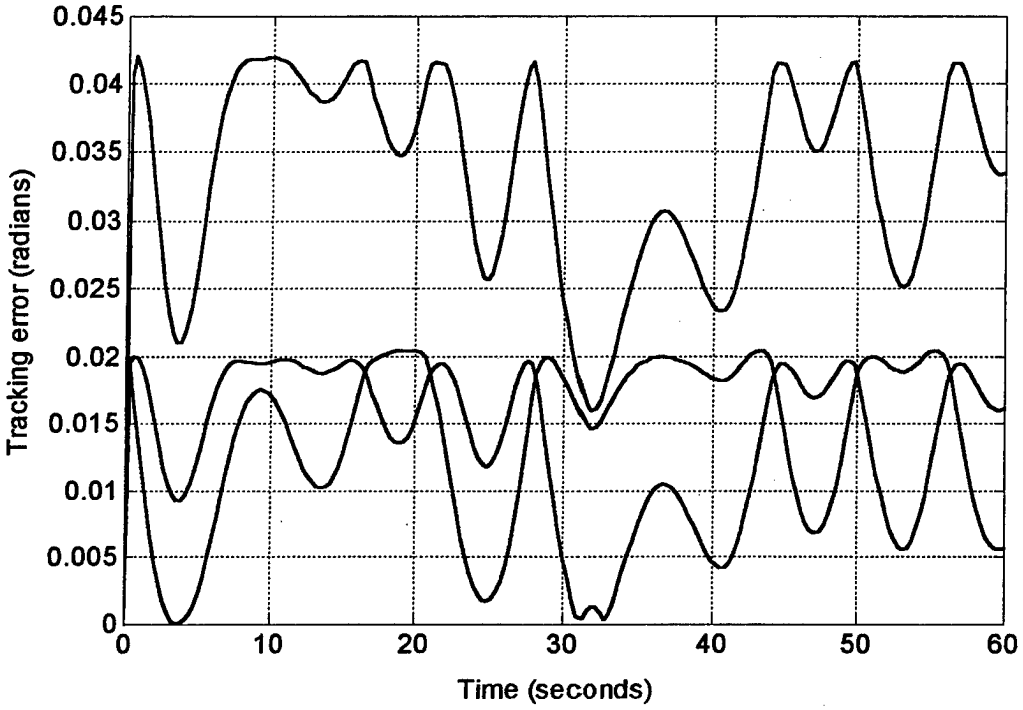


Figure 4.3 - Tracking errors for three different actual plants, M = 21.89 kg.

Figure 4.4 shows a typical control effort Δp for the above controller. It is evident that input pressure varies smoothly without any obvious chattering. Therefore, by using the sliding mode controller, the arm with PMs in bicep/tricep configuration achieves desired performance with good tracking precision and no obvious chattering for all three actual arms.

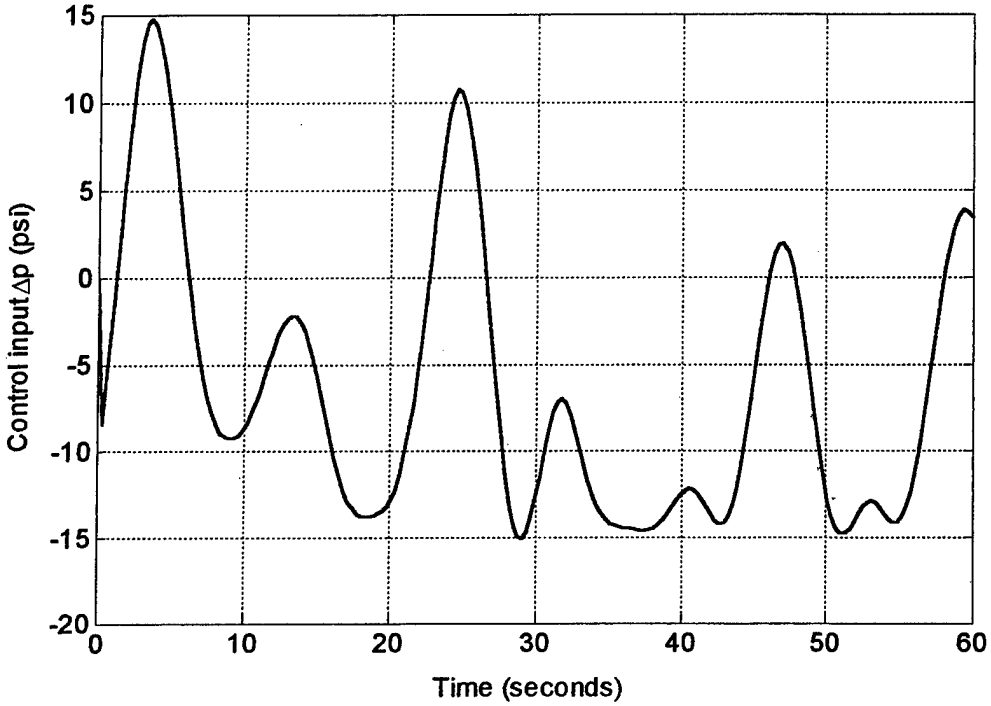


Figure 4.4 - Control effort, M = 21.89 kg.

In practical applications, it may be expected that the mass actuated by the arm will change. To investigate the robustness of the sliding controller to changing masses, we decreased the mass M by a factor of 3 to 7.30 kg and used the same controller as above. Figure 4.5 shows tracking errors for three different actual arms randomly chosen within the $\pm 50\%$ range. Tracking error is again within predicted bounds.

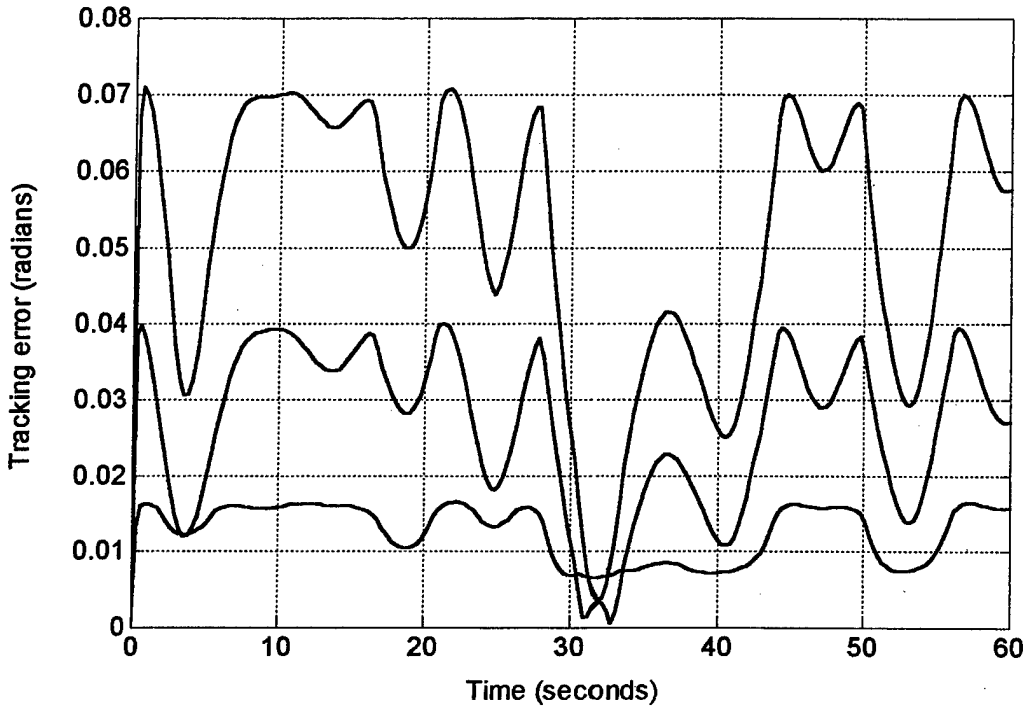


Figure 4.5 - Tracking errors for three different actual plants, M = 7.3 kg.

Figure 4.6 shows a typical control effort when $M = 7.3$ kg. Note that the control effort is smaller than the $M = 21.89$ kg case, which is to be expected since a lighter mass is being moved.

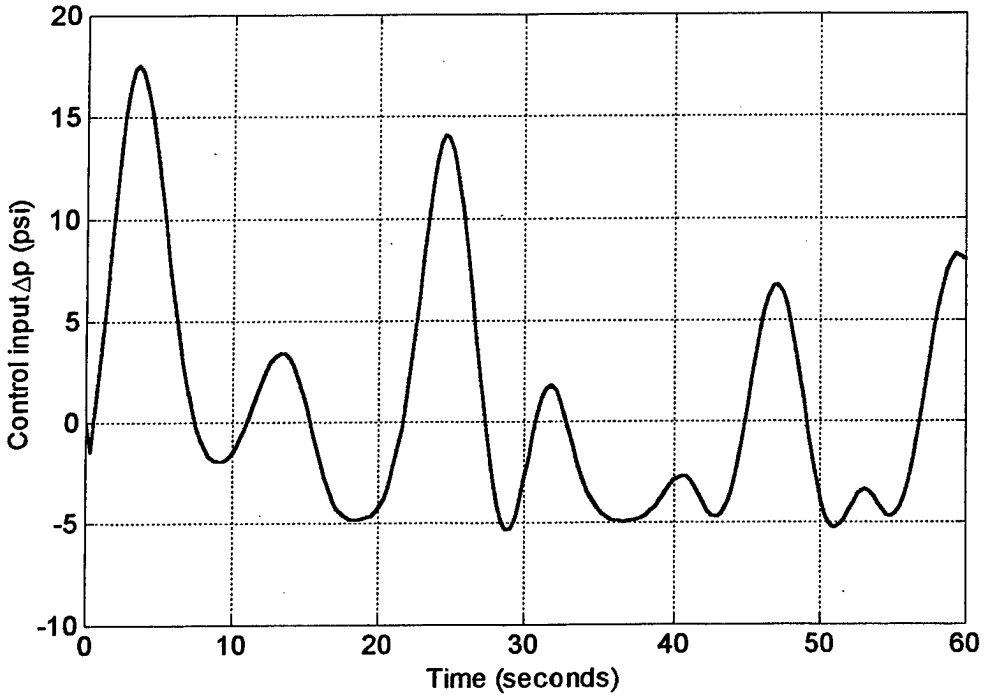


Figure 4.6 - Control effort, M = 7.30 kg.

The mass actuated by the arm will be practically limited, because very heavy masses would require the input pressure Δp to make the bicep and/or tricep pressures go outside the allowed range of PM internal pressure (206.84 - 896.32 kPa in this simulation). These limits would have the effect of a saturation nonlinearity on the PM input. The limitation on Δp is not the sliding controller's shortcoming; it is merely an acknowledgment of the practical constraint that the PM internal pressure must be kept within reasonable bounds to protect against actuator failure. If more force is desired, several PMs can be placed in parallel.

Case 2: To further verify the sliding mode controller, another simulation is performed to track a pseudo-square wave signal with a typical system within the ±50% range. Here, the desired trajectory is

$$\theta^*(t) = \begin{cases} 3\pi/4, & \sin(2\pi f_1(t-3)) \geq \frac{1}{4} \\ \frac{\pi}{2} + \pi\sin(2\pi f_1(t-3)), & |\sin(2\pi f_1(t-3)| \leq \frac{1}{4} \\ \pi/4, & \sin(2\pi f_1(t-3)) \leq -\frac{1}{4} \end{cases} \tag{4.5.2}$$

with $f_1 = 0.1$ Hz. This function transitions between constant values of $\pi/4$ and $3\pi/4$ smoothly rather than with discontinuous jumps. For the design parameters, we used

$\mu = 10$, $\epsilon = 10$, and $T = 0.3$. Therefore, the guaranteed tracking accuracy is $T/\mu = 0.03$ radians. We chose $P_{b0} = 637.29$ kPa and $P_{t0} = 206.84$ kPa for this simulation. These were chosen so that adequate tracking could be achieved with PM pressures in the permissible range.

Figures 4.7 and 4.8 show the tracking performance. Figure 4.8 indicates the tracking error exceeds that predicted by the theorem, i.e. the spikes in Figure 4.8 exceed the predicted maximum of $T/\mu = 0.03$. Figure 4.9 shows the control input Δp, which is seen to saturate at approximately 206.84 kPa and -689.48 kPa. These are values of Δp that cause the bicep or tricep PM pressure is go outside the allowed pressure range with the values of P_{b0} and P_{t0} above. The spikes in Figure 4.8 occur at times of input saturation, i.e. at these times the full input pressure dictated by the sliding mode controller is not applied and tracking accuracy is lost. For this reason, care should be taken to insure that PM pressure commanded by the controller will always be within the permissible range for the PM. This requirement on control effort is always present in any practical control system. Again, if more force is needed, several PMs should be used in parallel.

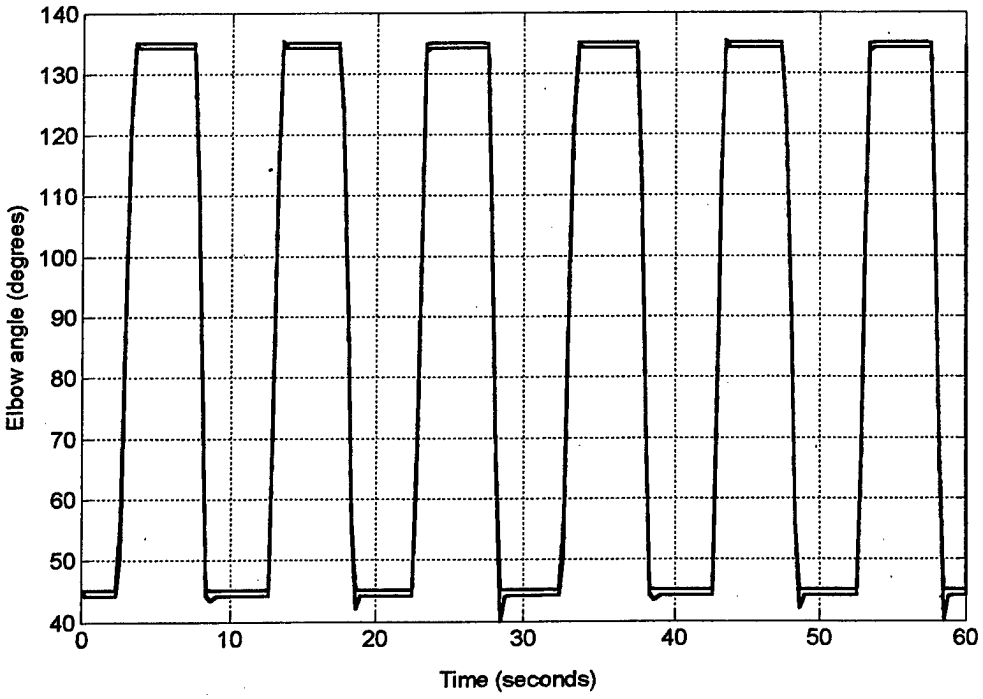


Figure 4.7 - Tracking performance, M = 21.89 kg.

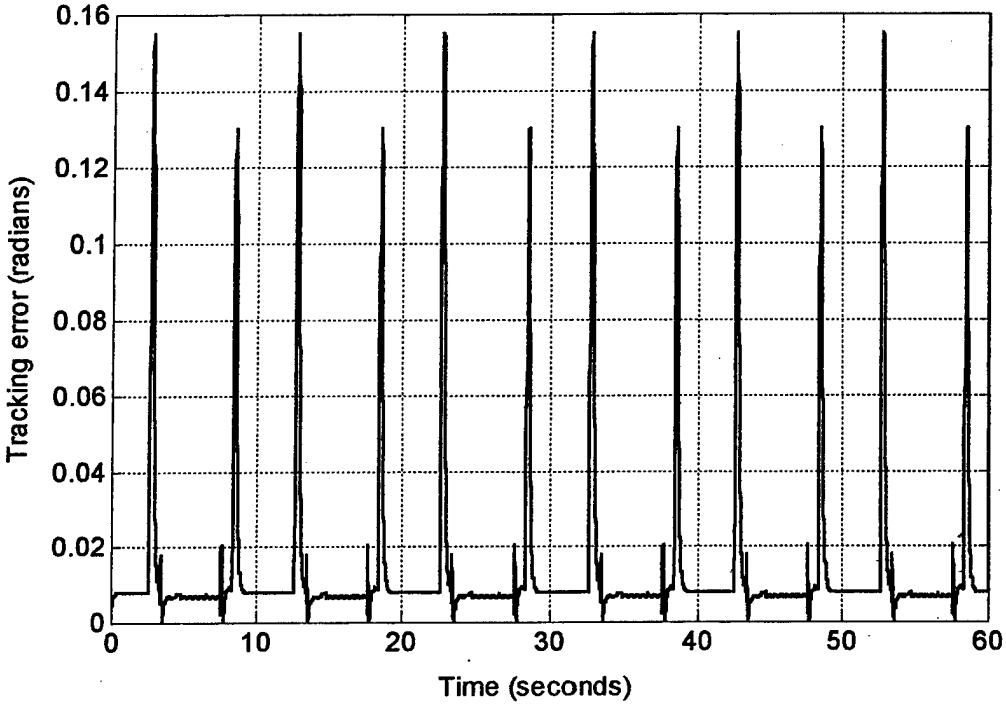


Figure 4.8 - Tracking error, M = 21.89 kg.

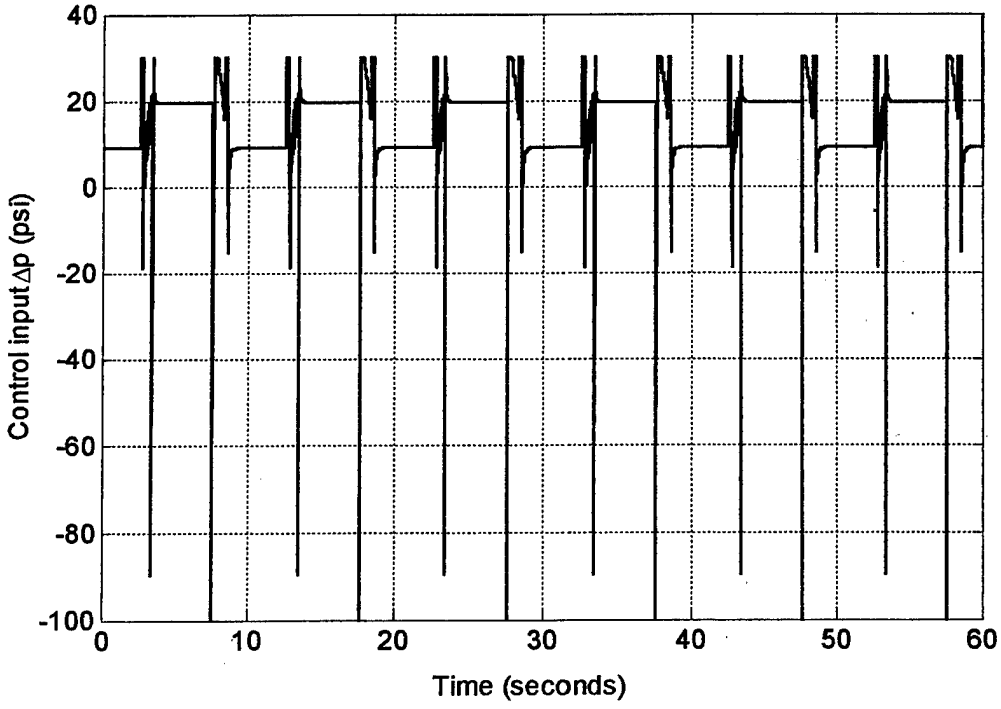


Figure 4.9 - Control effort, M = 21.89 kg

Heating Effects

When the PM is operated for extended periods of time, friction results in heating of the rubber bladder, changing its characteristics. We assume this has the effect of decreasing the B and K coefficients of the PM, thus decreasing the needed control effort to perform a task. This is due to the fact that the friction and spring effects opposing the PM motion are decreased as B and K decrease.

Assume the cold values of the PM coefficients are 1.5 times the nominal values given in Section 2, and over a period of 60 seconds of operation these values decrease linearly to 0.5 times the nominal values. Applying the sinusoidal reference signal of Case 1 above with $M = 21.89$ kg and the same controller as above, the elbow angle is nearly indistinguishable from the reference trajectory, i.e. tracking performance is excellent and well within predicted bounds. This is to be expected since with changing coefficients the arm becomes a slowly time-varying system. It is well-known that slowly varying systems pose no problem for sliding mode controllers [20]. Figure 4.10 shows the control input Δp. Comparing Figure 4.10 with Figure 4.4, the effects of the decaying B and K coefficients is seen as the control effort must be larger in the beginning due to larger B and K coefficients, but decreases toward the end as B and K decrease.

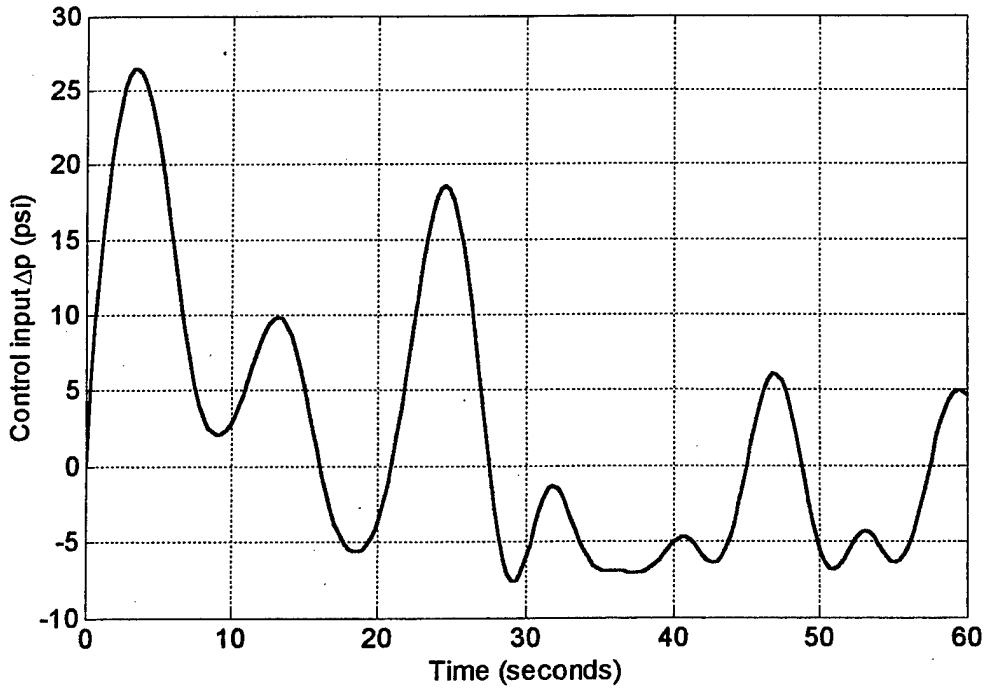


Figure 4.10 - Control effort with heat effects, M = 21.89 kg.

5 Sliding Mode Control of Planar Arm with Four PMs

5.1 Introduction

The high power/weight and power/volume ratios of PMs, in conjunction with their contractile nature (i.e. their inherent extensibility when activated), could make these actuators extremely useful in a variety of rehabilitation engineering applications for persons with neuromuscular or musculoskeletal pathologies that affect extremity function. Disorders that limit or hinder extremity function can include stroke, traumatic brain injury, amputation, and spinal cord injury. These conditions are associated with a number of functional deficits, including weakness, paralysis, limb loss, and joint contracture associated with spasticity and/or hypertonia. Individuals with such deficits could potentially benefit from powered devices that provide joint loading to generate or assist extremity motion, or maintain extension of muscles with contractures.

Many potential applications involve some type of exoskeletal or link segment configuration that attaches to existing anatomical body segments ([6], [23]-[26]). Devices of these types have often been relegated to use within a specific location (i.e. laboratory setting), or have limited actuation to small segments at the hand/wrist. Lower power/volume ratios of traditionally used power sources could hinder the use of powered exoskeletal orthoses, for larger extremity segments, in more general settings. Utilization of PMs to power exoskeletal devices, which interface with persons who have functional deficits, could potentially reduce size and weight sufficiently to facilitate more widespread use of such devices.

It should also be noted that traditional actuators, such as electric motors, are typically rigid in nature. Such actuators can create uncomfortable or painful conditions when interfacing with humans. For example, if an exoskeletal brace, actuated by a stepper motor pulley arrangement, is activated to extend a contracted joint to some desired angle, and a flexor spasticity episode occurs, the stepper motor will rigidly hold its position and not permit joint flexion. Such a response could result in pain and discomfort among patients with joint contractures associated with spasticity. Conversely, activated PMs maintain inherent extensibility, which could permit some joint flexion during muscular loading due to episodes of spasticity. While the elastic properties of PMs can complicate the control aspects of these actuators, they can potentially contribute to more comfortable devices when interfacing with human limb segments.

Sliding control has the advantage that it can provide accurate tracking with bounded error in the presence of model uncertainties. Sliding mode control is ideal for PM control because the PM model is usually poorly known, nonlinear, and time-varying, necessitating some type of robust control strategy. This accounts for the success of adaptive, variable-structure, and soft computing approaches also.

This chapter considers end-effector (hand) control of a planar arm actuated by four PMs. Two PMs in agonist/antagonist configuration actuate the shoulder joint, and two

PMs in agonist/antagonist configuration actuate the elbow. The contributions of this chapter include modeling of the arm with four PM actuators, formulating the model so that it is suitable for sliding mode control, determination of static internal pressures for stable arm behavior in the absence of a control signal, and extensive simulation results including the effects of changing temperatures.

5.2 Planar Arm Dynamic Model

In this chapter, the PM is modeled with the Reynolds model (2.3.1), (2.3.2). From (2.3.1), the total force exerted by the PM on the mass is

$$\phi = F(P) - B(P)\dot{x} - K(P)x \tag{5.2.1}$$

The internal pressure P of the PM is an independent control variable that can be externally commanded by adjusting the inflation and deflation solenoids. If several PMs are present, each one generally has its own F, K, and B coefficients, its own internal pressure P, and its own inflation or deflation status.

If a pair of such PMs is tied together around a pulley of radius r as in Figure 5.1, with the connecting line rigidly attached to the pulley to prevent slipping, the torque imparted to the pulley by the PM pair is

$$\tau_{\text{total}} = \tau_b - \tau_t = (\phi_b - \phi_t)r \tag{5.2.2}$$

where τ_b and τ_t are the torques due to each individual PM and are given by

$$\tau_b = (F_b - K_b x_b - B_b \dot{x}_b)r \tag{5.2.3a}$$

$$\tau_t = (F_t - K_t x_t - B_t \dot{x}_t)r \tag{5.2.3b}$$

In (5.2.3), x_b is the length of PM b and x_t is the length of PM t.

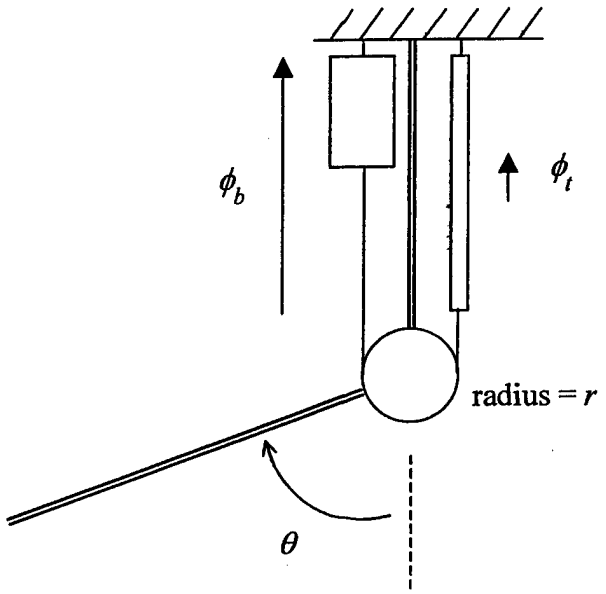


Figure 5.1 - Two PMs tied together around a pulley

The lengths of the arrows in Figure 5.1 are indicative of the forces exerted by the two PMs. Thus, $\phi_b > \phi_t$ and the torque exerted on the joint is clockwise. The total torque delivered to the pulley is given by

$$\tau_{\text{total}} = (F_b - K_b x_b - B_b \dot{x}_b - F_t + K_t x_t + B_t \dot{x}_t) r \qquad (5.2.4)$$

where F_b, K_b, and B_b depend on the internal pressure of PM b and F_t, K_t, and B_t depend on the input pressure of PM t according to (2.3.2).

The lengths x_b, x_t can be expressed in terms of the pulley angle θ since the pulley radius is known. We will find it advantageous to do so in order to formulate a two-input sliding mode controller for the planar arm actuated by opposing-pair PMs.

5.3 Dynamics of Planar Arm Actuated by PMs

Consider the manipulator configuration shown in Figure 5.2, which depicts a two-joint planar arm. In this figure, θ_i denotes the angle of joint i, m_i denotes the mass of link i, l_i denotes the length of link i, l_{ci} denotes the distance from the previous joint to the center of mass of link i (center of mass is denoted by a small dot), and I_i denotes the moment of inertia of link i about an axis coming out of the page, passing through the center of mass of link i.

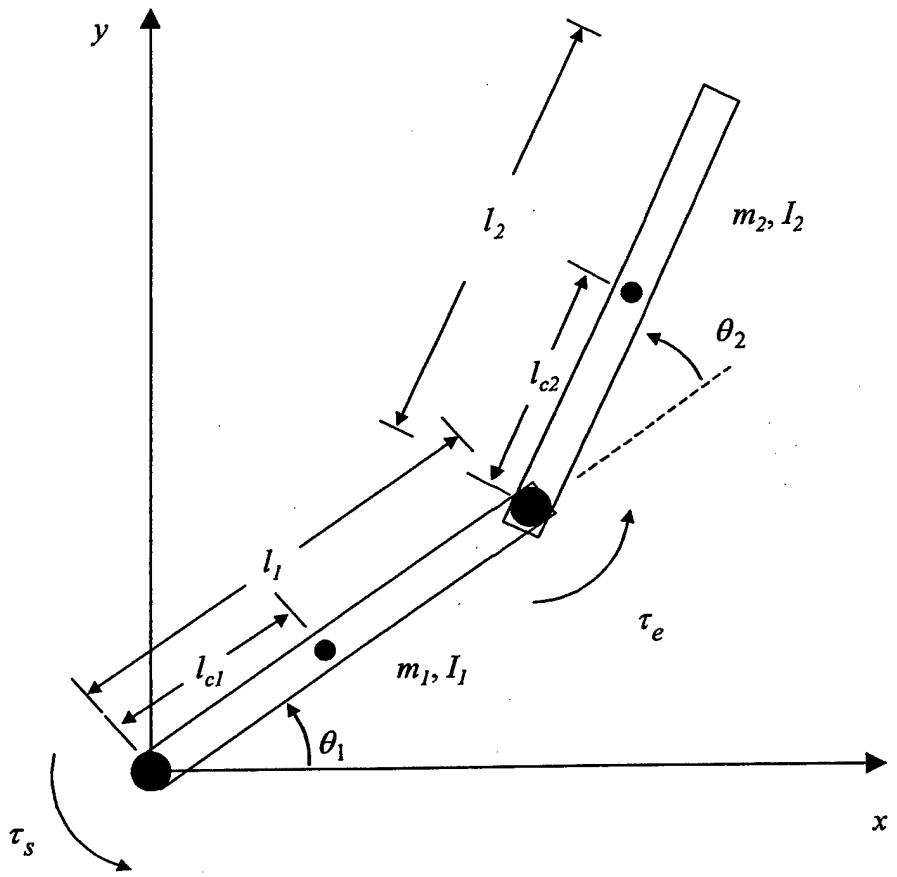


Figure 5.2 - Planar arm

The dynamics of this system are well-known [20] to be described by:

$$D(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + f(\theta) = \tau \tag{5.3.1}$$

where $\theta = [\theta_1, \theta_2]^T$ and $\tau = [\tau_s, \tau_e]^T$ is a matrix of input torques. The nonsingular inertia matrix $D(\theta)$ is

$$D(\theta) = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \tag{5.3.2}$$

where

$$d_{11} = m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2 + 2l_1 l_{c2}\cos\theta_2) + I_1 + I_2 \tag{5.3.3a}$$

$$d_{12} = d_{21} = m_2(l_{c2}^2 + l_1 l_{c2}\cos\theta_2) \tag{5.3.3b}$$

$$d_{22} = m_2 l_{c2}^2 + I_2 \tag{5.3.3c}$$

and $I_i = m_i l_{ci}^2$, $i = 1, 2$. The matrix $C(\theta, \dot{\theta})$ is given as

$$C(\theta, \dot{\theta}) = \begin{bmatrix} h\dot{\theta}_2 & h\dot{\theta}_2 + h\dot{\theta}_1 \\ -h\dot{\theta}_1 & 0 \end{bmatrix} \tag{5.3.4}$$

with $h = -m_2 l_1 l_{c2}\sin\theta_2$. The vector $f(\theta)$ is given by $f(\theta) = [f_1, f_2]^T$ where

$$f_1 = (m_1 l_{c1} + m_2 l_1)g\cos\theta_1 + m_2 l_{c2} g\cos(\theta_1 + \theta_2) \tag{5.3.5a}$$

$$f_2 = m_2 l_{c2} g\cos(\theta_1 + \theta_2) \tag{5.3.5b}$$

and g is the acceleration of gravity.

The arrangement of PMs on the manipulator is shown in Figure 5.3. The base (or torso) is fixed. At the top of the torso is the shoulder revolute joint, which is a pulley. The upper arm is attached to the shoulder joint, which is rotatable through an angle $-\frac{\pi}{2} < \theta_1 < \frac{\pi}{2}$. At the end of the upper arm is the elbow revolute joint, which is another pulley. The forearm is attached to the elbow joint, which is free to rotate through an angle $0 < \theta_2 < \pi$.

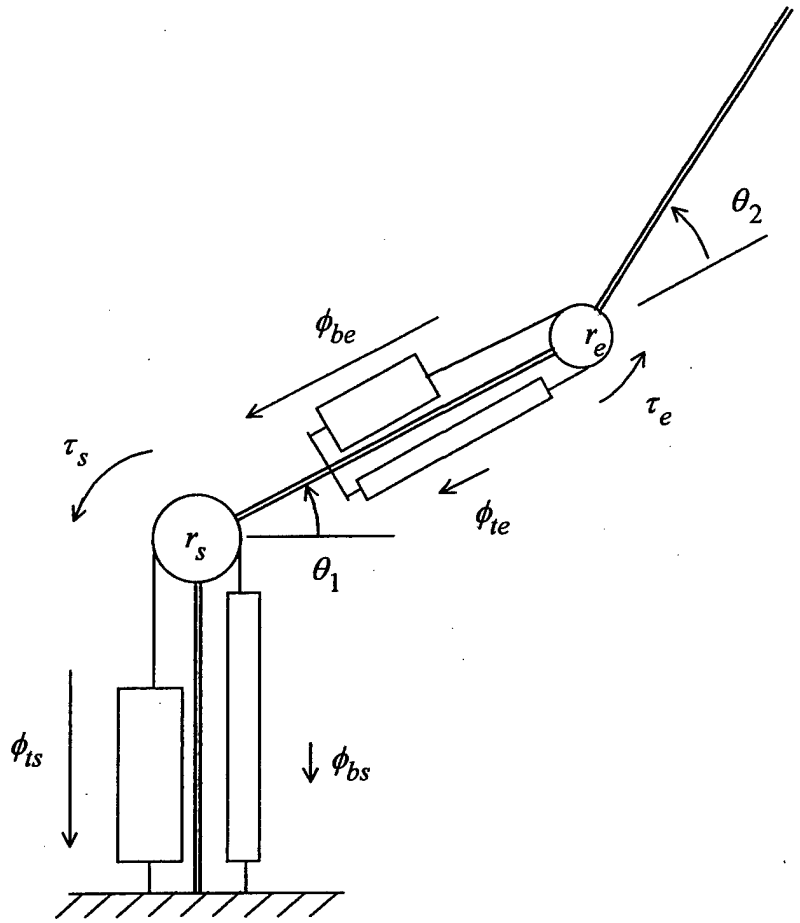


Figure 5.3 - Arm with PMs

Assume there are n_s pairs of matched PMs (i.e. all PMs have identical coefficients and lengths) tied together around the shoulder pulley (radius r_s) with all tricep PMs recieving the same input pressure P_{ts} and all bicep PMs recieving the same input pressure P_{bs}. Similarly, assume there are n_e pairs of matched PMs tied together around the elbow pulley (radius r_e) with all tricep PMs recieving the same input pressure P_{te} and all bicep PMs recieving the same input pressure P_{be}. We do not assume the elbow PMs are matched to the shoulder PMs, however. The assumption of matched PMs is rather idealistic. However, if care is taken to keep constrution of individual PMs the same and of the same materials and dimensions, the assumption of matched PMs may not be too erroneous.

In Figure 5.3, the shoulder and elbow torques τ_s and τ_e can be expressed as in (5.2.4):

$$\tau_s = n_s(F_s - K_s x_{ts} - B_{ts}\dot{x}_{ts} - F_s + K_s x_{bs} + B_{bs}\dot{x}_{bs})r_s \qquad (5.3.6a)$$

$$\tau_e = n_e(F_e - K_e x_{be} - B_{be}\dot{x}_{be} - F_e + K_e x_{te} + B_{te}\dot{x}_{te})r_e \qquad (5.3.6b)$$

where F_s, K_s, B_{ts}, and B_{bs} (F_e, K_e, B_{te}, and B_{be}) are the coefficients for the shoulder (elbow) PMs, t subscripts denote tricep PM quantities, b subscripts denote bicep PM quantities, s subscripts denote shoulder PM quantities (i.e. PMs located on the torso actuating the shoulder), and e subscripts denote elbow PM quantities (i.e. PMs located on the upper arm actuating the elbow).

A shoulder angle of $\theta_1 = -\pi/2$ corresponds to the shoulder tricep PM*s* being fully lengthened and the shoulder bicep PM*s* being fully shortened, and $\theta_1 = +\pi/2$ corresponds to the shoulder tricep PM*s* being fully shortened and the shoulder bicep PM*s* being fully lengthened. Therefore, the PM lengths x_{ts} and x_{bs} can be expressed in terms of θ_1 as

$$x_{ts} = r_s(\theta_1 + \tfrac{\pi}{2}) \qquad (5.3.7a)$$

$$x_{bs} = r_s(\tfrac{\pi}{2} - \theta_1) \qquad (5.3.7b)$$

Similarly, with $\theta_2 = 0$ corresponding to full shortening of the elbow tricep PM*s* and full lengthening of the elbow bicep PM*s*, and $\theta_2 = \pi$ corresponding to full lengthening of the elbow tricep PM*s* and full shortening of the elbow bicep PM*s*, the PM lengths x_{te} and x_{be} can be expressed in terms of θ_2 as

$$x_{te} = r_e(\pi - \theta_2) \qquad (5.3.8a)$$

$$x_{be} = r_e\theta_2 \qquad (5.3.8b)$$

Let the input pressure of the shoulder bicep and tricep PMs be

$$P_{bs} = P_{0bs} + \Delta p_s \qquad (5.3.9a)$$

$$P_{ts} = P_{0ts} - \Delta p_s \qquad (5.3.9b)$$

where P_{0bs} and P_{0ts} are arbitrary positive nominal constant pressures and Δp_s is an arbitrary function of time that is commanded by the controller. With these definitions, the set of n_s shoulder antagonist pair*s* becomes a single-input system with input Δp_s. When the bicep input pressure increases, the tricep input pressure decreases and vice versa, varying the torque on the shoulder joint. The nominal constant pressures P_{0bs} and P_{0ts} are arbitrary and can be chosen so that (1) the shoulder joint is well-behaved in the absence of a control signal Δp_s, and (2) desired joint stiffness is produced (see Section 5.4). Similarly, let the input pressures of the elbow PMs be defined as

$$P_{be} = P_{0be} + \Delta p_e \qquad (5.3.10a)$$

$$P_{te} = P_{0te} - \Delta p_e \tag{5.3.10b}$$

where P_{0be}, P_{0te} are arbitrary positive nominal constant pressures. With these definitions, the set of n_e elbow antagonist pairs becomes a single-input system with input Δp_e. Thus, with the PM input pressures defined as in (5.3.9) and (5.3.10), the 2-DOF planar arm of Figure 5.3 is a 2-input system with input Δp_s determining the shoulder torque τ_s and input Δp_e determining the elbow torque τ_e.

With definitions (5.3.6)-(5.3.10), we can write the shoulder and elbow torques as

$$\tau_s = \tau_{0s} + \tau_{1s}\Delta p_s \tag{5.3.11a}$$

$$\tau_e = \tau_{0e} + \tau_{1e}\Delta p_e \tag{5.3.11b}$$

where

$$\begin{aligned}\tau_{0s} = n_s\big[&F_{0s} + F_{1s}P_{0ts} - (K_{0s} + K_{1s}P_{0ts})x_{ts} - (B_{0ts} + B_{1ts}P_{0ts})\dot{x}_{ts} \\ &- F_{0s} - F_{1s}P_{0bs} + (K_{0s} + K_{1s}P_{0bs})x_{bs} + (B_{0bs} + B_{1bs}P_{0bs})\dot{x}_{bs}\big]r_s\end{aligned} \tag{5.3.12a}$$

$$\tau_{1s} = n_s\big[-F_{1s} + K_{1s}x_{ts} - B_{1ts}\dot{x}_{ts} - F_{1s} + K_{1s}x_{bs} + B_{1bs}\dot{x}_{bs}\big]r_s \tag{5.3.12b}$$

$$\begin{aligned}\tau_{0e} = n_e\big[&F_{0e} + F_{1e}P_{0be} - (K_{0e} + K_{1e}P_{0be})x_{be} - (B_{0be} + B_{1be}P_{0be})\dot{x}_{be} \\ &- F_{0e} - F_{1e}P_{0te} + (K_{0e} + K_{1e}P_{0te})x_{te} + (B_{0te} + B_{1te}P_{0te})\dot{x}_{te}\big]r_e\end{aligned} \tag{5.3.12c}$$

$$\tau_{1e} = n_e\big[F_{1e} + K_{1e}x_{be} - B_{1be}\dot{x}_{be} + F_{1e} - K_{1e}x_{te} + B_{1te}\dot{x}_{te}\big]r_e \tag{5.3.12d}$$

In (5.3.12), B_{0ts}, B_{1ts}, B_{0bs}, B_{1bs}, B_{0be}, B_{1be}, B_{0te} and B_{1te} are the appropriate coefficients from (2.3.2c), depending on whether the PMs are being inflated or deflated.

Combining (5.3.1) - (5.3.12), we can arrive at the following model for the planar arm actuated by four PM groups as in Figure 5.3:

$$\begin{bmatrix}\ddot{\theta}_1 \\ \ddot{\theta}_2\end{bmatrix} = \begin{bmatrix}a_1 \\ a_2\end{bmatrix} + G\begin{bmatrix}\Delta p_s \\ \Delta p_e\end{bmatrix} \tag{5.3.13}$$

where

$$\begin{bmatrix}a_1 \\ a_2\end{bmatrix} = D^{-1}\left(-C\dot{\theta} - f + \begin{bmatrix}\tau_{0s} \\ \tau_{0e}\end{bmatrix}\right) \tag{5.3.14}$$

and

$$G = D^{-1}\begin{bmatrix} \tau_{1s} & 0 \\ 0 & \tau_{1e} \end{bmatrix} \tag{5.3.15}$$

Note that a_1, a_2, and G are functions of θ_1, $\dot{\theta}_1$, θ_2, and $\dot{\theta}_2$.

The planar arm actuated by $n_s + n_e$ pairs of PMs in opposing pair configuration modeled as in (5.3.13) is now in a form which can be handled by multi-input sliding mode control methods.

5.4 PM Nominal Pressures for Desired Equilibrium Position of Planar Arm

In this section we find nominal pressures P_{0ts}, P_{0bs}, P_{0te}, P_{0be} such that the arm has an equilibrium point at a desired constant shoulder angle θ_{1eq} and an elbow angle θ_{2eq}. We do this in order that, if control is lost, the arm will revert to the desired equilibrium position. Another reason to do this is to produce desired stiffness in the joint. In order to find proper nominal pressures, we find relationships between them to balance the steady-state clockwise and counterclockwise torques about the shoulder and elbow as functions of the desired equilibrium joint angles. For this analysis, we assume all PM coefficients are exactly known. If they are not, see the end of this section.

From (5.3.6)-(5.3.8) and (5.3.12), the total steady-state clockwise and counterclockwise torques about the shoulder in Figure 5.3 are

$$\tau_{scw} = n_s[F_{0s} + F_{1s}P_{0bs} - (K_{0s} + K_{1s}P_{0bs})r_s(\tfrac{\pi}{2} - \theta_{1eq})]r_s + \tau_{12} \tag{5.4.1a}$$

$$\tau_{sccw} = n_s[F_{0s} + F_{1s}P_{0ts} - (K_{0s} + K_{1s}P_{0ts})r_s(\theta_{1eq} + \tfrac{\pi}{2})]r_s \tag{5.4.1b}$$

where

$$\tau_{12} = (m_1 l_{c1} + m_2 l_1)g\cos\theta_{1eq} + m_2 l_{c2} g\cos(\theta_{1eq} + \theta_{2eq}) \tag{5.4.2}$$

is the clockwise torque imparted to the shoulder by gravity, and r_s is the radius of the shoulder pulley.

Equating the clockwise and counterclockwise torques results in the following relationship between P_{0bs} and P_{0ts}:

$$P_{0ts} = m_s P_{0bs} + c_s \tag{5.4.3}$$

where

$$m_s = [F_{1s} - K_{1s} r_s (\frac{\pi}{2} - \theta_{1eq})]/\Delta_s \tag{5.4.4a}$$

$$c_s = (2K_{0s} r_s \theta_{1eq} + \tau_{12}/n_s r_s)/\Delta_s \tag{5.4.4b}$$

and

$$\Delta_s = F_{1s} - K_{1s} r_s (\theta_{1eq} + \frac{\pi}{2}) \tag{5.4.5}$$

Similarly, the total steady-state clockwise and counterclockwise torques about the elbow are

$$\tau_{ecw} = n_e[F_{1e} P_{0te} - (K_{0e} + K_{1e} P_{0te}) r_e (\pi - \theta_{2eq})] r_e + \tau_2 \tag{5.4.6a}$$

$$\tau_{eccw} = n_e[F_{1e} P_{0be} - (K_{0e} + K_{1e} P_{0be}) r_e \theta_{2eq}] r_e \tag{5.4.6b}$$

where

$$\tau_2 = m_2 l_{c2} g \cos(\theta_{1eq} + \theta_{2eq}) \tag{5.4.7}$$

is the clockwise torque imparted to the elbow by gravity, and r_e is the radius of the elbow pulley.

Equating these torques results in the following relationship between P_{0be} and P_{0te}:

$$P_{0be} = m_e P_{0te} + c_e \tag{5.4.8}$$

where

$$m_e = [F_{1e} - K_{1e} r_e (\pi - \theta_{2eq})]/\Delta_e \tag{5.4.9a}$$

$$c_e = [K_{0e} r_e (2\theta_{2eq} - \pi) + \tau_2/n_e r_e]/\Delta_e \tag{5.4.9b}$$

and

$$\Delta_e = F_{1e} - K_{1e} r_e \theta_{2eq} \tag{5.4.10}$$

Thus we have relationships between the nominal bicep and tricep pressures for shoulder and elbow joints that depend on the system coefficients and the desired equilibrium angles. Therefore, for a given shoulder equilibrium angle θ_{1eq}, one of the nominal shoulder pressures (say P_{0bs}) could be chosen arbitrarily in (5.4.3), and the other (P_{0ts}) determined by this choice. Similarly, for a given elbow equilibrium angle θ_{2eq}, P_{0te} could be chosen arbitrarily in (5.4.8), and P_{0be} determined by this choice. These nominal pressures will be used in equations (5.3.9) and (5.3.10) to determine the total PM pressures. The significance of these nominal pressures is that if the control inputs $\Delta p_s = \Delta p_e = 0$, the arm orientation reverts to the equilibrium angles θ_{1eq}, θ_{2eq}. Thus the arm is well-behaved even if the control loop is opened.

In practice, if the exact PM coefficients are unknown, these pressures could be easily determined experimentally, e.g. by first setting, say P_{0bs} and P_{0te} to some nominal values, then manually adjusting P_{0ts} and P_{0be} until the desired equilibrium arm orientation is obtained. In our simulations below, we arbitrarily choose $\theta_{1eq} = -\pi/4$, $\theta_{2eq} = \pi/2$.

It should be noted that larger nominal pressures increase joint stiffness and decrease joint compliance. An advantage of decreasing joint compliance is reduced susceptibility of the arm to impact disturbances such as unintended hitting of the arm or, in case of exoskeletons, unintended limb motions, which might decrease tracking accuracy. A disadvantage of decreased compliance is less flexibility in the arm, increasing the chances of discomfort and possible damage when the arm comes in contact with humans or equipment, or in the case of exoskeletons, greater discomfort in the event of limb spasms.

Nominal pressures are also important when considering the total PM pressures that will be necessary to accomplish a given task. PM pressures are practically limited to lie between maximum and minimum values, depending on the PM. However, a given task of the arm can be accomplished with any nominal pressure, if joint stiffness is not a concern. To keep all PM pressures within the allowable range, it is generally necessary to adjust the nominal PM input pressures as well as the number of parallel PMs actuating a joint. The necessary pressures will depend on the tracking task and the mass to be actuated by the arm. For instance, in the simulations at the end of this chapter, the link masses are both 10 kg. In order to accompish the tracking tasks while keeping all PM pressures within the allowable range for the PMs under consideration, i.e. $206.844 \leq P \leq 620.532$ kPa ($30 \leq P \leq 90$ psi), it is necessary to use 6 pairs of PMs actuating the shoulder and 3 pairs for the elbow, with $P_{0bs} = P_{0te} = 310.3$ kPa (45 psi), $P_{0ts} = 449.6$ kPa (65.2 psi), and $P_{0be} = 310.5$ kPa (45 psi), satisfying (5.4.3) and (5.4.8).

5.5 Two-input Sliding Mode Control for Planar Arm Actuated by Four Groups of PMs

Consider the model (5.3.13)-(5.3.15) of the planar arm actuated by four PMs. Because of our imperfect knowledge of coefficients F, K, and B for all PMs, we must assume that $a_1(\theta, \dot{\theta})$, $a_2(\theta, \dot{\theta})$, and $G(\theta, \dot{\theta})$ are imprecise. Let the extent of the imprecisions on a_1, a_2, and G be bounded by known continuous functions of θ_1, $\dot{\theta}_1$, θ_2, and $\dot{\theta}_2$. The control problem is to determine torques τ_s and τ_e to force the end effector (i.e. the end of the forearm) to follow a desired path in the spatial variables x and y (see Figure 5.2) in the presence of model imprecision on a_1, a_2, and G. It is straightforward to convert this into a tracking problem for the shoulder and elbow joint angles $\theta_1(t)$ and $\theta_2(t)$ using the inverse kinematics of the arm.

Assume we have estimates $\widehat{a}_1$, $\widehat{a}_2$ of a_1 and a_2 such that

$$|\widehat{a}_i - a_i| \leq A_i \tag{5.5.1}$$

for some known positive functions A_i, $i = 1, 2$. Further assume the control gain matrix G is unknown but that we have an estimate $\widehat{G}$ for it such that

$$G = (I + \Delta)\widehat{G} \tag{5.5.2}$$

with $|\Delta_{ij}| \leq \delta_{ij}$ for $i, j = 1, 2$ where δ_{ij} are known positive functions. Also we assume that G is nonsingular over the entire state space, and that $\widehat{G}$ is invertible, continuously dependent on the parametric uncertainty, and such that $\widehat{G} = G$ in the absence of parametric uncertainty.

Let $\theta_1^*(t)$ and $\theta_2^*(t)$ be smooth functions of time that represent the desired trajectories for the shoulder and elbow angles. Consider the two sliding surfaces $\sigma_i = 0$, $i = 1, 2$ with

$$\sigma_i = \dot{\widetilde{\theta}}_i + \mu_i \widetilde{\theta}_i = \dot{\theta}_i - \dot{\theta}_{ri} \tag{5.5.3}$$

where $\widetilde{\theta}_i = \theta_i - \theta_i^*$ are the two tracking errors, μ_1, μ_2 are positive scalar design parameters, and

$$\dot{\theta}_{ri} = \dot{\theta}_i^* - \mu_i \widetilde{\theta}_i \tag{5.5.4}$$

Then the tracking problem can be translated into finding inputs $[\Delta p_s \ \Delta p_e]^T$ that verify the individual sliding conditions

$$\frac{1}{2}\frac{d}{dt}\sigma_i^2 \leq -\eta_i|\sigma_i| \tag{5.5.5}$$

with $\eta_i > 0$ in the presence of parametric uncertainty.

Let the sliding mode control law be given by

$$\begin{bmatrix}\Delta p_s \\ \Delta p_e\end{bmatrix} = \widehat{G}^{-1}\left(\begin{bmatrix}\dot{\theta}_{r1} \\ \dot{\theta}_{r2}\end{bmatrix} - \begin{bmatrix}\widehat{a}_1 \\ \widehat{a}_2\end{bmatrix} - \begin{bmatrix}k_1\mathrm{sgn}(\sigma_1) \\ k_2\mathrm{sgn}(\sigma_2)\end{bmatrix}\right) \tag{5.5.6}$$

where k_1 and k_2 are positive constants. Then since

$$\begin{aligned}\dot{\sigma}_1 = \widehat{a}_1 - a_1 + \Delta_{11}(\dot{\theta}_{r1} - \widehat{a}_1) + \Delta_{12}(\dot{\theta}_{r1} - \widehat{a}_2) \\ - \Delta_{12}k_2\mathrm{sgn}(\sigma_2) - (1+\Delta_{11})k_1\mathrm{sgn}(\sigma_1)\end{aligned} \tag{5.5.7a}$$

$$\begin{aligned}\dot{\sigma}_2 = \widehat{a}_2 - a_2 + \Delta_{21}(\dot{\theta}_{r2} - \widehat{a}_2) + \Delta_{22}(\dot{\theta}_{r2} - \widehat{a}_2) \\ - \Delta_{21}k_1\mathrm{sgn}(\sigma_1) - (1+\Delta_{22})k_2\mathrm{sgn}(\sigma_2)\end{aligned} \tag{5.5.7b}$$

the sliding conditions (5.5.5) are verified if

$$(1-\delta_{11})k_1 \geq A_1 + \delta_{11}|\dot{\theta}_{r1} - \widehat{a}_1| + \delta_{12}|\dot{\theta}_{r1} - \widehat{a}_2| - \delta_{12}k_2 \tag{5.5.8a}$$

$$(1-\delta_{22})k_2 \geq A_2 + \delta_{21}|\dot{\theta}_{r2} - \widehat{a}_1| + \delta_{22}|\dot{\theta}_{r2} - \widehat{a}_2| - \delta_{21}k_1 \tag{5.5.8b}$$

and, in particular, if k_1 and k_2 are chosen such that

$$(1-\delta_{11})k_1 + \delta_{12}k_2 = A_1 + \delta_{11}|\dot{\theta}_{r1} - \widehat{a}_1| + \delta_{12}|\dot{\theta}_{r1} - \widehat{a}_2| + \eta_1 \tag{5.5.9a}$$

$$(1-\delta_{22})k_2 + \delta_{21}k_1 = A_2 + \delta_{21}|\dot{\theta}_{r2} - \widehat{a}_1| + \delta_{22}|\dot{\theta}_{r2} - \widehat{a}_2| + \eta_2 \tag{5.5.9b}$$

The Frobenius-Perron Theorem [28] guarantees that equations (5.5.9a) and (5.5.9b) have a unique nonnegative solution $[k_1, k_2]$. Therefore, the control law (5.5.6)

with k_1, k_2 defined by (5.5.9a) and (5.5.9b) satisfies the sliding conditions (5.5.5) in the presence of parametric uncertainties bounded as in (5.5.1), (5.5.2).

The control law (5.5.6) is known to cause chattering due to the discontinuities across the sliding surfaces. Chattering can cause excessive wear on the PM valves, and therefore is undesirable. To reduce this, we introduce boundary layers in the vicinity of the sliding surfaces. Inside the boundary layers, the control laws are linear and continuous. Outside the boundary layers, the control laws have the form given in (5.5.6). With boundary layers, the complete multi-input sliding mode control law for the arm is given by

$$\begin{bmatrix} \Delta p_s \\ \Delta p_e \end{bmatrix} = \widehat{G}^{-1} \left(\begin{bmatrix} \dot{\theta}_{r1} \\ \dot{\theta}_{r2} \end{bmatrix} - \begin{bmatrix} \widehat{a}_1 \\ \widehat{a}_2 \end{bmatrix} - \begin{bmatrix} k_1 \text{sat}(\sigma_1/\Gamma_1) \\ k_2 \text{sat}(\sigma_2/\Gamma_2) \end{bmatrix} \right) \tag{5.5.10}$$

where

$$\text{sat}(y) = \begin{cases} y, & |y| \leq 1 \\ \text{sgn}(y), & \text{otherwise} \end{cases} \tag{5.5.11}$$

and Γ_1 and Γ_2 are the boundary layer thicknesses, i.e. the boundary layers are defined by

$$L_1 = \{(\theta_1, \dot{\theta}_1) : |\sigma_1(\theta_1, \dot{\theta}_1)| \leq \Gamma_1\} \tag{5.5.12a}$$

$$L_2 = \{(\theta_2, \dot{\theta}_2) : |\sigma_2(\theta_2, \dot{\theta}_2)| \leq \Gamma_2\} \tag{5.5.12b}$$

Thus, when the state trajectories are outside their respective boundary layers, the trajectories approach and reach the boundary layers in finite times due to (5.5.7a) and (5.5.7b) when k_1, k_2 are chosen to satisfy (5.5.9a) and (5.5.9b). Once inside the boundary layers, the state trajectories θ_1 and θ_2 are governed by (5.5.3), i.e. taking the Laplace transform of $\widetilde{\theta}_i(t)$ we have

$$\widetilde{\theta}_i(s) = \frac{1}{s + \mu_i} \sigma_i(s) \tag{5.5.13}$$

which, together with the fact that $|\sigma_i| \leq \Gamma_i \; \forall t \geq t_1$ with t_1 finite, yields

$$\lim_{t_0 \to \infty} \sup_{t \geq t_0} |\theta_i(t) - \theta_i^*(t)| \leq \frac{\Gamma_i}{\mu_i} \tag{5.5.14}$$

for $i = 1, 2$.

Therefore, we have that the sliding mode control law (5.5.10) guarantees that the state trajectories reach their respective boundary layers in finite times whatever the initial states, and inside the boundary layers constrains trajectories to stay inside them for all later time and approach neighborhoods of $\tilde{\theta}_i = 0$ asymptotically as $t \to \infty$. Asymptotic tracking of the shoulder and elbow angles to within guaranteed accuracy is therefore obtained in spite of modeling errors which may be present in PM coefficients, masses, distances, etc. Thus the desired spatial path is followed by the end effector within an error bound.

5.6 Simulation Results

The planar arm of Figure 5.3 with opposing-pair PMs of the type described in Section 5.2 actuating the shoulder and elbow joints is simulated using a 4th-order Runge-Kutta algorithm with a step size of 0.01 seconds. Let $l_1 = l_2 = 0.46$ m, $l_{c1} = l_{c2} = 0.23$ m, $m_1 = m_2 = 10$ kg, $r_s = 7.62$ cm, and $r_e = 5.08$ cm, $n_s = 6$ and $n_e = 3$. For these simulations we assume all physical quantities of the arm, i.e. masses, lengths, etc. are exactly known, but that the PM coefficients, i.e. F, K, and B are not known with precision. Assume all 12 shoulder PMs (6 pairs) are matched to each other, but not to the elbow PMs. Similarly assume all 6 elbow PMs (3 pairs) are matched to each other, but not to the shoulder PMs.

The sliding mode controller is designed according to (5.5.10) with $\hat{a}_1$ and $\hat{a}_2$ given by (5.3.14) and $\hat{G}$ given by (5.3.15) using ideal values for all F, K, and B coefficients. To investigate robustness of the sliding mode controller, we randomly choose three sets of actual (nonideal) F, K, and B coefficients from a uniform distribution within $\pm 50\%$ of their nominal values. The coefficients used are listed in Table 5.1.

The coefficients listed in Table 5.1 are those that produced the greatest tracking error (while maintaining a random selection), so that we could see how close to the predicted error bounds (5.5.14) the actual tracking errors were. Other choices for coefficients produced less tracking error, so are not reported. In the process of choosing which sets of coefficients to use for the simulations, it was noticed that tracking error was by far the most sensitive to variations in the coefficient, F_1. This is perhaps not surprising, since from (5.2.1), (5.2.4) the PM contractile force F is directly proportional this coefficient, and F has a more direct effect on the PM force than the other coefficients (K_0, K_1, B_0, and B_1).

Table 5.1 - PM coefficient sets used for the simulations. $Si(Ei)$ = shoulder (elbow) PM coefficients for actual PM coefficient set i.

Coefficient	**Ideal**	$S1$	$E1$	$S2$	$E2$	$S3$	$E3$
$F_0(\times 10^2)$	1.79	1.53	2.58	0.984	2.34	1.69	2.40
F_1	1.39	0.763	1.67	1.49	0.812	1.99	0.722
K_0	5.71	7.17	7.70	6.86	8.25	5.52	6.75
$K_1(\times 10^{-2})$	3.07	4.28	2.18	4.49	4.35	2.82	2.70
B_{0i}	1.01	0.794	0.965	1.26	1.11	1.36	1.35
$B_{1i}(\times 10^{-3})$	6.91	5.19	4.02	8.57	5.21	7.08	6.93
$B_{0d}(\times 10^{-1})$	6.00	8.60	8.11	5.59	8.24	4.22	7.26
$B_{1d}(\times 10^{-4})$	-8.03	-5.07	-8.53	-9.11	-8.14	-9.41	-7.46

For all simulations, we use $\mu_i = 5.0$ and $\Gamma_i = 1.0$. From simulations with parametric uncertainties within the $\pm 50\%$ range, we find that $A_1 = 12.5$ and $A_2 = 15.0$ satisfy inequalities (5.5.1). From (5.5.2) we also have $\Delta = [\Delta_{ij}]$ where

$$|\Delta_{ij}| \leq \delta_{ij} = \begin{cases} 0.5, & i = j \\ 0, & i \neq j \end{cases} \tag{5.6.1}$$

Using these values and the simulated functions $|\dot{\theta}_{ri} - \widehat{a}_i|$, $i = 1, 2$, we find $k_i = 50$, $i = 1, 2$, satisfies (5.5.8), resulting in closed-loop stability and convergence of the trajectories to the interior of the boundary layers and a guaranteed tracking precision of $\Gamma_i/\mu_i = 0.2$ radians for both joints. Also let $\theta_{1eq} = -\pi/4$, $\theta_{2eq} = \pi/2$, $P_{0bs} = P_{0te} = 310.3\,\text{kPa}$ (45 psi), resulting in $P_{0ts} = 490.4\,\text{kPa}$ (71.1 psi) and $P_{0be} = 310.5$ kPa (45 psi), satisfying (5.4.3) and (5.4.8). With these nominal pressures and with n_s, n_e as above, all PM pressures remain in the allowable range $206.844 \leq P \leq 620.532$ kPa ($30 \leq P \leq 90$ psi) for all control tasks in this section.

We investigate controller performance for three tracking tasks for the end effector in x-y space: a sinusoidal spline, a vertical line, and a circle.

Sinusoidal spline

The desired spatial path is given by:

$$x_d(t) = 0.1524 + 0.1219t \text{ m} \tag{5.6.2a}$$

$$y_d(t) = 0.39624 + 0.24384\sin(0.4\pi t - \pi/2) \text{ m} \tag{5.6.2b}$$

where $0 \leq t \leq 5$ seconds.

The spatial tracking performance for the true plant with PM coefficients in set 1 is shown in Figure 5.4. Tracking performance when PMs are described by coefficient sets 2 and 3 are similar to Figure 5.4. It will be noted that there is some spatial tracking error, which is to be expected due to the parameter errors. An initial transient can also be seen, due to the initial pressure adjustment that is necessary to produce accurate tracking.

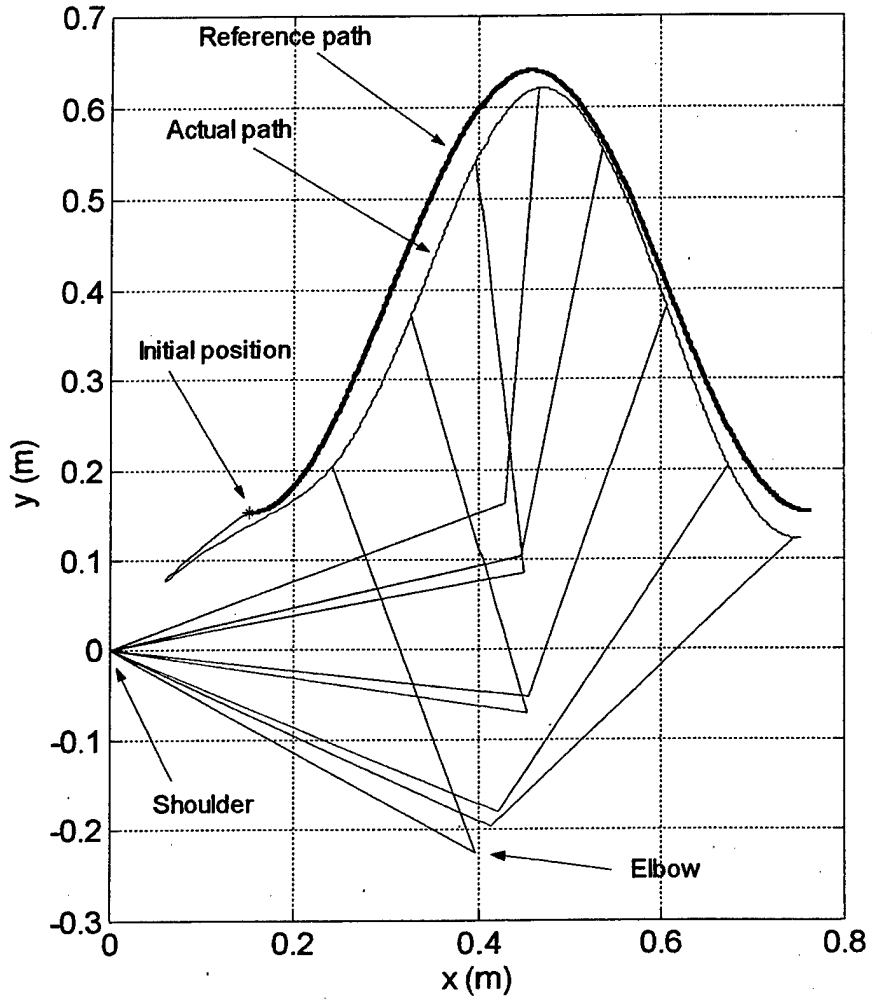


Figure 5.4 - Spatial tracking behavior, PMs in coefficient set 1, $m_1 = m_2 = 10$ kg.

Figure 5.5 shows the control efforts Δp_s, Δp_e that produced the tracking performance in Figure 5.4. It will be noted that with the nominal pressures P_{0ts}, P_{0bs}, P_{0te}, and P_{0be} given above, the PM input pressures remain within the allowable range for these PMs. The elbow angle tracking error (which is larger than the shoulder angle

tracking error in this case) for coefficient sets 1, 2, and 3 is shown in Figure 5.6. It is seen that for all systems the tracking error is within predicted bounds, i.e.

$$\lim_{t_0 \to \infty} \sup_{t \geq t_0} |\theta_i(t) - \theta_i^*(t)| \leq \frac{\Gamma_i}{\mu_i} = 0.2 \text{ radians} \tag{5.6.3}$$

This tracking error is obtained after the initial transients have died away.

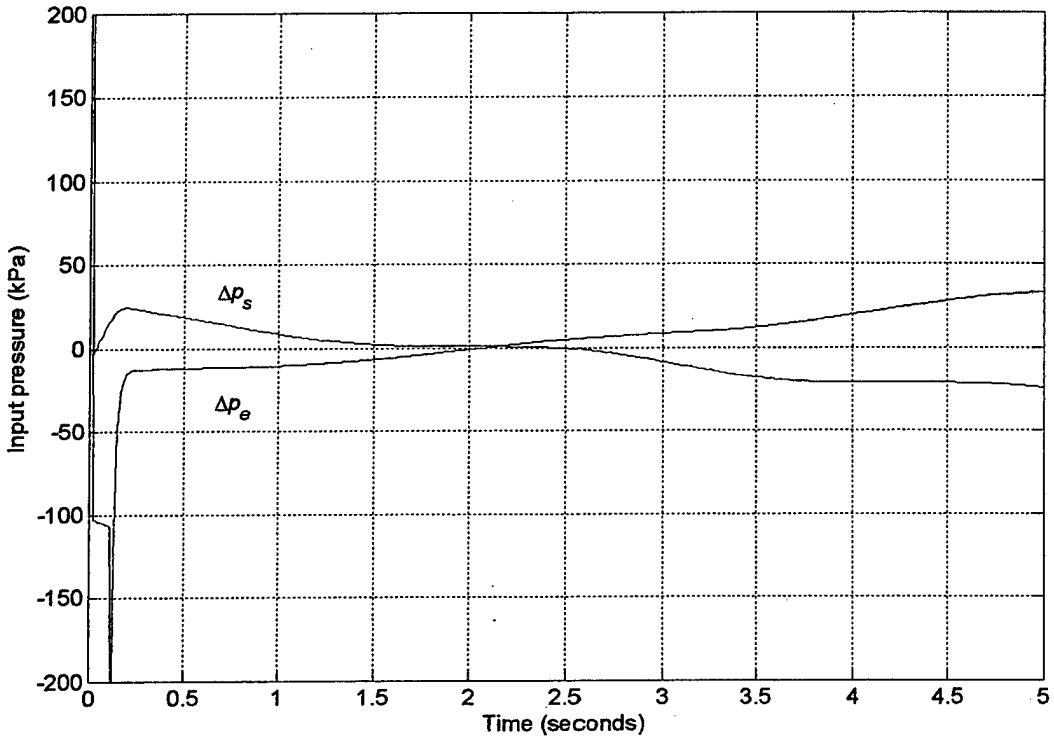


Figure 5.5 - Control effort producing tracking performance of Figure 5.4.

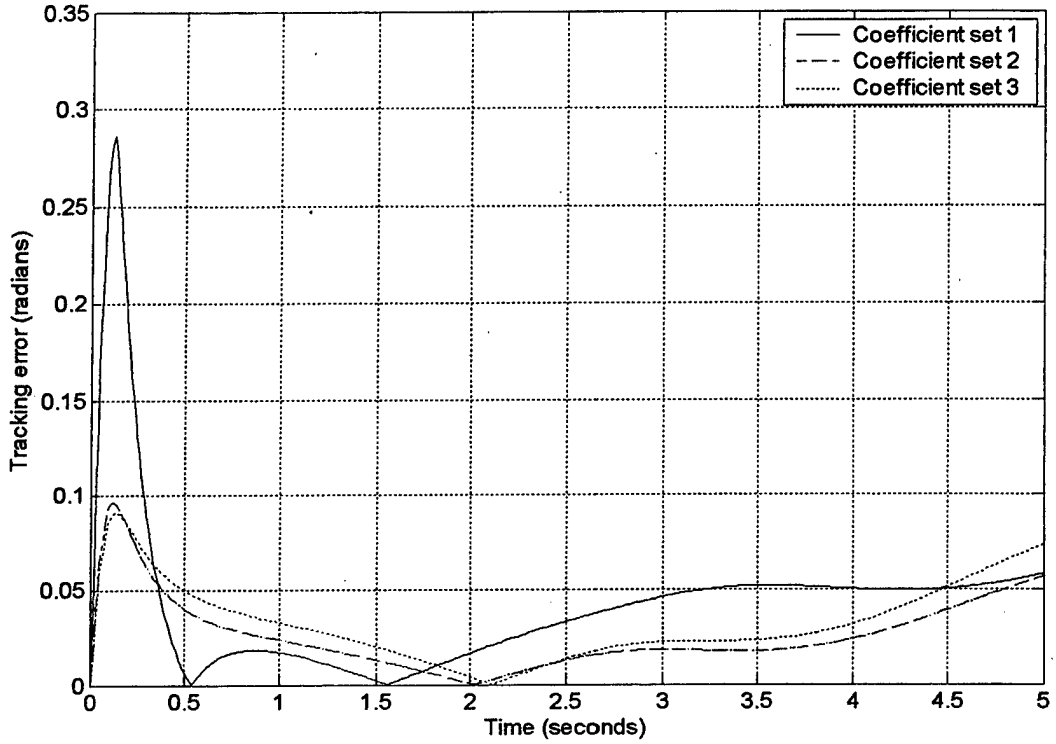


Figure 5.6 - Elbow angle tracking errors for 3 different plants (PM coefficient sets 1, 2, and 3), sinusoidal spline reference trajectory, $m_1 = m_2 = 10$ kg.

Vertical line

The desired spatial path for the vertical line is given by:

$$x_d(t) = 0.6096 \text{ m} \tag{5.6.4a}$$

$$y_d(t) = 0.39624 + 0.24384\sin(0.4\pi t - \pi/2) \text{ m} \tag{5.6.4b}$$

The spatial tracking performance for PMs with coefficients in set 3 is shown in Figure 5.7. Tracking performance for other sets is similar. The corresponding control effort is shown in Figure 5.8. The elbow angle tracking error (which is larger than the shoulder angle tracking error in this case) for PMs with coefficients in sets 1, 2, and 3 is shown in Figure 5.9. It is seen that for all systems the tracking error is within predicted bounds and PM pressures remain within the allowable range.

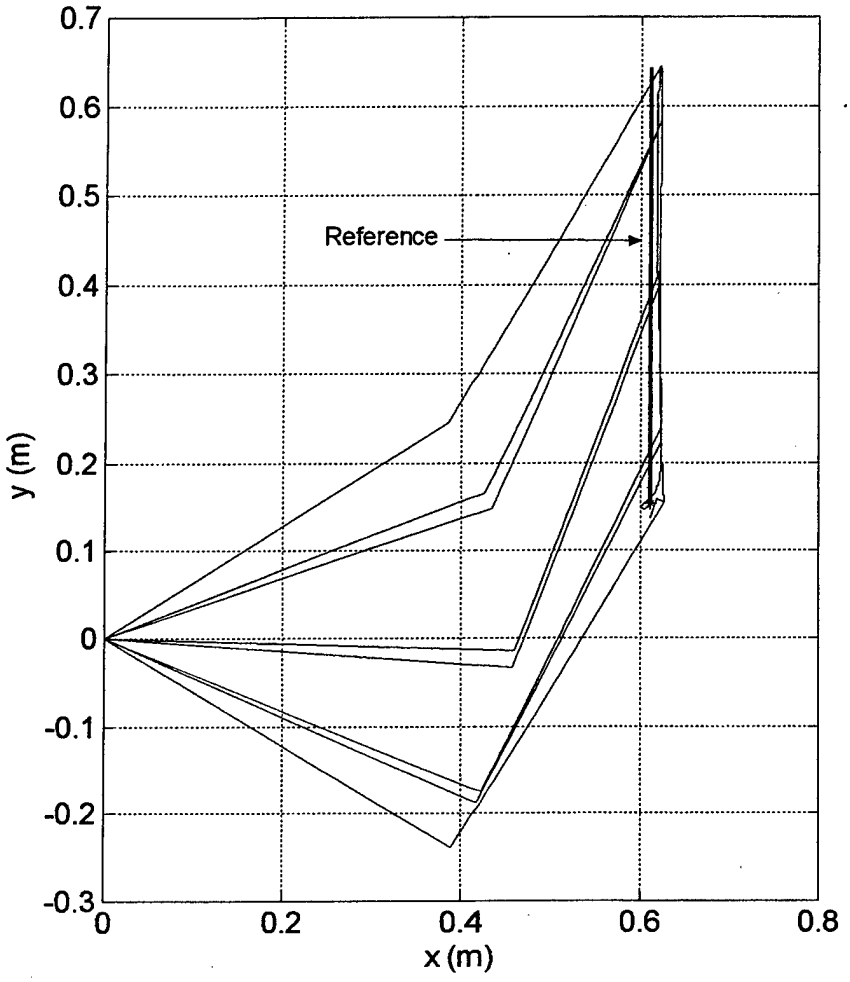


Figure 5.7 - Spatial tracking behavior, PMs in coefficient set 3, $m_1 = m_2 = 10$ kg

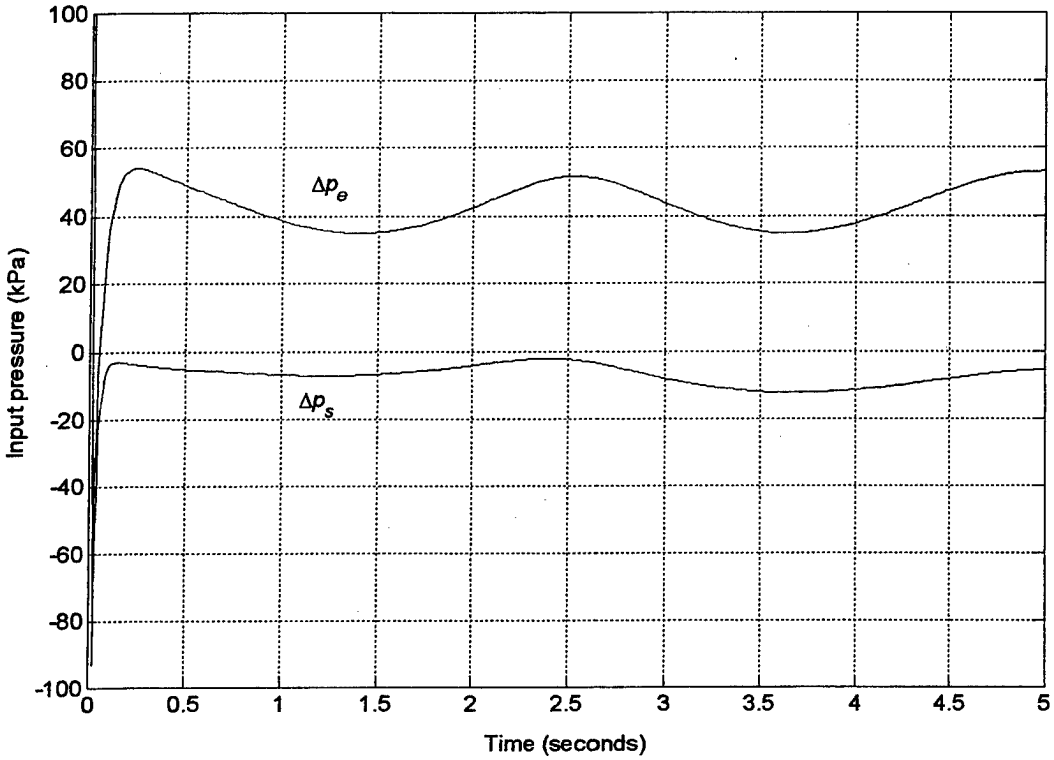


Figure 5.8 - Control effort producing tracking performance of Figure 5.7.

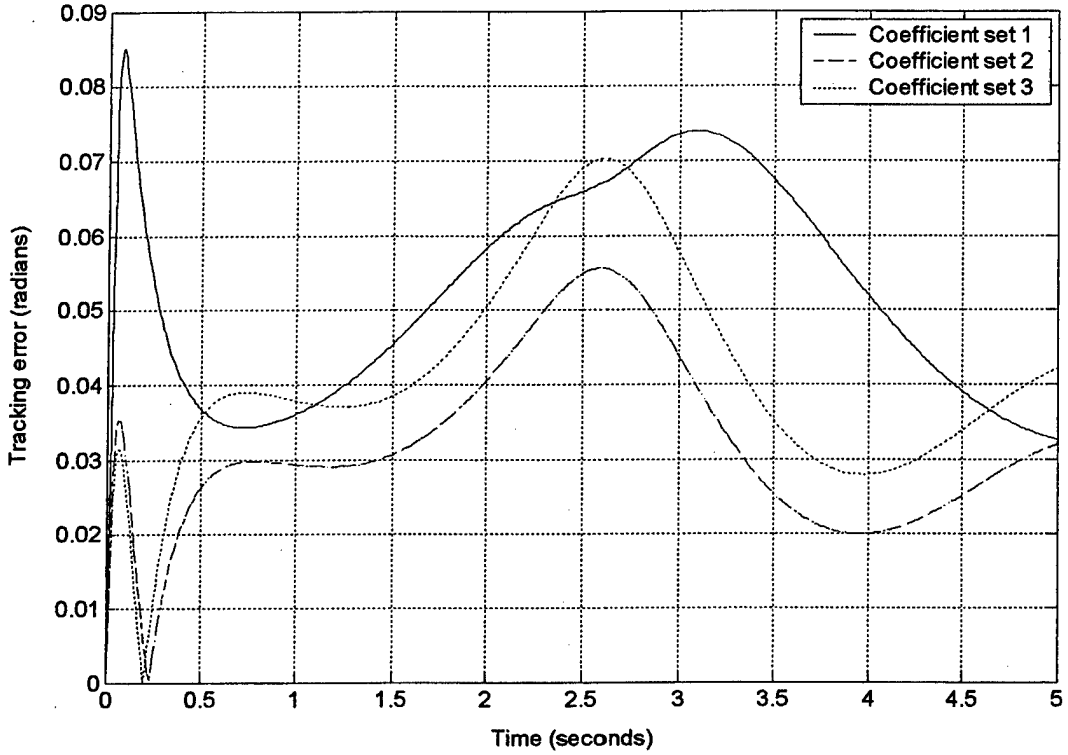


Figure 5.9 - Elbow angle tracking errors for 3 different plants (PM coefficient sets 1, 2, and 3), vertical line reference trajectory, $m_1 = m_2 = 10$ kg.

Circle

The desired spatial path for the circle is given by:

$$x_d(t) = 0.36576 + 0.3048\sin(0.4\pi t - 0.7754)) \text{ m} \qquad (5.6.5a)$$

$$y_d(t) = 0.36576 + 0.3048\cos(0.4\pi t + 2.3462) \text{ m} \qquad (5.6.5b)$$

The spatial tracking performance for PMs with coefficients in set 1 is shown in Figure 5.10. The corresponding control effort is shown in Figure 5.11. The elbow angle tracking error (which is larger than the shoulder angle tracking error in this case) for PMs with coefficients in sets 1, 2, and 3 is shown in Figure 5.12. It is seen that for all systems the tracking error is within predicted bounds and PM pressures remain within the allowable range.

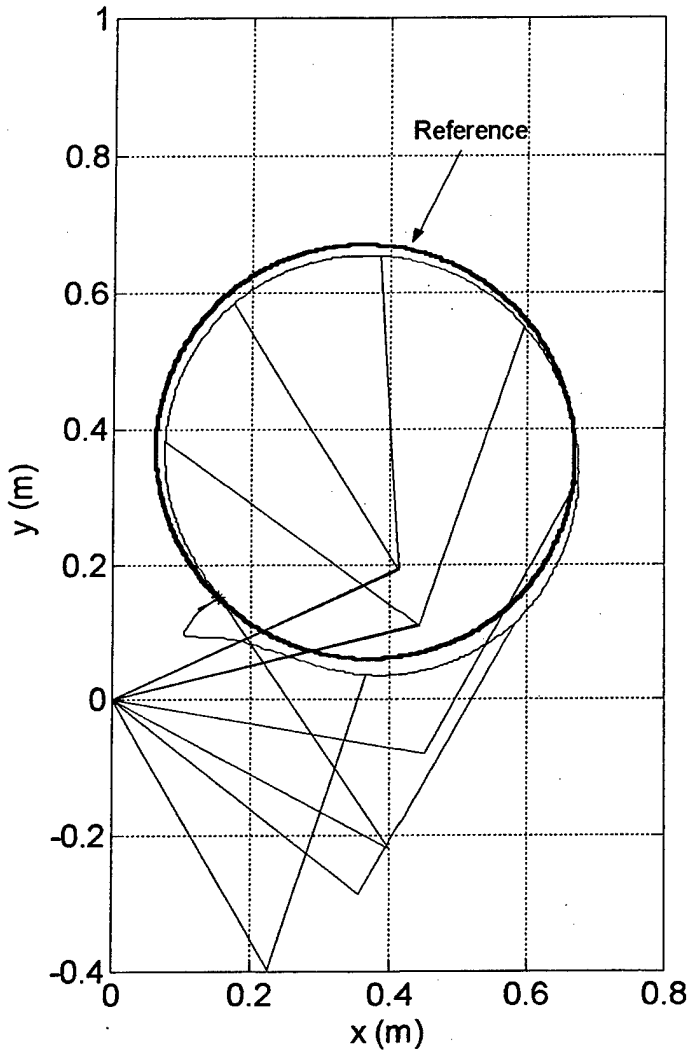


Figure 5.10 - Spatial tracking behavior, PMs in coefficient set 1, $m_1 = m_2 = 10$ kg

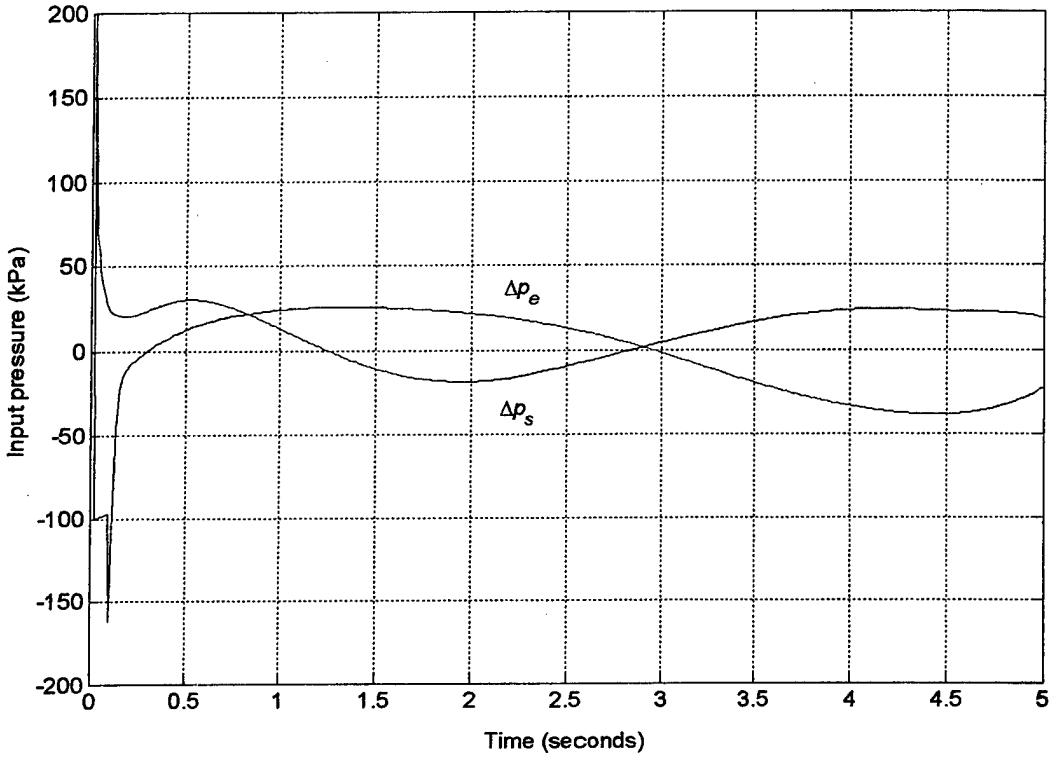


Figure 5.11 - Control effort producing tracking performance of Figure 5.10.

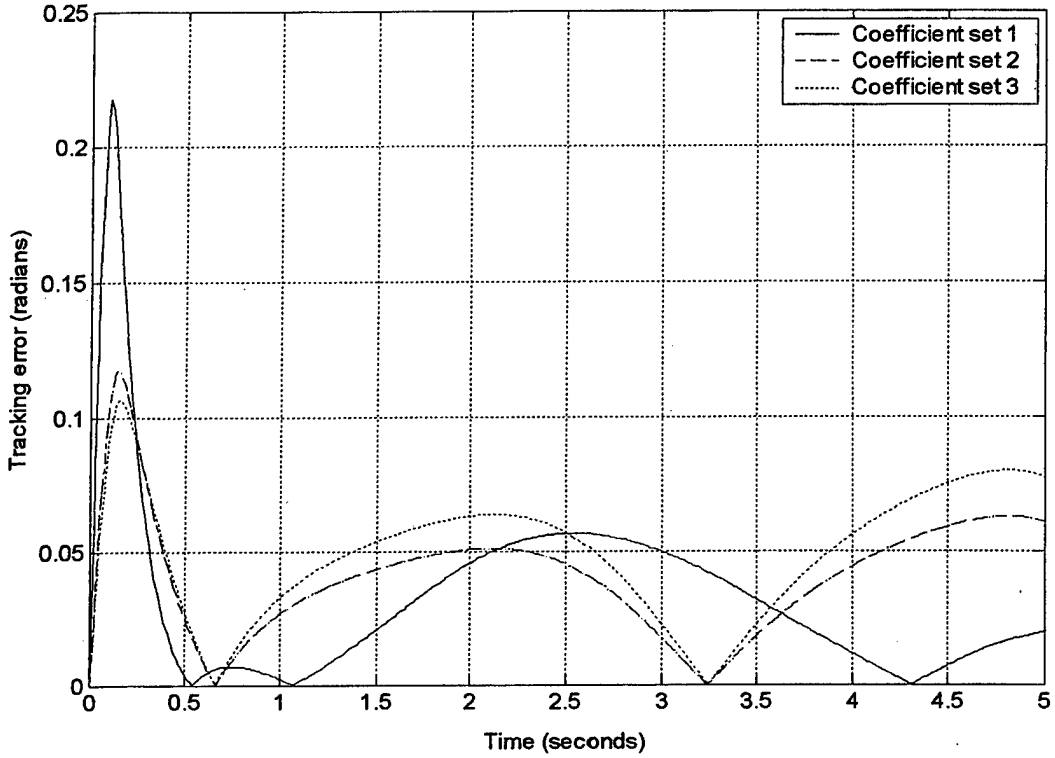


Figure 5.12 - Elbow angle tracking errors for 3 different plants (PM coefficient sets 1, 2, and 3), circle reference trajectory, $m_1 = m_2 = 10$ kg.

Sinusoidal spline, doubled mass

In practical applications, it may be expected that the mass actuated by the arm will change. To investigate the robustness of the sliding controller to changing masses, we increased the arm masses m_1, m_2 each by a factor of 2 to 20 kg and used the same controller as above to track the sinusoidal spline reference trajectory (5.6.2). Both the shoulder and elbow angle tracking errors are again within predicted bounds, indicating that the sliding mode controller is robust to changes in mass.

Figure 5.13 shows the control effort produced by the sliding controller for this plant. It is seen that with the above nominal pressures, the PM pressures remain within the allowable range for the duration of the control process.

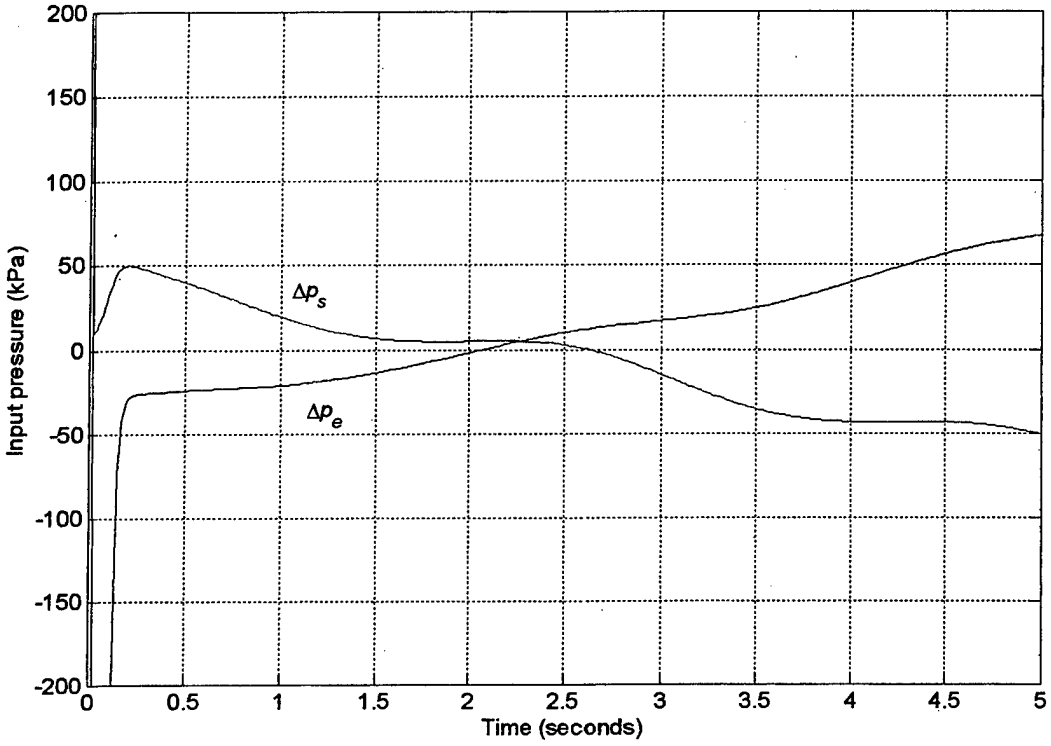


Figure 5.13 - Control effort, sinusoidal spline reference trajectory, coefficients in set 1, $m_1 = m_2 = 20$ kg.

5.7 Discussion

A two-input sliding mode controller has been designed for a 2 DOF planar arm assembly with highly nonlinear pneumatic muscle actuators in opposing pair configurations actuating the shoulder and elbow joints. Designation of these joints as shoulder and elbow is arbitrary and nominal; however, it is convenient for considering applications in which pneumatic muscle actuated devices could provide joint loading to generate or assist extremity motion, or maintain extension of muscles with contractures.

The control input for the planar assembly enters the process through nonlinear spring and friction coefficients and a nonlinear contractile force term that are contained within a mathematical model for the pneumatic muscle actuators. A dynamic model for the arm with four PM actuators is derived, and this is put in a form suitable for sliding mode control. A relationship between static internal pressures is derived to give stable arm behavior in the absence of any control signal.

Simulations of closed-loop tracking were performed with sinusoidal spline, vertical line, and circle paths desired for the assembly end effector. These paths are generic in nature; however, they can serve well as potential building blocks for an ample variety of more task-oriented end effector paths. Data from the motion biomechanics literature ([29],

[30]), as well as dedicated biomechanical experimentation, could be used to identify actual end effector paths for practical activities such as feeding and grooming.

Closed-loop tracking performance, resulting from simulations, is in line with theoretically predicted behavior. Closed-loop tracking with several arm models with coefficients within a ±50% range are shown to agree with theoretical results. The controller is shown to be robust for a 100% change in arm masses. The effects of heating are also presented. It is seen that some reasonable amount of heating may be tolerated. Thus, sliding mode control is shown to be a very promising method for control of systems containing pneumatic muscle actuators, including devices that could potentially benefit persons with neuromuscular or musculoskeletal pathologies.

6 Neuro-fuzzy Modeling of PM

6.1 Introduction

Recently, neural networks, fuzzy systems and combinations of these have obtained great success in modeling nonlinear dynamics. Of them, recurrent neuro-fuzzy systems are most effective and efficient ([31]-[36]). They combine the dynamic mapping capability of common recurrent neural networks with the human-like decision-making capability of fuzzy systems. Some theoretical results ([38], [39]) also prove their potential for approximating a large class of dynamics to an arbitrary degree of accuracy. Therefore, we implement this technique to develop a more accurate model for the pneumatic muscle by using soft-computing techniques, i.e. fuzzy logic combined with neural networks, based on recorded data.

6.2 Recurrent Neuro-Fuzzy Modeling for Pneumatic Muscle

According to studies of Reynolds et al. [17], the dynamics of a pneumatic muscle system can be described by a second-order nonlinear differential equation. Therefore, in order to model the PM with a neuro-fuzzy system, we choose a model of the form:

$$\hat{y}(k) = f\left(\hat{y}(k-1), \hat{y}(k-2), x(k-1), x(k-2)\right) + e(k) \tag{6.2.1}$$

where k is the time step, $\hat{y}$ is the estimated length of the muscle and x is the voltage input to the valve. The above equation can be approximated by a recurrent neuro-fuzzy inference system as shown in Fig. 6.1.

In the system of Figure 6.1, there are 81 Sugeno-type rules with the following expression used in the rule base:

$$\begin{aligned} &RULE\ i \quad i = 1, 2, \cdots, 81: \\ &IF\ \hat{y}(k-1)\ is \begin{Bmatrix} \mu_{11} \\ \mu_{12} \\ \mu_{13} \end{Bmatrix} AND\ \hat{y}(k-2)\ is \begin{Bmatrix} \mu_{21} \\ \mu_{22} \\ \mu_{23} \end{Bmatrix} AND\ x(k-1)\ is \begin{Bmatrix} \mu_{31} \\ \mu_{32} \\ \mu_{33} \end{Bmatrix} AND\ x(k-2)\ is \begin{Bmatrix} \mu_{41} \\ \mu_{42} \\ \mu_{43} \end{Bmatrix} \\ &THEN\ y = C_i \end{aligned} \tag{6.2.2}$$

where μ_{jm} $(j = 1, 2, 3, 4;\ m = 1, 2, 3)$ are input membership functions and C_i $(i = 1, \ldots 81)$ represents 81 local models in fuzzy partitions. In (6.2.2), AND means the fuzzy "AND" operator, or T-norm. In this chapter, the product T-norm is adopted for "AND."

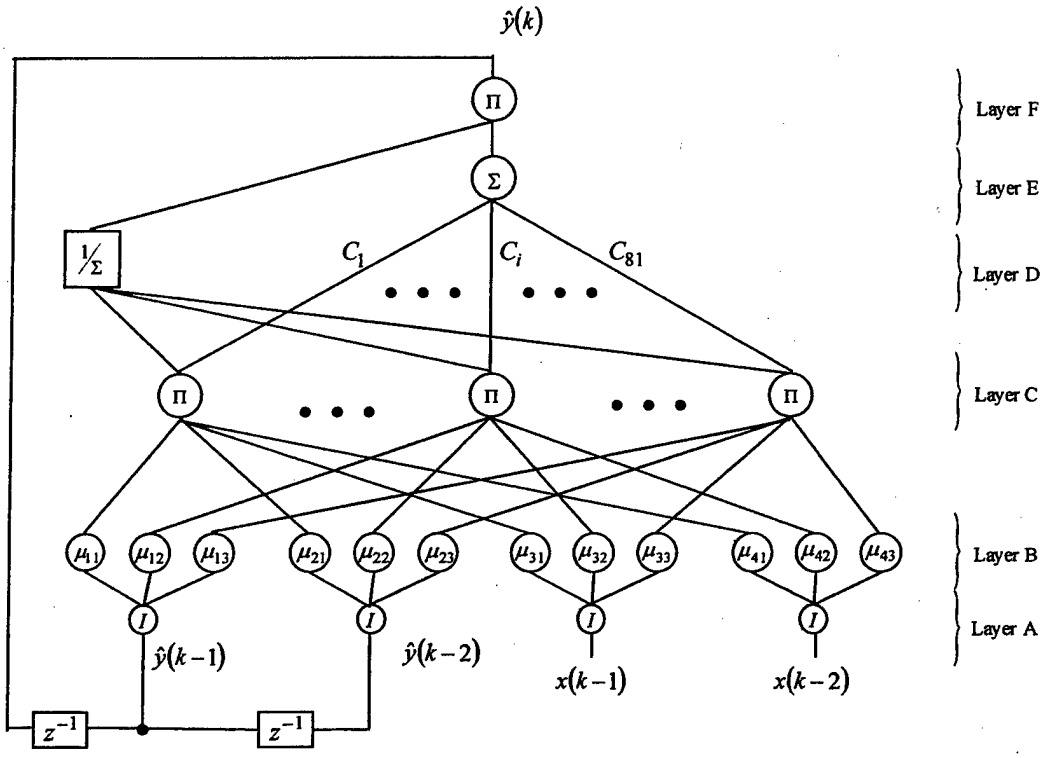


Fig. 6.1 - Recurrent neuro-fuzzy inference system

6.3 Topologies

The recurrent neuro-fuzzy inference system is six-layered. Layer A is an input layer, and neurons represent fuzzy variables found in the fuzzy inference machine. They are volts applied to the air-inlet valve and length of muscle estimate in the past two time steps. Layer B is a term layer. Nodes in this layer are called term nodes and correspond to linguistic values (small, medium and large) of the input variables. The membership functions μ are described below. Layer C is a rule layer. Nodes with label $\prod$ represent fuzzy reasoning rules collected in the rule base. Note that label $\prod$ means that the product "AND" operator is employed to calculate the degree of match for parts of the premise, which evaluates the degree of activation of rules. Layer D is a defuzzification layer. Its first neuron calculates the reciprocal of sum of degrees of match of rules, and connection weights on links between other neurons. The other neurons in layer D define local models C_i appearing in the consequents of rules. The neuron in layer E is an aggregation neuron, which aggregates the weighted local models. Layer F is the output layer. The output of the neuron in this layer represents the estimated length of muscle in the current instant.

In all, five different neurons are defined to support the network. These are described below, where v is the neuron's input and z is its output.

1. A unity neuron **I** with input-output relationship defined by

$$z = v \tag{6.3.1}$$

2. A sum neuron Σ in layer E implements a sum operation, which is used to aggregate the weighted local models:

$$z = \sum c_i v_i \tag{6.3.2}$$

3. A product neuron with label Π is used to implement a product "AND" operation

$$z = \prod w_i v_i \tag{6.3.3}$$

4. An inverse neuron with label $1/\Sigma$ in layer D implements the reciprocal operation in the defuzzification phase:

$$z = f\left(\sum w_i v_i\right) = \frac{1}{\sum w_i v_i} \tag{6.3.4}$$

5. A term neuron with label $\mu_{jm}\ (j = 1, 2, 3, 4;\ m = 1, 2, 3)$ in layer B implements a membership function defined below:

$$\mu_{jm}(v) = \begin{cases} \dfrac{v - a_{jm}}{b_{jm} - a_{jm}} & a_{jm} \le v \le b_{jm} \\ \dfrac{c_{jm} - v}{c_{jm} - b_{jm}} & b_{jm} \le v \le c_{jm} \\ 0 & \text{otherwise} \end{cases} \tag{6.3.5}$$

6.4 Structure Learning via VISIT

At the beginning, the RNFIS in Figure 6.1 is empty without any rules or fuzzy sets. The *structure* of the RNFIS refers to the number of rules, membership functions, and their properties (i.e. centers and spreads). A simple self-organizing algorithm known as the Variable Input Spread Inference Training algorithm (VISIT) [37], developed by the PI and coworkers, is used to perceive structure features from a sequence of training data. VISIT is a variation of the well-known Modified Learning From Examples (MLFE) algorithm for identification of fuzzy systems from data. In VISIT, the membership functions can be any shape in general, as long as they are convex. In this chapter, we will

use asymmetrical triangular input membership functions and singleton output membership functions. The fuzzy system determined by VISIT is then used as an initial condition for further tuning via backpropagation. This is done to improve BP's chances of converging to the global minimum of the error criterion rather than a local minimum, thus producing a more accurate model of the process.

To begin the VISIT algorithm, training constants σ_0, λ, ε, and w are specified by the user. The constant σ_0 is the initial value for the spread of the first membership function on each universe. The constant λ determines when a new membership function is created on a universe of discourse. If a new training point is within a λ-cut of an existing membership function on a universe, a new membership function is not created on that universe. The constant ε is the maximum identification error tolerated before a new rule and new memberships are added to the fuzzy system. Finally, w determines the amount of overlap between adjacent memberships on a universe. We now give the VISIT algorithm.

VISIT Algorithm for Function Approximation

1. Set $i=1$. Get the first training pair $tp^1=(x_1, x_2, \ldots, x_n, y)^1$. On each input universe of discourse form a fuzzy set A_j^1 characterized by a membership function $\mu_{A_j^1}(x_j^1)$ with center at $m_j^1=x_j^1$ and spread σ_0. On the output universe of discourse form a fuzzy set B^1 characterized by a singleton membership function with support y^1.

2. Add the rule

   If x_1 is A_1^1 and x_2 is A_2^1 and ... and x_n is A_n^1 then y is B^1.

3. If there are no more training pairs, the training cycle is completed. Otherwise, increment i and get the next training pair $tp^i=(x_1, x_2, \ldots, x_n, y)^i$.

4. If $\left|f(tp^i)-y^i\right|<\varepsilon$ where $f(tp^i)$ denotes the crisp output of the existing fuzzy system evaluated at the new training pair tp^i, discard tp^i and go to 3.

5. On the jth input universe, evaluate all membership functions at x_j^i. Call the fuzzy set whose membership function is maximum $A_j^{\max}$. If $x_j^i \notin \lambda-cut(A_j^{\max})$, form a new fuzzy set A_j^{new} on the jth universe characterized by a membership function

with center $m_j^{new} = x_j^i$ and left and right spreads σ_j^L and σ_j^R as in step 6 below. On the output universe form a new fuzzy set B^{new} characterized by a singleton membership function with support y^i.

6. On each input universe on which a new membership function was added in 5, the spreads of each side of the new membership function and its nearest neighbors on the left and right are re-calculated as follows. The right spread of the new membership function and the left spread of the nearest right neighbor are re-calculated as

$$\sigma_j^R = \frac{1}{w}\left|m_j^{new} - m_j^{nRn}\right| \tag{6.4.1}$$

The left spread of the new membership function and the right spread of the nearest left neighbor are re-calculated as

$$\sigma_j^L = \frac{1}{w}\left|m_j^{new} - m_j^{nLn}\right| \tag{6.4.2}$$

where m^{nRn} denotes the nearest existing center to the right of m_j^{new} and m_j^{nLn} denotes the nearest existing center to the left of m_j^{new}.

7. Add the rule

If x_1 is A_1 and x_2 is A_2 and ... and x_n is A_n then y is B^{new}

If there is no other rule in the rule base that is inconsistent (i.e. same premise, different consequent) with this rule, where the fuzzy sets in the premise are the ones maximized by the corresponding inputs

8. If there is another rule (rule p generated from tp^k) in the rule base that is inconsistent with the rule formed in 7 (rule q generated from the present training pair tp^l), define fuzzy system f_p (f_q) to be the fuzzy system with rule p (q) included in the rule base but rule q (p) omitted from the rule base. Retain rule p and omit rule q if

$$\left|f_p\left(tp^k\right) - y^k\right| + \left|f_p\left(tp^l\right) - y^l\right| < \left|f_q\left(tp^k\right) - y^k\right| + \left|f_q\left(tp^l\right) - y^l\right| \tag{6.4.3}$$

Otherwise, include rule q and omit rule p.

9. Go to 3.

Comment:

It is possible that the rule to be added in step 7 has the same premise but a different consequent than an existing rule, i.e. the newly created rule is *inconsistent* with an existing rule. In such a case, we need to determine which rule provides a better match to the data so we know which rule to retain in the rule base and which rule to omit. To do this, we create two fuzzy systems, each with only one of the inconsistent rules retained, but with the other inconsistent rule omitted. In (6.4.3), a comparison of the two fuzzy systems is made on both training pairs that generated the two inconsistent rules. The fuzzy system having less total identification error for both training pairs is retained and the other omitted. In this way, we determine which of the two inconsistent rules does a better job of describing the data and retain it in the rule base, omitting the other.

6.5 Backpropagation Training Algorithm

For fuzzy systems, training algorithms provide a means for refining inference rules in the form of (6.2.2). A general technique in most algorithms is that input and output data recorded from the actual process are used in the calculation of the current output estimate of the fuzzy system. Therefore, these methods *statically* refine fuzzy rules, and therefore are more suitable to excute static tasks such as function mapping and pattern classification rather than modeling of dynamics. In contrast to these static training techniques, the recurrent neuro-fuzzy inference system implies a kind of dynamic refinement for fuzzy inference rules, where past outputs of the fuzzy sytem itself and recorded input data from the process are used to calculate the current output esstimate of the fuzzy system due to the existence of some global feedback connections. As a result, rules are tuned in dynamic ways, and the resulting systems can better approximate the dynamics of a wide class of nonlinear systems with any degree of accuracy ([38], [39]).

Given N recorded input-output pairs from the process $\{x(n), y(n)\}, n = 1, \ldots, N$, we can train recurrent neuro-fuzzy systems by using the well-known error backpropagation (BP) training algorithm. To control the training process, the mean square error (MSE) is chosen as the performance index for evaluation of the effectiveness of the fuzzy system. If $y(n)$ is the nth recorded output (length) of the pneumatic muscle and $\hat{y}(n)$ be the model output, the MSE criterion is written as:

$$MSE = \left(\frac{1}{N}\right)\sum_{n=1}^{N} e^2(n) \tag{6.5.1}$$

where the training error $e(n)$ is defined as

$$e(n) = y(n) - \hat{y}(n) \tag{6.5.2}$$

The weights $W(t)$ of the neural network, which in this case consist of the membership function parameters, are adjusted by the well-known BP training algorithm, which is written as follows:

$$\begin{aligned} W(t+1) &= W(t) + \eta \frac{\partial e^2(n)}{\partial W} + \gamma \Delta W(t) \\ &= W(t) - 2\eta e(n) \frac{\partial \hat{y}(n)}{\partial W} + \gamma \Delta W(t) \end{aligned} \tag{6.5.3}$$

where $W = [a_{11}, b_{11}, c_{11}, \cdots, a_{13}, b_{13}, c_{13}, \cdots, a_{43}, b_{43}, c_{43}, C_1, \cdots, C_{81}]$ is a vector of tuning parameters in the recurrent neuro-fuzzy inference system, the integer t denotes the step during the training process, coefficients η and γ are learning parameters that control the training process, and $\Delta W(t) = W(t) - W(t-1)$. Larger η will accelerate the training process but may introduce oscillations. Larger γ will greatly smooth but slow the training. The parameters C_i, a_{jm}, b_{jm}, c_{jm} are updated according to the following rules

$$C_i(t+1) = C_i(t) - 2\eta e(n) \frac{\partial \hat{y}(n)}{\partial C_i} + \gamma \Delta C_i(t) \tag{6.5.4}$$

$$a_{jm}(t+1) = a_{jm}(t) - 2\eta e(n) \frac{\partial \hat{y}(n)}{\partial \mu_{jm}} \cdot \frac{\partial \mu_{jm}}{\partial a_{jm}} + \gamma \Delta a_{jm}(t) \tag{6.5.5}$$

$$b_{jm}(t+1) = b_{jm}(t) - 2\eta e(n) \frac{\partial \hat{y}(n)}{\partial \mu_{jm}} \cdot \frac{\partial \mu_{jm}}{\partial b_{jm}} + \gamma \Delta b_{jm}(t) \tag{6.5.6}$$

$$c_{jm}(t+1) = c_{jm}(t) - 2\eta e(n) \frac{\partial \hat{y}(n)}{\partial \mu_{jm}} \cdot \frac{\partial \mu_{jm}}{\partial c_{jm}} + \gamma \Delta c_{jm}(t) \tag{6.5.7}$$

The quantities $\frac{\partial \hat{y}(n)}{\partial \mu_{jm}}$ and $\frac{\partial \hat{y}(n)}{\partial C_i}$ are calculated by the following two formulas:

$$\begin{aligned} \frac{\partial \hat{y}(n)}{\partial C_i} &= \frac{\partial \hat{y}(n)}{\partial o_6^1} \cdot \frac{\partial o_6^1}{\partial net_6^1} \cdot \frac{\partial net_6^1}{\partial C_i} \\ &= \frac{\partial net_6^1}{\partial o_5^1} \cdot \frac{\partial o_5^1}{\partial net_5^1} \cdot \frac{\partial net_5^1}{\partial C_i} \\ &= o_4^1 \cdot \frac{\partial net_5^1}{\partial C_i} \\ &= o_4^1 \cdot o_4^{i+1} \end{aligned} \tag{6.5.8}$$

$$\begin{aligned}\frac{\partial \hat{y}(n)}{\partial \mu_{jm}} &= \frac{\partial \hat{y}(n)}{\partial o_6^1}\cdot\frac{\partial o_6^1}{\partial net_6^1}\cdot\frac{\partial net_6^1}{\partial \mu_{jm}} \\ &= o_4^1\cdot\frac{\partial o_5^1}{\partial net_5^1}\cdot\frac{\partial net_5^1}{\partial \mu_{jm}} + o_5^1\cdot\frac{\partial o_4^1}{\partial net_4^1}\cdot\frac{\partial net_4^1}{\partial \mu_{jm}} \\ &= o_4^1\cdot\frac{\partial net_5^1}{\partial \mu_{jm}} - o_5^1\cdot\frac{1}{(net_4^1)^2}\cdot\frac{\partial net_4^1}{\partial \mu_{jm}} \\ &= o_4^1\cdot\sum_i (C_i\cdot\frac{\partial o_3^i}{\partial \mu_{jm}}) - o_5^1\cdot\frac{1}{(net_4^1)^2}\cdot\sum_i(\frac{\partial o_3^i}{\partial \mu_{jm}})\end{aligned} \tag{6.5.9}$$

where o_h^i is the output of the ith neuron in the hth layer and net_h^j is the input of the jth neuron in the hth layer. This can be calculated by

$$net_j^h = \begin{cases}\prod_i w_i v_i, & \text{for product neurons} \\ \sum_i w_i v, & \text{otherwise}\end{cases} \tag{6.5.10}$$

The sensitivity of membership function μ_{jm} to changes in a_{jm} is derived as

$$\frac{\partial \mu_{jm}}{\partial a_{jm}} = \begin{cases}\dfrac{v - b_{jm}}{(b_{jm} - a_{jm})^2}, & a_{jm} < v \le b_{jm} \\ 0, & \text{otherwise}\end{cases} \tag{6.5.11}$$

Note that when $a_{jm} = v$, the sensitivity $\dfrac{\partial \mu_{jm}}{\partial a_{jm}}$ does not make sense. In all calculations involving $\dfrac{\partial \mu_{jm}}{\partial a_{jm}}$, we define

$$\left.\frac{\partial \mu_{jm}}{\partial a_{jm}}\right|_{a_{jm}=v} = \begin{cases}\dfrac{-1}{b_{jm} - a_{jm}}, & a_{jm} = v^+ \\ 0, & a_{jm} = v^-\end{cases} \tag{6.5.12}$$

Similarly, the sensitivity of membership function μ_{jm} to changes in b_{jm} and c_{jm} is derived as

$$\frac{\partial \mu_{jm}}{\partial b_{jm}} = \begin{cases} \dfrac{a_{jm} - v}{(b_{jm} - a_{jm})^2}, & a_{jm} \le v < b_{jm} \\ \dfrac{c_{j,m} - v}{(c_{jm} - b_{jm})^2}, & b_{jm} < v \le c_{jm} \\ 0, & \text{otherwise} \end{cases} \tag{6.5.13}$$

with

$$\left.\frac{\partial \mu_{jm}}{\partial b_{jm}}\right|_{b_{jm}=v} = \begin{cases} \dfrac{1}{c_{jm} - b_{jm}}, & b_{jm} = v^+ \\ \dfrac{-1}{b_{jm} - a_{jm}}, & b_{jm} = v^- \end{cases} \tag{6.5.14}$$

and

$$\frac{\partial \mu_{jm}}{\partial c_{jm}} = \begin{cases} \dfrac{b_{jm} - v}{(c_{jm} - b_{jm})^2}, & b_{jm} \le v < c_{jm} \\ 0, & \text{otherwise} \end{cases} \tag{6.5.15}$$

with

$$\left.\frac{\partial \mu_{jm}}{\partial c_{jm}}\right|_{c_{jm}=v} = \begin{cases} 0, & c_{jm} = v^+ \\ \dfrac{1}{c_{jm} - b_{jm}}, & c_{jm} = v^- \end{cases} \tag{6.5.16}$$

6.6 Dynamic Modeling of PM from Test Data

In the following, we model the dynamics of a real pneumatic muscle hanging vertically actuating a mass of approximately 20kg, as in Figure 2.2. We collected data from the Pneumatic Muscle Test Station in the Human Effectiveness Lab at Wright-Patterson Air Force Base. All algorithms in this chapter are developed in Matlab 5.3, and simulations are executed on a PC with 933MHz PIII CPU.

A sample of the normalized recorded input-output data is plotted in Figure 6.2. In the following, we model the pneumatic muscle using the recurrent neuro-fuzzy inference system presented in Section 6.2 and initiated via VISIT.

We collected several sets of input-output data from the PM and used these for training and test data for modeling. Using these data, a RNFIS model as in Figure 6.1

was developed. In the structure learning phase, we used VISIT with $\varepsilon = 0.1$, $w = 0.85$, $\lambda = 0.27$, and $\sigma_0 = 1.17$. These parameters were chosen to yield a fuzzy system with a relatively small number of rules and fuzzy sets on each universe while giving relatively small identification error. This provided a good initial system structure to be tuned further via BP.

In the parameter learning phase, we use the BP algorithm of Section 6.5 for tuning model parameters (i.e. centers and spreads of the model membership functions identified by VISIT). After approximately 20,000 iterations, the BP training was ended at a mean square error of MSE=0.0011, where

$$MSE = \frac{1}{K}\sum_{k=1}^{K}(\hat{y}(k) - y(k))^2 \tag{6.6.1}$$

In (6.6.1), $\hat{y}(k)$ is the output of the identified RNFIS model, $y(k)$ is the experimentally measured PM length (see Figure 6.2(a)), and K is the number of input-output pairs used. This tuning process yielded the following 32 rules together with the input membership functions plotted in Figure 6.3.

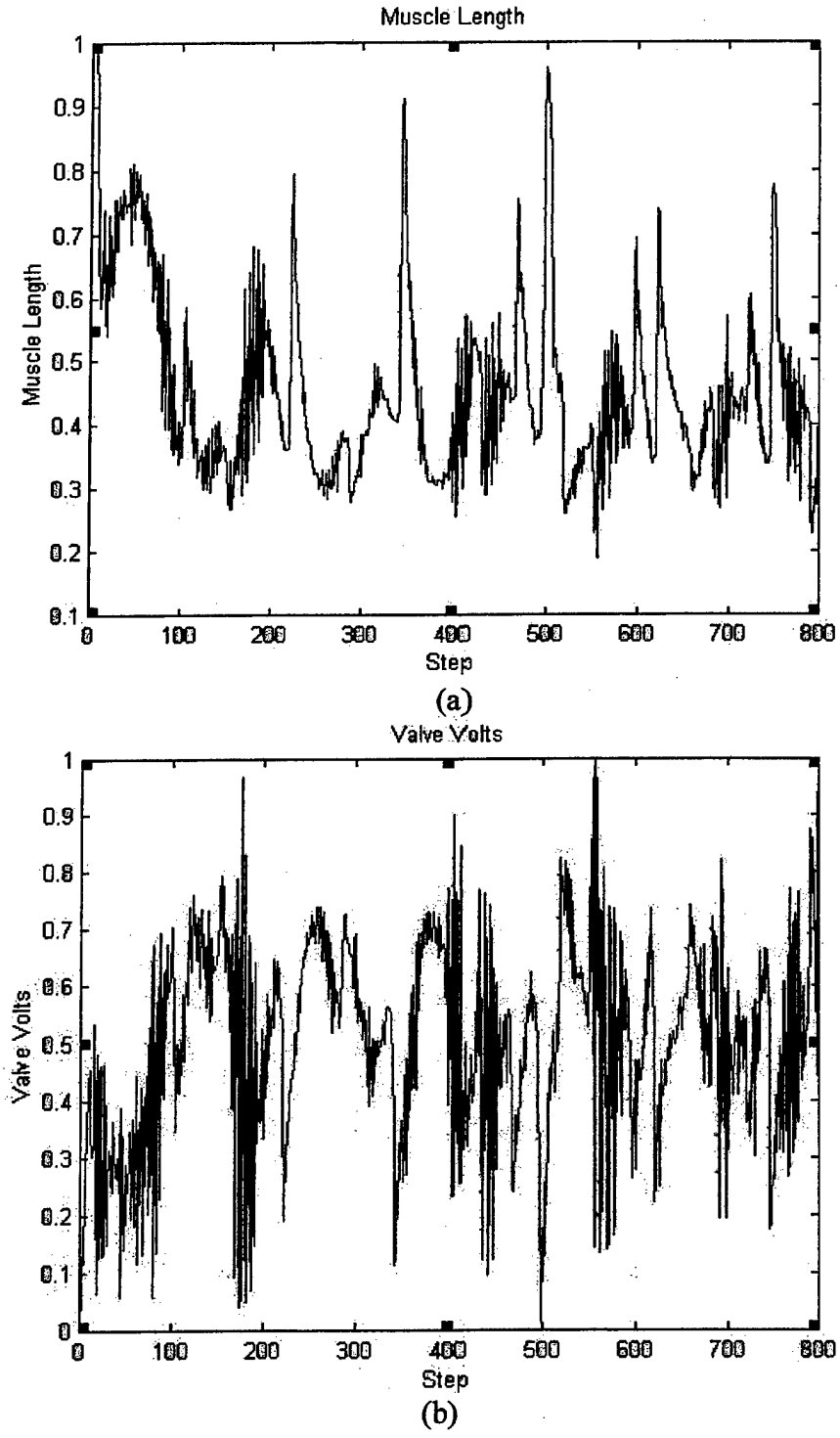


Figure 6.2 – Normalized training data used for neuro-fuzzy modeling: (a) muscle length (output), (b) valve volts (input).

Rule-base for RNFIS Model of PM

1. If $y(k\text{-}1)$ is mf_{11} and $y(k\text{-}2)$ is mf_{21} and $x(k\text{-}2)$ is mf_{31} and $x(k\text{-}1)$ is mf_{41} then $y(k) = 1.900$
2. If $y(k\text{-}1)$ is mf_{11} and $y(k\text{-}2)$ is mf_{21} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{41} then $y(k) = 2.154$
3. If $y(k\text{-}1)$ is mf_{11} and $y(k\text{-}2)$ is mf_{21} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{42} then $y(k) = 1.805$
4. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{21} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{42} then $y(k) = 0.160$
5. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{42} then $y(k) = 2.481$
6. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{41} then $y(k) = 2.478$
7. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{31} and $x(k\text{-}1)$ is mf_{42} then $y(k) = 0.668$
8. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{21} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{41} then $y(k) = 0.325$
9. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{31} and $x(k\text{-}1)$ is mf_{41} then $y(k) = 1.001$
10. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{43} then $y(k) = 0.425$
11. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{33} and $x(k\text{-}1)$ is mf_{41} then $y(k) = 2.179$
12. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{31} and $x(k\text{-}1)$ is mf_{43} then $y(k) = 1.163$
13. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{33} and $x(k\text{-}1)$ is mf_{42} then $y(k) = 0.461$
14. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{33} and $x(k\text{-}1)$ is mf_{43} then $y(k) = 0.876$
15. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{33} and $x(k\text{-}1)$ is mf_{43} then $y(k) = 0.573$
16. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{33} and $x(k\text{-}1)$ is mf_{41}then $y(k) = 0.829$
17. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{31} and $x(k\text{-}1)$ is mf_{43} then $y(k) = 3.594$
18. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{31} and $x(k\text{-}1)$ is mf_{44} then $y(k) = 0.611$
19. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{33} and $x(k\text{-}1)$ is mf_{42} then $y(k) = 1.071$
20. If $y(k\text{-}1)$ is mf_{11} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{31} and $x(k\text{-}1)$ is mf_{41} then $y(k) = 2.271$
21. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{23} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{44} then $y(k) = 1.154$

22. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{23} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{43} then $y(k) = 1.225$
23. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{23} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{42} then $y(k) = 1.169$
24. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{23} and $x(k\text{-}2)$ is mf_{31} and $x(k\text{-}1)$ is mf_{43} then $y(k) = 0.534$
25. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{42} then $y(k) = 0.685$
26. If $y(k\text{-}1)$ is mf_{11} and $y(k\text{-}2)$ is mf_{21} and $x(k\text{-}2)$ is mf_{31} and $x(k\text{-}1)$ is mf_{42} then $y(k) = 1.841$
27. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{23} and $x(k\text{-}2)$ is mf_{33} and $x(k\text{-}1)$ is mf_{43} then $y(k) = 0.587$
28. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{23} and $x(k\text{-}2)$ is mf_{33} and $x(k\text{-}1)$ is mf_{44} then $y(k) = 0.439$
29. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{23} and $x(k\text{-}2)$ is mf_{34} and $x(k\text{-}1)$ is mf_{42} then $y(k) = 0.607$
30. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{23} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{44} then $y(k) = 0.094$
31. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{23} and $x(k\text{-}2)$ is mf_{33} and $x(k\text{-}1)$ is mf_{42} then $y(k) = 0.893$
32. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{23} and $x(k\text{-}2)$ is mf_{33} and $x(k\text{-}1)$ is mf_{41} then $y(k) = 2.043$

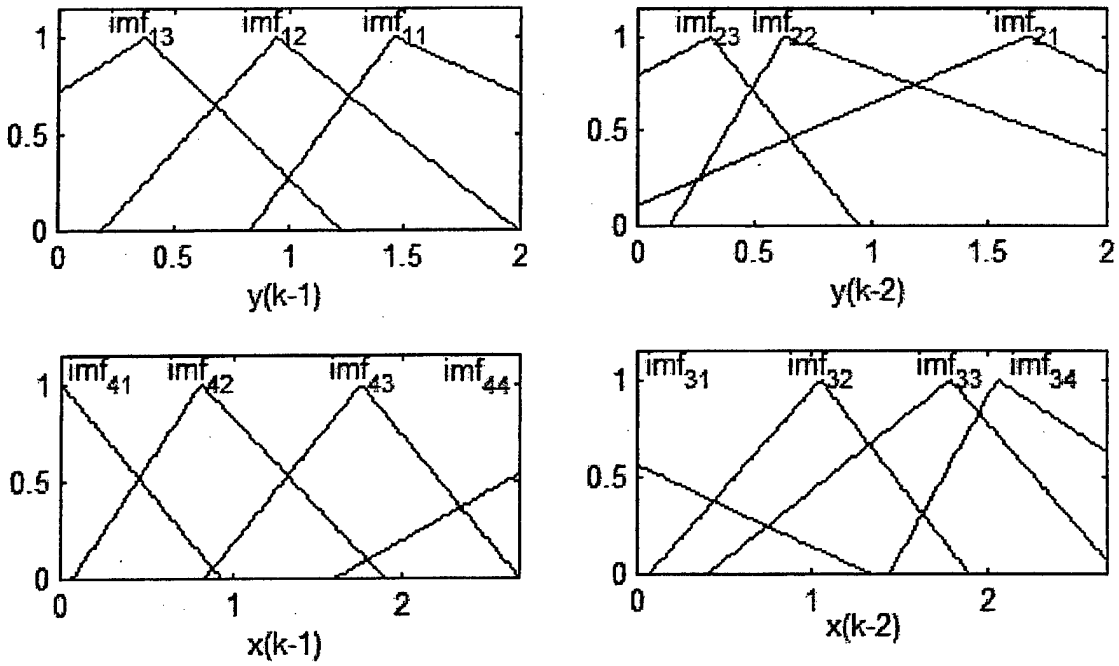
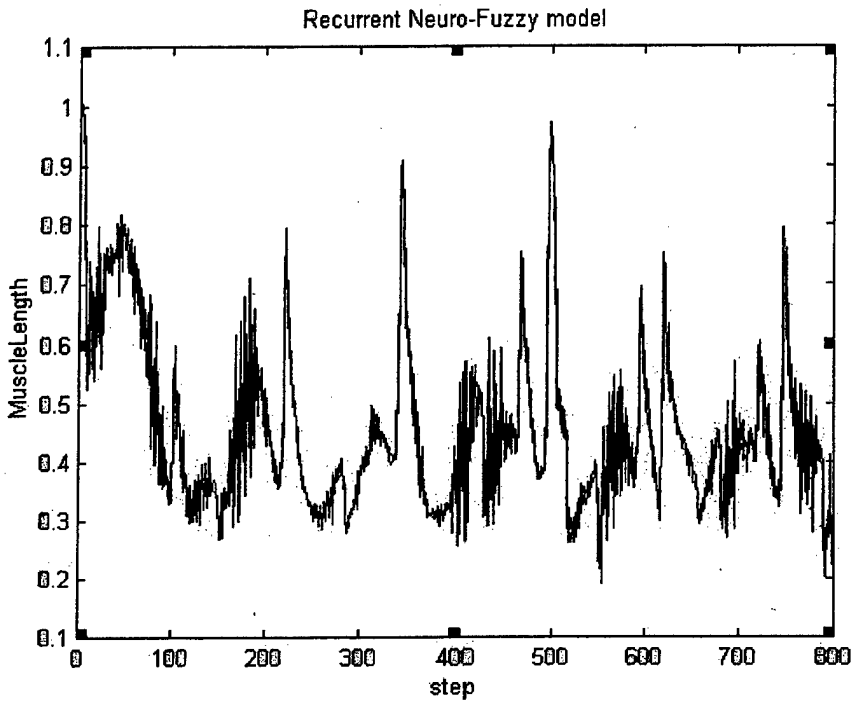


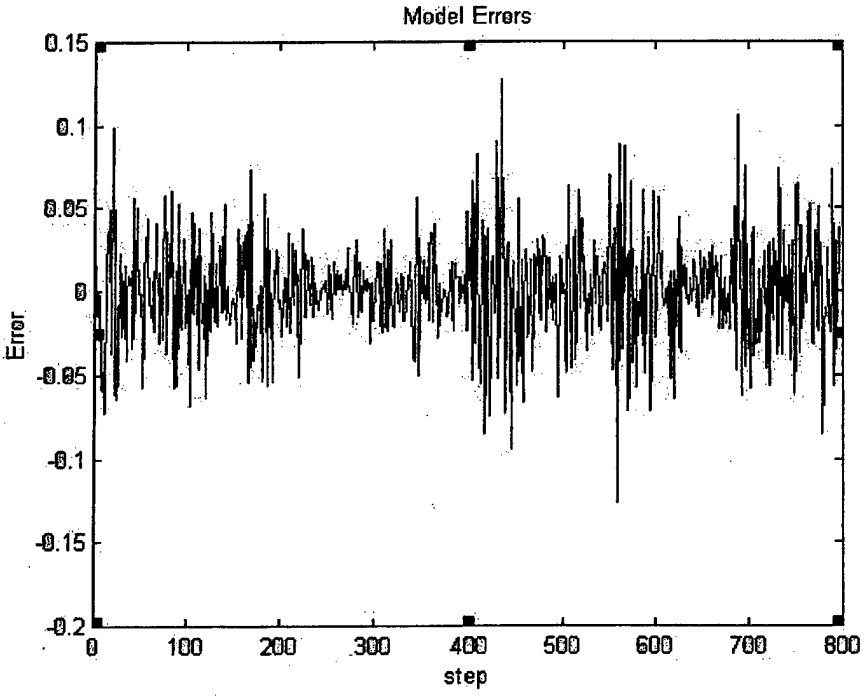
Figure 6.3 – Input membership functions for 32-rule fuzzy model of PM

The model output is compared with that of the true PM in Figure 6.4. The sampling time in Figure 6 is $1/64$ second, yielding $K \approx 3600$. Obviously from Figure 6.4, the model closely describes the PM dynamics.

To further verify the above model, we used another set of recorded data from the same pneumatic muscle to test the model. The results are plotted in Figure 6.5. From Fig.6.5a, we see that the PM length smoothly changed in this data set, and the above obtained model closely describes this behavior with small errors (MSE = 0.01622).

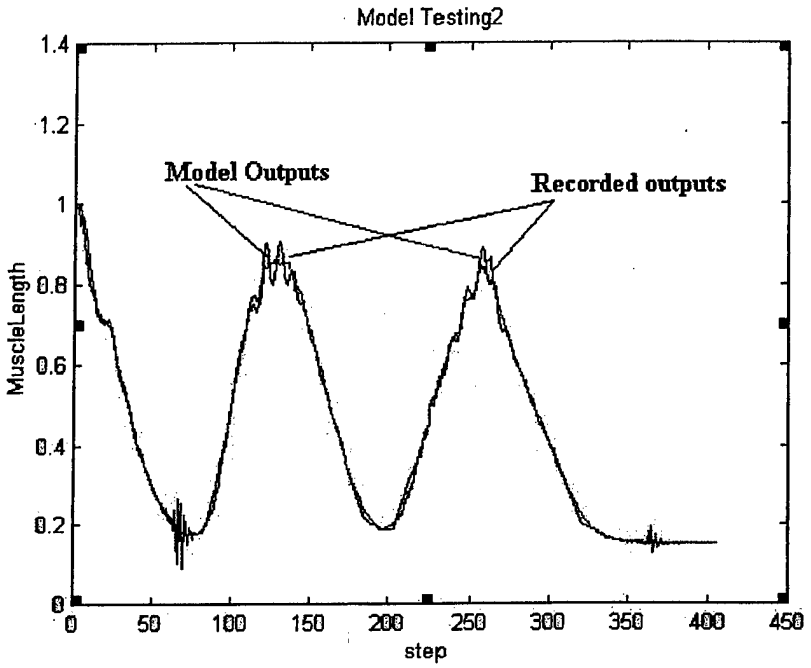


(a)

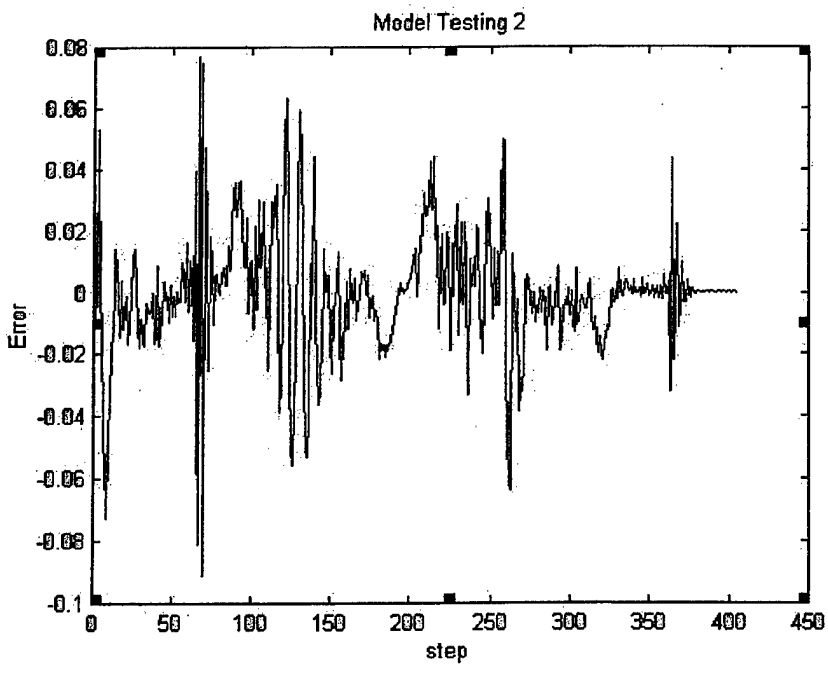


(b)

Figure 6.4 - Neuro-fuzzy dynamic modeling for the pneumatic muscle: (a) output of the fuzzy model, (b) modeling error



(a)



(b)

Figure 6.5 - Model validation: (a) model and recorded outputs, (b) model output error

7 Evolutionary Design of a Fuzzy Classifier from Data

This chapter presents results that are tangential to the main topic of the project, i.e. PM control. However, since the results grew directly out of our research on this project, we present them here.

7.1 Introduction

Pattern classification refers to the problem of partitioning a feature space into several regions and categorizing objects into classes defined on these regions. The problem of classifier design is to find an optimal mapping f from the feature space R^n into the decision space C, i.e. $f: R^n \rightarrow C$. There are many ways to construct classifiers, such as statistical models [40], neural networks [41], and fuzzy logic systems [42].

Fuzzy logic has been applied in several sophisticated classification systems [43] due to its powerful capabilities of handling uncertainty and vagueness. Fuzzy logic brings into classification systems the existence of overlapping classes and a soft decision mechanism. A pattern therefore can belong to several classes with different degrees of membership. Moreover, fuzzy classifiers consist of a set of fuzzy if-then rules, which provide insight into the classifier structure and improve interpretability [44].

Fuzzy classifiers such as those above generally lead to performance that is similar to alternative techniques such as those based on neural networks or statistics. Studies have shown that neural networks and fuzzy systems are equivalent and convertible ([45]-[47]). The advantage of fuzzy classifiers lies in their interpretable rule-base structure. The fuzzy classification rules are represented in linguistic forms that are easily interpreted and examined by users. The interpretability of results is related to the number of fuzzy rules. Fuzzy systems containing few fuzzy rules are always more interpretable than those with many fuzzy rules. Moreover, a small rule base generally improves performance by eliminating over-fitting, boosting generalization and enhancing robustness.

In this chapter, a new evolutionary approach is proposed for deriving a compact fuzzy classification system directly from data without any *a priori* knowledge of the distribution of the data. At the beginning, the fuzzy classifier is empty with no rules in the rule base and no membership functions assigned to fuzzy variables. Then, rules and membership functions are automatically created and optimized in an evolutionary process.

7.2 Fuzzy Classifier Architecture

The antecedent of a typical rule in a fuzzy classifier defines a decision region in the n-dimensional feature space; the rule consequent is a class label from the finite set $\{C_1 \;\; C_2 \;\; \ldots \;\; C_M\}$. A typical fuzzy classification rule is

$$\textit{Rule } l\text{: If } x_1 \text{ is } A_1^l \text{ and } x_2 \text{ is } A_2^l \text{ and } \cdots \text{ and } x_n \text{ is } A_n^l, \text{ then } y \text{ is } C_m$$

where $l = 1, \ldots, L$, L is the number of rules, $m \in 1, \ldots, M$, M is the number of classes, n is the number of features, and A_j^l (j = 1, 2, ..., n) is a fuzzy set associated with the feature variable x_j. Let A_j^l be characterized by the membership function $\mu_{A_j^l}(x_j)$. The membership function $\mu_{A_j^l}$ can be triangular, Gaussian, or any other shape. In this chapter, we consider asymmetric Gaussian membership functions, defined as:

$$\mu_{A_j^l}\left(x_j\right) = \begin{cases} e^{-\frac{1}{2}\left(\frac{x_j - m_j^l}{\sigma_{j,l}^L}\right)^2}, & x_j < m_j^l \\ e^{-\frac{1}{2}\left(\frac{x_j - m_j^l}{\sigma_{j,l}^R}\right)^2}, & x_j \geq m_j^l \end{cases} \tag{7.2.1}$$

where m_j^l denotes the membership function center and $\sigma_{j,l}^R$ and $\sigma_{j,l}^L$ represent the right and left spreads. If $\hat{x}_k = \lfloor x_1 \quad x_2 \quad \ldots \quad x_n \rfloor$ is a given feature vector, then using product for premise conjunction yields the following firing strength of rule R^l:

$$\mu^l(\hat{x}_k) = \mu_{A_1^l}(x_1)\mu_{A_2^l}(x_2)\cdots\mu_{A_n^l}(x_n) \tag{7.2.2}$$

The output of the fuzzy classifier is determined by the rule with the largest firing strength for a given feature vector. That is, if for pattern $\hat{x}_k$ the lth rule has the largest firing strength, $\hat{x}_k$ will be categorized into the class in the consequent of that rule. The classification error for pattern $\hat{x}_k$ is calculated as

$$e_k = \begin{cases} 0 & \text{if } \hat{x}_k \text{ is correctly classified} \\ 1 & \text{if } \hat{x}_k \text{ is incorrectly classified} \end{cases} \tag{7.2.3}$$

7.3 Extracting Fuzzy Rules via the VISIT Algorithm

The basic idea in many fuzzy learning systems is to separate the data into partitions by clustering techniques, to define the appropriate behavior on each partition, and then to tune the behaviors and/or the areas of partitions via such optimization techniques as the Recursive Least Squares (RLS), the Levenberg-Marquardt (LM), and evolutionary algorithms. The most straightforward and original methods used in constructing fuzzy systems from data are the Learning From Examples algorithm (LFE) [48] and the Modified Learning From Examples algorithm (MLFE) [49]. The Variable Input Spread Inference Training algorithm (VISIT) [37] is similar to LFE and MLFE and is most similar to MLFE. The basic idea of VISIT is that all adjacent membership functions on a given universe of discourse cross at the same level. VISIT shows some interesting features compared to its peers.

The LFE algorithm relies on a given set of input membership functions that may not adequately cover the universe of discourse or accurately describe the nature of the data clustering. If the membership functions do not adequately cover the universe of discourse, the classifier will not converge to an acceptable system and adequate performance will not be achievable. The rule base can grow unmanageably and unnecessarily large when the width of the chosen membership functions is significantly smaller than the relevant cluster size of the data.

The MLFE algorithm chooses its membership function widths as a function of learning constants and distance to the closest existing membership function on the current universe of discourse. The training can be very sensitive with respect to the learning constants. But the symmetrical nature of the membership functions precludes prudent choice of membership function widths on at least one side of each membership function. Each new membership function tends to decrease in width relative to earlier created membership functions. As a consequence, membership function widths can differ from each other significantly. With decreasing membership function widths, a rule explosion similar to that of LFE with narrowly specified membership functions could be realized.

The VISIT algorithm addresses the above problems by prudently choosing and actively adjusting the widths of asymmetrical membership functions. For asymmetrical triangular and several other membership function shapes, by choosing appropriate design parameters, these membership functions can form fuzzy partitions of unity. Because the membership function widths are created as a function of the training data, the input membership explosion experienced by LFE when cluster size exceeds membership width is significantly reduced or eliminated completely. VISIT adjusts membership function widths on each side of the membership function, based on the distance to the closest membership function on either side of the new membership function. At the same time, widths of existing nearest neighbor memberships are recalculated to account for the newly added membership function. A result of active adjustment of existing widths is that the membership function widths shrink at a much lower rate, one that is appropriate for a given data set. The membership widths differ from each other only as indicated by the nature of the data set, rather than as a function of the training algorithm.

Before starting VISIT, we first choose a vector of initial spreads

$\Sigma = \lfloor \sigma_1 \ \ \sigma_2 \ \ \ldots \ \ \sigma_n \rfloor$ for the first membership functions created, a vector $\mathbf{A} = \lfloor a_1 \ \ a_2 \ \ \ldots \ \ a_n \rfloor$ of alpha-cut values, and a vector $\mathbf{W} = \lfloor w_1 \ \ w_2 \ \ \ldots \ \ w_n \rfloor$ of parameters determining the degree of overlap between adjacent membership functions on each universe. We now give the VISIT algorithm for pattern classification.

VISIT Algorithm [37]

1. Set $i = 1$. Get the first training pair $tp^1 = (x_1, x_2, \ldots, x_n, y)^1$ where y^1 is the class of data pair 1 represented by a real number. On each input universe of discourse form a fuzzy set A_j^1 characterized by a symmetrical Gaussian membership function $\mu_{A_j^1}\left(x_j^1\right)$ with center at $m_{j,0} = x_j^1$ and spread σ_j. On the output universe of discourse form a fuzzy set B^1 characterized by a singleton membership function with support y^1.
2. Add the rule

   $$\text{If } x_1 \text{ is } A_1^1 \text{ and } x_2 \text{ is } A_2^1 \text{ and } \cdots \text{ and } x_n \text{ is } A_n^1 \text{ then } y \text{ is } B^1.$$

3. If there are no more training pairs, the training cycle is completed. Otherwise, increment i and get the next training pair $tp^i = (x_1, x_2, \ldots, x_n, y)^i$.
4. Evaluate tp^i with the current classifier. If classification is correct, go to step 3. Otherwise, go to step 5.
5. On the jth input universe $j = 1, \ldots, n$, evaluate all membership functions at x_j^i. Call the fuzzy set whose membership function is maximum $A_j^{\max}$ and its a_j-cut set $a_j - cut\left(A_j^{\max}\right)$. If $x_j^i \notin a_j - cut\left(A_j^{\max}\right)$, form a new fuzzy set A_j^{new} on the jth universe characterized by a membership function with center $m_j^{new} = x_j^i$ and left and right spreads σ_j^L and σ_j^R as in step 6 below. On the output universe, if the pattern tp^i belongs to a new class, add a new fuzzy set B^i characterized by a singleton membership function with support y^i. Otherwise, the class B^i already exists from previous data.
6. On each input universe on which a new membership function was added in step 5, the spread of each side of the new membership function is initialized as σ_j. However, if there exist neighboring memberships on the left and/or right of the new

one, the spread of the corresponding side of the new membership function and its nearest neighbors on the left and/or right are re-calculated as follows. The right spread of the new membership function and the left spread of the nearest right neighbor are re-calculated as

$$\sigma_j^R = \frac{1}{w_j}\left| m_j^{new} - m_j^{nRn} \right| \tag{7.3.1}$$

The left spread of the new membership function and the right spread of the nearest left neighbor are re-calculated as

$$\sigma_j^L = \frac{1}{w_j}\left| m_j^{new} - m_j^{nLn} \right| \tag{7.3.2}$$

where m^{nRn} denotes the nearest existing center to the right of m_j^{new} and m_j^{nLn} denotes the nearest existing center to the left of m_j^{new}.

7. If a new membership function was formed in step 6, consider the following candidate rule where the fuzzy sets in the premise are the ones maximized by the corresponding inputs:

   If x_1 is $A_1^{\max}$ and x_2 is $A_2^{\max}$ and $\cdots$ and x_n is $A_n^{\max}$ then y is B^i

   If there is no rule in the rule base that is inconsistent (i.e. same premise, different consequent) with this rule, add the above rule in the rule base. If there exists an inconsistent rule, discard tp^i and go to step 3.

8. If there is another training pattern, go to step 3. Otherwise, the training process ends.

Comments

Note that the above VISIT algorithm is somewhat different from the original one in [37]. In [37], the authors assumed that all universes of discourse were identical, hence scalar values of w, a, and σ were chosen to begin the algorithm. However, in many real systems, different features often work in different universes of discourse, so we apply three vector parameters $\mathbf{W}$, $\mathbf{A}$, and $\mathbf{\Sigma}$ for the VISIT algorithm. Choosing different values of $\mathbf{W}$, $\mathbf{A}$, and $\mathbf{\Sigma}$ results in different fuzzy systems from the same data set. The main problem in finding a good VISIT classifier amounts to finding good values of the parameters $\mathbf{W}$, $\mathbf{A}$, and $\mathbf{\Sigma}$. In the original VISIT, these parameters were chosen by trial and error.

The vector parameter $\mathbf{W}$ determines the degree of overlap between two neighboring

membership functions. Large w_j results in small overlaps in the jth input universe of discourse. It can be shown that two neighboring membership functions always overlap at $e^{-w_j^2/8}$ when asymmetric Gaussian membership functions are used. Therefore, the resulting fuzzy system will meet the so-called ε-completeness condition, i.e., there always exists at least one rule with firing strength of at least ε for any pattern in the operative region. It can be proved that the firing strength of any fuzzy rule generated by VISIT will be no less than $\prod_{i=1}^{n} e^{-w_i^2/8}$.

Parameters A determine when a new rule is added to the classifier, and Σ determines the initial spreads of the first membership functions. Larger a_j and smaller σ_j tend to increase the number of membership functions for the jth input, while smaller a_j and larger σ_j tend to decrease the number of membership functions. In some situations, if σ_j is large enough and a_j is small enough, a membership function will be generated only for the jth input, which implies feature reduction because the same membership function occurs in all rules.

When $a_j \le e^{-w_j^2/8}$, a new membership function is formed only if the new feature amount falls outside the a_j-cut. We can estimate the distance between two neighboring membership functions as:

$$\text{distance} \ge \sigma_j \sqrt{-2 \ln a_j} \tag{7.3.3}$$

If $m_{j,0}$ denotes the center of the first membership function for the jth input, we can derive other membership function centers recursively from the above equation.

$$m_{j,1} = \min_i \left(x_j^i > m_{j,0} + \sigma_j \sqrt{-2 \ln a_j} \right) \tag{7.3.4}$$

$$\vdots$$

$$m_{j,k} = \min_i \left(x_j^i > m_{j,k-1} + \sigma_j \sqrt{-2 \ln a_j} \right) \tag{7.3.5}$$

for membership function centers to the right of $m_{j,0}$ and

$$m_{j,-1} = \max_i \left(x_j^i < m_{j,0} - \sigma_j \sqrt{-2 \ln a_j} \right) \tag{7.3.6}$$

$$\vdots$$

$$m_{j,-k} = \max_i \left(x_j^i < m_{j,-k+1} - \sigma_j \sqrt{-2\ln a_j} \right) \tag{7.3.7}$$

for membership function centers to the left of $m_{j,0}$. Obviously, the final fuzzy system depends on $m_{j,0}$, which is traditionally equal to the value of the first feature amount fed into the learning machine. In other words, the original VISIT algorithm depends on the sequence of patterns fed in. In the sequel, we ease this limitation.

When $e^{-w_j^2/8} < a_j < 1$, VISIT will add a new membership function between two neighboring membership functions when a new pattern has an input x_j such that $x_j \in \left[m_{j,l} - \sigma_j \sqrt{-2\ln a_j},\ m_{j,l+1} + \sigma_j \sqrt{-2\ln a_j} \right]$ where $m_{j,l}$ and $m_{j,l+1}$ are the centers of two neighboring membership functions and $m_{j,l+1} > m_{j,l}$. Obviously, if a_j is large, this can relax the dependence on the sequence in which the patterns are fed to the learning mechanism.

7.4 Extraction of Fuzzy Rules via Evolutionary Algorithms

Consider the training data set $X = \left\{ \hat{x}_k = \left(x_1, x_2, \cdots x_n, y \right)^k \right\}$. The single data point $\hat{x}_k$ is an element of the space $R^n \times R$ for each k. The VISIT algorithm can be considered as defining a function mapping $F(\cdot)$ from the training data set X to the space of fuzzy systems *FIS*, i.e.

$$FIS = \underset{\mathbf{W,A,\Sigma,M}}{F}(X) \tag{7.4.1}$$

Note that an extra parameter vector $\mathbf{M} = \left[m_{1,0} \quad m_{2,0} \quad \ldots \quad m_{n,0} \right]^T$ consisting of the centers of the first membership functions has been added. Parameters $\mathbf{M}$ can relax the dependence of VISIT on the sequence in which the pattern is fed. The process for the creation of a "good" fuzzy classifier via VISIT is equivalent to looking for optimal parameters $\mathbf{W}$, $\mathbf{A}$, $\mathbf{\Sigma}$, and $\mathbf{M}$, where "goodness" is evaluated by some performance index J, such as J_1 and J_2 below. Therefore, the problem of generating a fuzzy system from data via VISIT can be mathematically expressed as a multi-objective optimization problem:

$$\min_{\mathbf{W,A,\Sigma,M}} (J) \tag{7.4.2}$$

subject to

$$w^{\min} < w_j < w^{\max} \tag{7.4.3}$$

$$0 < a_j < 1 \tag{7.4.4}$$

$$\sigma_j^{\min} < \sigma_j < \sigma_j^{\max} \tag{7.4.5}$$

$$x_j^{\min} < m_{j,0} < x_j^{\max} \tag{7.4.6}$$

In (7.4.3-7.4.6), $\left[w^{\min}, w^{\max}\right]$ defines the range of acceptable overlaps, and $\left[\sigma_j^{\min}, \sigma_j^{\max}\right]$ defines the range of acceptable initial membership function spreads for the jth feature, and $[x_j^{\min}, x_j^{\max}]$ is the jth universe of discourse.

J is the performance index for the evaluation of the fuzzy system. We often evaluate a fuzzy system in two ways:

1. Accuracy: the fuzzy system should accurately describe the behaviors embedded in the data.
2. Interpretability: the size of the fuzzy rule-base should be small enough to be easily understood. Interpretability is one of the features that distinguish a fuzzy classifier from its peers. Moreover, for a fuzzy system, good interpretability often implies robust performance on unseen data. The interpretability of a fuzzy system is inversely related to the number of membership functions, the number of rules, and premise length.

Traditional optimization methods are based on the assumption that the performance index, or objective function, is differentiable. Unfortunately in the above optimization problem, we cannot in general assume the objective function J is differentiable with respect to parameters $\mathbf{W}, \mathbf{A}, \mathbf{\Sigma}$, and $\mathbf{M}$. One way to overcome this problem is to use a genetic algorithm (GA) to optimize J. GAs provide powerful tools to efficiently search in poorly understood, irregular spaces, and are inspired by the mechanism of natural evolution. For some difficult optimization problems, GAs can perform better than traditional optimization techniques, such as hill-climbing methods, by means of exploitation of the best solutions and exploration of the search space simultaneously. The genetic algorithm used in this chapter is described as follows.

Step 1. Let $t = 0$. Initialize the first population P_t with N individuals (N even).

Step 2. Evaluate the fitness $f_i^t, i = 1, \ldots, N$ of the N individuals in P_t (fitness defined below).

Step 3. Select the $N/2$ fittest individuals in population P_t for mutation (defined below). Mutate the $N/2$ fittest individuals to generate $N/2$ new individuals P_t'.

Step 4. Randomly select $N/4$ pairs of individuals and apply a crossover operator on each pair to generate $N/2$ more new individuals P_t''.

Step 5. Select the N fittest individuals from the current population (P_t, P_t', and P_t''), to construct the next population P_{t+1}.

Step 6. Let $t = t+1$ and go to step 3. Repeat until $t > t_{end}$, where t_{end} is a predetermined end time.

In Step 5, the elitist selection method always preserves the N best individuals in the tth evolution.

Solution Representation

Chromosomes are used to describe individuals in the population. The VISIT algorithm provides an encoding method to create chromosomes for different individuals. The vector $P_t(i) = \left[w_1^t(i), \cdots, w_n^t(i), a_1^t(i), \cdots, a_n^t(i), \sigma_1^t(i), \cdots, \sigma_n^t(i), m_{1,0}^t(i), \cdots, m_{n,0}^t(i) \right] \in R^{4n \times 1}$ (chromosome) represents an individual in our genetic algorithm. In $P_t(i)$, t represents the generation number and i represents an individual in the population. Each element in $P_t(i)$ always corresponds to the same variable regardless of t. This is necessary for efficient utilization of the crossover operators. Moreover, the chromosome allows us to initialize individuals according to constraints, i.e.

$$w_j^0(i) = w^{\min} + r_{i,j}^1 (w^{\max} - w^{\min}) \tag{7.4.7}$$

$$a_j^0(i) = r_{i,j}^2 \tag{7.4.8}$$

$$\sigma_j^0(i) = s_j^{\min} + r_{i,j}^3 (\sigma_j^{\max} - \sigma_j^{\min}) \tag{7.4.9}$$

$$m_{j,0}^0(i) = x_j^{\min} + r_{i,j}^4 (x_j^{\max} - x_j^{\min}) \tag{7.4.10}$$

where $r_{i,j}^1$, $r_{i,j}^2$, $r_{i,j}^3$, and $r_{i,j}^4$ are random numbers uniformly distributed in [0, 1]. This initialization limits the search space, and makes the optimization problem more efficient.

Genetic Operators

There are two kinds of genetic operators used in the proposed genetic algorithm - whole arithmetic crossover and Gaussian mutation. The kth pair of selected individuals $P_t(i)$ and $P_t(j)$ are crossed via whole arithmetic crossover operators. The two resulting

offspring from this pair are

$$P_t'(1) = rP_t(i) + (1-r)P_t(j) \tag{7.4.11}$$

$$P_t'(2) = rP_t(j) + (1-r)P_t(i) \tag{7.4.12}$$

where r is a random number uniformly distributed in [0, 1]. As stated in step 4 above, this process is followed for $N/4$ randomly chosen pairs of individuals.

We define the following mutation operator to create a new individual from the ith individual in the tth generation $P_t(i)$ (step 3 above):

$$P_t'(i,j) = P_t(i,j) + \gamma_j \rho_j(i) \exp\left(-\frac{f_i^t}{f_{\max}^t}\right) \tag{7.4.13}$$

where j signifies a particular element of the chromosome, γ_j is a learning coefficient which determines the amount of maximum change of the jth element in the chromosome from generation to generation, $\rho_j(i)$ is a uniformly distributed random number $\in [-1, 1]$, f_i^t is the fitness of individual i in the tth generation, and $f_{\max}^t$ is the largest fitness in the tth generation.

Note that some infeasible offspring can be created via the mutation operator (7.4.13), so we have to examine each individual offspring to insure it meets the constraints of (7.4.3–7.4.6). If an offspring is infeasible we discard it, decrease γ_j by half, and apply the mutation operator again until a feasible offspring is obtained. Another simple method to generate a feasible offspring and replace the infeasible one is to use initialization equations (7.4.7-7.4.10).

Evaluation of Individual Fitness via a Fuzzy Expert System

In the above evolutionary algorithm, the fitness function used to evaluate the goodness of each individual is critical to the selection operation of the evolutionary process. For fuzzy systems, we care about resolution and interpretability, which are conflicting requirements. The resolution of the fuzzy classifier is related to the number of misclassifications. Many misclassifications indicate low resolution. The interpretability of the fuzzy system is related to the number of rules in the rule-base. Many rules impair interpretability. In general, we can separately use the Mean Classification Error (MCE) for measuring the resolution of the fuzzy classifier, and the number of rules as an inverse indicator of interpretability. The MCE criterion is as follows:

$$J_1 = \frac{1}{K} \cdot \sum_{k=1}^{K} e_k \tag{7.4.14}$$

where e_k is defined in (7.2.3) and K is the number of data points in the training set.

Interpretability of the fuzzy system can be inversely measured by:

$$J_2 = \beta(L - M) \tag{7.4.15}$$

where L is the number of rules and M is the total number of classes in the training set, which suggests the minimum number of rules in the fuzzy system. The parameter β is a factor to normalize the possible size of rule base into the range [0, 1]. In this chapter, we use $\beta = 0.01$.

The objective functions J_1 and J_2 always lie between 0 and 1. A good fuzzy classifier should have small MCE and few rules. However, a decrease in MCE does not imply a decrease in the number of rules, and vice versa. Usually, resolution is improved by increasing the number of rules. Thus it is necessary to seek a trade-off between resolution and interpretability in the design of fuzzy classifiers. In this chapter, the task of designing a fuzzy system is formulated as the following two-objective optimization problem:

$$\min_{\mathbf{W},\mathbf{A},\Sigma,\mathbf{M}} (J_1) \text{ and } \min_{\mathbf{W},\mathbf{A},\Sigma,\mathbf{M}} (J_2) \tag{7.4.16}$$

An important technique for solving a multi-objective optimization problem is to combine multiple objectives into one, reducing the complex problem into a standard single-objective one. A well-known tradeoff of multiple objectives is to define a single objective function that combines both J_1 and J_2 as

$$J = \eta \cdot J_1 + (1 - \eta) \cdot J_2 \tag{7.4.17}$$

where $0 < \eta < 1$. With this objective function, a fitness function *fit* could be defined as $1/(1+J)$. Possible difficulties with using this method are:

1. The result can vary significantly as η changes. It is difficult to determine an appropriate value of η when we do not know enough information about the problem. In general, we have to design η via trial and error, which is often time-consuming.
2. Due to the complexity of the classification problem, the factor η does not reliably reflect the relative importance of objectives. It is only a factor which, when varied, changes the result. The result is also dependent on the units in which the objective

functions are expressed. In practice, objectives often have different importance. If we want η to reflect the relative importance of the objectives, all objectives should be normalized in units of approximately the same numeric values.

For the above reasons, we do not use (7.4.17), but instead apply a fuzzy expert system for comprehensively evaluating the above two objectives for individuals in generations according to expert knowledge. The fuzzy expert system consists of two inputs (J_1 and J_2), one output (*fit*), and eleven T-S fuzzy rules. The output recommended by rule k in the fuzzy expert system is fit_k. Table 1 lists the rules in the fuzzy expert system, and Figure 1 shows the inputs' membership functions. The rule base and membership functions are the result of our general knowledge on a "good" fuzzy system.

Table 7.1 - Fitness rules

Rule # (k)	Membership Function J_1	Membership Function J_2	fit_k
1	*Zero*	Zero	80
2	Zero	*Small*	70
3	Zero	*Medium*	50
4	*Small*	Zero	60
5	*Small*	*Small*	40
6	*Small*	*Medium*	20
7	*Medium*	Zero	30
8	*Medium*	Small	10
9	*Medium*	*Medium*	1
10	—	*Large*	-20
11	*Large*	—	-20

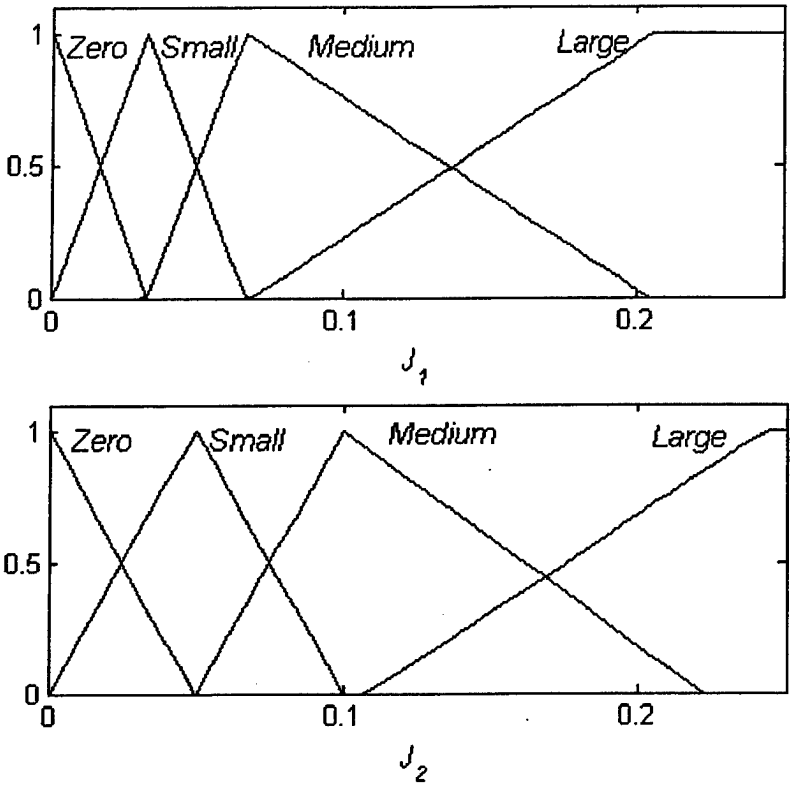


Figure 7.1 – Input membership functions for fuzzy expert system

The output of the fuzzy expert system, global fitness, is calculated as a weighted average of the local fitness functions:

$$fit = \frac{\sum_{k=1}^{11} \mu_k \cdot fit_k}{\sum_{k=1}^{11} \mu_k} \tag{7.4.18}$$

where fit_k is a local fitness taken from the last column of Table 1 and μ_k is the firing strength of the kth rule (7.2.2). The fitness surface is plotted in Figure 2.

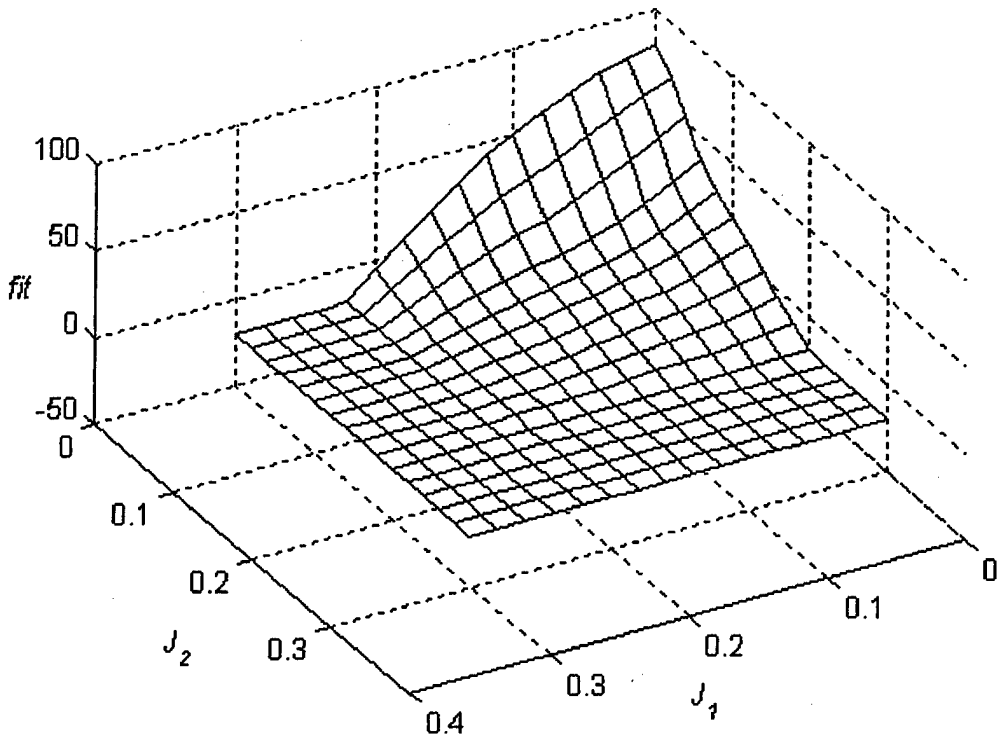


Figure 7.2 - Surface of fuzzy fitness function

7.5 Applications and Performance Evaluation

This section will evaluate the performance of the proposed algorithm on four benchmark data sets in classification, the iris data, the wine data, the Wisconsin breast cancer data, and the Pima Indians diabetes data. These data sets can be obtained from the machine-learning database at University of California, Irvine via an anonymous ftp server (ftp://ftp.ics.uci.edu/pub/machine-learning-databases/). It is to be noted that we corrected the errors in the UCI iris data [50]. All data sets are normalized to the range [0, 1].

Example1 - Wine data

The wine data contains the chemical analysis of 178 wines grown in the same region in Italy but derived from three different cultivars. Thirteen continuous attributes are measured on each wine: alcohol content (AL), Malic Acid content (MAC), ash content, alcalinity of ash (AA), magnesium content (MA), total phenols (TP), flavanoids (FL), nonflavanoids phenols (NFP), proanthocyaninsm (PR), color intensity (CI), hue, OD280/OD315 (O) of diluted wines, and praline (P). The numbers of patterns in three classes are 59, 71 and 48 respectively.

The wine data has been widely used to test the performance of classifier systems. Table 7.5 shows the results of some well-known classifier systems. Setnes et al. [51] applied a real-coded GA and a c-means clustering algorithm on all 178 patterns to design a TSK-type fuzzy classifier system. In their method, the c-means clustering technique

was used to generate an initial system with three rules. After repetitive simplification and optimization, only 9 features were selected and generated a fuzzy classifier system with only three rules and a high recognition rate of 98.3% (3 misclassifications).

Ishibuchi et al. [52] applied all 178 samples to the design of a fuzzy classifier via three-objective genetic-based machine learning techniques. At the beginning, there are 1834 candidate rules in the rule base. The obtained best result is a fuzzy system containing 6 rules and 9 term sets with a classification rate of 100%. In their recent work [82], they update their algorithm by integration of an evolutionary multi-objective algorithm (NSGA-II [83]), and a smaller fuzzy system was designed with 100 % correct classification. This fuzzy system only contained 4 fuzzy rules and 5 term sets.

Wang et al. [53] also used all 178 patterns to design their fuzzy classifier. They applied the Mapping-constraint Agglomerative Clustering (MAC) method to detect the structure of the fuzzy system, and improved the fuzzy classifier via a parameter learning method. Finally, several 3-rule classifier systems on 13 features were derived with the best recognition rate of 99.4% (1 misclassification).

Roubos et al. [54] proposed an iterative approach to design a fuzzy classifier system using all 178 patterns. To begin their algorithm, they built a 3-rule fuzzy system using a total of 13 features, then iteratively applied feature selection, rule base simplification and GA based parameter optimization to improve the performance and reduce the dimensionality of the classifiers. Finally, they developed three 5-feature fuzzy classifiers. The classification results of the three classifiers are 98.9%, 99.4%, and 98.3% correct.

Table 7.2 – VISIT classification rules for wine data

Rule	Al	MAC	Ash	AA	MA	TP	FP	NFP	PR	CI	Hue	O	P	Class
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	1	2	1	1	1	2	1	1	1	1	2	2	2
3	2	1	1	1	1	1	1	1	1	1	2	1	2	2
4	2	1	2	1	1	1	1	1	1	1	3	2	2	2
5	2	1	1	1	1	1	2	1	1	1	3	2	2	3

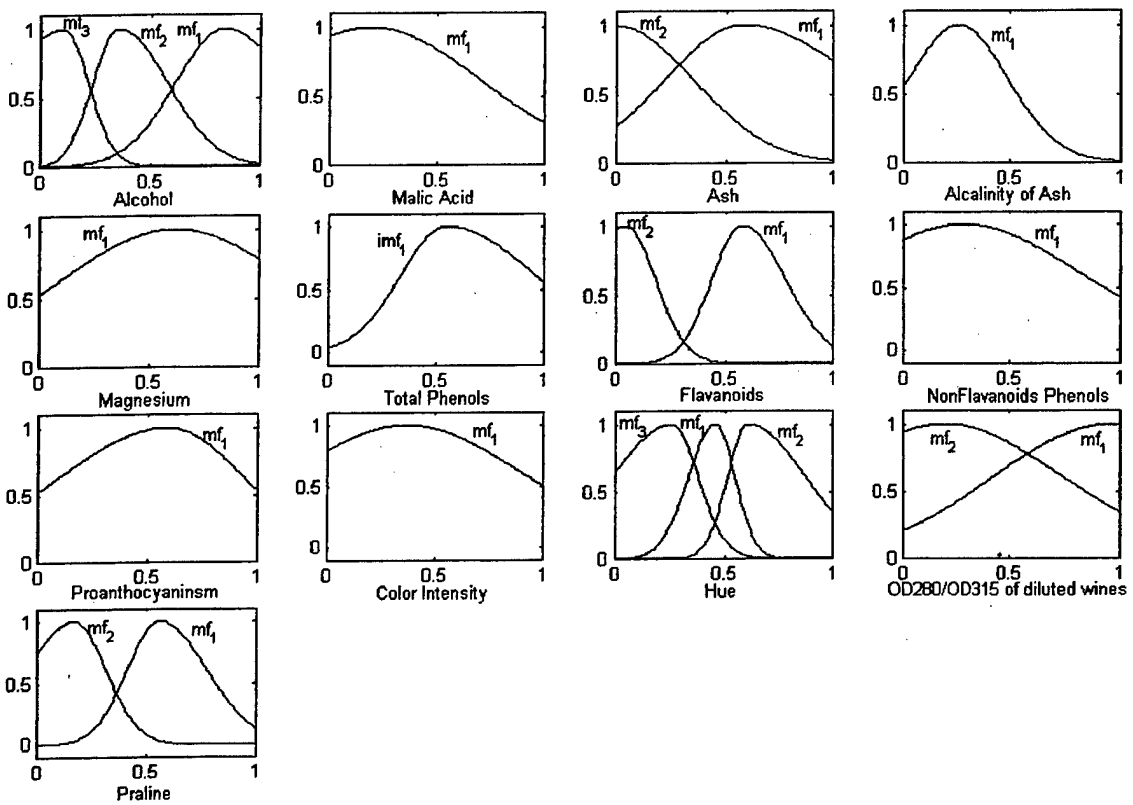


Figure 7.3 – VISIT membership functions for 13 features

Similarly, we also used all 178 data points to design our fuzzy classifier system. But differently from the above methods, we do not apply any clustering techniques to detect initial model structures. At the beginning, the fuzzy system is empty without any rules, and the proposed method is able to automatically extract and optimize a fuzzy rule base based on available patterns.

The VISIT algorithm is applied to create a fuzzy system from the wine data, and its parameters $\mathbf{W}=\lfloor w_1, w_2, \ldots, w_{13} \rfloor$, $\mathbf{A}=\lfloor a_1, a_2, \ldots, a_{13} \rfloor$, $\Sigma=\lfloor \sigma_1, \sigma_2, \ldots, \sigma_{13} \rfloor$, and $\mathbf{M}=\left[m_{1,0}, m_{2,0}, \ldots, m_{13,0} \right]$, are optimized by the evolutionary algorithm described in Section 7.4. The value of parameters **W**, **A**, **Σ**, and **M** were randomly initialized according to (7.4.7-7.4.10). The population size in each generation is 200, and the evolutionary process is stopped after 100 iterations. The classification rate for the resulting fuzzy system is 98.9% correct (2 misclassifications). Figure 7.3 shows the membership functions of the 13 features, and Table 7.2 lists all fuzzy rules. Table 7.3 shows the classification performance using VISIT. The fuzzy system can accurately recognize Class 3 with no errors.

Table 7.3 - VISIT classification on 3 classes

	Class1	Class2	Class3	Total
Total Patterns	59	71	48	178
Misclassifications	1	1	0	2

Note that the proposed algorithm creates only one membership function for 7 of the features: malic acid, alcalinity of ash, magnesium, total phenols, nonflavanoids phenols, proanthocyaninsm, and color intensity. Obviously, these 7 features are redundant to the obtained classifier system because they contribute the same to all rules. After deleting these redundant features, the membership functions and rule set of the final system is shown in Figure 7.4 and Table 7.4.

The proposed algorithm developed a compact fuzzy system without any assumption on the structure. Moreover, the classification rules are highly interpretable, utilizing only six features, five rules and no more than three membership functions associated with each feature. Table 7.5 shows a comparison of our obtained classifier with the above well-known classifier systems. Table 7.5 compares the various methods on the number of used features, the number of term sets, the number of rules, and the recognition rate.

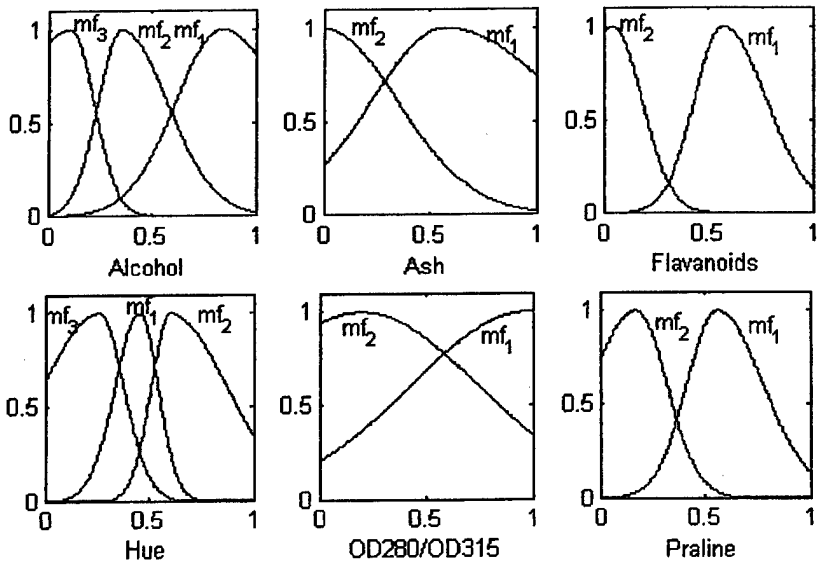


Figure 7.4 – VISIT membership functions for 6 features

Table 7.4 – VISIT classification rules for wine data (6 features)

Rule	Al	Ash	FP	Hue	O	P	Class
1	1	1	1	1	1	1	1
2	2	2	2	1	2	2	2
3	2	1	1	2	1	2	2
4	2	2	1	3	2	2	2
5	2	1	2	3	2	2	3

Table 7.5 – Comparison of results for wine data

	Features	Term sets	Rules	Recognition rate on total data set (%)
Setnes et al. [51]	9	21	3	98.3
Wang et al. [53]	13	34	3	99.4
Ishibuchi et al. [52]	-	9	6	100
Ishibuchi et al. [82]	-	5	4	100
Roubos et al. [54]	5	15,11,10	3	98.9, 98.3, 99.4
This chapter	6	13	5	98.9

For more reliable performance evaluation of the proposed algorithm, we ran it independently on the wine data ten times, with the random initializations of parameters $\mathbf{W}$, $\mathbf{A}$, $\boldsymbol{\Sigma}$, and $\mathbf{M}$ on each run. Table 7.6 reports the results of ten trials. The resultant system utilized an average of 6.7 feature variables, 16.7 membership functions and 6 fuzzy rules to get to an average recognition rate of 98.2% (3.3 misclassifications). The best recognition rate is 98.9% (2 misclassifications). The "best" system utilized 6 feature variables, 14 membership functions and 5 fuzzy rules. The worst recognition rate is 97.8% (4 misclassifications). The related system utilized 7 feature variables, 21 membership functions and 7 rules.

Table 7.6 – Results of ten runs on wine data

	1	2	3	4	5	6	7	8	9	10	Avg
Misclassifications	2	4	4	3	3	4	3	3	4	3	3.3
Recognition rate (%)	98.9	97.8	97.8	98.3	98.3	97.8	98.3	98.3	97.8	98.3	98.2
Number of features*	6	7	8	7	6	6	6	6	5	6	6.3
Number of rules	5	7	7	8	5	5	5	7	6	5	6
Number of term sets	14	21	23	18	14	14	14	15	13	13	15.9

* The number of features used in the obtained fuzzy system after deleting some one-membership-function features.

Example 2 - Iris Data

The Fisher iris data [50] consist of 4 measurements: sepal length (SL), sepal width (SW), petal length (PL) and petal width (PW), in 150 data sets. Three iris species are involved in the measurement: iris setosa, iris versicolor, and iris virgincia. Each species contains 50 samples.

The iris data is a famous benchmark to test the performance of classifier systems. Table 7.9 shows the results of some well-known classifier systems. For example, Shi et al. [55] applied an integer-code genetic algorithm to learn a Mamdani-type fuzzy system for classifying the iris data by training on all 150 patterns. After several trials with different learning options, a four-rule fuzzy system was obtained with 98% correct recognition (3 misclassifications). Abe et al. [56] discussed a fuzzy classifier with ellipsoidal regions. They applied clustering techniques to extract fuzzy rules, with one rule around one cluster center, and then they tuned the slopes of their membership functions to obtain a high recognition rate. Finally, they obtained a fuzzy classifier with a recognition rate of 98.7% (2 misclassifications). Russo [57] applied a hybrid GA neuro-fuzzy approach to learn a fuzzy model for the iris data. He derived a five-rule fuzzy system with 18 fuzzy sets and 0 misclassifications. Ishibuchi et al. [84] applied all 150 samples in the training process, and derived a fuzzy classifier with 7 term sets and 5 rules. The resolution was 98.0% correct and 3 misclassifications. Aboyi et al [85] proposed a new data-driven method to design compact fuzzy classifiers via combining a genetic algorithm, a decision-tree initialization, and a similarity-driven rule reduction technique. The final system had 3 fuzzy rules and 4 term sets. The accuracy is 96.11% correct (6 misclassifications).

Table 7.7 – Results of ten runs of VISIT on iris data

	1	2	3	4	5	6	7	8	9	10	Average
Misclassifications	1	3	1	2	1	2	2	2	1	2	1.7
Recognition rate (%)	99.3	98	99.3	98.7	99.3	98.7	98.7	98.7	99.3	98.7	98.87
Features	3	2	4	3	4	4	3	4	3	3	3.3
Rules	5	4	5	5	5	6	5	5	6	6	5.2
Term sets	7	6	9	9	8	9	8	9	11	8	8.4

For the iris example, we also used 150 patterns to design a fuzzy classifier system via the proposed evolutionary algorithm. The parameters $\mathbf{W}$, $\mathbf{A}$, $\mathbf{\Sigma}$, and $\mathbf{M}$ were initialized randomly. The population had 200 individuals, and the evolution processes was stopped after 100 iterations. Table 7.7 shows the results of ten runs (i.e. ten different initializations of parameters). The final fuzzy systems utilized an average of 3.3 feature variables, 8.4 membership functions and 5.2 rules to get an average recognition rate of 98.87% (1.7 misclassifications). The worst classification result is 98% correct (3 misclassifications). The "worst" system utilized 2 feature variables, 6 membership functions and 4 rules. The best classification result is 99.3% correct (1 misclassification). It misclassified a pattern in class Versicolor into class Virginica. The corresponding "best" system utilized 3 feature variables, 7 membership functions and 5 fuzzy rules. Figure 5 shows the membership functions in the best fuzzy classifier. Table 7.8 shows the fuzzy rules used in the best fuzzy classifier. The obtained fuzzy system is easily interpreted.

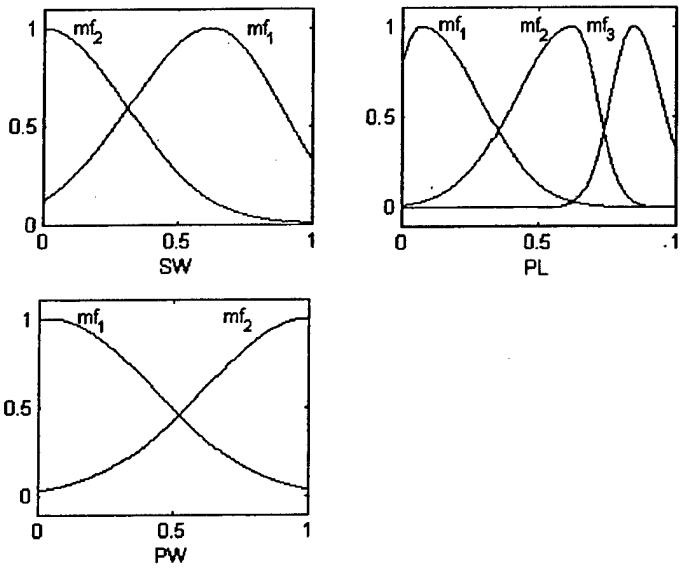


Figure 7.5 – VISIT membership functions for iris data

Table 7.8 – VISIT classification rules for iris data

Rule	SL	SW	PL	PW	Class
1	-	1	1	1	Setosa
2	-	1	2	1	Versicolor
3	-	2	2	1	Versicolor
4	-	1	3	2	Virginica
5	-	2	2	2	Virginica

Table 7.9 shows the comparison of results between the above fuzzy classifier system with other well-known classifier systems on total term sets, number of rules and recognition rate. The resulting system arrives at the highest degree of accuracy using the smallest number of term sets.

Table 7.9 – Comparison of results on iris data

	Term sets	Rules	Recognition rate on total data set (%)
Wang et al. [53]	11	3	97.5
Wu et al. [58]	9	3	96.2
Shi et al. [55]	12	4	98.0
Abe et al. [56]	—	3	98.7
Setnes et al. [51]	8 or 12	2or 3	99.3 or 98.9
Russo [57]	18	5	100
This chapter	7	5	99.3
Ishibuchi et al. [84]	7	5	98.0
Aboyi et al [85]	4	3	98.0

To estimate the performance of the proposed method on unseen data, the 5-fold cross-validation and the Leave-one-out experiments were performed on the iris data. In the 5-fold cross-validation experiment, the normalized iris data were divided into 5 disjoint groups containing 30 different patterns each, with 10 patterns belonging to each class. Then we derived fuzzy systems via the proposed method on all data outside one group and tested the resulting fuzzy classifier on the data inside that group. Finally, five fuzzy systems were derived. Table 7.10 reports the results of 5-fold cross validation. The average classification result is 98.8% correct (about 1.4 misclassifications) on the training data and 98.0% correct (about 0.6 misclassification) on the test data using 4.4 rules.

Table 7.10 – VISIT 5-fold cross-validation

	1	2	3	4	5	Avg
Rules	6	5	4	3	4	4.4
Training patterns	120	120	120	120	120	120
Misclassifications (training)	1	1	2	1	2	1.4
Recognition rate (training) (%)	99.2	99.2	98.3	99.2	98.3	98.8
Testing patterns	30	30	30	30	30	30
Misclassifications (testing)	0	0	1	1	1	0.6
Recognition rate (testing) (%)	100	100	96.7	96.7	96.7	98.0

In the Leave-one-out experiment, we iteratively apply the proposed method on the iris data set 150 times. Every time, we select a pattern as the testing set, and other 149 patterns as the training set. Table 7.11 contains the information of the derived fuzzy systems, as well as the corresponding training and testing errors. The last row in Table 7.11 demonstrates that the average result over 150 iterations is 98.0 % correct on the test set and 98.4% correct on the training set. This accuracy was obtained with an average of 4.75 fuzzy rules, 1.35 membership functions on SW, 1.43 mfs on SL, 3.17 mfs on PL and 3.03 mfs on PW.

Table 7.11– VISIT leave-one-out cross-validation on Iris data

	# Rules	SL	SW	PL	PW	Recognition Rate % (Training)	Recognition Rate % (Testing)
1	5.00	1.00	1.00	4.00	4.00	98.0	100
2	4.00	2.00	1.00	2.00	3.00	98.0	100
3	4.00	1.00	2.00	2.00	3.00	98.0	100
4	6.00	2.00	2.00	4.00	4.00	98.7	100
5	5.00	2.00	2.00	3.00	2.00	98.7	100
6	4.00	1.00	2.00	2.00	3.00	98.0	100

7	4.00	1.00	2.00	2.00	3.00	98.0	100
8	4.00	2.00	1.00	2.00	3.00	98.0	100
9	6.00	2.00	2.00	4.00	4.00	99.3	100
10	5.00	2.00	2.00	3.00	2.00	99.3	100
11	4.00	1.00	1.00	3.00	2.00	98.0	100
12	5.00	2.00	2.00	2.00	3.00	99.3	100
13	6.00	2.00	1.00	4.00	4.00	98.7	100
14	5.00	1.00	1.00	4.00	4.00	98.7	100
15	5.00	2.00	1.00	3.00	2.00	98.7	100
16	5.00	2.00	2.00	3.00	2.00	99.3	100
17	5.00	1.00	1.00	4.00	4.00	98.0	100
18	4.00	1.00	1.00	3.00	3.00	98.0	100
19	4.00	1.00	1.00	3.00	2.00	98.0	100
20	5.00	1.00	1.00	4.00	4.00	98.0	100
21	5.00	1.00	1.00	4.00	4.00	98.0	100
22	4.00	1.00	2.00	2.00	3.00	98.0	100
23	4.00	1.00	2.00	2.00	3.00	98.0	100
24	5.00	1.00	2.00	3.00	2.00	99.3	100
25	5.00	1.00	1.00	4.00	4.00	98.0	100
26	4.00	1.00	2.00	2.00	3.00	98.0	100
27	5.00	1.00	1.00	4.00	4.00	98.0	100
28	6.00	1.00	2.00	4.00	4.00	99.3	100
29	4.00	2.00	1.00	2.00	3.00	98.0	100
30	4.00	1.00	1.00	3.00	3.00	98.0	100
31	4.00	1.00	1.00	3.00	3.00	98.0	100
32	5.00	1.00	1.00	4.00	4.00	98.0	100
33	5.00	1.00	2.00	3.00	2.00	98.7	100
34	4.00	1.00	1.00	3.00	3.00	98.0	100
35	5.00	1.00	1.00	4.00	4.00	98.0	100
36	4.00	1.00	1.00	2.00	3.00	98.0	100
37	6.00	2.00	2.00	4.00	2.00	99.3	100
38	5.00	2.00	1.00	3.00	2.00	98.7	100
39	5.00	1.00	2.00	3.00	2.00	99.3	100
40	5.00	1.00	1.00	4.00	4.00	98.0	100
41	4.00	1.00	1.00	3.00	2.00	98.0	100
42	6.00	1.00	2.00	4.00	4.00	98.7	100
43	6.00	2.00	2.00	4.00	4.00	98.7	100
44	4.00	1.00	1.00	3.00	3.00	98.0	100
45	5.00	2.00	2.00	2.00	3.00	99.3	100
46	4.00	1.00	1.00	3.00	3.00	98.0	100
47	4.00	2.00	1.00	2.00	3.00	98.0	100
48	5.00	2.00	2.00	3.00	2.00	99.3	100

49	5.00	1.00	1.00	4.00	4.00	98.0	100
50	5.00	1.00	1.00	4.00	4.00	98.0	100
51	5.00	3.00	2.00	5.00	5.00	99.3	100
52	5.00	1.00	2.00	2.00	4.00	98.0	100
53	4.00	1.00	1.00	3.00	2.00	98.0	100
54	5.00	1.00	2.00	2.00	4.00	98.0	100
55	6.00	2.00	2.00	5.00	4.00	98.7	100
56	4.00	1.00	1.00	3.00	3.00	98.0	100
57	4.00	1.00	2.00	2.00	3.00	98.0	100
58	4.00	1.00	1.00	3.00	3.00	98.0	100
59	5.00	2.00	1.00	3.00	2.00	98.7	100
60	5.00	1.00	1.00	4.00	4.00	98.7	100
61	4.00	1.00	2.00	2.00	3.00	98.0	100
62	5.00	1.00	1.00	4.00	4.00	98.0	100
63	5.00	1.00	1.00	4.00	4.00	98.0	100
64	6.00	2.00	1.00	4.00	2.00	98.7	100
65	5.00	1.00	2.00	3.00	2.00	99.3	100
66	6.00	2.00	1.00	4.00	2.00	98.7	100
67	5.00	1.00	1.00	4.00	4.00	98.7	100
68	4.00	1.00	1.00	2.00	3.00	98.0	100
69	5.00	1.00	1.00	4.00	4.00	98.0	100
70	4.00	1.00	1.00	3.00	3.00	98.0	100
71	5.00	1.00	1.00	4.00	4.00	98.7	0
72	4.00	1.00	1.00	3.00	3.00	98.0	100
73	5.00	1.00	2.00	3.00	2.00	99.3	100
74	5.00	1.00	2.00	3.00	2.00	99.3	100
75	5.00	1.00	1.00	4.00	4.00	98.0	100
76	4.00	1.00	1.00	3.00	2.00	98.0	100
77	4.00	1.00	1.00	3.00	3.00	98.0	100
78	6.00	2.00	2.00	4.00	4.00	99.3	100
79	5.00	2.00	1.00	3.00	2.00	98.7	100
80	4.00	1.00	1.00	3.00	3.00	98.0	100
81	5.00	1.00	1.00	4.00	4.00	98.0	100
82	5.00	1.00	1.00	4.00	4.00	98.0	100
83	5.00	1.00	2.00	3.00	2.00	98.7	100
84	5.00	1.00	2.00	3.00	2.00	100	0
85	4.00	1.00	2.00	2.00	3.00	98.0	100
86	5.00	1.00	1.00	4.00	4.00	98.0	100
87	6.00	2.00	2.00	4.00	4.00	99.3	100
88	5.00	1.00	1.00	4.00	4.00	98.0	100
89	4.00	1.00	1.00	3.00	3.00	98.0	100

90	4.00	1.00	2.00	2.00	3.00	98.0	100
91	5.00	1.00	1.00	4.00	4.00	98.0	100
92	5.00	2.00	2.00	3.00	2.00	99.3	100
93	4.00	1.00	1.00	3.00	3.00	98.0	100
94	6.00	2.00	2.00	5.00	4.00	98.7	100
95	6.00	2.00	2.00	5.00	4.00	98.7	100
96	4.00	1.00	1.00	3.00	2.00	98.0	100
97	5.00	1.00	1.00	4.00	4.00	98.0	100
98	5.00	2.00	1.00	3.00	2.00	98.7	100
99	5.00	1.00	2.00	2.00	3.00	98.0	100
100	5.00	2.00	1.00	4.00	4.00	98.0	100
101	4.00	3.00	2.00	4.00	4.00	98.7	100
102	5.00	1.00	1.00	4.00	4.00	98.0	100
103	4.00	1.00	2.00	2.00	3.00	98.0	100
104	6.00	2.00	2.00	4.00	2.00	99.3	100
105	5.00	2.00	2.00	3.00	2.00	98.7	100
106	6.00	2.00	1.00	4.00	2.00	98.0	100
107	6.00	1.00	2.00	4.00	2.00	99.3	100
108	4.00	1.00	2.00	2.00	3.00	98.0	100
109	6.00	1.00	2.00	3.00	4.00	98.7	100
110	4.00	1.00	2.00	2.00	3.00	98.0	100
111	4.00	1.00	1.00	3.00	2.00	98.0	100
112	4.00	1.00	1.00	3.00	3.00	98.0	100
113	5.00	2.00	2.00	3.00	2.00	99.3	100
114	6.00	2.00	1.00	4.00	4.00	98.7	100
115	4.00	1.00	1.00	3.00	2.00	98.7	100
116	4.00	1.00	1.00	3.00	3.00	98.0	100
117	4.00	2.00	2.00	2.00	4.00	98.7	100
118	4.00	1.00	2.00	2.00	3.00	98.0	100
119	4.00	1.00	1.00	3.00	3.00	98.0	100
120	3.00	1.00	1.00	2.00	3.00	98.0	0
121	4.00	1.00	1.00	3.00	3.00	98.0	100
122	6.00	1.00	2.00	3.00	4.00	98.7	100
123	5.00	1.00	2.00	3.00	2.00	98.0	100
124	6.00	2.00	2.00	4.00	2.00	99.3	100
125	5.00	2.00	1.00	4.00	4.00	98.0	100
126	5.00	2.00	1.00	3.00	2.00	98.7	100
127	4.00	1.00	1.00	3.00	2.00	98.0	100
128	6.00	2.00	2.00	4.00	4.00	99.3	100
129	5.00	1.00	2.00	3.00	2.00	99.3	100
130	4.00	1.00	1.00	3.00	2.00	98.0	100
131	4.00	2.00	1.00	2.00	3.00	98.0	100

132	5.00	2.00	2.00	2.00	3.00	99.3	100
133	4.00	2.00	1.00	2.00	3.00	98.0	100
134	6.00	2.00	1.00	4.00	2.00	98.7	100
135	5.00	2.00	2.00	3.00	2.00	99.3	100
136	6.00	2.00	1.00	5.00	5.00	98.7	100
137	5.00	1.00	1.00	4.00	4.00	98.0	100
138	4.00	1.00	2.00	2.00	3.00	98.0	100
139	4.00	1.00	2.00	2.00	3.00	98.0	100
140	4.00	1.00	1.00	3.00	3.00	98.0	100
141	5.00	1.00	1.00	4.00	4.00	98.7	100
142	5.00	1.00	2.00	3.00	2.00	98.7	100
143	4.00	1.00	1.00	3.00	2.00	98.0	100
144	4.00	1.00	2.00	2.00	3.00	98.0	100
145	5.00	2.00	1.00	3.00	2.00	98.7	100
146	4.00	1.00	1.00	3.00	3.00	98.0	100
147	5.00	2.00	1.00	3.00	2.00	98.7	100
148	5.00	1.00	2.00	3.00	2.00	99.3	100
149	5.00	2.00	2.00	3.00	2.00	99.3	100
150	4.00	1.00	1.00	3.00	3.00	98.0	100
Average	4.75	1.35	1.43	3.17	3.03	98.4	98

Example 3 - Wisconsin Breast Cancer Data

The Wisconsin breast cancer data contains 699 patterns for two cancer states, "benign" and "malignant." The nine features involved are clump thickness (CT), uniformity of cell size (UC), uniformity of cell shape (UCS), marginal adhesion (MA), single epithelial cell size (SECS), bare nuclei (BN), bland chromatin (BC), normal nuclei (NN) and mitoses (MI). A total of 683 patterns are used to evaluate the performance because the other 16 patterns contain missing features. Of 683 valid patterns, 444 patterns pertain to benign class, and 239 patterns to malignant class. The proposed method is applied to design a fuzzy classifier for the Wisconsin breast cancer data. The population size was 200. To compare the performance with other classifiers, the first 400 patterns are used as the training set in the simulation, and the other 283 patterns as the testing set. After 100 iterations, the evolutionary process was terminated.

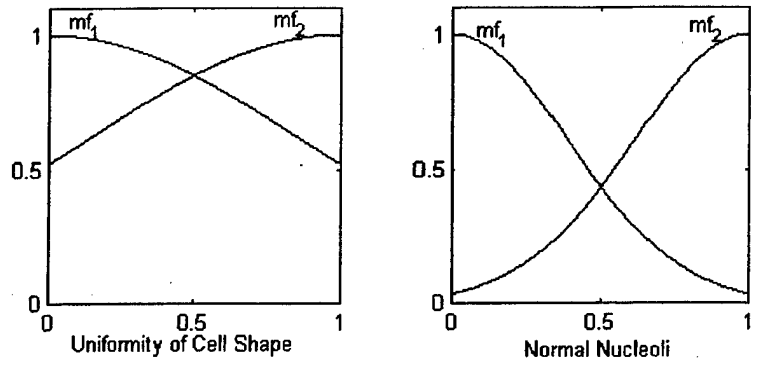


Figure 7.6 – VISIT membership functions for Wisconsin breast cancer data

Table 7.12 – VISIT classification rules for Wisconsin breast cancer data

CT	UC	UCS	MA	SECS	BN	BC	NN	MI	Class
1		1					1		1
2		2					1		2
3		1					2		2

The proposed algorithm created only a single membership function for features clump thickness (CT), uniformity of cell size (UC), marginal adhesion (MA), single epithelial cell size (SECS), bare nuclei (BN), bland chromatin (BC) and mitoses (MI). After deleting these redundant features, the resulting fuzzy system is quite simple with only two input features and three rules. Figure 7.6 shows the membership functions for features UCS and MN. Table 7.12 shows the three fuzzy rules. The classifier system generates 14 misclassifications (97.5% correct) on 400 training patterns, and 10 misclassifications (96.5% correct) on 283 testing patterns. This leads to a recognition rate of 96.5% (24 misclassifications) on the total data set.

Table 7.13 shows a comparison between the proposed method and other published models. Wang and Lee [53] applied half of the total 683 data sets as the training set and the other half as the testing data. They developed a 2-rule and 18-membership-function classifier system with a recognition rate of 96.3 % (about 25 misclassifications) in 10 trials performed on 10 different training sets. Nauck et al. [59] applied all patterns to training their fuzzy classifier. They initialized the NEFCLASS system with 3 fuzzy rules obtained by a modified GK clustering algorithm called "GK parallel." After 80 epochs of training, the classification rate was improved to 92.7% correct (50 misclassifications). Ishibuchi et al [82] also applied ten-fold cross validation procedures on the breast cancer dataset to test their NSGA-II based three-objective genetic rule selection method. The best result from this method is a fuzzy system with 4 rules of average length 1.5. The classification rate for this system is 97.6% correct (15 misclassifications) on the training data and 95.6% (3 misclassifications) correct on the test data. In [85], Aboyi et al also utilized their data-driven method to design fuzzy classifiers for classifying cancer data.

The best result is one with 2 rules, 3 term sets, and 96.5% accuracy (24 misclassifications).

Table 7.13 – Comparison results on Wisconsin breast cancer data

	Term sets	Rules	Recognition rate on total data set (%)
Wang et al. [53]	18	2	96.3
Nauck et al. [59]	27	3	92.7
Ishibuchi et al [82]	>4*	4	97.4*
Aboyi et al [85]	3	2	96.5
This chapter	4	3	96.5

* These data are estimated via the information from [82]

Example 4 - Pima Indians Diabetes Data

The Pima Indian diabetes data set contains 768 patterns with 8 features. This data set has significant class overlap, hence classification is somewhat more difficult in this case. The dataset is divided into two classes, indicating whether a Pima Indian individual is diabetes positive or negative. The features are based on measurements of Number of Times Pregnant (NTP), Plasma Glucose Concentration (PGC), Diastolic Blood Pressure (DBP), Triceps Skin Fold Thickness (TSFT), 2-Hour Serum Insulin (2HSI), Body Mass Index (BMI), Diabetes Pedigree Function (DPF), and Age (AGE).

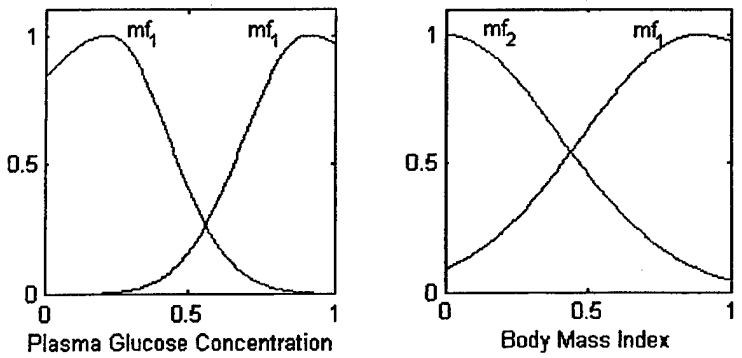


Figure 7.7 – VISIT membership functions for Pima data

In our experiment, the first 400 data pairs are used for the training set, and the other 368 data pairs are used for testing. The population size was 200, and the training process is stopped after100 iterations. After deleting redundant features, the resulting fuzzy system is compact and simple, having only two inputs, three fuzzy rules, and a total of four term sets. Figure 7.7 shows the membership functions for feature variables NTP and 2HSI. Table 7.14 shows the three fuzzy rules. The recognition rate of the classifier system is 75.8% correct on 400 training patterns, 78.2% correct on 368 testing patterns and 77.0% on the total data set.

Table 7.14 – VISIT classification rules for Pima data

	NTP	PGC	DBP	TSFT	2HSI	BMI	DPF	AGE	Class
1		1				1			1
2		1				2			2
3		2				2			2

Aboyi et al. [85] also utilized the Pima dataset to test their algorithm. In their 5-fold cross validation experiments, the average fuzzy system had 11.2 fuzzy rules, 40 fuzzy sets and an accuracy rate of 73.05%. These results indicate that our algorithm performs well even when there is significant class overlap.

7.6 Discussion and Conclusion

This chapter presents an automatic method to design fuzzy systems for classification via evolutionary optimization. Different fuzzy systems are defined on the same data set via the VISIT algorithm by randomly assigning different initial parameters $\mathbf{W}$, $\mathbf{A}$, $\mathbf{\Sigma}$, and $\mathbf{M}$. The evolutionary algorithm finds the optimal fuzzy system through simultaneously optimizing the parameters. Without any *a priori* knowledge of the distribution of the training data, the proposed approach can automatically develop a parsimonious fuzzy system from scratch. Simulation results performed on the wine data, iris data, Wisconsin breast cancer data, and Pima Indian diabetes data illustrate the power of the proposed approach.

The proposed algorithm is simple, and it does not require any techniques such as various clustering algorithms to provide the initial models as the basis for further tuning. Many other methods in Tables 7.5, 7.9 and 7.11 do need initial structure learning to determine the initial structure of the fuzzy system, including the rules, the shape of membership functions, the number of rules and the number of membership functions associated with each feature. For example, Wang and Lee [53] applied a MAC algorithm to find the initial structure and use a fast recursive linear/nonlinear least-squares algorithm to tune parameters. Setnes et al. [51] applied a fuzzy clustering technique to develop the structure of the fuzzy model, and then tuned the structure and parameters via a real-code GA combined with similarity-driven simplification. Shi et al. [55] utilized LVQ to detect the structure of the fuzzy system, then tuned rules and membership functions by an integer-coded GA.

For the evolutionary algorithm used in our method, a fuzzy expert system acts as the fitness function to evaluate individuals. The value of the MCE criterion and the number of rules are the inputs to the fuzzy expert system. Therefore, the evolutionary algorithm can investigate the accuracy and size of the rule base of the fuzzy system simultaneously, and leads to an accurate fuzzy system with a compact rule base. Moreover, we can shift the surface of the fitness function according to our preference or expert knowledge by adjusting the constant consequents in the fuzzy expert system rules and its membership functions.

8 Fuzzy PD+I Learning Control for a Pneumatic Muscle

8.1 Introduction

Fuzzy control [60]-[63] has been used in commercial and industrial applications in recent years. The fuzzy model reference learning control (FMRLC) algorithm developed from the general ideas in linguistic self-organizing control and conventional model reference adaptive control [64]. FMRLC has a learning ability that enhances the performance over time, so that it can improve the closed-loop system by using measured data from the plant's output to update the controller. The FMRLC technique has been used in the control of the cargo ship steering [65] and it has also been employed to improve the performance of the anti-skid braking system in various road conditions [66].

In this chapter, we utilize ideas from FMRLC to design a learning controller that results in length tracking for a pneumatic muscle. The controller is an adjustable PD type with a nonfuzzy parallel integral branch. The controller consequent singleton locations are dynamically adjusted by a fuzzy inverse model. The inverse model has for one of its inputs the interior pressure of the PM. This is effective in this case because the PM behaves differently when inflated as compared to when deflated.

8.2 PM Experimental Apparatus

The laboratory equipment used in connection with the experiments reported in this chapter includes the PM itself, the supporting frame, the inflation/deflation apparatus, the interfacing electronics, and the computer hardware and software used to implement the controller.

The pneumatic muscle is constructed of a rubber inner bladder made from a section of 7/8 inch diameter bicycle tire tubing enclosed in a plastic sheath material. The bicycle tubing is size 27-s-143, and the plastic mesh shell is 1.25 inch plastic sheathing used in electrical coaxial cable. For the experiments reported in this chapter, the PM assembly is hung vertically from the supporting structure. The upper end of the PM is hose-clamped to an adapter, which is attached to a source of compressed gas. The lower end is attached to a mass that is freely suspended. The upper end is also connected to air supply and drain proportional valves that regulate the amount of gas in the PM at any time. The valves used are two-way proportional valves that respond to voltage signals (Festo Corp. model MPPE-3-1/8-10-0101B)

The supporting stricture consists of an aluminum cage surrounded by plexiglass. This structure supports and surrounds the PM and the mass. The inflation/deflation apparatus consists of an outside source of compressed gas connected to the PM by an input gas line.

The computer hardware for the results in this chapter consists of a 200MHz PC with 16Mb memory interfacing with a general purpose ISA bus I/O card with digital I/O

and ADCs, along with a custom electronic interface control box with connectors. This box can select sampling rates of 1/64 second to 1/14400 second. Sensors available on the system include a pressure transducer for measuring PM interior pressure, a force transducer measuring the force applied by the PM, and a linear distance transducer used to measure the PM length.

The control programs are implemented using Borland Turbo C/C++ with a sampling time of 1/64 second. A sampling time of 1/128 second also works, but performance is essentially identical to 1/64 second.

8.3 Fuzzy Learning Control

Fuzzy control offers several advantages over conventional control techniques. An important advantage for problems such as tracking control of PMs is that the plant model is not required to be in any particular form, or even that the form is known. Fuzzy control is rather based on plant performance. An extension of fuzzy control is the fuzzy learning control methodology. In this method, the fuzzy controller is not fixed, but rather it is dynamically adjusted by a fuzzy inverse model. The inverse model incorporates expert knowledge about how the plant should behave and how the controller should be adjusted if the plant does not behave as desired. The inverse model dynamically adjusts the locations of the consequent centers of the fuzzy controller according to the measured behavior of the plant.

Our fuzzy learning controller is shown in Fig.8.1. It has two main parts: the fuzzy controller and the learning mechanism, which includes the fuzzy inverse model and knowledge-base modifier. We explain each of these below.

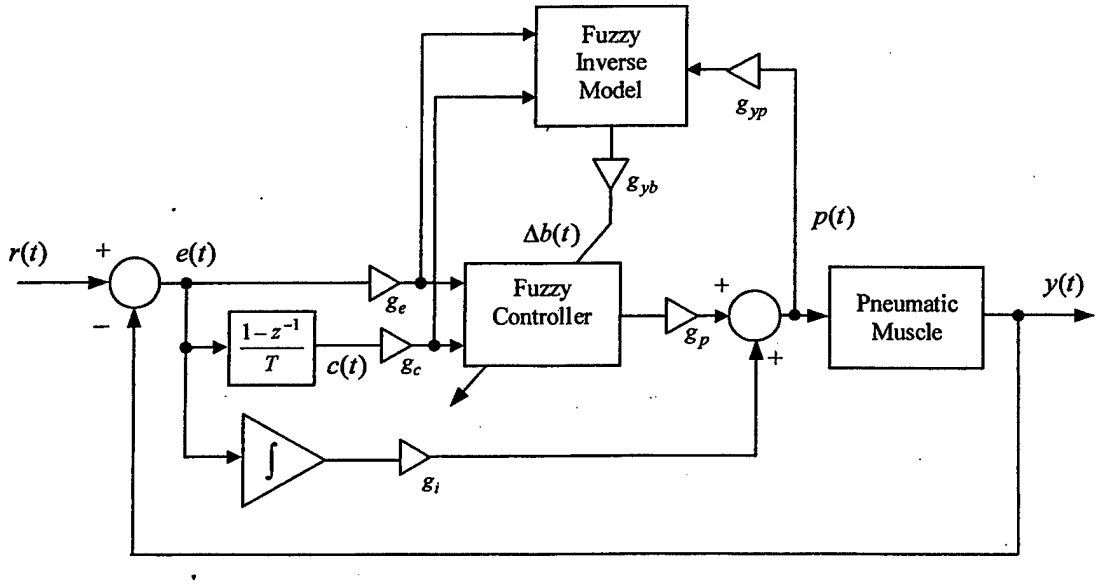


Figure 8.1- Fuzzy learning PD + I tracking controller

Fuzzy PD+I Controller

The controller is of PID type with a fuzzy PD part and a nonfuzzy integral branch. The fuzzy part has inputs $e(kT) = r(kT) - y(kT)$ and $c(kT) = (e(kT) - e((k-1)T))/T$ where $r(kT)$ is the reference signal to be tracked and T is the sampling interval. The controller output is the commanded PM pressure $p(kT)$. The integral branch of the controller is nonfuzzy because we have no expert knowledge about how to adjust consequent centers for this branch and fuzziness is not needed for any other reason (e.g. making the integral path nonlinear).

For the fuzzy part of the controller, we define 11 fuzzy sets on the e universe of discourse $(A_e^i(e), i = 1,\ldots,11)$ and 11 fuzzy sets on the c universe of discourse $(A_c^i(c), i = 1,\ldots,11)$. These fuzzy sets are characterized by 11 equally-spaced symmetrical triangular membership functions centered at [-1, -.8, -.6, -.4, -.2, 0, .2, .4, .6, .8, 1] shown in Figure 8.2.

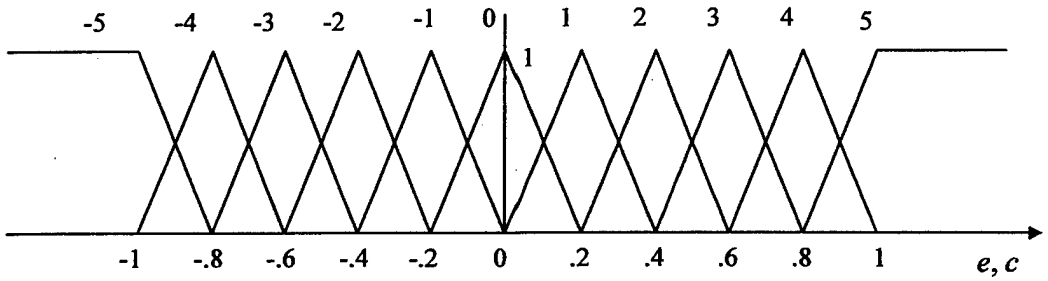


Figure 8.2 - e, c input fuzzy sets for controller and inverse model (linguistic-numeric values above memberships)

The choice of scaling gains g_e, g_c, and g_p generally come from the experience of the designer. As a first guess, the gain g_e is chosen so that the range of $e(kT)$ will lie in the interval [-1, +1], i.e. so that the values of $e(kT)$ do not result in the saturation of the outermost membership functions. The gain g_c is chosen in the same way, so that the value of e_c will not saturate in its outermost membership functions. We choose g_p to insure that signals input to the plant from the controller will not exceed the allowed range of the input to the plant.

The knowledge-base in the fuzzy controller has two parts, the rule-base and the fuzzy sets. The rule-base for the fuzzy controller contains the IF-THEN control rules of the form:

$$\text{Rule } i\text{: } \textbf{If } \tilde{e} \text{ is } \tilde{E}^j \textbf{ and } \tilde{c} \text{ is } \tilde{C}^k \textbf{ then } \tilde{p} \text{ is } \tilde{P}^l \qquad (8.3.1)$$

where $\tilde{e}$ and $\tilde{c}$ denote the linguistic variables associated with the fuzzy controller inputs $e(kT)$ and $c(kT)$, $\tilde{p}$ denotes the linguistic variable associated with the fuzzy controller output $p(kT)$, $\tilde{E}^j$ and $\tilde{C}^k$ denote the j^{th} and k^{th} linguistic value associated with $\tilde{e}$ and $\tilde{c}$, and $\tilde{P}^l$ denotes the l^{th} linguistic value associated with $\tilde{p}$.

The controller rule base consists of $11\times11=121$ rules, each rule with a separate consequent. Therefore we will have 121 singleton output memberships with the consequent membership for rule i at $b_i, i=1,\ldots,121$. The input membership functions are fixed and are not tuned. The locations of the output singletons are assumed to be unknown, and will be tuned by the fuzzy inverse model.

The degree to which a rule is "on" at time kT is calculated using the min T-norm. This gives rise to a premise membership function for Rule i (8.3.1) calculated as

$$\mu_{premise(i)}(e(kT),c(kT))=\min(\mu_{A_e^j}(e(kT)),\mu_{A_c^k}(c(kT))) \tag{8.3.2}$$

for $i=1,\ldots,121$. Finally we use the center-average defuzzification technique for the inference mechanism. Therefore, the input to the PM, which is pressure, is calculated through the defuzzification process as

$$p(kT)=\frac{\sum_i b_i(kT)\mu_{premise(i)}(e(kT),c(kT))}{\sum_i \mu_{premise(i)}(e(kT),c(kT))} \tag{8.3.3}$$

where the summation is taken over all 121 rules and $b_i(kT)$ denotes the center of the membership function as a consequence of the i^{th} rule at time kT. The centers $b_i(kT), i=1,\ldots,121$ are dynamically adjusted by the fuzzy inverse model and the learning mechanism.

The integral branch is a standard nonfuzzy integral function of the tracking error. A constant multiple of the integral of the error is added to the fuzzy PD output, making the controller essentially a PID controller in which the P and D parts are fuzzy and adjustable, while the I part is nonfuzzy and fixed. The addition of the integral of the tracking error serves to reduce steady-state error. The integral gain g_i is chosen so that steady-state error is small. If g_i is too large, however, the tracking performance becomes sluggish.

Incidentally, a controller similar to this was tried also, in which the integral part as well as the P and D part was fuzzy, making a true fuzzy PID learning controller. However, not much success resulted from this controller due to the fact that tuning was much more difficult with the integral branch of the controller implemented as a fuzzy system. Therefore, the I branch has been left fixed and nonadjustable in this research.

Learning Mechanism

The learning mechanism consists of a fuzzy inverse model and knowledge-base modifier. The fuzzy inverse model is a fuzzy system with inputs $g_e e(kT)$, $g_c c(kT)$, and PM internal pressure $p(kT)$. The inverse model output $\Delta b(kT)$ determines the amount that the fuzzy controller output memberships are changed at each time step. We choose pressure to be the third inverse model input since pressure has a direct effect on PM characteristics (2.3). Simulations indicate that controller P and D gains should be larger at low PM pressures than at high PM pressures. This is due to the fact that it requires more pressure input to effect a given amount of shortening when the PM is deflated than when it is inflated

The e and c input universes for the inverse model each have 11 fuzzy sets characterized by 11 equally-spaced symmetrical triangular membership functions identical to those of the controller (see Figure 8.2). The third input p to the inverse model has a universe of discourse consisting of 11 fuzzy sets characterized by 11 equally-spaced symmetrical triangular membership functions centered at [0, .1, .2, .3, .4, .5, .6, .7, .8, .9, 1], shown in Figure 8.3. Therefore, the inverse model has $11 \times 11 \times 11 = 1331$ rules.

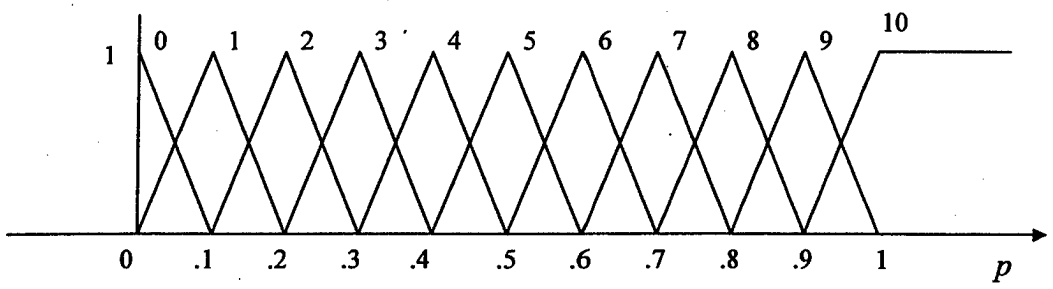


Figure 8.3 - Fuzzy sets for p input to inverse model
(linguistic-numeric values above memberships)

The output fuzzy sets for the inverse model are 17 singletons spaced as [-1, -.8, -.6, -.5, -.4, -.3, -.2, -.1, 0, .1, .2, .3, .4, .5, .6, .8, 1] in units of pressure (see Figure 8.4). The spacing is asymmetrical to give more change in pressure when the PM is running at low pressures (i.e. inflated a small amount, which corresponds to longer lengths) and less at high pressures (more inflation, shorter lengths). The output $\Delta b(kT)$ of the inverse model is calculated similarly to the output of the fuzzy controller of the previous section, i.e. minimum T-norm and center average defuzzification.

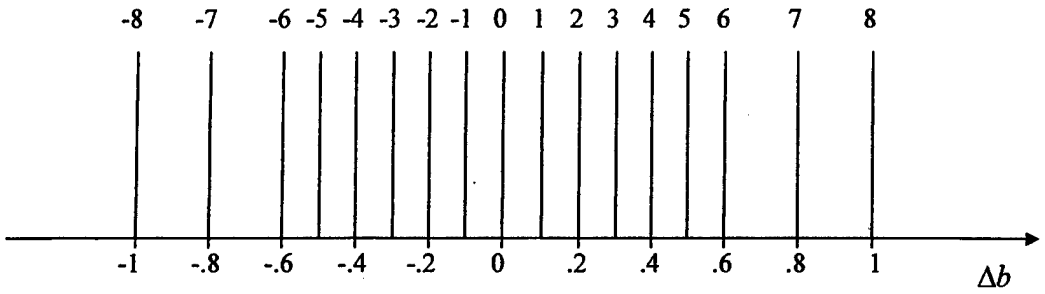


Figure 8.4 - Output singleton fuzzy sets for inverse model
(linguistic-numeric values above memberships)

Two representative parts of the inverse model rule base are shown in Figures 8.5a and 8.5b. Note that the tables list the linguistic-numeric values for the 17 output membership functions of the inverse model that are defined on the universe of discourse between the interval [-1, +1]. For example, when the error and derivative error are small and positive, and the pressure is low, we will find the output membership function center is at +0.8 (see highlighted box in Figure 8.5a). This means a relatively large positive increment should be added on the process input $p(kT)$ to insure the $y(kT)$ will not continue to decrease. On the other hand, for the same values in error and derivative error, but the pressure is now high, the center is at +0.4 (see highlighted box in Figure 8.5b), which means we only need half as much adjustment when the PM is running in high pressure.

		c										
Δb		-5	-4	-3	-2	-1	0	1	2	3	4	5
e	-5	-8	-8	-8	-8	-8	-8	-7	-6	-4	-2	0
	-4	-8	-8	-8	-8	-8	-7	-6	-4	-2	0	2
	-3	-8	-8	-8	-8	-7	-6	-4	-2	0	2	4
	-2	-8	-8	-8	-7	-6	-4	-2	0	2	4	6
	-1	-8	-8	-7	-6	-4	-2	0	2	4	6	7
	0	-8	-7	-6	-4	-2	0	2	4	6	7	8
	1	-7	-6	-4	-2	0	2	4	6	7	8	8
	2	-6	-4	-2	0	2	4	6	7	8	8	8
	3	-4	-2	0	2	4	6	7	8	8	8	8
	4	-2	0	2	4	6	7	8	8	8	8	8
	5	0	2	4	6	7	8	8	8	8	8	8

Figure 8.5a – One page of inverse model rule base for low PM internal pressure
(*p* linguistic-numeric values 0 – 4)

Δb		c										
		-5	-4	-3	-2	-1	0	1	2	3	4	5
e	-5	-5	-5	-5	-5	-5	-5	-4	-3	-2	-1	0
	-4	-5	-5	-5	-5	-5	-4	-3	-2	-1	0	1
	-3	-5	-5	-5	-5	-4	-3	-2	-1	0	1	2
	-2	-5	-5	-5	-4	-3	-2	-1	0	1	2	3
	-1	-5	-5	-4	-3	-2	-1	0	1	2	3	4
	0	-5	-4	-3	-2	-1	0	1	2	3	4	5
	1	-4	-3	-2	-1	0	1	2	3	4	5	5
	2	-3	-2	-1	0	1	2	3	4	5	5	5
	3	-2	-1	0	1	2	3	4	5	5	5	5
	4	-1	0	1	2	3	4	5	5	5	5	5
	5	0	1	2	3	4	5	5	5	5	5	5

Figure 8.5b – One page of inverse model rule base for high PM internal pressure (p linguistic-numeric values 5 – 10)

The knowledge-base modifier performs the modification of the PD part of the fuzzy controller to improve tracking performance. The knowledge-base modifier changes the knowledge-base of the fuzzy controller by adjusting the output centers of the rules that are "on" at time kT for the PD part of the controller according to the update law:

$$b_i(kT) = b_i((k-1)T) + \Delta b(kT) \tag{8.3.4}$$

Other update laws include (i) tempering the amount of update $\Delta b(kT)$ by the degree to which that particular rule is "on" at a given time, and (ii) updating the consequents for rules that were "on" n time steps in the past, where n is the estimated time delay of the plant. Neither of these schemes provided any improvement over the update law in (8.3.4).

Note that at each time step only a few output membership function centers are changed or updated, and the others are left unchanged. In every single time step, the centers changed only correspond to the rules that are "on." Therefore no more than 4 centers are changed for a single time step in our learning controller. This is because only two triangular membership functions can be enabled in each universe of discourse in one time step (i.e., 2 inputs will only turn on a maximum of $2^2 = 4$ rules at any one time).

Actually, the controller learns a new situation and also remembers it for the next time this situation is encountered. Hence, the next time we get similar values for the error, the controller can handle it better. This is the reason we call it "learning," rather than adaptive control.

8.4 Tracking Results

In this section we compare simulation and actual experimental results for tracking control of the pneumatic muscle. The simulation is carried out to conform as closely as possible to laboratory conditions at the PM test station in the Human Effectiveness lab at Wright-Patterson Air Force Base in Ohio.

Simulation Results

We can write equation (2.3.1) in state variable form as follows:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{1}{M}\left(-Kx_1 - Bx_2 + F - Mg\right) \end{aligned} \tag{8.4.1}$$

where $x_1 = y$, $x_2 = \dot{y}$, $g = 9.8$ m/sec^2, and $M = 29.9$ kg. These equations are simulated using a 4th-order Runge-Kutta algorithm with a step size of 0.001 seconds.

The fuzzy PD+I learning controller is simulated with a sampling interval of 1/64 seconds. The controller gains are chosen to be $g_e = 1$, $g_i = 0.1$, $g_c = 0.01$ and $g_p = 100$. The inverse model gains are chosen to be $g_{yp} = .01$ and $g_{yb} = 0.02$.

The initial position of the mass is 0 inch, corresponding to the PM fully deflated and extended. The command is a combination sinusoid/triangle signal. The tracking performance of the simulation is shown in Figure 8.6.

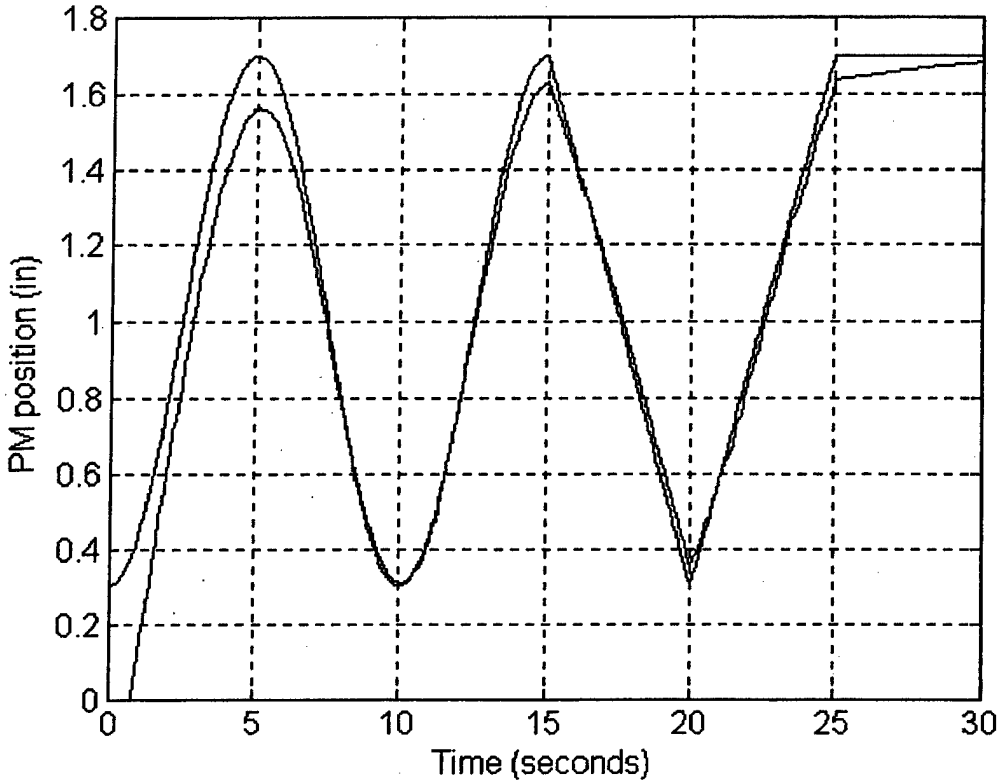


Figure 8.6 – Simulation tracking performance

Experimental Results

The PM was loaded with a mass of 29.9 kg. The fuzzy learning controller was implemented with a sampling time of 1/64 seconds. By extensive tuning, the best controller gains were determined to be $g_e = 1$, $g_i = 0.1$, $g_c = 0.01$ and $g_p = 100$. The inverse model gains are chosen to be $g_{yp} = .01$ and $g_{yb} = 0.02$. Thus the controller exactly conforms to the controller of the simulation.

The initial position of the mass is 0 inch, corresponding to the PM fully deflated and extended. The command is the same combination sinusoid/triangle signal used in the simulation. The experimental tracking performance is shown in Fig. 8.7.

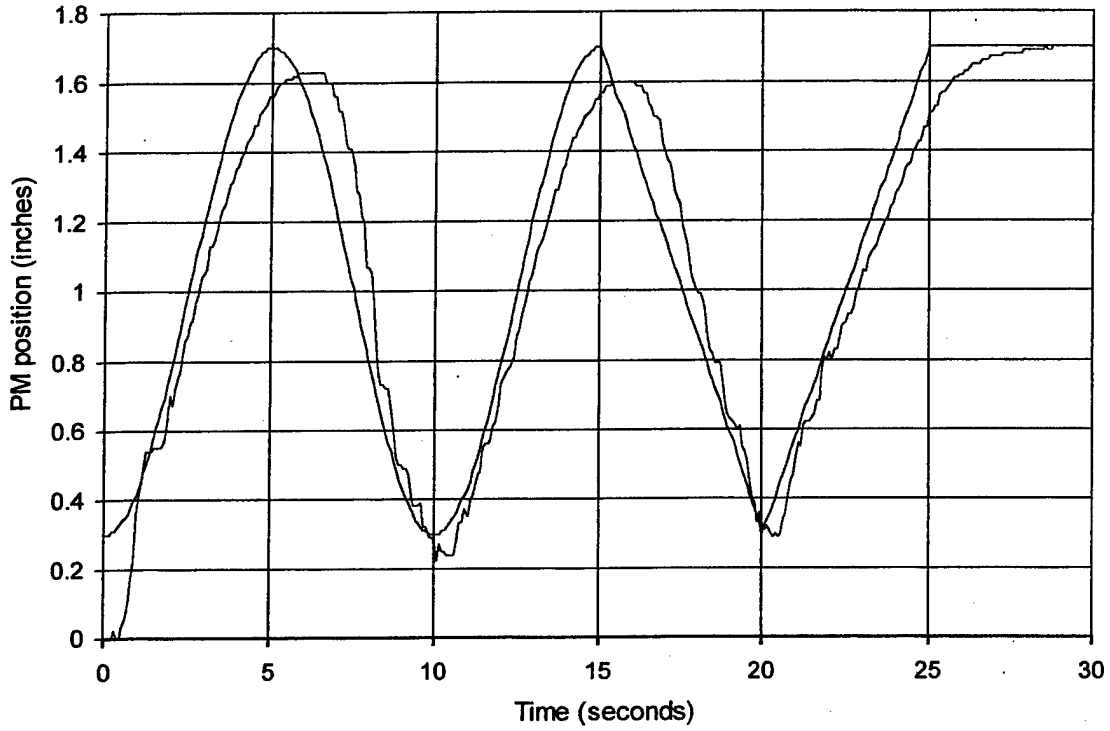


Figure 8.7 - Experimental tracking performance

8.5 Discussion

Our simulation and experimental results are fairly close, lending credibility to the model derived in [7]. The tracking performance in the simulation can be improved over that of the experimental results by increasing the scaling gains. However, these gains do not provide acceptable tracking when applied to the actual PM in the laboratory, and so are not presented here.

Experimental performance is inevitably different from simulation performance due to nonideal conditions present in the laboratory which are not accounted for in the simulation. Nonideal conditions in the lab include the fact that the supporting structure for the PM assembly is not absolutely rigid, giving rise to vibrations in the PM motion (see Figure 8.7 especially at deflation), the fact that PM length and pressure measurements are not exact, and the PM valves are not ideal.

It can be observed from Figures 8.6 and 8.7 that initial learning occurs during the first 1 - 2 seconds. After this, tracking is fairly accurate. Tracking performance can be expected to improve over time as more different situations (i.e. combinations of e, c, and p) are encountered and learning becomes more complete.

8.6 Conclusions

The problem addressed in this chapter is position tracking control for a pneumatic muscle actuator. Since the coefficients in the PM are poorly known and vary with time, we use a fuzzy PID-type tracking controller with learning ability. The controller consists of an adjustable PD fuzzy part and a parallel nonfuzzy integral branch. A comparison of simulation and experimental results shows general agreement between the two, lending credibility to a recently-derived PM model.

9 Fuzzy Control for Pneumatic Muscle Tracking via Evolutionary Tuning

9.1 Introduction

In general, fuzzy controllers have many parameters to be determined. The tuning of these parameters is typically done by a time-consuming trial and error process. Therefore, automatic tuning of design parameters is advantageous. Some studies ([67]-[69]) on evolutionary computation have shown that evolutionary algorithms (EAs) provide powerful ways to efficiently search in poorly understood, irregular spaces inspired by the mechanism of natural evolution. Following their successful applications in many difficult optimization problems, EAs are becoming increasingly used in the design of fuzzy systems. Work in EA-based design of fuzzy systems can be divided into two categories: rule-base construction and rule-base tuning. For rule-base construction, EAs are used for finding rules ([55], [70]) and determining the number of rules [71] from a data set. For rule-base tuning, EAs are used for optimizing membership functions in the fuzzy system according to some performance criterion [72].

In this chapter, we design a fuzzy P+ID controller for an actual pneumatic muscle system located in the Human Sensory Feedback Laboratory at Wright-Patterson Air Force Base in Dayton, Ohio. For this purpose, a recurrent neuro-fuzzy model of the PM is utilized, and on the basis of this model a modified evolutionary algorithm is used to tune the controller parameters. The resulting controller is then applied to the actual PM with excellent tracking results.

9.2 Evolutionary Fuzzy P+ID Control

Fuzzy controllers are gaining popularity in the control community due to their capabilities of dealing with uncertainty, but their systematic design is still an open problem. A typical fuzzy controller found in the literature often has many parameters, some of which are intercorrelated and co-dependent. It is often impossible to perceive the relationship between closed-loop performance and the parameters of the fuzzy controller. This increases the difficulty of design of fuzzy controllers. For these reasons, we propose an evolutionary fuzzy P+ID controller for PM control.

Fuzzy P+ID Controller

A fuzzy P+ID controller, shown in Figure 9.1, is constructed using an incremental fuzzy logic controller with output $\Delta x_f(k) = FLC(e, \dot{e})$ in place of the proportional term in a conventional PID controller. The PM input is a voltage $x(k)$ which is expressed in incremental form as

$$\Delta x(k) = x(k) - x(k-1)$$
$$= K_P S_{dx} \Delta x_f(k) + K_I T e(k) - K_D \frac{y(k) - 2y(k-1) + y(k-2)}{T} \qquad (9.2.1)$$

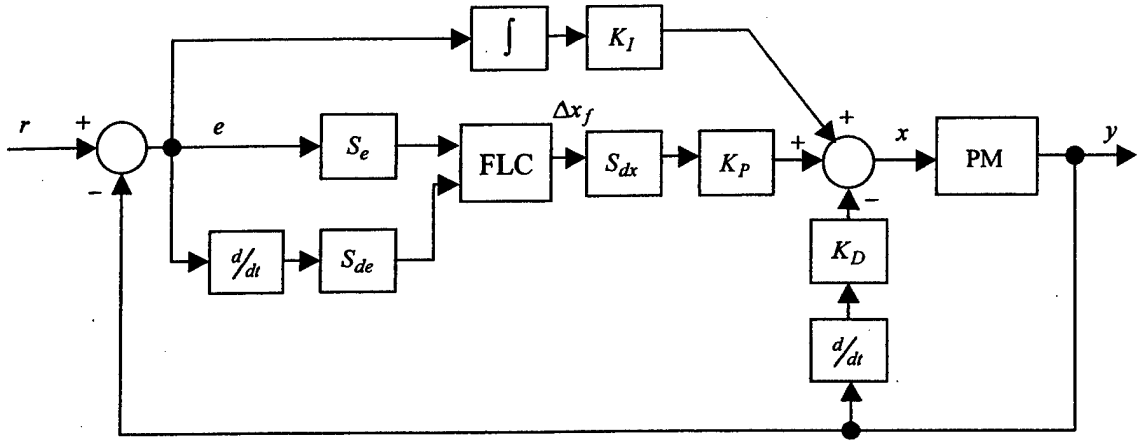


Figure 9.1 - Fuzzy P+ID controller for PM.

In (9.2.1), K_P, K_I, and K_D are PID control gains, S_{dx} is the output scaling gain of the fuzzy-P subsystem, and T is the step size. In general, scaling gains S_e and S_{de} in Figure 9.1 are chosen so that values of $S_e e$ and $S_{de}\dot{e}$ lie in the interval [-1, 1], and S_{dx} is often chosen to be equal to the reciprocal of S_{de}. In our approach, these scaling gains will be determined by an evolutionary algorithm.

To reduce the number of control parameters, we design membership functions for the fuzzy-P controller with the following assumptions:

1. $e(k)$ and $\dot{e}(k)$ universes of discourse have five triangular membership functions, i.e. NB, NS, ZO, PS and PB. These attain their maximum value of unity at $-\beta$, $-\alpha$, 0, α, and β respectively and form a partition of unity. This number of memberships is chosen to keep the evolutionary optimization task relatively simple while maintaining good control performance.

2. $\Delta x_f(k)$ universe has five singleton membership functions located at $-\beta$(NB), $-\alpha$(NS), 0(ZO), α(PS) and β(PB).

A general rule in the fuzzy-P part of the P+ID controller can be written as:

RULE i $(i = 1, 2, \ldots, 25)$:

If $e(k)$ *is* $\mu_e^j(e(k))$ *and* $\dot{e}(k)$ *is* $\mu_{de}^k(\dot{e}(k))$ *then* $\Delta x_f(k+1) = C^i$

where μ_e^j and μ_{de}^k are one of the fuzzy sets NB, NS, ZO, PS and PB. The consequent C^i takes on values in the set $\{-\beta, -\alpha, 0, \alpha, \beta\}$. The rule base of the fuzzy-P part of the controller is determined from expert knowledge and shown in Table 9.1. The crisp output of the fuzzy-P system then can be calculated as:

$$\Delta x_f(k+1) = \frac{\sum_i C^i \mu_i}{\sum_i \mu_i} \tag{9.2.2}$$

where $i = 1, \ldots 25$ and μ_i is the premise value of rule *i*.

Table 9.1 – Rule base for fuzzy P+ID controller

Δx_f		e				
		NB	NS	ZO	PS	PB
$\dot{e}$	PB	ZO	PS	PB	PB	PB
	PS	NS	ZO	PS	PB	PB
	ZO	NB	NS	ZO	PS	PB
	NS	NB	NB	NS	ZO	PS
	NB	NB	NB	NB	NS	ZO

There are three kinds of parameters in the fuzzy P+ID controller: fuzzy control parameters, PID parameters, and scaling gains for the fuzzy-P subsystem. The quantities α and β are fuzzy control parameters. The proper tuning of α and β will greatly improve performance. In general, α is related to steady-state accuracy and β is related to the response speed. Small α implies high steady-state accuracy; large β speeds up the transient response [73]. The PID parameters K_P, K_I, and K_D determine the control resolution and the stability of the controlled system. According to [73], the proven sufficient conditions show that the system stability is not destroyed when a fuzzy P+ID controller takes the place of a conventional PID controller. This means that closed-loop stability is guaranteed regardless of the choice of α and β. The quantities S_e, S_{de}, and S_{dx} are scaling gains for the fuzzy-P subsystem. Control parameters and scaling gains will be determined by an evolutionary algorithm. The PID parameters will be determined using the Ziegler-Nichols tuning method.

Evolutionary Design of Fuzzy P+ID Controller

The design of fuzzy controllers is complicated by the fact that we do not know the relationship between membership functions and the control performance. To avoid a tedious trial and error tuning process, the following evolutionary algorithm is proposed to automatically tune the five parameters α, β, S_e, S_{de}, and S_{dx} of the fuzzy P +ID controller.

Evolution Algorithm:

1. Initialize randomly an even number of P individuals.
2. Evaluate the fitness of the P individuals using the fitness function f defined in (9.2.3) below.
3. Select the $P/2$ fittest individuals and copy them as the first $P/2$ individuals of the next generation. Discard the other individuals.
4. Apply the mutation operator defined in (9.2.4) below to the individuals selected in 3 and generate another $P/2$ individuals for the next generation.
5. Go to Step 2 and repeat until the stopping criterion is met.

A real vector $p(n) = [\alpha(n), \beta(n), S_e(n), S_{de}(n), S_{dx}(n)] \in \Re^5$ is used in the above algorithm to represent the nth individual in the population, which can be evolved by a set of mutation operations and the elitist selection operation. In each generation, the individuals with fitness in the top 50% are used to create the population for the next generation. For control applications, the well-known ITAE criterion is often used to form the fitness function:

$$f = \frac{1}{\sum_{k=1}^{K} \left| \frac{e(k)}{r(k)} \right| kT} \tag{9.2.3}$$

where K is the total number of time steps in the calculation, $e(k)$ is the tracking error at the kth time instant, $r(k)$ is the desired output at the kth time instant, and T is the step size. Note that the reference signal $r(k)$ is always greater than zero. Thus, the larger the value of the fitness function, the better the tracking performance.

We define the following mutation operator for creation of new individuals from the fittest of the previous generation:

$$p_{j,m+1}(n + P/2) = p_{j,m}(n) + \sigma_j \rho_j(n) \exp\left(-\frac{f_n^m}{f_{\max}^m} \right) \tag{9.2.4}$$

where $j = 1, \ldots, 5$ indexes parameters in individuals, m is the generation number, $n = 1, \ldots, P/2$ is the index for individuals with the $P/2$ largest fitnesses in the current generation, P is the population size, σ_j is a learning coefficient which determines the amount of change of parameter j from generation to generation, ρ_j is a Gaussian random number $\in [-1, 1]$, f_n^m is the fitness of individual n in the mth generation, and $f_{\max}^m$ is the largest fitness in the mth generation.

In consideration of the constraints $0 \le \alpha < \beta \le 1$, the following techniques are used to deal with illegal individuals possibly generated by the mutation operators:

1. If $\alpha_{m+1}(n + P/2) < 0$, $\alpha_{m+1}(n + P/2) = \rho_\alpha(n)$, where ρ_α is a random number uniformly distributed $\in [0, 1]$.
2. If $\beta_{m+1}(n + P/2) > 1$, $\beta_{m+1}(n + P/2) = \rho_\beta(n)$, where ρ_β is a random number uniformly distributed $\in [0, 1]$.
3. If $\alpha_{m+1}(n + P/2) > \beta_{m+1}(n + P/2)$, exchange the values of $\alpha_{m+1}(n + P/2)$ and $\beta_{m+1}(n + P/2)$.

9.3 Experimental Results and Discussion

In this section, we design a fuzzy P+ID controller for a PM hanging vertically actuating a mass as shown in Figure 2.2. The system is located in the Human Sensory Feedback Laboratory at Wright-Patterson Air Force Base. The design procedure for the controller is as follows. First, a recurrent neuro-fuzzy model of the PM is derived from experimental data taken from the physical system using the VISIT and BP algorithms as outlined in Chapter 6.

This RNFIS model is then used in place of the actual PM system in the tuning of the five parameters of the fuzzy controller ($\alpha, \beta, S_e, S_{de}, S_{dx}$) using the evolutionary method discussed in Section 9.2 and also to determine the PID gains K_P, K_I, and K_D. The PID gains are found via the well-known Ziegler-Nichols tuning method applied to the identified recurrent neuro-fuzzy system. We discuss each of these below.

Evolutionary Design of Fuzzy P+ID Controller Using Neuro-fuzzy Model

An optimal fuzzy P+ID controller is designed for the pneumatic muscle via the evolutionary algorithm of the previous section based on the RNFIS model for the PM obtained in Chapter 6. Note that parameters α, β, S_e, S_{de}, and S_{dx} were optimized, so we needn't specify scaling gains in advance according to the actual ranges of e and $\dot{e}$. PID parameters were set as $K_P = -0.7$, $K_I = -0.22$, and $K_D = -0.5$ using the Ziegler-Nichols tuning method based on open-loop tests with the identified RNFIS model.

After 400 generations with a population of $P=120$, the following optimal parameters are obtained: $\alpha^* = 0.203$, $\beta^* = 0.632$, $S_e^* = 0.9982$, $S_{de}^* = 0.2232$, and $S_{dx}^* = 1.002$. Corresponding membership functions of the fuzzy controller are plotted in Figure 9.2, and its control surface is plotted in Figure 9.3.

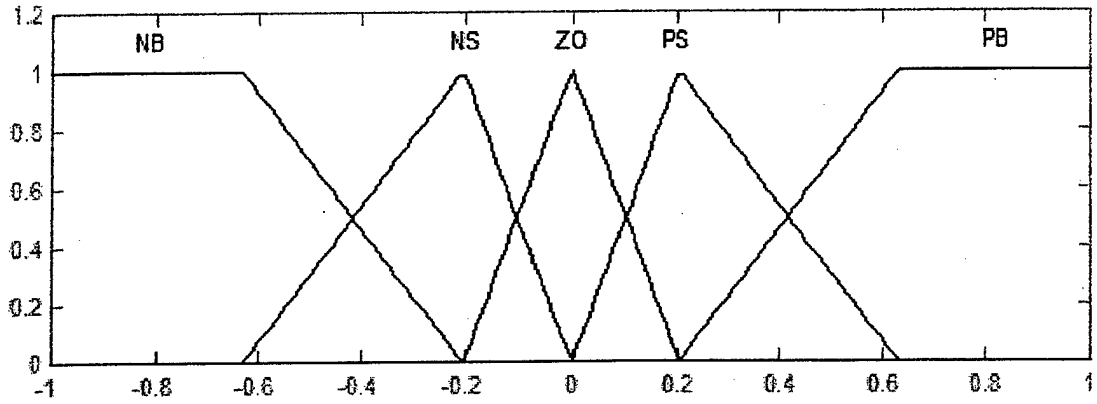


Figure 9.2 - Membership functions for e and $\dot{e}$

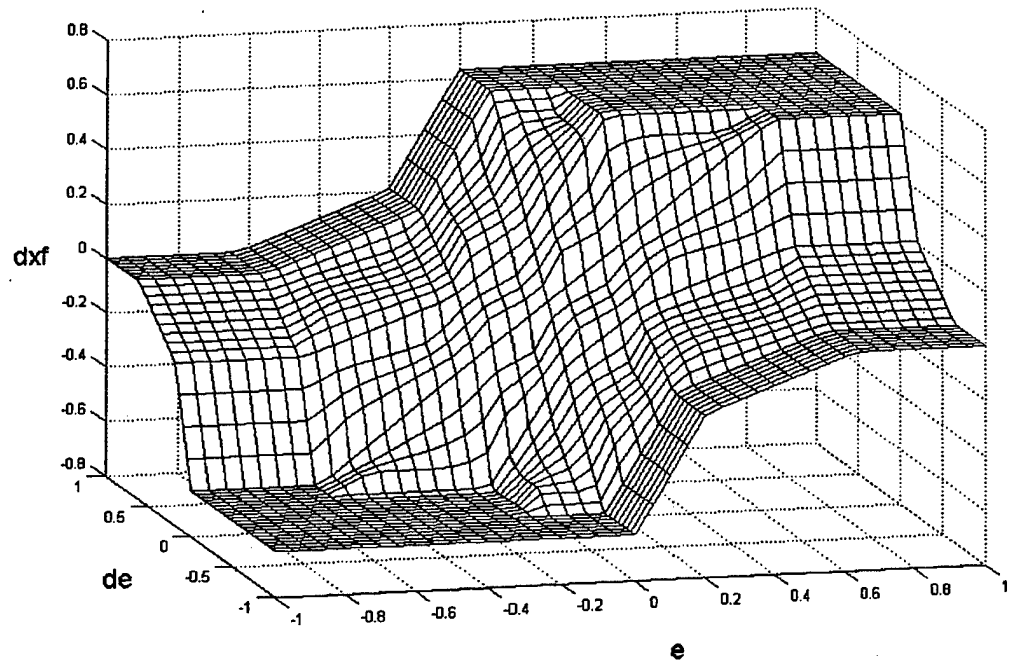
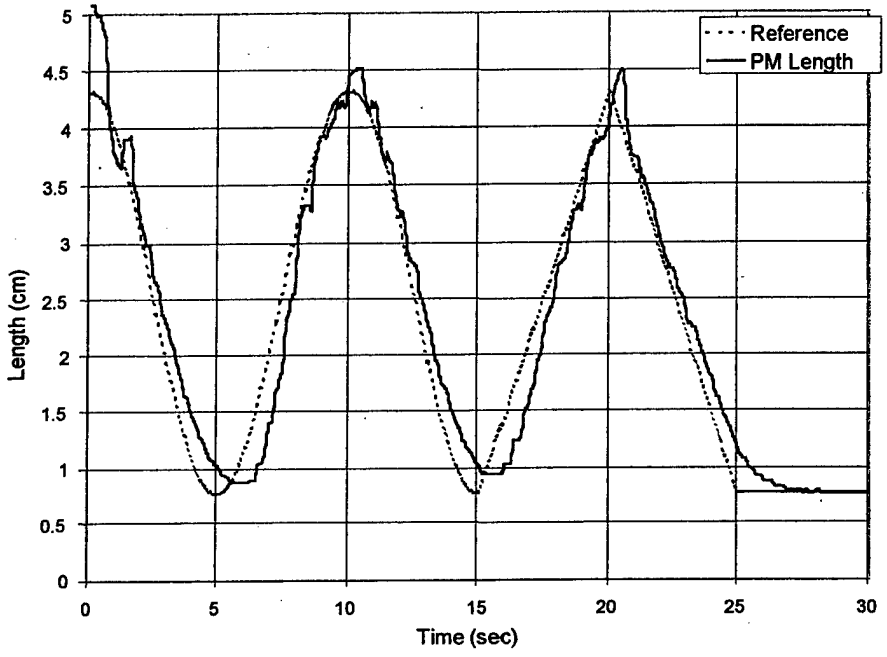
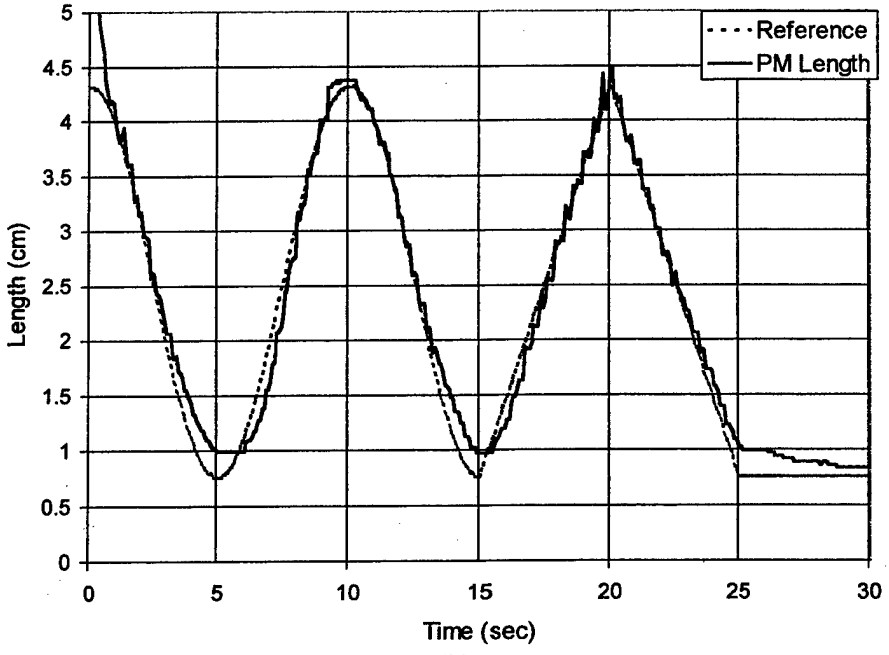


Figure 9.3 - Control surface for fuzzy-P part of controller

We tested the controller's ability to force the PM to track a reference signal. For these experiments the pneumatic muscle was loaded with a mass of 20 kg. The initial length of the muscle is 5 cm, corresponding to the PM fully deflated and extended. For comparison, the P+ID controller performance is compared to that of the fuzzy model reference learning controller (FMRLC) which was tuned by trial and error for best performance in Chapter 8. The tracking performance of both controllers is plotted in Figure 9.4, and a comparison of the tracking errors is shown in Figure 9.5. The reference signal is a combination sinusoid/triangle function.



(a)



(b)

Figure 9.4 - Tracking performance with (a) FMRLC controller, (b) fuzzy P+ID controller.

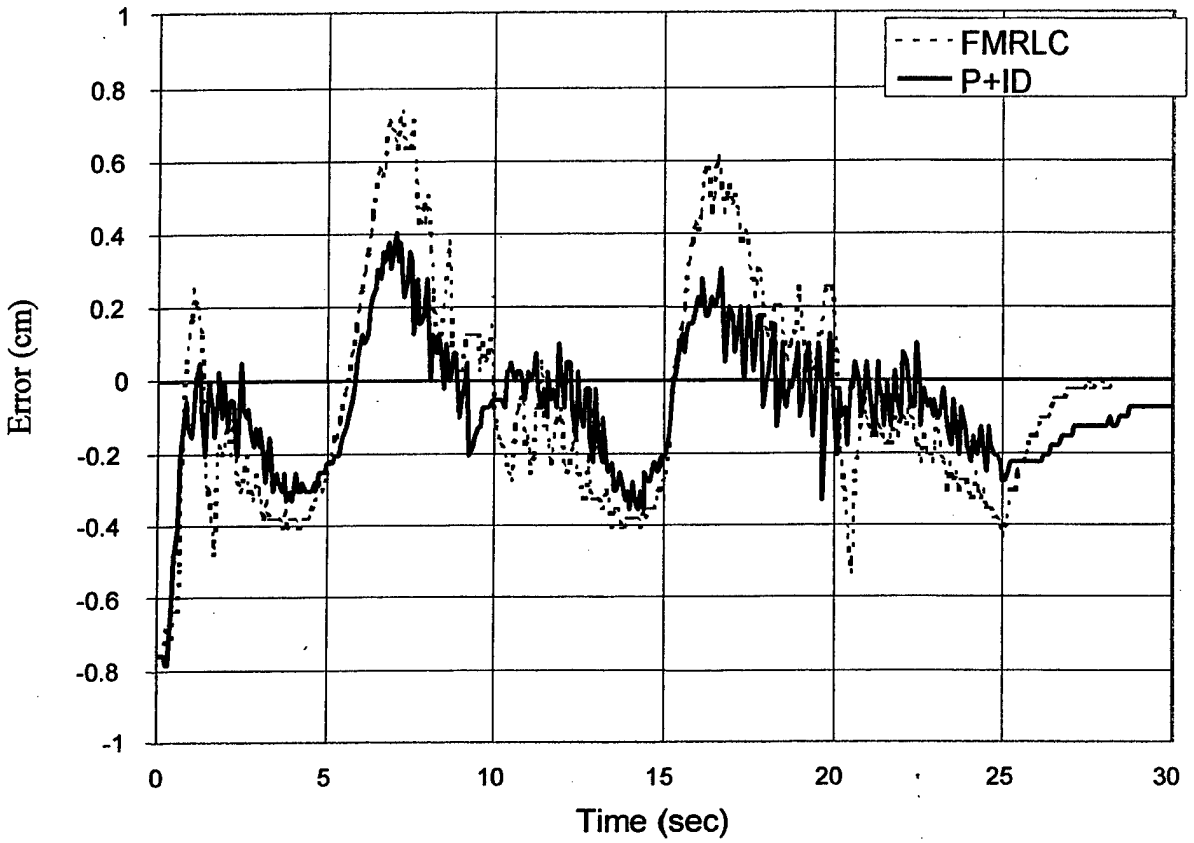


Figure 9.5 - Comparison of tracking errors

From Figure 9.5, we see that the evolutionary fuzzy P+ID controller performance is superior to that of the FMRLC i.e. the tracking error is less. To compare the tracking performances rigorously, we use the maximal deviation, defined as

$$MD = \min_{k}\left(\left|r(k) - y(k)\right|\right) \tag{9.3.1}$$

and the average deviation, defined as

$$AD = \frac{1}{K1}\left(\sum_{k}\left|r(k) - y(k)\right|\right) \tag{9.3.2}$$

where k ranges over all time samples, $r(k)$ is the reference signal, $y(k)$ is the PM length, and $K1$ is the total number of time steps. For the FMRLC, the maximal deviation is MD = 0.41 and the average deviation is AD = 0.1173. For the evolutionary fuzzy P+ID controller, we have MD = 0.16 and AD = 0.0558, demonstrating the superiority of the evolutionary fuzzy P+ID controller.

The control actions exerted by the two controllers in the above tracking performances are plotted in Figure 9.6. The control input from the evolutionary fuzzy P+ID controller is seen to be less than that from the FMRLC. Therefore, the evolutionary fuzzy P+ID controller attains better tracking performance using smaller control effort than the FMRLC. Because the control signal in the PM system corresponds to the volts supplied to the valve, this means that the fuzzy P+ID controller is more efficient, i.e. can use less energy and get better results. This can be valuable in applications where energy is limited, such as use in remote locations, underwater or spacecraft applications.

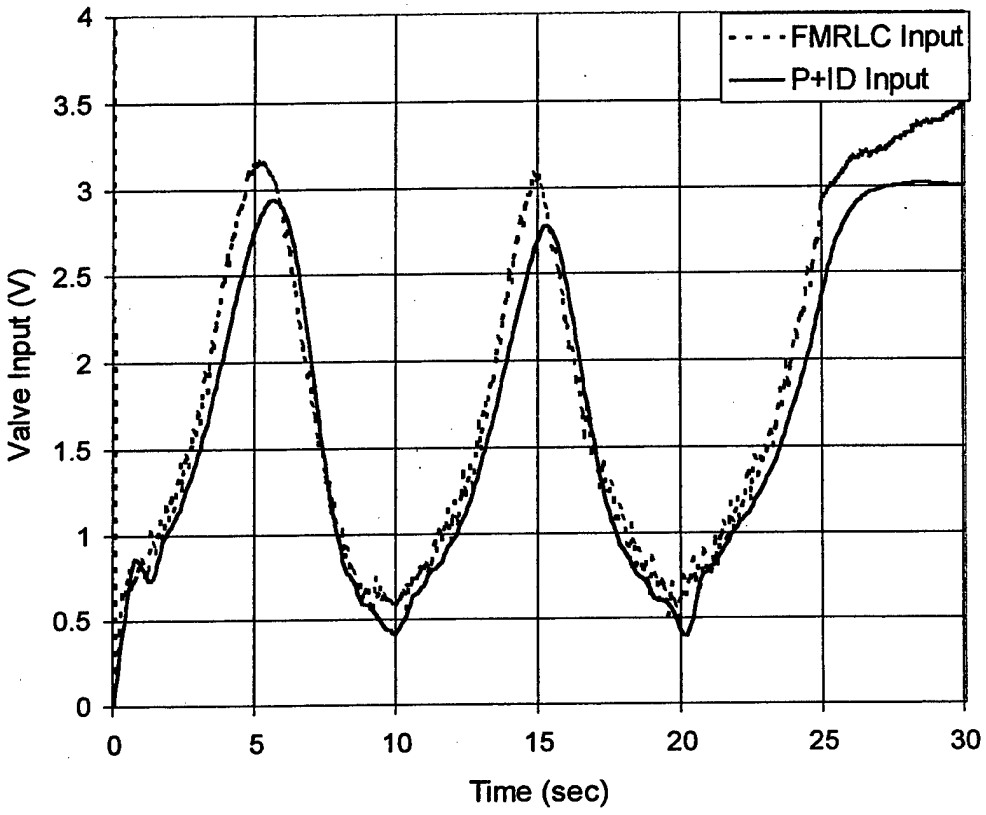


Figure 9.6 - Comparison of control actions

It should be noted that the FMRLC used for these experiments was tuned over many trials to yield good tracking. To be fair, however, these FMRLC results may not be the best attainable; better tracking may result from further tuning. In contrast, the excellent tracking results shown in Figure 9.4(b) for the evolutionary fuzzy P+ID controller were obtained upon first application of the controller, i.e. no tuning was required. Most fuzzy systems require extensive tuning before they perform satisfactorily. Therefore, the combination of neuro-fuzzy modeling and evolutionary design based on this model is a powerful design tool for fuzzy control.

10 Fuzzy Model Predictive Control for a Planar Arm Actuated by Four Pneumatic Muscle Groups

10.1 Introduction

Model predictive control (MPC) is based on the optimization of a finite-time functional of the plant tracking error and the control effort. It was originally developed for process control applications, which are typically highly nonlinear with slow dynamics. This is because of the typically high computational burden of MPC. With the advent of faster computation capabilities, MPC has found applications in control of plants with faster dynamics as well. MPC is especially advantageous for the control of highly nonlinear systems, such as robotic systems actuated by pneumatic muscles ([74]-[81]).

In this chapter, MPC is implemented by discretizing a continuous-time model of the arm actuated by PMs, then formulating an 81-rule Takagi-Sugeno model based on linearization of the arm model at 81 points in the state space. The controller is derived by finding the input sequence that minimizes a finite-time horizon of the tracking error. Simulations showing tracking performance of the arm demonstrate the effectiveness of the method. Also, we compare the performance of the MPC to that of the sliding mode controller of Chapter 5.

10.2 Takagi-Sugeno Model of Planar Arm Actuated by Four Groups of PMs

For development of the fuzzy MPC, we use the model of the planar arm developed in Chapter 5 (i.e. 5.3.13 - 5.3.15). Define the state vector for the arm $x = [\, x_1 \quad x_2 \quad x_3 \quad x_4 \,]^T = [\theta_1 \; \theta_2 \; \dot{\theta}_1 \; \dot{\theta}_2]^T$. Then from (5.3.13), the arm model can be written as

$$\dot{x} = g(x) + m(x)u \tag{10.2.1}$$

where

$$g(x) = \begin{bmatrix} x_3 \\ x_4 \\ a_1(x) \\ a_2(x) \end{bmatrix} \tag{10.2.2a}$$

$$m(x) = \begin{bmatrix} 0_{2\times 2} \\ G(x) \end{bmatrix} \tag{10.2.2b}$$

$$u = \begin{bmatrix} \Delta p_s \\ \Delta p_e \end{bmatrix} \tag{10.2.2c}$$

Also define an output y consisting of the two joint angles, i.e.

$$y = Cx \tag{10.2.3}$$

where

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{10.2.4}$$

Linearizing the model (10.2.1) about an operating point x^i gives

$$\dot{x} = A_i x + B_i u + D_i \tag{10.2.5}$$

where

$$A_i = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{\partial a_1}{\partial x_1} & \frac{\partial a_1}{\partial x_2} & \frac{\partial a_1}{\partial x_3} & \frac{\partial a_1}{\partial x_4} \\ \frac{\partial a_2}{\partial x_1} & \frac{\partial a_2}{\partial x_2} & \frac{\partial a_2}{\partial x_3} & \frac{\partial a_2}{\partial x_4} \end{bmatrix}_{x=x^i} \tag{10.2.6a}$$

$$B_i = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ G_{11}(x) & G_{12}(x) \\ G_{21}(x) & G_{22}(x) \end{bmatrix}_{x=x^i} \tag{10.2.6b}$$

and

$$D_i = \left[\begin{array}{c} 0 \\ 0 \\ a_1 - A_{i31}x_1 - A_{i32}x_2 - A_{i33}x_3 - A_{i34}x_4 \\ a_2 - A_{i41}x_1 - A_{i42}x_2 - A_{i43}x_3 - A_{i44}x_4 \end{array} \right]_{x=x^i} \tag{10.2.6c}$$

This linearized model is valid in the vicinity of the point x^i.

Defining a sample time T and approximating the derivative as

$$\dot{x}(t) = \dot{x}(kT) \approx \frac{x((k+1)T) - x(kT)}{T} \tag{10.2.7}$$

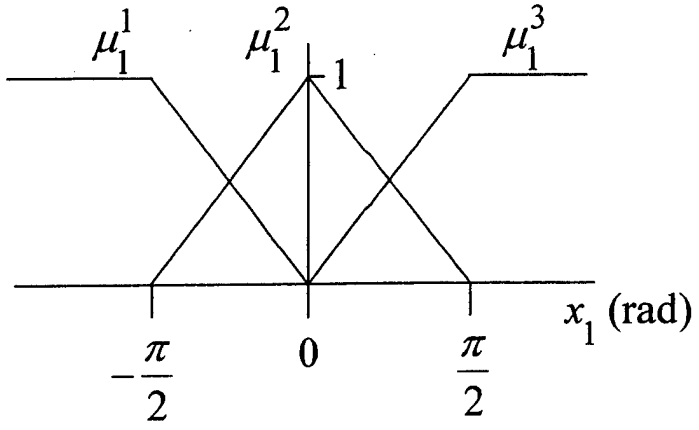
where k is an integer, the linear continuous-time model (10.2.5) can be approximated in the vicinity of x^i by the linear discrete-time system:

$$x(k+1) = (I + TA_i)x(k) + TB_iu(k) + TD_i \tag{10.2.8}$$

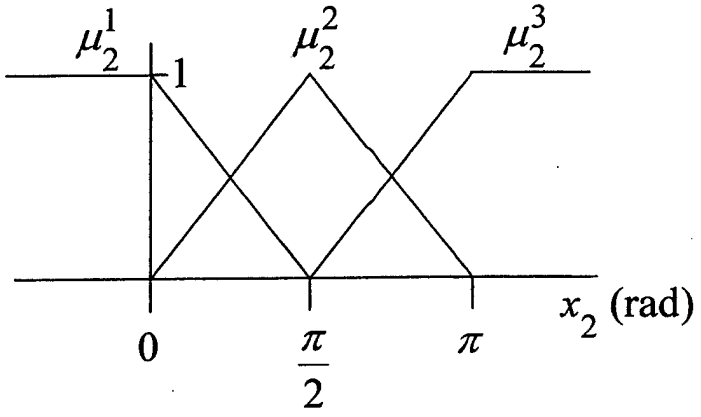
Now define three fuzzy sets on x_1's and x_2's universes of discourse characterized by the membership functions in Figures 10.1a and 10.1b. Similarly, define three fuzzy sets on x_3's and x_4's universes of discourse characterized by the membership functions in Figure 10.1c. Consider all possible combinations of the fuzzy set centers, defining $3^4 = 81$ points x^i, $i = 1, \ldots, 81$ in the state space about which to linearize (10.2.1). Discretizing these 81 linearized systems yields an 81-rule discrete-time TS fuzzy model of the arm. A rule in this model is

***Rule i* ($i = 1, \ldots, 81$) :**

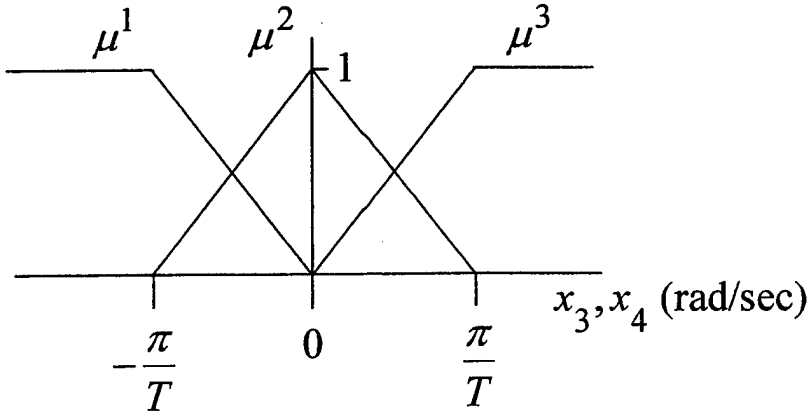
If $x_1(k)$ is μ_1^i and $x_2(k)$ is μ_2^i and $x_3(k)$ is μ_3^i and $x_4(k)$ is μ_4^i then $x(k+1) = (I + TA_i)x(k) + TB_iu(k) + TD_i$, $y(k) = Cx(k)$.



(a)



(b)



(c)

Figure 10.1 - Input membership functions: (a) x_1 universe, (b) x_2 universe, (c) x_3, x_4 universes

Note that the linearization is performed about the points listed in Table 10.1:

Table 10.1 - Linearization points for the 81 rules in rule base

Rule	x_1	x_2	x_3	x_4
1	$-\frac{\pi}{2}$	0	$-\frac{\pi}{T}$	$-\frac{\pi}{T}$
2	$-\frac{\pi}{2}$	0	$-\frac{\pi}{T}$	0
3	$-\frac{\pi}{2}$	0	$-\frac{\pi}{T}$	$\frac{\pi}{T}$
4	$-\frac{\pi}{2}$	0	0	$-\frac{\pi}{T}$
5	$-\frac{\pi}{2}$	0	0	0
6	$-\frac{\pi}{2}$	0	0	$\frac{\pi}{T}$
7	$-\frac{\pi}{2}$	0	$\frac{\pi}{T}$	$-\frac{\pi}{T}$
8	$-\frac{\pi}{2}$	0	$\frac{\pi}{T}$	0
9	$-\frac{\pi}{2}$	0	$\frac{\pi}{T}$	$\frac{\pi}{T}$
10	$-\frac{\pi}{2}$	$\frac{\pi}{2}$	$-\frac{\pi}{T}$	$-\frac{\pi}{T}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
81	$\frac{\pi}{2}$	π	$\frac{\pi}{T}$	$\frac{\pi}{T}$

The resulting global TS fuzzy model is given by

$$x(k+1) = \sum_{i=1}^{81} w_i(I + TA_i)x(k) + \sum_{i=1}^{81} w_i TB_i u(k) + \sum_{i=1}^{81} w_i TD_i \qquad (10.2.9)$$

where $w_i(x)$ is the degree of match of rule i for an input x, calculated as

$$w_i(x) = \frac{\mu_1^i(x)\mu_2^i(x)\mu_3^i(x)\mu_4^i(x)}{\sum_{j=1}^{81}\mu_1^j(x)\mu_2^j(x)\mu_3^j(x)\mu_4^j(x)} \qquad (10.2.10)$$

Therefore, the global TS fuzzy model is given by

$$x(k+1) = Ax(k) + Bu(k) + D \qquad (10.2.11)$$

where

$$A = \sum_{i=1}^{81} w_i (I + TA_i) \tag{10.2.12a}$$

$$B = \sum_{i=1}^{81} w_i TB_i \tag{10.2.12b}$$

$$D = \sum_{i=1}^{81} w_i TD_i \tag{10.2.12c}$$

In the next section, the matrices A, B, and D will be labeled $A(k)$, $B(k)$, and $D(k)$ because they change with every time step according to w_i, which depends on the values of the state x.

It should be noted that positive values of x_3 imply the shoulder tricep is inflating and the shoulder bicep is deflating, while negative values of x_3 imply the opposite. Similarly, positive values of x_4 imply the elbow bicep is inflating and the elbow tricep is deflating, while negative values of x_4 imply the opposite. This determines which values of B_0 and B_1 (2.3.2) to use when performing the linearization in (10.2.6).

The matrices for the 81-rule TS fuzzy model of the arm are listed in Appendix A of this report. In practice, a TS fuzzy model could be derived from experiments on the actual arm, rather than by linearizing a mathematical model.

10.3 Model Predictive Control of Planar Arm Actuated by Four Groups of PMs

Define the values of matrices A, B, and D (10.2.12) at time step k to be $A(k)$, $B(k)$, and $D(k)$. Then the TS fuzzy model of the arm actuated by PMs is

$$x(k+1) = A(k)x(k) + B(k)u(k) + D(k) \tag{10.3.1a}$$
$$y(k) = Cx(k) \tag{10.3.1b}$$

Since $CB(k) = 0_{2\times 2}\ \forall k$, from (10.3.1), we have

$$y(k+2) = CA(k+1)A(k)x(k) + CA(k+1)B(k)u(k) + CA(k+1)D(k) \tag{10.3.2}$$

where we have used the fact that $CB = CD = 0$.

Let $r(k)$ be a vector of desired reference input trajectories and consider the performance index

$$J = [y(k+2) - r(k+2)]^T P[y(k+2) - r(k+2)] + u^T(k)Qu(k) \quad (10.3.3)$$

where P and Q are symmetric positive definite weighting matrices. Using (10.3.2) and setting $\partial J/\partial u(k) = 0$ yields the optimal control

$$u^*(k) = [B^T(k)A^T(k+1)C^T PCA(k+1)B(k) + Q]^{-1} \cdot$$

$$\cdot B^T(k)A^T(k+1)C^T P[r(k+2) - CA(k+1)A(k)x(k) - CA(k+1)D(k)] \quad (10.3.4)$$

Since $A(k+1)$ is not yet known, we initially use $A(k+1) = A(k)$ in (10.3.4) and use this to calculate an interim value for $u(k)$ via (10.3.4). Now this interim value of $u(k)$ is used to calculate a predicted value for $x(k+1)$ on the basis of the TS fuzzy model (10.3.1). From this predicted $x(k+1)$, a predicted value for $A(k+1)$ can be obtained as in (10.2.10), (10.2.12). This predicted value for $A(k+1)$ along with the known $A(k)$, $B(k)$, and $D(k)$ and the true plant state $x(k)$ are then used again in (10.3.4) to calculate a final optimal control at time k, i.e. $u^*(k)$. This process is repeated at every time step.

10.4 Simulation Results

The planar arm of Figure 5.3 with opposing-pair PM groups actuating the shoulder and elbow joints is simulated using a 4th-order Runge-Kutta algorithm with a step size of 0.01 seconds. Let $l_1 = l_2 = 0.46$ m, $l_{c1} = l_{c2} = 0.23$ m, $m_1 = m_2 = 10$ kg, $r_s = 7.62$ cm, and $r_e = 5.08$ cm, $n_s = 6$ and $n_e = 3$. Thus the conditions are identical to those of the Chapter 5 simulations. For these simulations we assume all physical quantities of the arm, i.e. masses, lengths, etc. are exactly known, but that the PM coefficients, i.e. F, K, and B are not known with precision. Assume all 12 shoulder PMs (6 pairs) are matched to each other, but not to the elbow PMs. Similarly assume all 6 elbow PMs (3 pairs) are matched to each other, but not to the shoulder PMs.

The model predictive controller is designed according to (10.3.4) using ideal values for all F, K, and B coefficients (see Section 2.3). To investigate robustness of the model predictive controller and compare it to the sliding mode controller, we randomly choose three sets of actual (nonideal) F, K, and B coefficients from a uniform distribution within $\pm 50\%$ of their nominal values. The coefficients used are listed in Table 5.1. Note that, for comparison purposes, we use the same coefficients for the MPC that were used for the sliding mode controller of Chapter 5.

For all simulations, we use $P = 3 \times 10^6 I_{2\times 2}$, $Q = 10^{-1} I_{2\times 2}$, and $T = 0.01$ sec. Also let $P_{0bs} = P_{0te} = 310.3\,\text{kPa}$ (45 psi), $P_{0ts} = 490.4\,\text{kPa}$ (71.1 psi) and $P_{0be} = 310.5$

kPa (45 psi). These noiminal pressures insure that if control is lost, shoudler and elbow angles will revert to the equilibrium positions $\theta_{1eq} = -\pi/4$, $\theta_{2eq} = \pi/2$ (see Chapter 5). With these nominal pressures and with n_s, n_e as above, all PM pressures remain in the allowable range $206.844 \leq P \leq 620.532$ kPa ($30 \leq P \leq 90$ psi) for all control tasks in this section.

We investigate controller performance in tracking a sinusoidal spline in x-y space (5.6.2). The spatial tracking performance for the true plant with PM coefficients in set 1 is shown in Figure 10.2. Tracking performance when PMs are described by coefficient sets 2 and 3 are similar to Figure 10.2 and are not shown. It will be noted that there is some spatial tracking error, which is to be expected due to the parameter errors. For comparison purposes, tracking performance for the identical plant using sliding mode control (Chapter 5) is shown in Figure 5.4.

The MPC elbow angle tracking error (which is larger than the shoulder angle tracking error in this case) for coefficient sets 1, 2, and 3 is shown in Figure 10.3. For comparison purposes, the elbow tracking errors for the same three coefficient sets under sliding mode control from Chapter 5 are shown in Figure 5.6.

Figure 10.4 shows the control efforts Δp_s, Δp_e that produced the tracking performance in Figure 10.2. It will be noted that with the nominal pressures P_{0ts}, P_{0bs}, P_{0te}, and P_{0be} given above, the PM input pressures remain within the allowable range for these PMs. For comparison purposes, the control efforts from sliding mode control of the identical plant are shown in Figure 5.5.

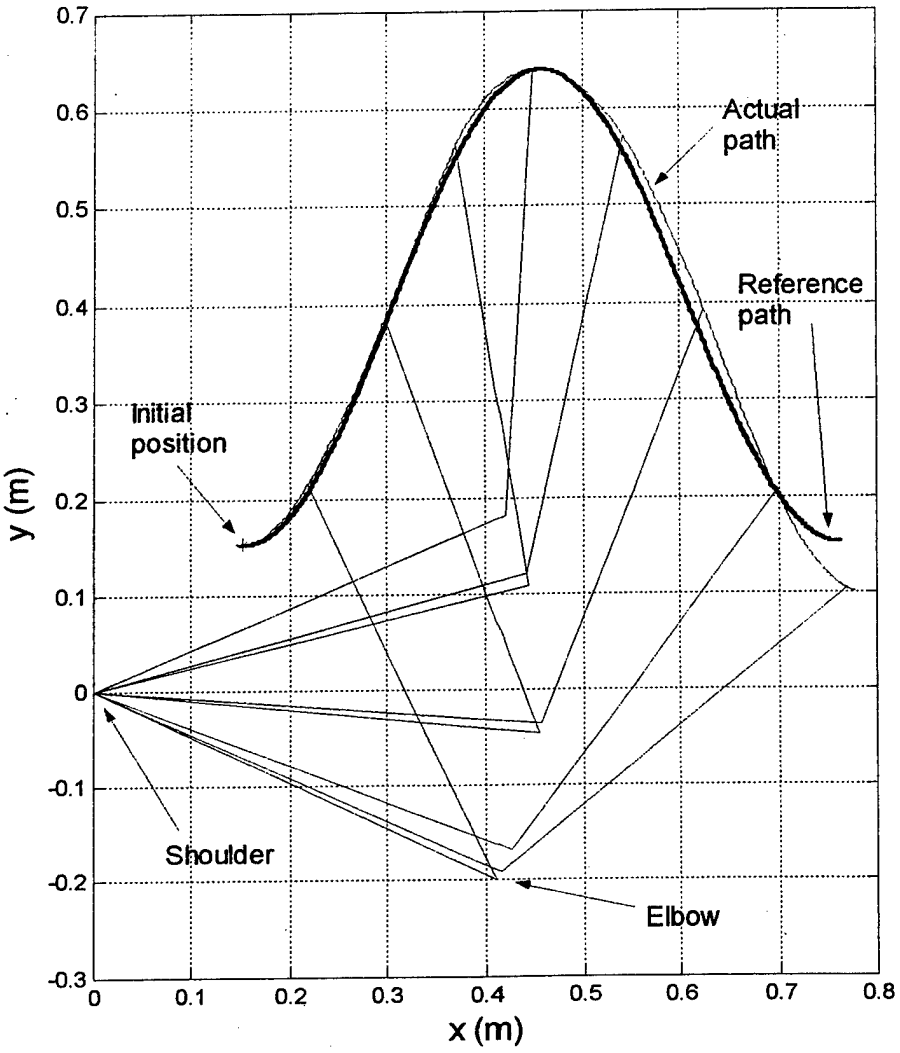


Figure 10.2 - Fuzzy MPC spatial tracking behavior, PMs in coefficient set 1, $m_1 = m_2 = 10$ kg.

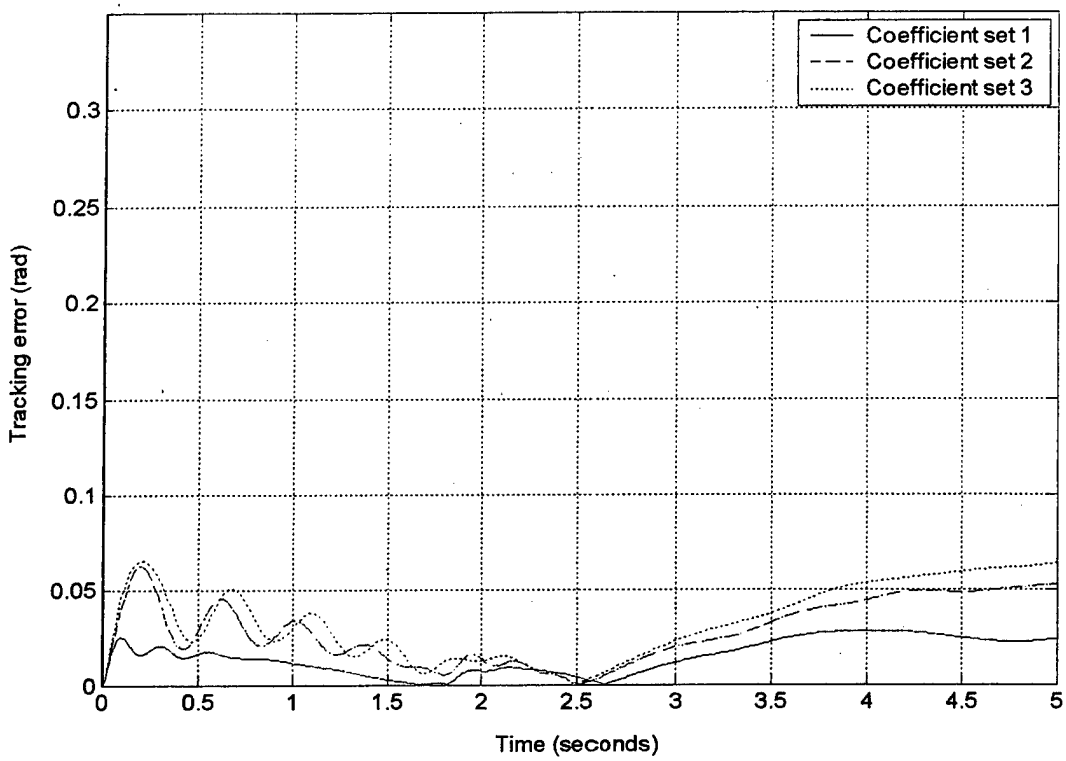


Figure 10.3 - MPC elbow angle tracking errors for 3 different plants (PM coefficient sets 1, 2, and 3), sinusoidal spline reference trajectory, $m_1 = m_2 = 10$ kg.

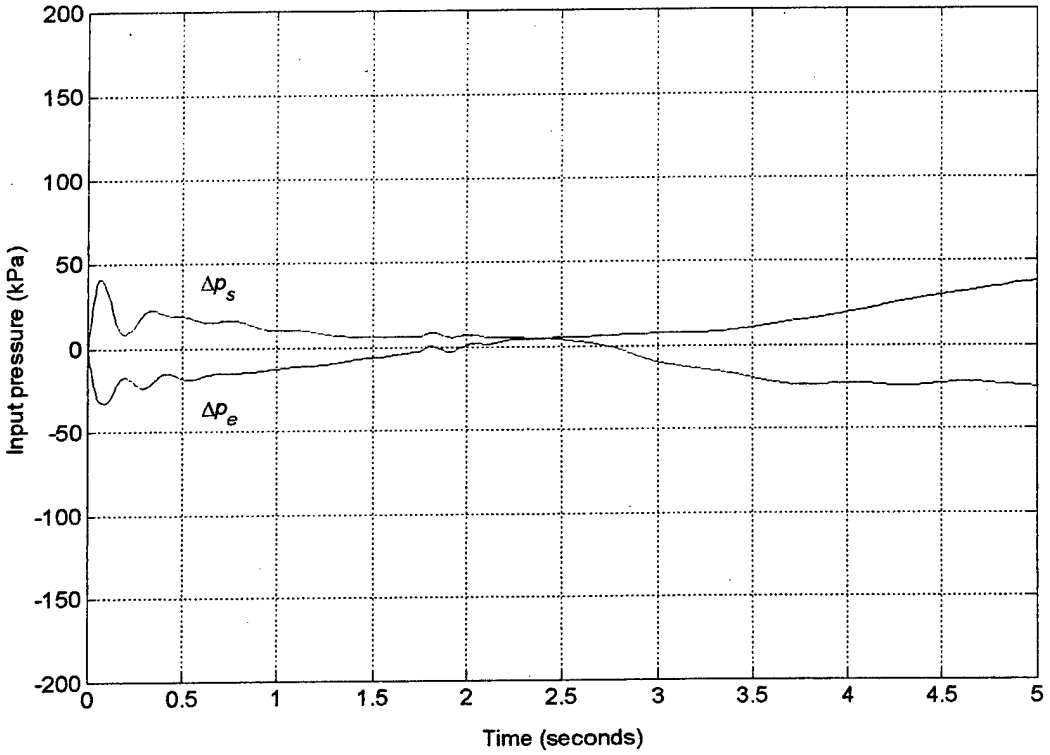


Figure 10.4 - MPC control effort producing tracking performance of Figure 10.2.

Sinusoidal spline, doubled mass

In practical applications, it may be expected that the mass actuated by the arm will change. To investigate the robustness of the sliding controller to changing masses, we increased the arm masses m_1, m_2 each by a factor of 2 to 20 kg and used the same model predictive controller as above to track the sinusoidal spline reference trajectory. Both the shoulder and elbow angle tracking errors are similar to those in Figure 10.3, indicating that the fuzzy model predictive controller is robust to changes in mass.

Figure 10.5 shows the control effort produced by the fuzzy model predictive controller for this plant. Naturally, the control effort is larger due to the increased arm masses. It is seen that with the above nominal pressures, the PM pressures remain within the allowable range for the duration of the control process. In Figure 5.13, for comparison purposes, we show the corresponding control efforts resulting from sliding mode control of the identical plant.

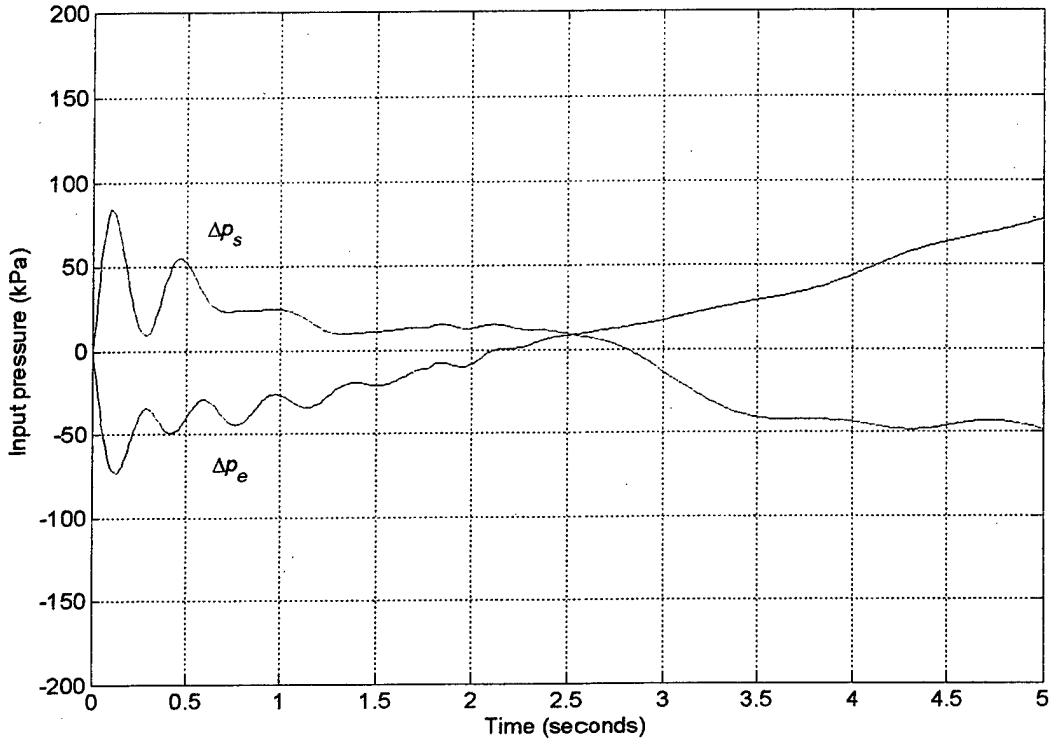


Figure 10.5 - MPC control effort, sinusoidal spline reference trajectory, coefficients in set 1, $m_1 = m_2 = 20$ kg.

10.5 Discussion and Conclusions

A fuzzy model predictive controller has been designed for a 2 DOF planar arm assembly with highly nonlinear pneumatic muscle actuators in opposing pair configurations actuating the shoulder and elbow joints. The control input for the planar assembly enters the process through nonlinear spring and friction coefficients and a nonlinear contractile force term that are contained within a mathematical model for the pneumatic muscle actuators. A dynamic model for the arm with four groups of PM actuators is derived. This model is linearized about 81 operating points in the state space to produce an 81-rule TS fuzzy model of the arm. A two-input model predictive controller is designed on the basis of this TS fuzzy model.

Simulations of closed-loop tracking were performed with a sinusoidal spline spatial reference path desired for the end effector. Closed-loop tracking performance, resulting from simulations, is seen to be in line with corresponding results obtained with sliding mode contoller for the identical plant. Closed-loop tracking with several sets of PM coefficients within a ±50% range of ideal are compared. The model predictive controller is also shown to be robust for a 100% change in arm masses.

It should be noted that P and Q chosen for our MPC were not the best possible. They were chosen to give similar results to those of Chapter 5. The MPC tracking performance can always be improved by adjusting P and Q appropriately. Similar statements could be made about the sliding mode controller. Sliding mode control has several user-chosen parameters that can profoundly affect performance. For this reason, it is not meaningful to attempt to determine which of the two controllers is "best."

However, the following can be stated from a comparison of sliding mode and MPC:

1. From a comparison of Figures 5.4 and 10.3, we see more oscillatory arm movement with the fuzzy MPC designed above than with sliding mode controller of Chapter 5, especially in the initial stages. This can be a disadvantage for flexible systems where unmodeled higher frequency modes can be excited by vibrations.
2. The sliding mode controller produces greater initial tracking errors but smoother overall arm movement, while fuzzy MPC produces smaller initial errors but greater errors in the middle and end of the task..
3. From Figures 5.6, 10.4, 5.5, and 10.5, we see greater initial control effort (i.e. pressure input to the PMs) with sliding mode control than with fuzzy MPC. This can be a significant factor, since input pressure is always limited by the supply pressure, which is typically less than 1000 kPa.
4. After initial transients, sliding mode control produces a much smoother pressure command than does the fuzzy MPC designed above. This can be a significant consideration, since pressure variations necessitate excessive control valve action, which can shorten the life of the valve.
5. Fuzzy model predictive control is a simple linear control law. It is much easier to implement than sliding mode, which is nonlinear, time-varying, and has boundary layers across which the control law changes.
6. Tracking performance of both controllers can be made as good as desired by adjusting the user-defined parameters appropriately. The price paid for this greater accuracy is usually greater control effort.
7. In practice, the TS fuzzy model could be obtained from experiments on the actual arm, rather than by linerizing a mathematical model.

11 Conclusions

This research has consisted of applying nonlinear control techniques to the difficult problem of controlling systems actuated by pneumatic muscle actuators (PMs). PMs are lightweight, cheap, powerful actuators that are particularly suitable for robot actuation and for use in exoskeletons because of their similarity to human muscle. Their advantage lies in the fact that in robotic and exoskeletal systems, the actuators must be moved along with the system itself, as is the case for a robot whose actuators are mounted on the links, or an exoskeleton whose actuators are mounted on the wearer's limbs. If the actuators are heavy (as in the case of electric motors), this can significantly subtract from the system's payload. PMs have the highest power/weight and power/volume ratios of any actuator.

PM control is complicated by the fact that they are highly nonlinear and time-varying. Their nonlinearity arises because of their construction, and their time-varying nature is a result of heat generation due to friction, which is a natural consequence of the PM's repeated inflation and deflation. The control philosophies chosen for this research are adaptive control, sliding mode control, model predictive control, and neuro-fuzzy control. The methods were chosen because of their ability to handle these conditions.

Some of our algorithms were implemented on the Pneumatic Muscle Test Station in the Human Effectiveness Laboratory (HECP)) at Wright Patterson Air Force Base, which is under the direction of Dr. D. W. Repperger. This enabled us to test these algorithms on actual PMs, which gave us invaluable experience and insight into their properties. Other algorithms involved too many PMs to test in HECP. For these systems, we relied on simulations to investigate our controllers. We took great pains to insure that, to the greatest extent possible, the simulations were accurate and agreed with the behavior of actual PM-actuated systems.

Our results fall into two broad categories: (a) position control of a single PM hanging vertically actuating a mass, and (b) control of planar robotic manipulators via groups of PMs. The former could be carried out on an actual PM in the lab, while the latter was done in simulation. The research on systems containing more than one PM was concentrated on planar arm configurations because we wanted to specifically address anthropomorphic systems, as would be encountered in exoskeletons.

The control methods used for (a) above include the Fuzzy Model Reference Learning Control (FMRLC) strategy (Chapter 8), and the Fuzzy P+ID strategy (Chapter 9). In addition, a neuro-fuzzy model of the PM was obtained from data taken in the lab (Chapter 6). This model was necessary for the control method of Chapter 9, which utilizes an evolutionary algorithm in conjunction with the Zeigler-Nichols tuning method for PID compensators to arrive at a suitable controller. Both methods controlled the PM well, but the Fuzzy P+ID method (Chapter 9) was superior to the FMRLC (Chapter 8) in that more accurate tracking resulted while using less control effort. Control effort is always a concern in any real-world control task. However, an advantage of the FMRLC is that it was not necessary to take data prior to designing the controller and no *a priori* modeling was necessary, as was necessary for the Fuzzy P+ID method. Instead, the

FMRLC required extensive tuning, which was not necessary for the Fuzzy P+ID controller

The control methods used for (b) above, i.e. control of planar robotic manipulators via groups of PMs, include adaptive control, single- and multi-input sliding control, and multi-input Fuzzy Model Predictive Control (MPC). In our opinion, the sliding mode approach (Chapters 4 and 5) is particularly advantageous for PM control because it can handle any degree of nonlinearity and it is specifically designed assuming modeling errors. These errors are always present when dealing with PMs, since PMs are very difficult to model accurately. In addition, bounds on state tracking errors are available with the sliding mode approach. For planar arm robotic systems or exoskeleton systems, these state errors can be translated into guaranteed spatial tracking accuracies.

The adaptive approach (Chapter 3), which is derived from the sliding mode approach, has the advantage of being capable of adapting to slowly-varying dynamics. The dynamics of systems containing PMs do vary with time, due to heat generated by friction. Therefore, this capability of adaptive control is valuable. However, the adaptive approach requires a somewhat more accurate model of the system dynamics. This is a disadvantage of the adaptive approach, since the dynamics of systems containing PMs are always poorly known.

Model Predictive Control (MPC) has been used for years in the control of industrial processes. These processes are highly nonlinear, MIMO, and contain time delays, therefore are very difficult to control. MPC has traditionally suffered from high computational load, and has therefore been relegated to processes with slow dynamics. This situation has changed in recent years with the advent of faster, more compact computers with more memory, hence MPC is considered in this research for PM control.

As with adaptive and sliding mode control, MPC requires some kind of model of the process. An advantage of MPC is that this model can be of any kind. In this research, we have modeled the planar arm actuated by four groups of PMs with an 81-rule Takagi-Sugeno fuzzy system. This model was derived from a 4-state nonlinear DE model of the arm. If desired, the fuzzy model could be determined experimentally, obviating the need for a DE model. This is a further advantage of fuzzy MPC – no DE model of the process is necessary, and a fuzzy model of the nonlinear system can be determined experimentally. Also, MPC has well-known robustness properties, which are useful for systems involving PMs. The only control method considered in this research that required no model of the process was FMRLC (Chapter 8).

An additional outcome of this research is the Evolutionary Variable Input Spread Inference Training (EVISIT) algorithm. EVISIT is a novel method for constructing a fuzzy system describing a process on the basis of data taken from the process. While not absolutely necessary for PM control, EVISIT can be used for obtaining a fuzzy model of the PM, and this fuzzy model can then be used for control. EVISIT is also quite useful for pattern classification problems (Chapter 7).

In conclusion, we have investigated several methods for controlling systems containing PMs. The methods include adaptive, sliding mode, model predictive, and neuro-fuzzy control. Each method has its advantages and disadvantages. The methods are immediately useful for and applicable to autonomous robotic systems actuated by PMs because these systems are relatively straightforward control problems with no human interaction. Exoskeletal systems will require more investigation, since they must

interact in real-time with the wearer. Thus the exoskeleton problem is more challenging and interesting. It appears, however, that the control methods investigated in this research may be useful for control of exoskeletons actuated by PMs as well. Further research should concentrate on exoskeleton actuation taking into account wearer interaction.

References

[1] D. G. Caldwell, G. A. Medrano-Cerda, and M. Goodwin, "Control of pneumatic muscle actuators," *IEEE Control Systems Magazine*, February 1995.

[2] C-P. Chou and B. Hannaford, "Static and dynamic characteristics of McKibben pneumatic artificial muscles," *Proc. 1994 IEEE Robotics and Automation Conference*, 1994.

[3] H. F. Schulte, Jr., "The characteristics of the McKibben artificial muscle," in *The Application of External Power in Prosthetics and Orthotics*, Washington, D. C.: National Academy of Sciences, National Research Council, 1961.

[4] K. Inoue, "Rubber actuators and applications for robotics," *Robotics Research: The 4th International Symposium*, R. Bolles and B. Roth, Eds., Cambridge, MA: MIT Press, 1988.

[5] B. Hannaford and J. M. Winters, "Actuator properties and movement control: biological and technological muscles," in *Multiple Muscle Systems*, J. Winters and S. Woo, Eds., New York: Springer-Verlag, 1990.

[6] D. W. Repperger, C. A. Phillips, D. C. Johnson, R. D. Harmon, and K. Johnson, "A study of pneumatic muscle technology for possible assistance in mobility," *Proceedings of the 19th Annual International Conference on the IEEE Engineering in Medicine and Biology Society*, Chicago, IL, November 1997.

[7] D. W. Repperger, K. R. Johnson, and C. A. Phillips, "Nonlinear feedback controller design of a pneumatic muscle actuator system," *Proc. 1999 American Control Conference*, San Diego, CA, June 1999.

[8] T. Noritsugu and T. Tanaka, "Application of rubber artificial muscle manipulator as a rehabilitation robot," *IEEE/ASME Transactions on Mechatronics*, Vol. 2, No. 4, December 1997.

[9] D. G. Caldwell, G. A. Medrano-Cerda, and M. Goodwin, "Braided pneumatic muscle actuator control of a multi-jointed manipulator," *IEEE SMC Conf., Vol. 1*, Le Touquet, France, 1993.

[10] D. G. Caldwell, A. Razak, and M. J. Goodwin, "Braided pneumatic muscle actuators," *Proc. IFAC Conf. on Intelligent Autonomous Vehicles*, Southampton, UK, April, 1993.

[11] S. D. Prior, P. R. Warner, A. S. White, J. T. Parsons, and R. Gill, "Actuators for rehabilitation robots," *Mechatronics - Special Issue on Robot Actuators*, Vol. 3, No. 3, June 1993.

[12] J. H. Cocatre-Zilgien, F. Delcomyn, and J. M. Hart, "Performance of a muscle-like 'leaky' pneumatic actuator powered by modulated air pulses," *J. Robot. Syst.*, Vol. 13, No. 6, 1996.

[13] G. J. Klute, J. M. Czerniecki, and B. Hannaford, "McKibben artificial muscles: pneumatic actuators with biomechanical intelligence," *Proc. IEEE/ASME 1999 International Conference on Advanced Intelligent Mechatronics (AIM '99)*, Atlanta, GA, Sept. 1999.

[14] G. J. Klute and B. Hannaford, "Fatigue characteristics of McKibben artificial muscle actuators," *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Victoria, B. C. Canada, October 1998.

[15] R. T. Pack, J. L. Christopher Jr., and K. Kawamura, "A rubbertuator-based structure-climbing inspection robot," *Proc. IEEE 1997 International Conference on Robotics and Automation*, Albuquerque, NM, April 1997.

[16] M. Bishay, M. E. Cambron, K. Negishi, R. A. Peters II, and K. Kawamura, "Visual servoing in ISAC, a decentralized robot system for feeding the disabled," *Proc. 1995 IEEE International Symposium on Computer Vision*, November 1995.

[17] D. B. Reynolds, D. W. Repperger, C. A. Philips, and G. Bandry, "Dynamic characteristics of pneumatic muscle," *Annals of Biomedical Engineering*, March 2003.

[18] S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence, and Robustness*, Englewood Cliffs, NJ: Prentice-Hall, 1989.

[19] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*, Englewood Cliffs, NJ: Prentice-Hall, 1989.

[20] J-J. Slotine and W. Li, *Applied Nonlinear Control*, Englewood Cliffs, NJ: Prentice-Hall, 1991.

[21] J-J. Slotine and J. A. Coetsee, "Adaptive sliding controller synthesis for nonlinear systems," *Int. J. Control*, Vol. 43, No. 6, 1986.

[22] C. M. Close and D. K. Frederick, *Modeling and Analysis of Dynamic Systems*, Boston, MA: Houghton Mifflin, 1978.

[23] B. J. Ruthenberg, N. A. Wasylewski, and J. E. Beard, "An experimental device for investigating the force and power requirements of a powered gait orthosis," *Journal of Rehabilitation Research & Development*, Vol. 34. No. 2, pp. 203-13, 1997.

[24] G. R. Johnson, and M. A. Buckley, "Development of a new motorized upper limb orthotic system (MULOS)," *Proceedings of the Rehabilitation Engineering Society of North America*, Pittsburgh, PA, June 1997, pp. 399-401.

[25] T. R. Lunsford, "Advanced contracture reduction orthosis (CRO)," *Proceedings of the 19th Annual Symposium of the American Academy of Orthotists and Prosthetists*, Las Vegas, NV, 1993.

[26] G. Colombo, M. Joerg, R. Schreier, and V. Dietz, "Treadmill training of paraplegic patients using a robotic orthosis," *Journal of Rehabilitation Research and Development*, Vol. 37, No. 6, pp. 693-700, 2000.

[27] J.-J. E. Slotine, "The robust control of robot manipulators," *Int. J. Robotics Research*, Vol. 4, No. 2, 1985.

[28] D. G. Luenberger, *Introduction to Dynamic Systems*, J. Wiley and Sons, New York, 1979.

[29] T. L. Palmieri, K. Petuskey, A. Bagley, S. Takashiba, D. G. Greenhalgh, and G. T. Rab, "Alterations in functional movement after axillary burn scar contracture: a motion analysis study," *Journal of Burn Care & Rehabilitation*, Vol. 24, No. 2, pp. 104-8, 2003.

[30] N. Yang, M. Zhang, C. Huang, and D. Jin, "Synergic analysis of upper limb target-reaching movements," *Journal of Biomechanics*, Vol. 35, No. 6, pp. 739-46, 2002.

[31] V. Gorrini and H. Bersini, "Recurrent fuzzy system," *Proc. IEEE International Conference on Fuzzy Systems*, 1994.

[32] C. F. Juang and C. T. Lin, "A recurrent self-organizing neural fuzzy inference network," *IEEE Transactions on Neural Networks*, Vol. 10, No. 4, pp. 828-845, 1999.

[33] J. Zhang, A. J. Morris, "Recurrent neuro-fuzzy networks for nonlinear process modeling," *IEEE Transactions on Neural Networks*, Vol. 10, No. 2, pp. 313-326, 1999.

[34] X. Chang and Wei Li, "Dynamic behavior modeling of stoker-fired boilers by recurrent neuro-fuzzy networks," *Neural Networks & Advanced Control Strategies*, M. Mohammadian, Ed., pp.190-195, IOS Press, Netherlands, 1999.

[35] A. Blanco, M. Delgado, and M. C. Pegalajar, "Identification of fuzzy systems using max-min recurrent neural networks," *Fuzzy Sets and Systems*, Vol. 122, No. 3, pp. 451-467, 2001.

[36] C-H. Lee and C-C. Teng, "Identification and control of dynamic systems using recurrent fuzzy neural networks," *IEEE Transactions on Fuzzy Systems*, Vol. 8, No. 4, pp. 349-366, 2000.

[37] J. Branson, "A trainable fuzzy system incorporating dynamically variable asymmetric spreads and negative rule defuzzification," Ph.D. Dissertation, University of Louisville, Louisville, KY, May 2000.

[38] L. Jin, P. N. Nikiforuk, and M. Gupta, "Approximation of discrete-time state space trajectories using dynamic recurrent neural networks," *IEEE Transactions on Automatic Control*, Vol. 40, No. 10, pp. 1266-1270, 1995.

[39] S. Adwankar and R. N. Banavar, "A recurrent network for dynamic system identification," *International Journal of Systems Science*, 28:12, 1239-1250, 1997.

[40] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, New York, Academic Press, 1972.

[41] B. D. Ripley, *Pattern Recognition and Neural Networks*, New York, Cambridge University Press, 1996.

[42] J. C. Bezdek and S. K. Pal, *Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data*, New York, IEEE Press, 1992.

[43] A. Baraldi and P. Blonda, "Survey of fuzzy clustering algorithms for pattern recognition - Parts I and II," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol.29, No. 6, pp.778-785, 1999.

[44] J. Valente, "Semantic constraints for membership function optimization," *IEEE Transactions on Fuzzy Systems*, Vol. 7, No. 1, pp. 128–138, 1999.

[45] J. J. Buckley, Y. Hayashi, and E. Czogala, " On the equivalence of neural networks and fuzzy systems," *Proc. IJCNN-92*, Vol. 2, pp. 691-695, Baltimore, MD, 1992.

[46] R. Jiang and C. T. Sun, "Functional equivalence between radial basis function networks and fuzzy inference systems," *IEEE Transactions on Neural Networks*, Vol. 4, No. 1, pp. 156-159, 1993.

[47] A. Gaweda and J. M. Zurada, "Equivalence between neural networks and fuzzy systems," *Proc. International Joint Conference on Neural Networks*, Washington, D.C., July 16-19, 2001.

[48] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 22, No. 6, Nov. 1992.

[49] E. G. Laukonen and K. M. Passino, "Training fuzzy systems to perform estimation and identification," *Engineering Applications of Artificial Intelligence*, Vol. 8, No. 5, 1995.

[50] R. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugen.*, Vol. 7, No. 2, pp. 179-188, 1936.

[51] M. Setnes and H. Roubos, "GA-fuzzy modeling and classification: complexity and performance," *IEEE Transactions On Fuzzy Systems*, Vol. 8, No. 5, pp. 509-522, Oct. 2000.

[52] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Selecting fuzzy if–then rules for classification problems using genetic algorithms," *IEEE Trans. Fuzzy Systems*, Vol. 3, pp. 260– 270, 1995.

[53] J. S. Wang and G. C. S. Lee, "Self-adaptive neuro-fuzzy inference system for classification application," *IEEE Transaction on Fuzzy Systems*, Vol. 10, No. 6, pp. 790-802, Dec. 2002.

[54] J. A. Roubos, M. Setnes, and J. Abonyi, "Learning fuzzy classification rules from labeled data," *Information Sciences*, in press.

[55] Y. Shi, R. Eberhart, and Y. Chen, "Implementation of evolutionary fuzzy system," *IEEE Transactions on Fuzzy Systems*, Vol. 7, No. 2, pp. 109-119, April 1999.

[56] S. Abe and R. Thawonmas, "A fuzzy classifier with ellipsoidal regions," *IEEE Transactions on Fuzzy Systems*, Vol. 5, No. 3, pp. 358-368, August 1997.

[57] M. Russo, "Genetic fuzzy learning," *IEEE Transactions On Evolutionary Computation*, Vol. 4, No. 3, pp. 259-273, Sept. 2000.

[58] T. P. Wu and S. M. Chen, "A new method for constructing membership functions and fuzzy rules from training examples," *IEEE Transaction on System, Man and Cybernetics - part B*, Vol. 29, No. 1, pp. 25-40, Feb. 1999.

[59] D. Nauck and R. Kruse, "A neuro-fuzzy method to learn fuzzy classification rules from data," *Fuzzy Sets and Systems*, Vol. 89, No. 3, pp. 277-288, 1997.

[60] E. V. D. Rhee, H. V. N. Lemke, and J. Dijkman, "Knowledge based fuzzy control of systems," *IEEE Trans. Automatic Control*, Vol. 35, February 1990.

[61] C. Lee, "Fuzzy logic in control systems: fuzzy logic controller – part I," *IEEE Trans. Syst. Man. Cybern.*, Vol. 20, March/April 1990.

[62] C. Lee, "Fuzzy logic in control systems: fuzzy logic controller - part II," *IEEE Trans. Syst. Man. Cybern.*, Vol. 20, March/April 1990.

[63] K. M. Passino and S. Yurkovich, *Fuzzy Control*, Addison-Wesley, 1998.

[64] J. R. Layne and K. M. Passino, "Fuzzy model reference learning control," *Journal of Intelligent and Fuzzy Systems*, Vol. 4, No. 1, 1996.

[65] J. R. Layne and K. M. Passino, "Fuzzy model reference learning control for cargo ship steering," *IEEE Control Systems Magazine*, Vol. 13, No. 6, December 1993.

[66] J. R. Layne, K. M. Passino and S. Yurkovich, "Fuzzy learning control for antiskid braking systems," *IEEE Trans. on Control Systems Technology*, Vol. 1, No. 2. June 1993.

[67] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
[68] H. P. Schewefel, *Evolution and Optimum Seeking*, John Wiley & Sons, New York, 1995.
[69] L. J. Fogel, *Intelligence Through Simulated Evolution - Forty Years of Evolutionary Programming*, John Wiley & Sons, New York, 1999.
[70] S. J. Kang, C. H. Woo, H. S. Hwang, and K. B. Woo, "Evolutionary design of fuzzy rule base for nonlinear system modeling and control," *IEEE Transactions on Fuzzy Systems*, Vol. 8, No. 1, pp. 37-45, 2000.
[71] O. Cordon and F. Herrera, "A two stage evolutionary process for designing TSK fuzzy rule-based systems," *IEEE Transactions on Systems, Man and Cybernetics: Part B*, Vol. 29, pp. 703-715, Dec.1999.
[72] D. Spiegel and T. Sudkmp, "Employment locality in the evolutionary generation of fuzzy rule base," *IEEE Transactions on Systems, Man and Cybernetics- Part B*, Vol. 32, No. 3, pp. 296-305, 2002.
[73] X. Chang, "Fuzzy P+ID controller and behavior modeling for complex systems," Ph.D. Dissertation, Sept. 2000.
[74] I. Skrjanc and D. Matko, " Predictive functional control based on fuzzy model for heat-exchanger pilot plant," *IEEE Transactions on Fuzzy Systems*, Vol. 8, No. 6, June 2000.
[75] M. Mahfouf, S. Kandiah and D. A. Linkens, "Fuzzy model-based predictive control using an ARX structure with feedforward," *Fuzzy Sets and Systems*, Vol. 125, No. 1, Jan. 2002.
[76] L. Huaguang and C. Zhang, "Multivariable fuzzy generalized predictive control," *Cybernetics and Systems*, Vol. 33, No. 1, January/February, 2002.
[77] I. Skrjanc and D. Matko, "Fuzzy predictive functional control in the state space domain," *Journal of Intelligent and Robotic Systems: Theory and Applications*, Vol. 31, No. 1-3, 2001.
[78] P. D. Vukovic, "One-step ahead predictive fuzzy controller," *Fuzzy Sets and Systems*, Vol. 122, No. 1, 2001.
[79] L. F. Baptista and J. M. Sousa, "Fuzzy predictive algorithms applied to real-time force control," *Control Engineering Practice*, Vol. 9, No. 4, April 2001.
[80] J. Q. Hu and E. Rose, "Generalized predictive control using a neuro-fuzzy model," *International Journal of Systems Science*, Vol. 30, No. 1, January 1998.
[81] J. A. Roubos, S. Mollov, R. Babuska, and H. B. Verbruggen, "Fuzzy model-based predictive control using Takagi-Sugeno models," *International Journal of Approximate Reasoning*, Vol. 22, No. 1-2, Sept. 1999.
[82] H. Ishibuchi and T. Yamamoto, "Effects of Three-Objective Genetic Rule Selection on the Generalization Ability of Fuzzy Rule-Based Systems," Proc. of Second International Conference on Evolutionary Multi-Criterion Optimization (Faro, Portugal), pp. 608-622, April 8-11, 2003
[83] Deb K., Pratap A., Agarwal S. and Meyarivan T. "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol.6, no. 2, pp.182-197, 2002.

[84] Ishibuchi, H., Nakashima, T., and Murata, T., "Three-objective genetic-based machine learning for linguistic rule extraction," *Information Science*, vol. 136, pp. 109-133, 2001.

[85] Aboyi, J., Roubos, J. A., and Szeifert, F., "Data-driven generation of compact, accurate, and linguistically sound fuzzy classifiers based on a decision-tree initialization," *International Journal of Approximate Reasoning*, vol.32, no.1, pp. 1–21, 2003.

Appendix A
Matrices for TS Fuzzy Model of Planar Arm

The TS fuzzy model of the planar arm is an 81-rule fuzzy system given in Section 10.2. The matrices $(I + TA_i)$, B_i, and D_i, $i = 1, \ldots 81$ are given by

$$(I + TA_i) = \begin{bmatrix} 1.0 & 0 & 0.01 & 0 \\ 0 & 1.0 & 0 & 0.01 \\ \epsilon_{31} & \epsilon_{32} & \epsilon_{33} & \epsilon_{34} \\ \epsilon_{41} & \epsilon_{42} & \epsilon_{43} & \epsilon_{44} \end{bmatrix}_i \tag{A.1}$$

$$B_i = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \eta_{31} & \eta_{32} \\ \eta_{41} & \eta_{42} \end{bmatrix}_i \tag{A.2}$$

$$D_i = \begin{bmatrix} 0 \\ 0 \\ \gamma_3 \\ \gamma_4 \end{bmatrix}_i \tag{A.3}$$

Define

$$\alpha_i \equiv \begin{bmatrix} \epsilon_{31} & \epsilon_{32} & \epsilon_{33} & \epsilon_{34} \\ \epsilon_{41} & \epsilon_{42} & \epsilon_{43} & \epsilon_{44} \end{bmatrix}_i \tag{A.4}$$

$$\beta_i = \begin{bmatrix} \eta_{31} & \eta_{32} \\ \eta_{41} & \eta_{42} \end{bmatrix}_i \tag{A.5}$$

$$\delta_i = \begin{bmatrix} \gamma_3 \\ \gamma_4 \end{bmatrix}_i \tag{A.6}$$

Then the matrices $(I + TA_i)$, B_i, D_i are given by (A.1 - A.6) with

$$\alpha_1 = \begin{bmatrix} -1.45e-1 & +1.18e+3 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -2.76e+3 & +4.51e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_2 = \begin{bmatrix} -1.45e-1 & +3.95e+2 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -1.58e+3 & +4.51e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_3 = \begin{bmatrix} -1.45e-1 & +1.32e+2 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -1.18e+3 & +4.51e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_4 = \begin{bmatrix} -1.45e-1 & +2.63e+2 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -3.95e+2 & +5.52e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_5 = \begin{bmatrix} -1.45e-1 & +2.93e-2 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -2.59e-1 & +5.52e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_6 = \begin{bmatrix} -1.45e-1 & +2.63e+2 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -3.95e+2 & +5.52e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_7 = \begin{bmatrix} -1.45e-1 & +1.32e+2 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -1.18e+3 & +5.52e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_8 = \begin{bmatrix} -1.45e-1 & +3.95e+2 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -1.58e+3 & +5.52e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_9 = \begin{bmatrix} -1.45e-1 & +1.18e+3 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -2.76e+3 & +5.52e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{10} = \begin{bmatrix} -1.73e-1 & -1.75e+1 & -1.51e+0 & -3.35e+0 \\ +8.67e-2 & +9.30e+2 & +7.54e+0 & +2.68e+0 \end{bmatrix}$$

$$\alpha_{11} = \begin{bmatrix} -1.73e-1 & -2.28e+2 & +1.62e-1 & -1.68e+0 \\ +8.67e-2 & +2.46e+2 & +6.70e+0 & +1.84e+0 \end{bmatrix}$$

$$\alpha_{12} = \begin{bmatrix} -1.73e-1 & -2.98e+2 & +1.84e+0 & +3.42e-5 \\ +8.67e-2 & +1.78e+1 & +5.86e+0 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{13} = \begin{bmatrix} -1.73e-1 & +7.02e+1 & -6.76e-1 & -1.68e+0 \\ +8.67e-2 & +2.28e+2 & +8.38e-1 & +1.84e+0 \end{bmatrix}$$

$$\alpha_{14} = \begin{bmatrix} -1.73e-1 & -2.05e-2 & +1.00e+0 & +3.42e-5 \\ +8.67e-2 & +1.43e-1 & +1.84e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{15} = \begin{bmatrix} -1.73e-1 & +7.01e+1 & +2.68e+0 & +1.68e+0 \\ +8.67e-2 & +2.28e+2 & -8.38e-1 & +1.62e-1 \end{bmatrix}$$

$$\alpha_{16} = \begin{bmatrix} -1.73e-1 & -2.98e+2 & +1.62e-1 & +3.42e-5 \\ +8.67e-2 & +1.76e+1 & -5.86e+0 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{17} = \begin{bmatrix} -1.73e-1 & -2.28e+2 & +1.84e+0 & +1.68e+0 \\ +8.67e-2 & +2.46e+2 & -6.70e+0 & +1.62e-1 \end{bmatrix}$$

$$\alpha_{18} = \begin{bmatrix} -1.73e-1 & -1.76e+1 & +3.51e+0 & +3.35e+0 \\ +8.67e-2 & +9.30e+2 & -7.54e+0 & -6.76e-1 \end{bmatrix}$$

$$\alpha_{19} = \begin{bmatrix} -1.89e-1 & -1.41e+3 & +1.00e+0 & -7.33e-5 \\ +1.19e-1 & +2.82e+2 & -3.22e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{20} = \begin{bmatrix} -1.89e-1 & +2.82e+2 & +1.00e+0 & -7.33e-5 \\ +1.19e-1 & +1.13e+3 & -3.22e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{21} = \begin{bmatrix} -1.89e-1 & +8.46e+2 & +1.00e+0 & -7.33e-5 \\ +1.19e-1 & +1.41e+3 & -3.22e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{22} = \begin{bmatrix} -1.89e-1 & -5.64e+2 & +1.00e+0 & -7.33e-5 \\ +1.19e-1 & -2.82e+2 & -3.94e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{23} = \begin{bmatrix} -1.89e-1 & +1.82e-1 & +1.00e+0 & -7.33e-5 \\ +1.19e-1 & +3.02e-1 & -3.94e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{24} = \begin{bmatrix} -1.89e-1 & -5.64e+2 & +1.00e+0 & -7.33e-5 \\ +1.19e-1 & -2.82e+2 & -3.94e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{25} = \begin{bmatrix} -1.89e-1 & +8.46e+2 & +1.00e+0 & -7.33e-5 \\ +1.19e-1 & +1.41e+3 & -3.94e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{26} = \begin{bmatrix} -1.89e-1 & +2.82e+2 & +1.00e+0 & -7.33e-5 \\ +1.19e-1 & +1.13e+3 & -3.94e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{27} = \begin{bmatrix} -1.89e-1 & -1.41e+3 & +1.00e+0 & -7.33e-5 \\ +1.19e-1 & +2.82e+2 & -3.94e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{28} = \begin{bmatrix} -2.94e-3 & +1.18e+3 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -2.76e+3 & +4.51e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{29} = \begin{bmatrix} -2.94e-3 & +3.95e+2 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -1.58e+3 & +4.51e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{30} = \begin{bmatrix} -2.94e-3 & +1.32e+2 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -1.18e+3 & +4.51e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{31} = \begin{bmatrix} -2.94e-3 & +2.63e+2 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -3.95e+2 & +5.52e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{32} = \begin{bmatrix} -2.94e-3 & +8.92e-4 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -3.57e-3 & +5.52e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{33} = \begin{bmatrix} -2.94e-3 & +2.63e+2 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -3.95e+2 & +5.52e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{34} = \begin{bmatrix} -2.94e-3 & +1.32e+2 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -1.18e+3 & +5.52e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{35} = \begin{bmatrix} -2.94e-3 & +3.95e+3 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -1.58e+3 & +5.52e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{36} = \begin{bmatrix} -2.94e-3 & +1.18e+3 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -2.76e+3 & +5.52e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{37} = \begin{bmatrix} +2.55e-2 & -1.75e+1 & -1.51e+0 & -3.35e+0 \\ +2.00e-1 & +9.30e+2 & +7.54e+0 & +2.68e+0 \end{bmatrix}$$

$$\alpha_{38} = \begin{bmatrix} +2.55e-2 & -2.28e+2 & +1.62e-1 & -1.68e+0 \\ +2.00e-1 & +2.46e+2 & +6.70e+0 & +1.84e+0 \end{bmatrix}$$

$$\alpha_{39} = \begin{bmatrix} +2.55e-2 & -2.98e+2 & +1.84e+0 & +3.42e-5 \\ +2.00e-1 & +1.78e+1 & +5.86e+0 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{40} = \begin{bmatrix} +2.55e-2 & +7.02e+1 & -6.76e-1 & -1.68e+0 \\ +2.00e-1 & +2.28e+2 & +8.38e-1 & +1.84e+0 \end{bmatrix}$$

$$\alpha_{41} = \begin{bmatrix} +2.55e-2 & +2.56e-2 & +1.00e+0 & +3.42e-5 \\ +2.00e-1 & +1.86e-1 & +1.84e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{42} = \begin{bmatrix} +2.55e-2 & +7.02e+1 & +2.68e+0 & +1.68e+0 \\ +2.00e-1 & +2.28e+2 & -8.38e-1 & +1.62e-1 \end{bmatrix}$$

$$\alpha_{43} = \begin{bmatrix} +2.55e-2 & -2.98e+2 & +1.62e-1 & +3.42e-5 \\ +2.00e-1 & +1.76e+1 & -5.86e+0 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{44} = \begin{bmatrix} +2.55e-2 & -2.28e+2 & +1.84e+0 & +1.68e+0 \\ +2.00e-1 & +2.46e+2 & -6.70e+0 & +1.62e-1 \end{bmatrix}$$

$$\alpha_{45} = \begin{bmatrix} +2.55e-2 & -1.76e+1 & +3.51e+0 & +3.35e+0 \\ +2.00e-1 & +9.30e+2 & -7.54e+0 & -6.76e-1 \end{bmatrix}$$

$$\alpha_{46} = \begin{bmatrix} -6.31e-3 & -1.41e+3 & +1.00e+0 & -7.33e-5 \\ -3.15e-3 & +2.82e+2 & -3.22e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{47} = \begin{bmatrix} -6.31e-3 & +2.82e+2 & +1.00e+0 & -7.33e-5 \\ -3.15e-3 & +1.13e+3 & -3.22e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{48} = \begin{bmatrix} -6.31e-3 & +8.46e+2 & +1.00e+0 & -7.33e-5 \\ -3.15e-3 & +1.41e+3 & -3.22e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{49} = \begin{bmatrix} -6.31e-3 & -5.64e+2 & +1.00e+0 & -7.33e-5 \\ -3.15e-3 & -2.82e+2 & -3.94e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{50} = \begin{bmatrix} -6.31e-3 & -6.37e-4 & +1.00e+0 & -7.33e-5 \\ -3.15e-3 & -2.55e-3 & -3.94e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{51} = \begin{bmatrix} -6.31e-3 & -5.64e+2 & +1.00e+0 & -7.33e-5 \\ -3.15e-3 & -2.82e+2 & -3.94e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{52} = \begin{bmatrix} -6.31e-3 & +8.46e+2 & +1.00e+0 & -7.33e-5 \\ -3.15e-3 & +1.41e+3 & -3.94e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{53} = \begin{bmatrix} -6.31e-3 & +2.82e+2 & +1.00e+0 & -7.33e-5 \\ -3.15e-3 & +1.13e+3 & -3.94e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{54} = \begin{bmatrix} -6.31e-3 & -1.41e+3 & +1.00e+0 & -7.33e-5 \\ -3.15e-3 & +2.82e+2 & -3.94e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{55} = \begin{bmatrix} +1.39e-1 & +1.18e+3 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -2.76e+3 & +4.51e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{56} = \begin{bmatrix} +1.39e-1 & +3.95e+2 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -1.58e+3 & +4.51e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{57} = \begin{bmatrix} +1.39e-1 & +1.32e+2 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -1.18e+3 & +4.51e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{58} = \begin{bmatrix} +1.39e-1 & +2.63e+2 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -3.95e+2 & +5.52e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{59} = \begin{bmatrix} +1.39e-1 & -2.75e-2 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & +2.52e-1 & +5.52e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{60} = \begin{bmatrix} +1.39e-1 & +2.63e+2 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -3.95e+2 & +5.52e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{61} = \begin{bmatrix} +1.39e-1 & +1.32e+2 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -1.18e+3 & +5.52e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{62} = \begin{bmatrix} +1.39e-1 & +3.95e+2 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -1.58e+3 & +5.52e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{63} = \begin{bmatrix} +1.39e-1 & +1.18e+3 & +1.00e+0 & +1.03e-4 \\ +4.41e-3 & -2.76e+3 & +5.52e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{64} = \begin{bmatrix} +1.67e-1 & -1.74e+1 & -1.51e+0 & -3.35e+0 \\ -8.37e-2 & +9.30e+2 & +7.54e+0 & +2.68e+0 \end{bmatrix}$$

$$\alpha_{65} = \begin{bmatrix} +1.67e-1 & -2.28e+2 & -1.62e-1 & -1.68e+0 \\ -8.37e-2 & +2.46e+2 & +6.70e+0 & +1.84e+0 \end{bmatrix}$$

$$\alpha_{66} = \begin{bmatrix} +1.67e-1 & -2.98e+2 & +1.84e+0 & +3.42e-5 \\ -8.37e-2 & +1.78e+1 & +5.86e+0 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{67} = \begin{bmatrix} +1.67e-1 & +7.03e+1 & -6.76e-1 & -1.68e+0 \\ -8.37e-2 & +2.28e+2 & +8.38e-1 & +1.84e+0 \end{bmatrix}$$

$$\alpha_{68} = \begin{bmatrix} +1.67e-1 & +1.06e-1 & +1.00e+0 & +3.42e-5 \\ -8.37e-2 & +1.27e-1 & +1.84e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{69} = \begin{bmatrix} +1.67e-1 & +7.03e+1 & +2.68e+0 & +1.68e+0 \\ -8.37e-2 & +2.28e+2 & -8.38e-1 & +1.62e-1 \end{bmatrix}$$

$$\alpha_{70} = \begin{bmatrix} +1.67e-1 & -2.98e+2 & +1.62e-1 & +3.42e-5 \\ -8.37e-2 & +1.76e+1 & -5.86e+0 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{71} = \begin{bmatrix} +1.67e-1 & -2.28e+2 & +1.84e+0 & +1.68e+0 \\ -8.37e-2 & +2.46e+2 & -6.70e+0 & +1.62e-1 \end{bmatrix}$$

$$\alpha_{72} = \begin{bmatrix} +1.67e-1 & -1.75e+1 & +3.51e+0 & +3.35e+0 \\ -8.37e-2 & +9.30e+2 & -7.54e+0 & -6.76e-1 \end{bmatrix}$$

$$\alpha_{73} = \begin{bmatrix} +1.76e-1 & -1.41e+3 & +1.00e+0 & -7.33e-5 \\ -1.25e-1 & +2.82e+2 & -3.22e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{74} = \begin{bmatrix} +1.76e-1 & +2.82e+2 & +1.00e+0 & -7.33e-5 \\ -1.25e-1 & +1.13e+3 & -3.22e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{75} = \begin{bmatrix} +1.76e-1 & +8.46e+2 & +1.00e+0 & -7.33e-5 \\ -1.25e-1 & +1.41e+3 & -3.22e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{76} = \begin{bmatrix} +1.76e-1 & -5.64e+2 & +1.00e+0 & -7.33e-5 \\ -1.25e-1 & -2.82e+2 & -3.94e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{77} = \begin{bmatrix} +1.76e-1 & -1.83e-1 & +1.00e+0 & -7.33e-5 \\ -1.25e-1 & -3.07e-1 & -3.94e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{78} = \begin{bmatrix} +1.76e-1 & -5.64e+2 & +1.00e+0 & -7.33e-5 \\ -1.25e-1 & -2.82e+2 & -3.94e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{79} = \begin{bmatrix} +1.76e-1 & +8.46e+2 & +1.00e+0 & -7.33e-5 \\ -1.25e-1 & +1.41e+3 & -3.94e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{80} = \begin{bmatrix} +1.76e-1 & +2.82e+2 & +1.00e+0 & -7.33e-5 \\ -1.25e-1 & +1.13e+3 & -3.94e-4 & +1.00e+0 \end{bmatrix}$$

$$\alpha_{81} = \begin{bmatrix} +1.76e-1 & -1.41e+3 & +1.00e+0 & -7.33e-5 \\ -1.25e-1 & +2.82e+2 & -3.94e-4 & +1.00e+0 \end{bmatrix}$$

$$\beta_1 = \begin{bmatrix} -3.03e-3 & -1.66e-3 \\ +4.54e-3 & +6.62e-3 \end{bmatrix}$$

$$\beta_2 = \begin{bmatrix} -3.03e-3 & -1.60e-3 \\ +4.54e-3 & +6.40e-3 \end{bmatrix}$$

$$\beta_3 = \begin{bmatrix} -3.03e-3 & -1.54e-3 \\ +4.54e-3 & +6.17e-3 \end{bmatrix}$$

$$\beta_4 = \begin{bmatrix} -3.20e-3 & -1.66e-3 \\ +4.79e-3 & +6.62e-3 \end{bmatrix}$$

$$\beta_5 = \begin{bmatrix} -3.20e-3 & -1.60e-3 \\ +4.79e-3 & +6.40e-3 \end{bmatrix}$$

$$\beta_6 = \begin{bmatrix} -3.20e-3 & -1.54e-3 \\ +4.79e-3 & +6.17e-3 \end{bmatrix}$$

$$\beta_7 = \begin{bmatrix} -3.36e-3 & -1.66e-3 \\ +5.05e-3 & +6.62e-3 \end{bmatrix}$$

$$\beta_8 = \begin{bmatrix} -3.36e-3 & -1.60e-3 \\ +5.05e-3 & +6.40e-3 \end{bmatrix}$$

$$\beta_9 = \begin{bmatrix} -3.36e-3 & -1.54e-3 \\ +5.05e-3 & +6.17e-3 \end{bmatrix}$$

$$\beta_{10} = \begin{bmatrix} -3.03e-3 & -5.53e-4 \\ +1.51e-3 & +4.42e-3 \end{bmatrix}$$

$$\beta_{11} = \begin{bmatrix} -3.03e-3 & -5.34e-4 \\ +1.51e-3 & +4.27e-3 \end{bmatrix}$$

$$\beta_{12} = \begin{bmatrix} -3.03e-3 & -5.15e-4 \\ +1.51e-3 & +4.12e-3 \end{bmatrix}$$

$$\beta_{13} = \begin{bmatrix} -3.20e-3 & -5.53e-4 \\ +1.60e-3 & +4.42e-3 \end{bmatrix}$$

$$\beta_{14} = \begin{bmatrix} -3.20e-3 & -5.34e-4 \\ +1.60e-3 & +4.27e-3 \end{bmatrix}$$

$$\beta_{15} = \begin{bmatrix} -3.20e-3 & -5.15e-4 \\ +1.60e-3 & +4.12e-3 \end{bmatrix}$$

$$\beta_{16} = \begin{bmatrix} -3.36e-3 & -5.53e-4 \\ +1.68e-3 & +4.42e-3 \end{bmatrix}$$

$$\beta_{17} = \begin{bmatrix} -3.36e-3 & -5.34e-4 \\ +1.68e-3 & +4.27e-3 \end{bmatrix}$$

$$\beta_{18} = \begin{bmatrix} -3.36e-3 & -5.15e-4 \\ +1.68e-3 & +4.12e-3 \end{bmatrix}$$

$$\beta_{19} = \begin{bmatrix} -6.49e-3 & +1.19e-3 \\ -3.24e-3 & +4.75e-3 \end{bmatrix}$$

$$\beta_{20} = \begin{bmatrix} -6.49e-3 & +1.15e-3 \\ -3.24e-3 & +4.58e-3 \end{bmatrix}$$

$$\beta_{21} = \begin{bmatrix} -6.49e-3 & +1.11e-3 \\ -3.24e-3 & +4.42e-3 \end{bmatrix}$$

$$\beta_{22} = \begin{bmatrix} -6.85e-3 & +1.19e-3 \\ -3.42e-3 & +4.75e-3 \end{bmatrix}$$

$$\beta_{23} = \begin{bmatrix} -6.85e-3 & +1.15e-3 \\ -3.42e-3 & +4.58e-3 \end{bmatrix}$$

$$\beta_{24} = \begin{bmatrix} -6.85e-3 & +1.11e-3 \\ -3.42e-3 & +4.42e-3 \end{bmatrix}$$

$$\beta_{25} = \begin{bmatrix} -7.21e-3 & +1.19e-3 \\ -3.60e-3 & +4.75e-3 \end{bmatrix}$$

$$\beta_{26} = \begin{bmatrix} -7.21e-3 & +1.15e-3 \\ -3.60e-3 & +4.58e-3 \end{bmatrix}$$

$$\beta_{27} = \begin{bmatrix} -7.21e-3 & +1.11e-3 \\ -3.60e-3 & +4.42e-3 \end{bmatrix}$$

$$\beta_{28} = \begin{bmatrix} -3.03e-3 & -1.66e-3 \\ +4.54e-3 & +6.62e-3 \end{bmatrix}$$

$$\beta_{29} = \begin{bmatrix} -3.03e-3 & -1.60e-3 \\ +4.54e-3 & +6.40e-3 \end{bmatrix}$$

$$\beta_{30} = \begin{bmatrix} -3.03e-3 & -1.54e-3 \\ +4.54e-3 & +6.17e-3 \end{bmatrix}$$

$$\beta_{31} = \begin{bmatrix} -3.20e-3 & -1.66e-3 \\ +4.79e-3 & +6.62e-3 \end{bmatrix}$$

$$\beta_{32} = \begin{bmatrix} -3.20e-3 & -1.60e-3 \\ +4.79e-3 & +6.40e-3 \end{bmatrix}$$

$$\beta_{33} = \begin{bmatrix} -3.20e-3 & -1.54e-3 \\ +4.79e-3 & +6.17e-3 \end{bmatrix}$$

$$\beta_{34} = \begin{bmatrix} -3.36e-3 & -1.66e-3 \\ +5.05e-3 & +6.62e-3 \end{bmatrix}$$

$$\beta_{35} = \begin{bmatrix} -3.36e-3 & -1.60e-3 \\ +5.05e-3 & +6.40e-3 \end{bmatrix}$$

$$\beta_{36} = \begin{bmatrix} -3.36e-3 & -1.54e-3 \\ +5.05e-3 & +6.17e-3 \end{bmatrix}$$

$$\beta_{37} = \begin{bmatrix} -3.03e-3 & -5.53e-4 \\ +1.51e-3 & +4.42e-3 \end{bmatrix}$$

$$\beta_{38} = \begin{bmatrix} -3.03e-3 & -5.34e-4 \\ +1.51e-3 & +4.27e-3 \end{bmatrix}$$

$$\beta_{39} = \begin{bmatrix} -3.03e-3 & -5.15e-4 \\ +1.51e-3 & +4.12e-3 \end{bmatrix}$$

$$\beta_{40} = \begin{bmatrix} -3.20e-3 & -5.53e-4 \\ +1.60e-3 & +4.42e-3 \end{bmatrix}$$

$$\beta_{41} = \begin{bmatrix} -3.20e-3 & -5.34e-4 \\ +1.60e-3 & +4.27e-3 \end{bmatrix}$$

$$\beta_{42} = \begin{bmatrix} -3.20e-3 & -5.15e-4 \\ +1.60e-3 & +4.12e-3 \end{bmatrix}$$

$$\beta_{43} = \begin{bmatrix} -3.36e-3 & -5.53e-4 \\ +1.68e-3 & +4.42e-3 \end{bmatrix}$$

$$\beta_{44} = \begin{bmatrix} -3.36e-3 & -5.34e-4 \\ +1.68e-3 & +4.27e-3 \end{bmatrix}$$

$$\beta_{45} = \begin{bmatrix} -3.36e-3 & -5.15e-4 \\ +1.68e-3 & +4.12e-3 \end{bmatrix}$$

$$\beta_{46} = \begin{bmatrix} -6.49e-3 & +1.19e-3 \\ -3.24e-3 & +4.75e-3 \end{bmatrix}$$

$$\beta_{47} = \begin{bmatrix} -6.49e-3 & +1.15e-3 \\ -3.24e-3 & +4.58e-3 \end{bmatrix}$$

$$\beta_{48} = \begin{bmatrix} -6.49e-3 & +1.11e-3 \\ -3.24e-3 & +4.42e-3 \end{bmatrix}$$

$$\beta_{49} = \begin{bmatrix} -6.85e-3 & +1.19e-3 \\ -3.42e-3 & +4.75e-3 \end{bmatrix}$$

$$\beta_{50} = \begin{bmatrix} -6.85e-3 & +1.15e-3 \\ -3.42e-3 & +4.58e-3 \end{bmatrix}$$

$$\beta_{51} = \begin{bmatrix} -6.85e-3 & +1.11e-3 \\ -3.42e-3 & +4.42e-3 \end{bmatrix}$$

$$\beta_{52} = \begin{bmatrix} -7.21e-3 & +1.19e-3 \\ -3.60e-3 & +4.75e-3 \end{bmatrix}$$

$$\beta_{53} = \begin{bmatrix} -7.21e-3 & +1.15e-3 \\ -3.60e-3 & +4.58e-3 \end{bmatrix}$$

$$\beta_{54} = \begin{bmatrix} -7.21e-3 & +1.11e-3 \\ -3.60e-3 & +4.42e-3 \end{bmatrix}$$

$$\beta_{55} = \begin{bmatrix} -3.03e-3 & -1.66e-3 \\ +4.54e-3 & +6.62e-3 \end{bmatrix}$$

$$\beta_{56} = \begin{bmatrix} -3.03e-3 & -1.60e-3 \\ +4.54e-3 & +6.40e-3 \end{bmatrix}$$

$$\beta_{57} = \begin{bmatrix} -3.03e-3 & -1.54e-3 \\ +4.54e-3 & +6.17e-3 \end{bmatrix}$$

$$\beta_{58} = \begin{bmatrix} -3.20e-3 & -1.66e-3 \\ +4.79e-3 & +6.62e-3 \end{bmatrix}$$

$$\beta_{59} = \begin{bmatrix} -3.20e-3 & -1.60e-3 \\ +4.79e-3 & +6.40e-3 \end{bmatrix}$$

$$\beta_{60} = \begin{bmatrix} -3.20e-3 & -1.54e-3 \\ +4.79e-3 & +6.17e-3 \end{bmatrix}$$

$$\beta_{61} = \begin{bmatrix} -3.36e-3 & -1.66e-3 \\ +5.05e-3 & +6.62e-3 \end{bmatrix}$$

$$\beta_{62} = \begin{bmatrix} -3.36e-3 & -1.60e-3 \\ +5.05e-3 & +6.40e-3 \end{bmatrix}$$

$$\beta_{63} = \begin{bmatrix} -3.36e-3 & -1.54e-3 \\ +5.05e-3 & +6.17e-3 \end{bmatrix}$$

$$\beta_{64} = \begin{bmatrix} -3.03e-3 & -5.53e-4 \\ +1.51e-3 & +4.42e-3 \end{bmatrix}$$

$$\beta_{65} = \begin{bmatrix} -3.03e-3 & -5.34e-4 \\ +1.51e-3 & +4.27e-3 \end{bmatrix}$$

$$\beta_{66} = \begin{bmatrix} -3.03e-3 & -5.15e-4 \\ +1.51e-3 & +4.12e-3 \end{bmatrix}$$

$$\beta_{67} = \begin{bmatrix} -3.20e-3 & -5.53e-4 \\ +1.60e-3 & +4.42e-3 \end{bmatrix}$$

$$\beta_{68} = \begin{bmatrix} -3.20e-3 & -5.34e-4 \\ +1.60e-3 & +4.27e-3 \end{bmatrix}$$

$$\beta_{69} = \begin{bmatrix} -3.20e-3 & -5.15e-4 \\ +1.60e-3 & +4.12e-3 \end{bmatrix}$$

$$\beta_{70} = \begin{bmatrix} -3.36e-3 & -5.53e-4 \\ +1.68e-3 & +4.42e-3 \end{bmatrix}$$

$$\beta_{71} = \begin{bmatrix} -3.36e-3 & -5.34e-4 \\ +1.68e-3 & +4.27e-3 \end{bmatrix}$$

$$\beta_{72} = \begin{bmatrix} -3.36e-3 & -5.15e-4 \\ +1.68e-3 & +4.12e-3 \end{bmatrix}$$

$$\beta_{73} = \begin{bmatrix} -6.49e-3 & +1.19e-3 \\ -3.24e-3 & +4.75e-3 \end{bmatrix}$$

$$\beta_{74} = \begin{bmatrix} -6.49e-3 & +1.15e-3 \\ -3.24e-3 & +4.58e-3 \end{bmatrix}$$

$$\beta_{75} = \begin{bmatrix} -6.49e-3 & +1.11e-3 \\ -3.24e-3 & +4.42e-3 \end{bmatrix}$$

$$\beta_{76} = \begin{bmatrix} -6.85e-3 & +1.19e-3 \\ -3.42e-3 & +4.75e-3 \end{bmatrix}$$

$$\beta_{77} = \begin{bmatrix} -6.85e-3 & +1.15e-3 \\ -3.42e-3 & +4.58e-3 \end{bmatrix}$$

$$\beta_{78} = \begin{bmatrix} -6.85e-3 & +1.11e-3 \\ -3.42e-3 & +4.42e-3 \end{bmatrix}$$

$$\beta_{79} = \begin{bmatrix} -7.21e-3 & +1.19e-3 \\ -3.60e-3 & +4.75e-3 \end{bmatrix}$$

$$\beta_{80} = \begin{bmatrix} -7.21e-3 & +1.15e-3 \\ -3.60e-3 & +4.58e-3 \end{bmatrix}$$

$$\beta_{81} = \begin{bmatrix} -7.21e-3 & +1.11e-3 \\ -3.60e-3 & +4.42e-3 \end{bmatrix}$$

$$\delta_1 = \begin{bmatrix} -6.63e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_2 = \begin{bmatrix} -6.63e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_3 = \begin{bmatrix} -6.63e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_4 = \begin{bmatrix} -6.63e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_5 = \begin{bmatrix} -6.63e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_6 = \begin{bmatrix} -6.63e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_7 = \begin{bmatrix} -6.63e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_8 = \begin{bmatrix} -6.63e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_9 = \begin{bmatrix} -6.63e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_{10} = \begin{bmatrix} -8.94e+2 \\ -1.37e+1 \end{bmatrix}$$

$$\delta_{11} = \begin{bmatrix} +2.27e+2 \\ +6.66e+2 \end{bmatrix}$$

$$\delta_{12} = \begin{bmatrix} +6.00e+2 \\ +8.93e+2 \end{bmatrix}$$

$$\delta_{13} = \begin{bmatrix} -3.74e+2 \\ -2.27e+2 \end{bmatrix}$$

$$\delta_{14} = \begin{bmatrix} -1.06e-1 \\ -3.68e-1 \end{bmatrix}$$

$$\delta_{15} = \begin{bmatrix} -3.74e+2 \\ -2.27e+2 \end{bmatrix}$$

$$\delta_{16} = \begin{bmatrix} +6.00e+2 \\ +8.93e+2 \end{bmatrix}$$

$$\delta_{17} = \begin{bmatrix} +2.67e+2 \\ +6.67e+2 \end{bmatrix}$$

$$\delta_{18} = \begin{bmatrix} -8.94e+2 \\ -1.34e+1 \end{bmatrix}$$

$$\delta_{19} = \begin{bmatrix} +4.43e+3 \\ -8.86e+2 \end{bmatrix}$$

$$\delta_{20} = \begin{bmatrix} -8.86e+2 \\ -3.54e+3 \end{bmatrix}$$

$$\delta_{21} = \begin{bmatrix} -2.66e+3 \\ -4.43e+3 \end{bmatrix}$$

$$\delta_{22} = \begin{bmatrix} +1.77e+3 \\ +8.85e+2 \end{bmatrix}$$

$$\delta_{23} = \begin{bmatrix} -5.20e-1 \\ -5.91e-1 \end{bmatrix}$$

$$\delta_{24} = \begin{bmatrix} +1.77e+3 \\ +8.85e+2 \end{bmatrix}$$

$$\delta_{25} = \begin{bmatrix} -2.66e+3 \\ -4.43e+3 \end{bmatrix}$$

$$\delta_{26} = \begin{bmatrix} -8.86e+2 \\ -3.54e+3 \end{bmatrix}$$

$$\delta_{27} = \begin{bmatrix} +4.43e+3 \\ -8.86e+2 \end{bmatrix}$$

$$\delta_{28} = \begin{bmatrix} +1.48e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_{29} = \begin{bmatrix} +1.48e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_{30} = \begin{bmatrix} +1.48e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_{31} = \begin{bmatrix} +1.48e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_{32} = \begin{bmatrix} +1.48e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_{33} = \begin{bmatrix} +1.48e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_{34} = \begin{bmatrix} +1.48e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_{35} = \begin{bmatrix} +1.48e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_{36} = \begin{bmatrix} +1.48e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_{37} = \begin{bmatrix} -8.94e+2 \\ -1.36e+1 \end{bmatrix}$$

$$\delta_{38} = \begin{bmatrix} +2.27e+2 \\ +6.66e+2 \end{bmatrix}$$

$$\delta_{39} = \begin{bmatrix} +6.00e+2 \\ +8.93e+2 \end{bmatrix}$$

$$\delta_{40} = \begin{bmatrix} -3.74e+2 \\ -2.27e+2 \end{bmatrix}$$

$$\delta_{41} = \begin{bmatrix} -5.23e-2 \\ -2.86e-1 \end{bmatrix}$$

$$\delta_{42} = \begin{bmatrix} -3.73e+2 \\ -2.27e+2 \end{bmatrix}$$

$$\delta_{43} = \begin{bmatrix} +6.00e+2 \\ +8.94e+2 \end{bmatrix}$$

$$\delta_{44} = \begin{bmatrix} +2.27e+2 \\ +6.67e+2 \end{bmatrix}$$

$$\delta_{45} = \begin{bmatrix} -8.94e+2 \\ -1.34e+1 \end{bmatrix}$$

$$\delta_{46} = \begin{bmatrix} +4.43e+3 \\ -8.86e+2 \end{bmatrix}$$

$$\delta_{47} = \begin{bmatrix} -8.86e+2 \\ -3.54e+3 \end{bmatrix}$$

$$\delta_{48} = \begin{bmatrix} -2.66e+3 \\ -4.43e+3 \end{bmatrix}$$

$$\delta_{49} = \begin{bmatrix} +1.77e+3 \\ +8.86e+2 \end{bmatrix}$$

$$\delta_{50} = \begin{bmatrix} +1.58e-1 \\ +2.96e-1 \end{bmatrix}$$

$$\delta_{51} = \begin{bmatrix} +1.77e+3 \\ +8.86e+2 \end{bmatrix}$$

$$\delta_{52} = \begin{bmatrix} -2.66e+3 \\ -4.43e+3 \end{bmatrix}$$

$$\delta_{53} = \begin{bmatrix} -8.86e+2 \\ -3.54e+3 \end{bmatrix}$$

$$\delta_{54} = \begin{bmatrix} +4.43e+3 \\ -8.86e+2 \end{bmatrix}$$

$$\delta_{55} = \begin{bmatrix} -6.63e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_{56} = \begin{bmatrix} -6.63e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_{57} = \begin{bmatrix} -6.63e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_{58} = \begin{bmatrix} -6.63e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_{59} = \begin{bmatrix} -6.63e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_{60} = \begin{bmatrix} -6.63e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_{61} = \begin{bmatrix} -6.63e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_{62} = \begin{bmatrix} -6.63e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_{63} = \begin{bmatrix} -6.63e-2 \\ -2.31e-1 \end{bmatrix}$$

$$\delta_{64} = \begin{bmatrix} -8.94e+2 \\ -1.32e+1 \end{bmatrix}$$

$$\delta_{65} = \begin{bmatrix} +2.26e+2 \\ +6.67e+2 \end{bmatrix}$$

$$\delta_{66} = \begin{bmatrix} +6.00e+2 \\ +8.93e+2 \end{bmatrix}$$

$$\delta_{67} = \begin{bmatrix} -3.74e+2 \\ -2.27e+2 \end{bmatrix}$$

$$\delta_{68} = \begin{bmatrix} -2.47e-1 \\ +5.45e-2 \end{bmatrix}$$

$$\delta_{69} = \begin{bmatrix} -3.74e+2 \\ -2.27e+2 \end{bmatrix}$$

$$\delta_{70} = \begin{bmatrix} +6.00e+2 \\ +8.94e+2 \end{bmatrix}$$

$$\delta_{71} = \begin{bmatrix} +2.27e+2 \\ +6.67e+2 \end{bmatrix}$$

$$\delta_{72} = \begin{bmatrix} -8.94e+2 \\ -1.30e+1 \end{bmatrix}$$

$$\delta_{73} = \begin{bmatrix} +4.43e+3 \\ -8.85e+2 \end{bmatrix}$$

$$\delta_{74} = \begin{bmatrix} -8.85e+2 \\ -3.54e+3 \end{bmatrix}$$

$$\delta_{75} = \begin{bmatrix} -2.66e+3 \\ -4.43e+3 \end{bmatrix}$$

$$\delta_{76} = \begin{bmatrix} +1.77e+3 \\ +8.87e+2 \end{bmatrix}$$

$$\delta_{77} = \begin{bmatrix} +6.27e-1 \\ +1.32e+0 \end{bmatrix}$$

$$\delta_{78} = \begin{bmatrix} +1.77e+3 \\ +8.87e+2 \end{bmatrix}$$

$$\delta_{79} = \begin{bmatrix} -2.66e+3 \\ -4.43e+3 \end{bmatrix}$$

$$\delta_{80} = \begin{bmatrix} -8.85e+2 \\ -3.54e+3 \end{bmatrix}$$

$$\delta_{81} = \begin{bmatrix} +4.43e+3 \\ -8.85e+2 \end{bmatrix}$$

Appendix B - Publications

X. Chang and J. H. Lilly, "Fuzzy control for pneumatic muscle tracking via evolutionary tuning," *Intelligent Automation and Soft Computing*, Vol. 9, No. 4, pp. 227-244, 2003 (attached).

J. H. Lilly, "Adaptive tracking for pneumatic muscle actuators in bicep and tricep configurations," *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, Vol. 11, No. 3, September, 2003 (attached).

S. W. Chan, J. H. Lilly, D. W. Repperger, and J. E. Berlin, "Fuzzy PD+I learning control for a pneumatic muscle," *IEEE International Conference on Fuzzy Systems*, St. Louis, MO, March, 2003 (attached).

L. Yang and J. H. Lilly, "Sliding mode tracking for pneumatic muscle actuators in bicep/tricep pair configuration," *American Control Conference*, Denver, CO, June 2003 (attached).

J. H. Lilly and P. M. Quesada, "A two-input sliding mode controller for a planar arm actuated by four pneumatic muscle groups," *IEEE Trans. on Neural Systems and Rehabilitation Engineering* (in review).

J. H.. Lilly and L. Yang, "Sliding mode tracking for pneumatic muscle actuators in opposing pair configuration," *IEEE Trans. on Control System Technology*, (in review).

X. Chang and J. H. Lilly, "The evolutionary design of a fuzzy classifier from data," *IEEE Trans. on Systems, Man, and Cybernetics* (in review).

J. H. Lilly and X. Chang, "Fuzzy model predictive control for a planar arm actuated by four pneumatic muscle groups," *Journal of Intelligent and Fuzzy Systems* (in review).

Fuzzy PD+I Learning Control for a Pneumatic Muscle

S. W. Chan
John H. Lilly
Department of Electrical and Computer Engineering
University of Louisville

Daniel W. Repperger
James E. Berlin
Air Force Research Laboratory
Wright-Patterson AFB, Ohio

Abstract - **A fuzzy learning control technique is used for position tracking involving the vertical movement of a mass attached to a pneumatic muscle. Because the pneumatic muscle is nonlinear and time varying, conventional fixed controllers are less effective than the fuzzy controller proposed in this paper. The controller is of a PID type, with an adaptive fuzzy PD part and a nonfuzzy integral branch. A novelty of the controller is that the fuzzy inverse model, which dynamically adjusts the PD part of the controller, incorporates the internal PM pressure as an input. Experimental results are presented from the pneumatic muscle test facility in the Human Effectiveness Laboratory at Wright-Patterson Air Force Base.**

I. Introduction

A pneumatic muscle (PM) actuator [1-3] is a type of artificial muscle that has been used in robotic applications. The PM construction consists of an ellipsoidal rubber tube inside a very strong braided plastic sheath (Fig. 1). When the tube is inflated due to an increase in pressure, it widens and the muscle shortens, providing force to move a mass in the direction of shortening.

Compared to either hydraulic or electric motor actuation systems, the PM actuator has two important advantages [1] which are extremely high power/weight ratio (1 W/g) and power/volume ratio (1 W/cm^3). Another advantage of the PM is that it is a "soft actuator" which, if it fails (i.e. bursts), does so in a relatively safe manner. Since the PM consists of a small gas volume enclosed in a soft material, if a failure occurs, the risk of human injury is minimal.

The main disadvantage of PMs is that they are highly nonlinear and time varying, since they are made of rubber, the characteristics of which vary with temperature. Therefore it is difficult to control them precisely in a position or force control sense.

This research was supported by AFOSR grant #F49620-00-1-0300, and by ONR grant #N00014-98-1-0568.

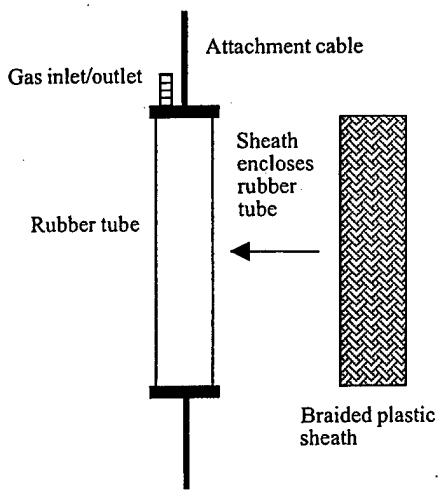


Figure 1 - Construction of pneumatic muscle actuator

Efforts to actuate pneumatic muscles have taken several approaches. In [4], a sliding mode controller is used for tracking. Inoue [5] attempts to control the PM using a PID-like control scheme. Lilly [6] presents proof of stability and asymptotic tracking for a nonlinear adaptive controller for a joint with PMs arranged in bicep and tricep configurations. Repperger et al. [7] design a variable structure controller for a PM to perform asymptotic position tracking.

Fuzzy control [8 – 11] has been successfully used in many commercial and industrial applications in recent years. The fuzzy model reference learning control (FMRLC) methodology was developed from the general ideas in linguistic self-organizing control and conventional model reference adaptive control [12]. FMRLC has a learning ability that improves the closed-loop performance over time

by using measured data from the plant operation to update the controller. The FMRLC technique has been used in the control of cargo ship steering [13] and has also been employed to improve the performance of an anti-skid braking system [14].

In this paper, we utilize ideas from FMRLC to design a learning controller that results in length tracking for a pneumatic muscle. The controller is essentially a PID type, with adjustable fuzzy proportional and derivative parts, and a nonfuzzy integral branch. The fuzzy PD consequent singleton locations are dynamically adjusted by a fuzzy inverse model. A novelty of this controller is that the inverse model has for one of its inputs the interior pressure of the PM. This helps closed-loop performance in this case because the PM inflation dynamics are different from its deflation dynamics.

This paper is arranged as follows. Section 2 gives a currently-derived mathematical model of the PM. Section 3 describes the details of the fuzzy learning controller used for position tracking. Section 4 presents the experimental results and discussion, and Section 5 contains conclusions.

II. Dynamic Model of Pneumatic Muscle

Extensive efforts to model the PM have been underway in the Human Effectiveness Laboratory at Wright Patterson AFB in Ohio [15]. In this study, the PM was modeled by a 3-element mechanical model shown in Fig. 2, consisting of a contractile element, dashpot, and spring. These three elements all have pressure-dependent coefficients.

The PM was hung vertically with a mass attached at the lower end. Fig. 3 portrays the PM being inflated due to an increase in pressure. When inflated, it shortens which exerts force to lift the mass. Let $y = 0$ be the position of the mass when the PM is completely deflated. It is shown in [15] that the dynamic motion of the PM can be modeled as

$$M\ddot{y} + B\dot{y} + Ky = F - Mg \tag{1}$$

where M is the mass lifted by the PM, B is the coefficient of viscous friction, K is the spring coefficient, F is the force exerted by the contractile element, and g is the acceleration of the gravity. From (1), when $Mg = F$ and $y = \dot{y} = 0$, the mass is idle, i.e. no motion of the mass occurs. For $F > Mg$, the right hand side of (1) becomes the forcing function for the system and the mass rises.

According to [15], B, K and F are dependent on the internal pressure of the PM and are given as:

$$F = 3.77P - 0.0138P^2 \tag{2a}$$

$$K = 32.6 + 1.21P \tag{2b}$$

$$B = \begin{cases} 5.75 + 0.272P \text{ (Inflation)} \\ 3.41 - 0.0316P \text{ (Deflation)} \end{cases} \tag{2c}$$

where P is the gauge pressure inside the PM in psi. The coefficients specified in (2) are valid in the range $0 < P < 90$ psi. Note that the coefficient B in deflation is smaller than in inflation. The reason is because in deflation

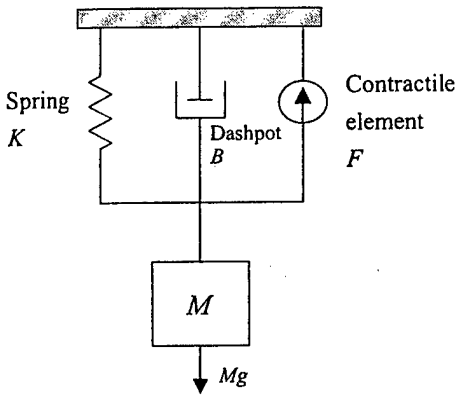


Figure 2 - Three element system used to model PM

the PM system vents against a constant atmospheric pressure. During inflation, however, the pressure buildup is in a closed volume and the forcing function has to fight against itself as it inflates. Thus, during inflation, the net B term is larger.

Note from (2) that the contractile force F, viscous damping coefficient B and spring coefficient K are functions of P, which is the control variable, i.e. P is the independent variable which can be commanded by the controller. Therefore, in order to express the PM model in terms of the control variable P, we rewrite (1) as

$$M\ddot{y} + B(P)\dot{y} + K(P)y = F(P) - Mg \tag{3}$$

which becomes

$$M\ddot{y} + B(P)\dot{y} + (32.6 + 1.21P)y = 3.77P - 0.0138P^2 - Mg \tag{4}$$

with B as in (2c).

We observe that (4) is not in a form that is solvable by standard nonlinear adaptive control techniques. For instance, many nonlinear adaptive control results rely on the use of parameter adaptive control for obtaining asymptotically exact cancellation in linearizing control laws [16], [17]. Well-known results exist for plants that are decouplable by static state feedback, i.e. plants of the form

$$\dot{x} = f(x) + g(x)u \tag{5a}$$

$$y = h(x) \tag{5b}$$

(2)

where $x \in \Re^n$, u and y are the scalar input and output respectively and f, g are infinitely differentiable functions. However, these results are not applicable to (4) due to the P^2 term and the fact that B is different for inflation than for deflation. Hence, it is difficult to control this PM by using conventional nonlinear adaptive control techniques.

III. Fuzzy Learning Control

Fuzzy control offers several advantages over conventional control techniques. An important advantage for problems

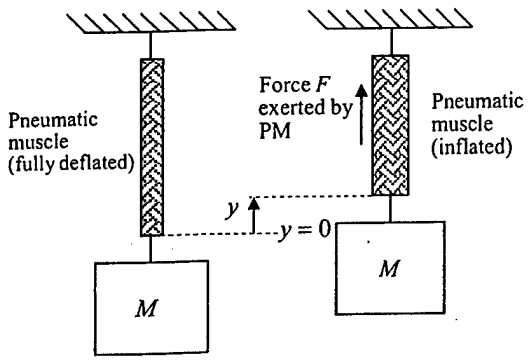


Figure 3 – Pneumatic muscle driving a mass.

such as tracking control of PMs is that the plant model is not required to be in any particular form, or even that the form is known. Fuzzy control is rather based on plant performance. An extension of fuzzy control is the fuzzy learning control methodology. In this method, the fuzzy controller is not fixed, but rather it is dynamically adjusted by a fuzzy inverse model. The inverse model incorporates expert knowledge about how the plant should behave and how the controller should be adjusted if the plant does not behave as desired. The inverse model dynamically adjusts the locations of the consequent centers of the fuzzy controller according to the measured behavior of the plant.

Our fuzzy learning controller is shown in Fig. 4. It has two main parts: the fuzzy controller and the learning mechanism, which includes the fuzzy inverse model and knowledge-base modifier. We explain each of these below.

A. Fuzzy PD+I Learning Controller

The controller is of PID type with a fuzzy PD part and a nonfuzzy integral branch. The fuzzy part has inputs $e(kT) = r(kT) - y(kT)$ and $c(kT) = (e(kT) - e((k-1)T))/T$ where $r(kT)$ is the reference signal to be tracked, $y(kT)$ is the PM length, and T is the sampling interval. The controller output is the commanded PM pressure $p(kT)$. The integral branch of the controller is nonfuzzy because we have no expert knowledge about how to adjust consequent centers for this branch and fuzziness is not needed for any other reason (e.g. making the integral path nonlinear).

For the fuzzy (i.e. PD) part of the controller, we define 11 fuzzy sets on the e universe of discourse ($A_e^i(e), i = 1,\ldots,11$) and 11 fuzzy sets on the c universe of discourse ($A_c^i(c), i = 1,\ldots,11$). These fuzzy sets are characterized by 11 equally-spaced symmetrical triangular membership functions centered at [-1, -.8, -.6, -.4, -.2, 0, .2, .4, .6, .8, 1] which form partitions of unity on their respective universes.

The controller rule base therefore consists of 121 rules, each rule with a separate consequent. Therefore we will have 121 singleton output memberships with the consequent membership for rule i at $b_i, i = 1,\ldots,121$. The input membership functions are fixed and are not tuned.

Using minimum T-norm for calculation of premise values and center-average defuzzification, the output of the PD part of the controller is calculated as

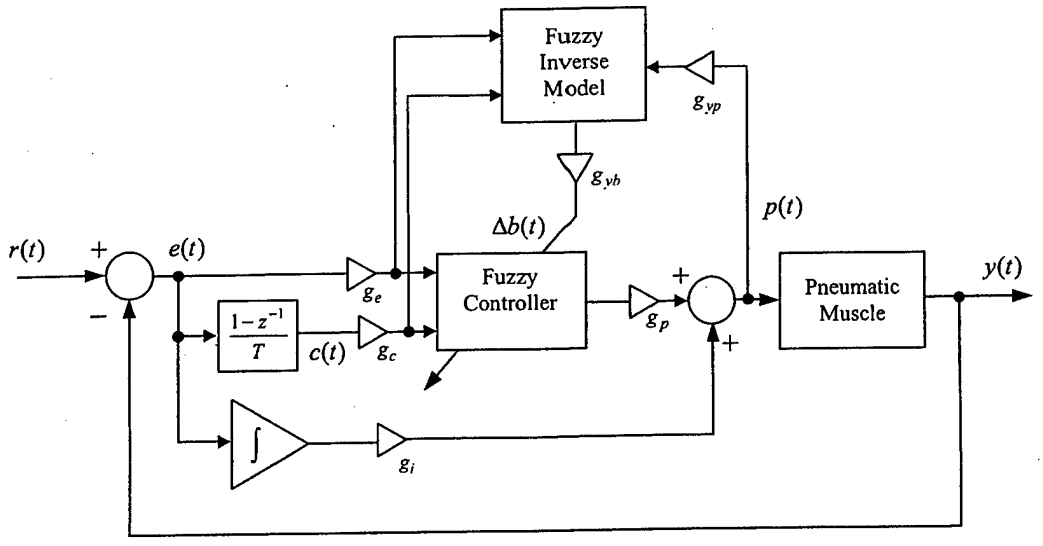


Figure 4 - Fuzzy learning PD + I tracking controller

$$p_{PD}(kT) = \frac{\sum_i b_i(kT)\mu_{premise(i)}(e(kT), c(kT))}{\sum_i \mu_{premise(i)}(e(kT), c(kT))} \quad (6)$$

where the summation is taken over all 121 rules and $b_i(kT)$ denotes the output membership function for the i^{th} rule at time kT. The centers $b_i(kT), i = 1, \ldots, 121$ are dynamically adjusted by the fuzzy inverse model and the learning mechanism.

The integral branch is a standard nonfuzzy integral function of the tracking error. A constant multiple of the integral of the error is added to the fuzzy PD output, making the controller essentially a PID controller in which the P and D parts are fuzzy and adjustable, while the I part is nonfuzzy and fixed.

B. Learning Mechanism

The learning mechanism consists of a fuzzy inverse model and knowledge-base modifier. The fuzzy inverse model is a fuzzy system with inputs $g_e e(kT)$, $g_c c(kT)$, and PM internal pressure $g_{yp} p(kT)$. The inverse model output $\Delta b(kT)$ determines the amount that the fuzzy controller output singletons are changed at each time step. We choose pressure to be the third inverse model input since pressure has a direct effect on PM characteristics (2). Simulations indicate that controller P and D gains should be larger at low PM pressures than at high PM pressures. This is due to the fact that it requires more pressure input to effect a given amount of shortening when the PM is deflated than when it is inflated

The e and c input universes for the inverse model each have 11 fuzzy sets characterized by 11 equally-spaced symmetrical triangular membership functions identical to those of the controller. The third input p to the inverse model has a universe of discourse consisting of 11 fuzzy sets characterized by 11 equally-spaced symmetrical triangular membership functions centered at [0, .1, .2, .3, .4, .5, .6, .7, .8, .9, 1] forming a partition of unity. Therefore, the inverse model has $11 \times 11 \times 11 = 1331$ rules.

The output fuzzy sets for the inverse model are 17 singletons spaced as [-1, -.75, -.5, -.4, -.3, -.2, -.1, -.05, 0, .05, .1, .2, .3, .4, .5, .75, 1] (see Fig. 5). The spacing is nonuniform to give more change in commanded input pressure when the PM internal pressure is low (i.e. inflated a small amount, which corresponds to longer lengths) and less at high pressures (more inflation, shorter lengths). The output $\Delta b(kT)$ of the inverse model is calculated similarly to the output of the fuzzy controller of the previous section, i.e. minimum T-norm and center average defuzzification.

Two representative parts of the inverse model rule base are shown in Figs. 6a and 6b. For example, when the error and derivative error are small and positive and the pressure is low, the output membership function linguistic-numeric value is 7, corresponding to a positive addition of +0.75 (see highlighted box in Fig. 6a). This means the controller output singletons should be increased by a relatively large positive increment to insure $y(kT)$ will not continue to decrease. On the other hand, for the same values in error and derivative error, but high internal pressure, the value is only 4, corresponding to an addition of only +0.3 (see highlighted box in Fig. 6b), which means we need less than half as much adjustment in controller output singletons when the PM internal pressure is high.

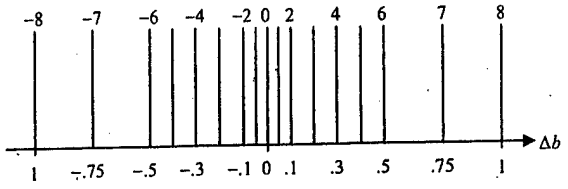


Figure 5 - Output singleton fuzzy sets for inverse model (linguistic-numeric values above memberships)

The knowledge-base modifier modifies the PD part of the fuzzy controller to improve tracking performance. The knowledge-base modifier changes the knowledge-base of the fuzzy controller by adjusting only the output singletons of the rules that are "on" at time kT for the PD part of the controller according to the update law:

$$b_i(kT) = b_i((k-1)T) + \Delta b(kT) \quad (7)$$

Note that at each time step only a few controller output singletons are updated, and the others are left unchanged. Specifically, no more than 4 singletons are changed for a single time step in our learning controller.

IV. Experimental Results and Discussion

The PM was loaded with a weight of 65 pounds. The fuzzy learning controller was implemented with a sampling time of 1/64 seconds. The controller gains used were $g_e = 1$, $g_i = 0.1$, $g_c = 0.01$ and $g_p = 100$. The inverse model gains were chosen to be $g_{yp} = .01$ and $g_{yb} = 0.02$.

The initial position of the mass is 0 in., corresponding to the PM fully deflated and extended. Figs. 7 and 8 show typical tracking performance.

The error in tracking performance is due to the fact that the supporting structure for the PM assembly is not absolutely rigid, giving rise to vibrations in the PM motion (see Figure 7 especially at deflation), the fact that PM length and pressure measurements are not exact, and the fact that the valves used for inflation and deflation are not ideal. It should be remembered that PMs are typically used in opposing pairs (e.g. bicep/tricep), which would probably remove most of the bouncing seen in Figure 7. Therefore, the setup used in these experiments, i.e. a single PM hanging vertically, is somewhat unrealistic.

It can be observed from Figs. 7 and 8 that fairly accurate tracking begins during the first 1 - 2 seconds. In general, *learning* (as opposed to tracking) occurs over a longer time,

and controller output singletons converge after 30 – 45 seconds, depending on the input. Output singleton convergence is not necessary for accurate tracking. This phenomenon is also present in nonfuzzy model reference adaptive controllers [18]. Tracking performance can be expected to improve over time as different situations (i.e. combinations of *e*, *c*, and *p*) are encountered and learning becomes more complete.

V. Conclusions

The problem addressed in this paper is position tracking control for a pneumatic muscle actuator. Since the coefficients in the PM are poorly known and vary with time, we use a fuzzy PID-type tracking controller with learning ability. The controller consists of an adjustable PD fuzzy part and a parallel nonfuzzy integral branch. Tracking is accurate after a few seconds of operation, although singleton convergence has not yet occurred.

VI. References

[1] D. G. Caldwell, G. A. Medrano-Cerda and M. Goodwin, "Control of pneumatic muscle actuators," *IEEE Control Systems Magazine*, February 1995.

[2] C-P Chou and B. Hannaford, "Measurement and modeling of McKibben pneumatic artificial muscles," *IEEE Transactions on Robotics and Automation*, vol.12, no.1, February 1996.

[3] T. Noritsugu and T. Tanaka, "Application of rubber artificial muscle manipulator as a rehabilitation robot," *IEEE/ASME Transactions on Mechatronics*, vol.2, no.4, December 1997.

[4] H. Sira-Ramirez, P. Lopez and B. Tondu, "Robust stabilization and tracking for robotic manipulators with artificial muscles," *International Journal of Systems Science*, vol. 27, no.11, 1996.

[5] K. Inoue, "Rubber actuators and applications in robotics," *in Robotic Research: The 4th International Symposium*, R. Bolles and B. Roth, Eds. Cambridge, MA, MIT Press, 1998.

[6] J. H. Lilly, "Adaptive tracking for pneumatic muscle actuators in bicep and tricep configuration," *IEEE Trans. on Neural Systems and Rehabilitation*. To appear.

[7] D. W. Repperger, K.R. Johnson, and C. A. Philips, "A VSC position tracking system involving a large scale pneumatic muscle actuator," *Proceedings of the 1998 Conference on Decision and Control*, December 1998, Tampa, Florida.

[8] E. V. D. Rhee, H. V. N. Lemke and J. Dijkman, "Knowledge based fuzzy control of systems," *IEEE Trans. Automatic Control*, vol. 35, February 1990.

[9] C. Lee, "Fuzzy logic in control systems: fuzzy logic controller – part I," *IEEE Trans. Syst. Man. Cybern.*, vol. 20, March/April 1990.

[10] C. Lee, "Fuzzy logic in control systems: fuzzy logic controller - part II," *IEEE Trans. Syst. Man. Cybern.*, vol. 20, March/April 1990.

[11] K. M. Passino and S. Yurkovich, *Fuzzy Control*, Addison-Wesley, 1998.

[12] J. R. Layne and K. M. Passino, "Fuzzy model reference learning control," *Journal of Intelligent and Fuzzy Systems*, vol. 4, no. 1, 1996.

[13] J. R. Layne and K. M. Passino, "Fuzzy model reference learning control for cargo ship steering," *IEEE Control Systems Magazine*, vol. 13, no. 6, December 1993.

[14] J. R. Layne, K. M. Passino and S. Yurkovich, "Fuzzy learning control for antiskid braking systems," *IEEE Trans. on Control Systems Technology*, vol. 1, no. 2, June 1993.

[15] D. B. Reynolds, D. W. Repperger, C. A. Philips and G. Bandry, "Dynamic characteristics of pneumatic muscle," *Computers in Biology and Medicine*. Submitted.

[16] S. Sastry and M. Bodson, *Adaptive Control*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[17] J-J. Slotine and W. Li, *Applied Nonlinear Control*, Englewood Cliffs, NJ: Prentice-Hall, 1991.

[18] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*, Englewood Cliffs, NJ: Prentice-Hall, 1989.

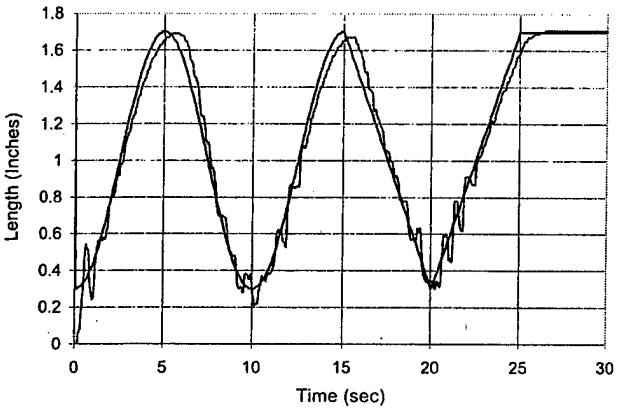


Figure 7 – Experimental tracking performance.

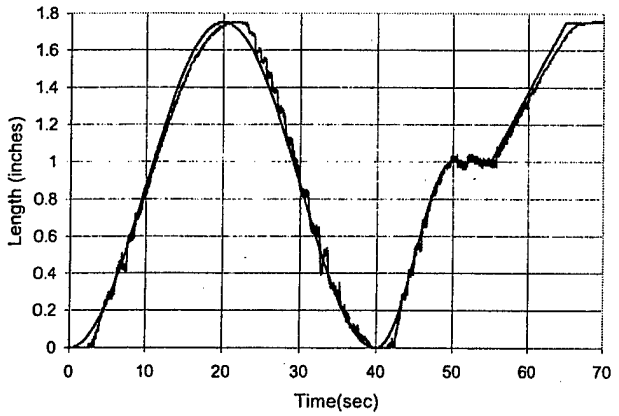


Figure 8 – Experimental tracking performance

Δb		c										
		-5	-4	-3	-2	-1	0	1	2	3	4	5
e	-5	-8	-8	-8	-8	-8	-8	-7	-6	-4	-2	0
	-4	-8	-8	-8	-8	-8	-7	-6	-4	-2	0	2
	-3	-8	-8	-8	-8	-7	-6	-4	-2	0	2	4
	-2	-8	-8	-8	-7	-6	-4	-2	0	2	4	6
	-1	-8	-8	-7	-6	-4	-2	0	2	4	6	7
	0	-8	-7	-6	-4	-2	0	2	4	6	7	8
	1	-7	-6	-4	-2	0	2	4	6	7	8	8
	2	-6	-4	-2	0	2	4	6	7	8	8	8
	3	-4	-2	0	2	4	6	7	8	8	8	8
	4	-2	0	2	4	6	7	8	8	8	8	8
	5	0	2	4	6	7	8	8	8	8	8	8

Figure 6a – One page of inverse model rule base for low PM internal pressure
(p linguistic-numeric values 0 – 4)

Δb		c										
		-5	-4	-3	-2	-1	0	1	2	3	4	5
e	-5	-5	-5	-5	-5	-5	-5	-4	-3	-2	-1	0
	-4	-5	-5	-5	-5	-5	-4	-3	-2	-1	0	1
	-3	-5	-5	-5	-5	-4	-3	-2	-1	0	1	2
	-2	-5	-5	-5	-4	-3	-2	-1	0	1	2	3
	-1	-5	-5	-4	-3	-2	-1	0	1	2	3	4
	0	-5	-4	-3	-2	-1	0	1	2	3	4	5
	1	-4	-3	-2	-1	0	1	2	3	4	5	5
	2	-3	-2	-1	0	1	2	3	4	5	5	5
	3	-2	-1	0	1	2	3	4	5	5	5	5
	4	-1	0	1	2	3	4	5	5	5	5	5
	5	0	1	2	3	4	5	5	5	5	5	5

Figure 6b – One page of inverse model rule base for high PM internal pressure
(p linguistic-numeric values 5 – 10)

Sliding Mode Tracking for Pneumatic Muscle Actuators in Bicep/Tricep Pair Configuration[1]

Liang Yang
John H. Lilly
Department of Electrical and Computer Engineering
University of Louisville
Louisville, KY, USA 40292

Abstract - **This paper presents robust motion control of pneumatic muscle actuators arranged in bicep/tricep pair configuration to deal with system and environmental uncertainties using the sliding mode approach. A mathematical model is derived for an arm with PMs in bicep/tricep pair configuration. A sliding mode controller is designed to yield asymptotic tracking of elbow angle. The results of simulations are presented to demonstrate the success of the proposed controller.**

I. Introduction

A pneumatic muscle actuator (PM) consists of a cylindrical, flexible rubber or plastic airtight tube inside a braided plastic sheath [1]. Its characteristics are similar to those of human skeletal muscles, i.e. PMs are similar in size and force capability, and force is only created through the action of PM contraction. When the tube is inflated, it widens and due to the braided sheath, the entire assembly shortens. The force exerted when the PM shortens is quite large in proportion to the PM's weight. Pneumatic muscles have the highest power/weight ratio (1W/g) and power/volume ratio (1W/cm^3) of any actuator. Both these ratios are about 5 times higher than electric motors or hydraulic actuators.

For these reasons, PMs have found application in robotics, and for strength assistive devices for humans. PMs have the attractive property that humans are at a very low level of risk of injury in event of failure (i.e. bursting). This is not the case for hydraulic or electric motor actuators, which produce a far greater risk of injury to the human user around the device in the event of actuator failure.

However, since PMs are made of flexible rubber or plastic, their characteristics vary with temperature and their temperature varies with use, which makes them more difficult to control. In PM control, there are typically significant discrepancies between the actual plant and the mathematical model developed for controller design. This difficulty is inherent in PM technology.

To ensure the required performance levels exist despite the existence of plant/model mismatches, we use variable structure control methodology for PM control [2]. Sliding mode control [3-9] is one of these methods. Sliding mode is a high-speed switching feedback control that switches between two values based upon some switching criteria. Sliding mode control drives the nonlinear plant's state trajectory onto a specified surface, which is called the sliding or switching surface.

Since sliding controller design provides a systematic approach to the problem of maintaining stability and consistent performance in the face of modeling imprecision, it is currently under investigation for the control of PMs. In this paper, we model an arm with PM actuators in bicep/tricep pair configuration and utilize sliding mode tracking control to make the elbow angle asymptotically track a desired reference function of time.

This paper is arranged as follows. Section 2 contains the derivation of a mathematical model of an arm actuated by PMs in bicep/tricep pair configuration. Section 3 addresses sliding mode control of limbs actuated by PMs and proposes a sliding mode controller for PMs in bicep/tricep pair configuration. Section 4 presents simulation results of the closed-loop sliding mode angle control of the arm with PMs in bicep/tricep pair configuration. Section 5 presents a discussion of the results, and Section 6 contains conclusions.

II. Modeling of Limbs with PM in Bicep/Tricep Pair Configuration

Figure 1 shows a PM hanging vertically actuating a mass. When inflated, the PM shortens which exerts force to lift the mass. Let $y = 0$ be the position of the mass when the PM is completely deflated. If the PM is modeled as a parallel combination of a spring, a dashpot, and a contractile element (as in [10]), the dynamic motion of the system of Figure 1 can be modeled as

$$M\ddot{y} + B\dot{y} + Ky = F - Mg \qquad (2.1)$$

where M is the mass lifted by the PM, B is the coefficient of viscous friction, K is the spring coefficient, F is the force exerted by the contractile element, and g is the acceleration of the gravity. Thus

[1] This research was supported by AFOSR grant #F49620-00-1-0300.

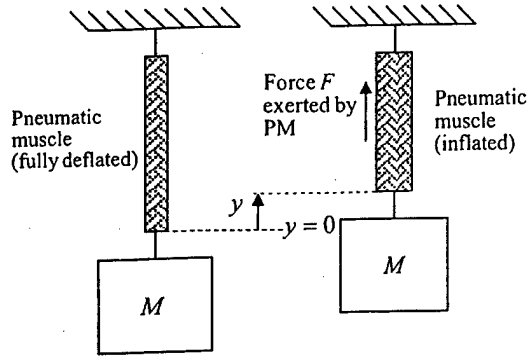


Figure 1 – PM hanging vertically actuating a mass

the total upward force exerted on the mass by the PM is $F - B\dot{y} - Ky$.

According to [10], B, K and F are dependent on the internal pressure of the PM and are given as:

$$F = 3.77P - 0.138P^2 \qquad (2.2a)$$

$$K = 32.58 + 1.209P \qquad (2.2b)$$

$$B = \begin{cases} 5.748 + 0.2719P \text{ (Inflation)} \\ 3.41 - 0.0316P \text{ (Deflation)} \end{cases} \qquad (2.2c)$$

where P is the internal gauge pressure of the PM.

Figure 2 shows an arm actuating a mass, with PMs in the position of a bicep/tricep pair. The upper arm remains stationary as the PMs expand and contract, moving the forearm. The upper ends of the bicep and tricep are attached to a motionless reference point. The mass is held at the end of the forearm (i.e. hand). The forearm, which is considered massless, is attached to the upper arm by a frictionless planar revolute joint. The PMs are attached to the forearm at point A, which is a distance a from the joint. The distance from the center of mass of the load to the joint is L. The forearm is free

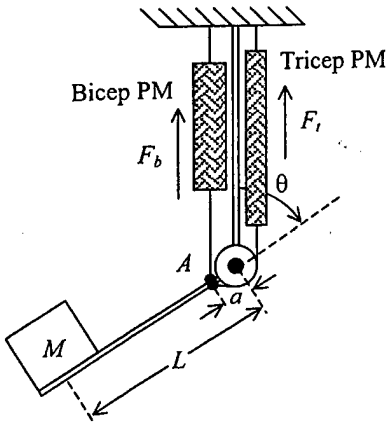


Figure 2 – Arm with PMs in bicep/tricep pair configuration

to rotate through an angle θ, where $\theta = 0^\circ$ corresponds to the tricep being fully shortened while bicep is fully lengthened, and $\theta = 180^\circ$ corresponds to the tricep being fully lengthened while the bicep is fully shortened. For simplicity, we will assume the PM force always acts parallel to the forearm. This is valid as long as θ is not close to either of its extremes.

Let subscripts b denote bicep PM coefficients and subscripts t denote tricep PM coefficients. Also let x_b denote bicep PM length and x_t denote tricep PM length. Since the total clockwise torque exerted by the bicep on the elbow is $(F_b - B_b\dot{x}_b - K_b x_b)a\sin\theta$, the total counterclockwise torque exerted by the tricep on the elbow is $(F_t - B_t\dot{x}_t - K_t x_t)r$ and the counterclockwise torque imparted to the elbow by gravity is $MgL\sin\theta$, the dynamics of the system of Figure 2 are described by:

$$\begin{aligned} I\ddot{\theta} = {} & (F_b - B_b\dot{x}_b - K_b x_b)a\sin\theta \\ & -(F - B_t\dot{x}_t - K_t x_t)r - MgL\sin\theta \end{aligned} \qquad (2.3)$$

where $I = ML^2$ is the moment of inertia of the mass about the elbow and g is the acceleration of gravity. Note that, since the bicep force is multiplied by $a\sin\theta$ the bicep loses controllability at $\theta = 0^\circ$ and $\theta = 180^\circ$. Thus, the arm should be kept away from these extremes. The tricep does not have this limitation because its cable always makes an angle of $\alpha = \sin^{-1}\left(r/a\right)$ with the arm regardless of θ.

As in (2.2), we use $F = F_1 P$, $K = K_0 + K_1 P$, and $B = B_0 + B_1 P$ where $F_1 = 3.77$, $K_0 = 32.58$, $K_1 = 1.209$, and B_0, B_1 depend on whether the PM in question is being inflated or deflated, as follows:

$$B_0 = \begin{cases} 5.748, \text{inflation} \\ 3.41, \text{deflation} \end{cases}, \quad B_1 = \begin{cases} 0.2719, \text{inflation} \\ -0.0316, \text{deflation} \end{cases}$$

We have neglected the P^2 term in F because it tends to be negligible in the pressure range of operation of the PM. The internal bicep and tricep pressures P_b and P_t are the control variables that can be independently commanded by the controller as inputs to the system. Thus this is a 2-input system. Note that the PM dynamics depend on whether the PM is being inflated or deflated. We see that (2.3) is in an unusual form for control since the control inputs enter into the system through the coefficients F, B, and K and not as a separate term.

Proceedings of the American Control Conference
Denver, Colorado June 4-6, 2003

To convert this 2-input system to a single-input system, let us assume that the bicep and tricep internal pressures are given by

$$P_b = P_0 + \Delta p \quad (2.4a)$$

$$P_t = P_0 - \Delta p \quad (2.4b)$$

where P_0 is a nominal constant pressure and Δp is the change in pressure which is now the independent control input. Note that, with PM pressure defined as in (2.4), one PM inflating always corresponds to the other deflating. Therefore, one set of B parameters (say inflation) will apply to one of the PMs while the other set (deflation) applies to the other PM at a given time. When the inflation status of the PMs changes, they trade B parameters. We denote the bicep B coefficients as B_{0b} and B_{1b}, and the tricep B coefficients as B_{0t} and B_{1t}.

When either PM is fully lengthened, its length is defined as zero, and when it is fully shortened, its length is defined as $-2a$ (i.e. x is the amount of PM shortening). Therefore, from Figure 2, the bicep length is $x_b = a(\cos\theta - 1)$ and the tricep length is $x_t = -a(1 + \cos\theta)$. Combining (2.3) with the above relationships for F, B, and K, we obtain the following 2nd order equation describing the system of Figure 2:

$$\ddot{\theta} = f(\theta, \dot{\theta}) + b(\theta, \dot{\theta})\Delta p \quad (2.5)$$

where

$$f(\theta,\dot{\theta}) = \sum_{i=1}^{6} c_i f_i(\theta,\dot{\theta}) \quad (2.6a)$$

$$b(\theta,\dot{\theta}) = \sum_{i=1}^{6} d_i f_i(\theta,\dot{\theta}) \quad (2.6b)$$

In (2.6), $f_1 = \sin\theta$, $f_2 = \sin\theta(\cos\theta - 1)$, $f_3 = \dot{\theta}\sin^2\theta$, $f_4 = 1$, $f_5 = 1 + \cos\theta$, $f_6 = \dot{\theta}\sin\theta$, $c_1 = (aF_1P_0 - MgL)/I$, $c_2 = (K_0 + K_1P_0)a^2/I$, $c_3 = (-B_{0b} - B_{1b}P_0)a^2/I$, $c_4 = -F_1P_0r/I$, $c_5 = (K_0 + K_1P_0)ar/I$, $c_6 = (-B_{0t} - B_{1t}P_0)ar/I$, $d_1 = F_1a/I$, $d_2 = K_1a^2/I$, $d_3 = -B_{1b}a^2/I$, $d_4 = F_1r/I$, $d_5 = -K_1ar/I$, and $d_6 = B_{1t}ar/I$.

The model (2.5) is now in a form suitable for sliding mode control.

III. Sliding Mode Controller for PMs in Bicep/Tricep Configuration

Pneumatic muscle actuators are time-variant, nonlinear and uncertain due to compressibility of air, static and Coulomb friction, and payload variations. The mass M manipulated by the PM can be expected to vary significantly from use to use. The coefficients B and K vary with PM temperature. Also, the physical distances r, a, and L may not be precisely known. Therefore, it is difficult to develop a precise model for PM systems such as that of Figure 2. Linear control methods often lead to poor performance for pneumatic systems.

Sliding mode control (SMC) has been used for robot control, including robotics utilizing PMs as actuators [11], [12]. In SMC, rather than controlling the states of the system directly, the desired error behavior of the system is specified in terms of a sliding surface in the state space. A discontinuous, so-called variable structure control law is used to drive the state to the sliding surface. The state then moves along this surface so that the desired tracking performance is asymptotically achieved. This method is robust to model uncertainty, because of the discontinuous feedback.

In practice, imperfect switching can result in unacceptably high control activity and chattering which can excite unmodeled dynamics. For this reason continuous feedback is often used in a neighborhood of the sliding surface so that the state remains within a so-called boundary layer near the sliding surface.

The typical sliding mode controller is composed of a nominal part, similar to a feedback linearizing or inverse control law, and additional terms aimed at dealing with model uncertainty. The problem of designing a robust control law by the sliding mode approach can be stated as follows: given a desired sliding manifold function of the system's states $(s(x) = 0)$, determine a control input which satisfies the sliding condition. Then the desired performance can be achieved by an involved reduced-order dynamics in the sliding regime. We show that sliding mode gives a viable alternative for high performance robotics applications involving PMs.

Because of our imperfect knowledge of coefficients F_1, K_0, K_1, B_0, and B_1, we must assume that $f(\theta,\dot{\theta})$ and $b(\theta,\dot{\theta})$ in (2.5) are imprecise. Assume the extent of the imprecision on $f(\theta,\dot{\theta})$ can be bounded by a known continuous function of θ and $\dot{\theta}$. Similarly, we assume that the extent of the imprecision on $b(\theta,\dot{\theta})$ can be bounded by a known, continuous function of θ and $\dot{\theta}$. The control problem is to get the joint angle $\theta(t)$ to track a desired trajectory $\theta_d(t)$ in the presence of model imprecision on $f(\theta,\dot{\theta})$ and $b(\theta,\dot{\theta})$.

Let $\hat{f}(\theta,\dot{\theta})$ be the estimate of f and let $F(\theta,\dot{\theta})$ be a positive function such that

$$\left|\hat{f}-f\right|\le F \tag{3.1}$$

Further assume the control gain $b(\theta,\dot{\theta})$ is unknown but of known bounds ($0<b_{\min}\le b\le b_{\max}$) where $b_{\min}$ and $b_{\max}$ may depend on θ and $\dot{\theta}$. Then define the gain margin

$$\beta=\sqrt{b_{\max}/b_{\min}} \tag{3.2}$$

Let the estimation of b be $\hat{b}=\sqrt{b_{\min}b_{\max}}$.

Let $\theta_d(t)$ be a smooth function of time that represents a desirable angular trajectory for the PM-actuated arm. Consider the sliding surface $s=0$, $s=\dot{\tilde{\theta}}+\lambda\tilde{\theta}$, where $\tilde{\theta}=\theta-\theta_d$ is the tracking error and λ is a scalar design parameter.

Let 'sat' denote the saturation function defined as

$$sat(y)=\begin{cases} y, & |y|\le 1 \\ \text{sgn}(y), & \text{otherwise}\end{cases} \tag{3.3}$$

Then the sliding mode control law is given by

$$\Delta p=\left(\Delta\hat{p}-k\,\text{sat}(s/\Phi)\right)\hat{b}^{-1} \tag{3.4}$$

where $\Delta\hat{p}=(-\hat{f}+\ddot{\theta}_d-\lambda\dot{\tilde{\theta}})$, $k\ge\beta(F+\eta)+(\beta-1)\left|\Delta\hat{p}\right|$, η is a positive constant, and Φ is the thickness of a "boundary layer," which is a neighborhood of the sliding surface introduced to reduce control chattering.

It can be shown [4] that the control law (3.4) guarantees that the system trajectory reaches the boundary layer in a finite time whatever the initial state, and inside the boundary layer constrains trajectories to stay inside it for all later time and approach a neighborhood of $\tilde{\theta}=0$ asymptotically as t goes to infinity. Asymptotic tracking to within a guaranteed accuracy is therefore obtained in spite of modeling errors.

IV. Simulation Results

The system of Figure 2 with PMs in bicep/tricep pair configuration is simulated using a 4th-order Runge-Kutta algorithm with a step size of 0.01seconds. Let L = 1.5 ft, a = 3 inch, M = 1 slug, and r = 2 inch. Since a = 3 inch, the full travel of the forearm from $\theta=0$ (arm fully straightened) to $\theta=\pi$ (arm fully bent) corresponds to a maximum change in length of the PM of 6 inches.

Case 1: The desired trajectory for the joint is

$$\theta_d(t)=\frac{\pi}{2}+0.5(\sin(2\pi f_1 t)+\sin(2\pi f_2 t)+\sin(2\pi f_3 t)) \tag{4.1}$$

with $f_1=0.02$Hz, $f_2=0.05$Hz, and $f_3=0.09$z. This trajectory spans joint angles from approximately 30° to 150° during the period $t=0-60$ sec.

Let λ = 10 and η = 10 (chosen by trial and error to yield good performance). We choose the boundary layer thickness as $\Phi=1$. It can be shown [4] that $\varepsilon=\frac{\Phi}{\lambda}$ can be regarded as the guaranteed tracking precision. Therefore, for this simulation we have $\varepsilon=0.1$.

Assume that the true values of $f(\theta,\dot{\theta})$ and $b(\theta,\dot{\theta})$ in (2.5) are known to fall within $\pm30\%$ of the best estimates we have of them, i.e. $\hat{f}(\theta,\dot{\theta})$ and $\hat{b}(\theta,\dot{\theta})$. Then we have $F=0.3\left|\hat{f}\right|$, $b_{\max}=1.3\hat{b}$, $b_{\min}=0.7\hat{b}$, and the gain margin β is determined as 1.86 by (3.2).

The sliding control input to the PM is given in (3.4) with parameters defined as above. For the simulation, the actual $f(\theta,\dot{\theta})$ and $b(\theta,\dot{\theta})$ terms were randomly chosen to lie within ±30% of their modeled values. Figure 3 shows the tracking errors for three different sets of f and b within this range. It is seen that for all systems the tracking error is within predicted bounds, with areas of maximum error corresponding to $\dot{\theta}$ changing signs, i.e. places where the arm motion has to change direction. This is especially noticeable when the arm must change from downward motion to upward motion.

Figure 4 shows a typical control effort Δp with $P_0=50$ lb/in^2. It is evident that input pressure varies smoothly without any obvious chattering. Therefore, by using the sliding mode controller, the PM system achieves desired performance with good tracking precision and no obvious chattering for all three systems which may represent the true arm with PMs in bicep/tricep pair configuration.

To investigate the robustness of the sliding controller to changing masses, we increased the mass M to 2 slugs, i.e. an increase of 100%. Figure 5 shows tracking errors for three different actual arms randomly chosen within the ±30% range. Tracking is again within predicted bounds. Figure 6 shows a typical control effort when $M=2$. Note that the control effort is larger than the $M=1$ case, which is to be expected

since a heavier mass is being moved. The mass M could be increased more, but very heavy masses require the input pressure to be outside the allowed range of PM internal pressure (0–130 psi). This limitation is not the sliding controller's shortcoming however; it is merely an acknowledgement that the PM internal pressure must be kept within reasonable bounds to protect against actuator failure (bursting). If more force is desired, several PMs can always be placed in parallel.

It is noted from Figure 6 that since $P_0 = 50$ psi, the values of Δp would require tricep pressure P_t to be negative (2.4b). This is impossible, and in such a case, we would simply set $P_t = 0$. The simulation reflects this. The fact that tricep pressure is mostly zero when $M = 2$ results from the heavier mass exerting enough downward force to track the downward parts of the reference trajectory without needing the tricep to help pull the arm down.

Case 2: To further verify the sliding mode controller, another simulation is performed to track a pseudo-square wave signal with a typical system within the ±30% range. Here, the desired trajectory is

$$\theta_d(t) = \begin{cases} \frac{3}{4}\pi, & \sin(2\pi f(t-3)) \geq \frac{1}{4} \\ \frac{\pi}{2} + \pi \sin(2\pi f(t-3)), & \left|\sin(2\pi f(t-3))\right| \leq \frac{1}{4} \\ \frac{1}{4}\pi, & \sin(2\pi f(t-3)) \leq -\frac{1}{4} \end{cases} \tag{4.2}$$

with f = 0.1Hz. This function transitions between constant values of $\pi/4$ and $3\pi/4$ smoothly rather than with discontinuous jumps. For the design parameters, we used $\lambda = 10$, $\eta = 10$, and $\Phi = 0.3$. Therefore, the tracking accuracy is $\varepsilon = \frac{\Phi}{\lambda} = 0.03$.

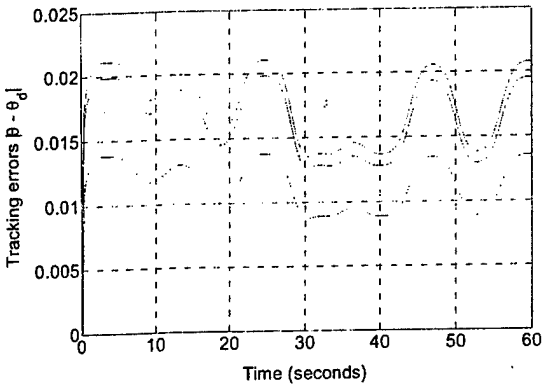


Figure 3 – Tracking errors for three possible actual arms, $M = 1$ (Case 1)

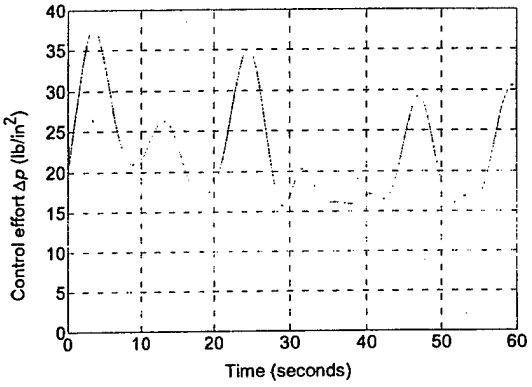


Figure 4 – Typical control effort Δp, $M = 1$ (Case 1)

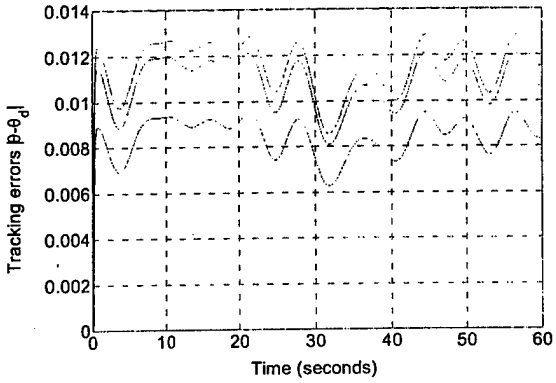


Figure 5 – Tracking errors for three possible actual arms, $M = 2$ (Case 1)

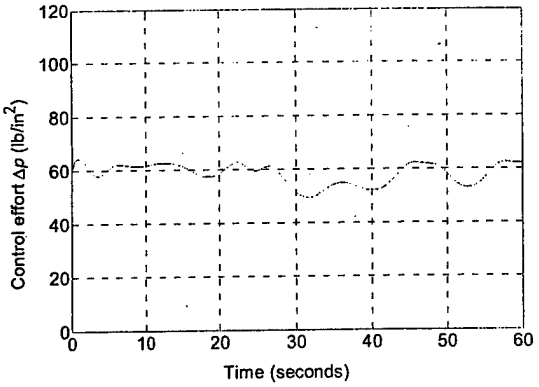


Figure 6 – Typical control effort Δp, $M = 2$ (Case 1)

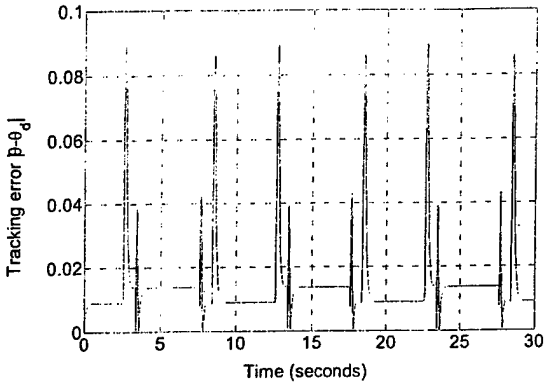


Figure 7 – Tracking error (Case 2)

From Figure 7, the joint angle trajectory is seen to follow the desired one with acceptable error except at the times of rapid transition between the two constant values. This is attributed to the fact that in the simulation the PM pressures are constrained to lie within the range $[0,130]$psi to better conform to actual PM operation. Therefore, the needed input pressure dictated by the sliding mode controller is not applied and tracking accuracy is lost.

V. Discussion

The properties of PMs in bicep and tricep configurations have already been studied individually in the context of nonlinear adaptive control [13]. In the present paper, actuation via bicep/tricep pair is developed. Sliding mode control methods have been applied to this problem since fixed controllers are less robust to parameter changes than sliding mode ones. In the case of time-invariant and well-known coefficients, traditional methods, i.e. PID may give good results. However, if coefficients or physical quantities change significantly, the fixed PID cannot stabilize the system. With the sliding mode controller given in Section 3, good tracking performance is obtained even in the presence of modeling uncertainties. The two trajectories considered in this paper are used because they mimic two common working situation of the PMs. Trajectory (4.1) represents a movement of the mass in a smooth trajectory. Trajectory (4.2) represents the task of holding the mass in a stationary position and then lifting it up or dropping it down suddenly. In both cases, the sliding mode controller can work with desirable performance of good tracking precision and little chattering.

VI. Conclusions

A dynamic model for an arm actuated by two pneumatic muscles in bicep/tricep pair configuration has been derived. Assuming certain degrees of inaccuracy in our knowledge of the PM coefficients, a sliding mode controller was designed to achieve good tracking performance. In order to eliminate chattering, the control action was also designed to be smoothed to achieve a trade-off between control bandwidth and tracking precision. Simulation results demonstrate the effectiveness of sliding mode control for PM applications.

VII. References

[1] D. G. Caldwell, G. A. Medrano-Cerda and M. Goodwin, "Control of pneumatic muscle actuators," *IEEE Control Systems Magazine*, February 1995.

[2] R. A. DeCarlo, S. H. Zak, and G. P. Matthews, "Variable structure control of nonlinear multi-variation systems: a tutorial," *Proceedings of the IEEE*, Vol. 76, No. 3, pp. 212-232, March 1988.

[3] V. I. Utkin, *Sliding Modes and Their Application in Variable Structure Systems*, Moscow: MIR, 1978.

[4] J. Slotine and W.Li, *Applied Nonlinear Control*, Englewood Cliffs, NJ: Prentice Hall 1991.

[5] W. Perruquetti, *Sliding mode control in engineering*, New York: Marcel Dekker, Inc., 2002.

[6] J-J. Slotine and S. Sastry, "Tracking control of nonlinear system using sliding surface with application to robot manipulators," *Int. J of Control*, Vol. 38,1983.

[7] C. Ünsal and P. Kachroo, "Sliding mode measurement feedback control for antilock braking systems," *IEEE Trans. on Control Systems Technology*, March 1999.

[8] J. Zhou and T. Hagiwara, "Existence conditions and applications of shifting sliding mode control," *Proc. of the 40th IEEE Conference on Decision and Control*, 2001.

[9] A. Levant, "Universal SISO sliding-mode controllers with finite-time convergence," *IEEE Trans. on Automatic Control*, Vol. 46, No.9, 2001.

[10] D. B. Reynolds, D. W. Repperger, C. A. Philips and G. Bandry, "Dynamic characteristics of pneumatic muscle," *Computers in Biology and Medicine*. Submitted.

[11] A. Lanzon, "Trajectory/force control for robotic manipulators using sliding mode and adaptive control," *Proc. American Control Conference*, 1999.

[12] B. W. Bekit, "Fuzzy sliding mode control for a robot manipulator," *Proc. of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 1997.

[13] J. H. Lilly, "Adaptive tracking for pneumatic muscle actuators in bicep and tricep configuration," *IEEE Trans. on Neural Systems and Rehabilitation*. To appear.

Intelligent Automation and Soft Computing, Vol. 9, No. 4, pp. 227-244, 2003

FUZZY CONTROL FOR PNEUMATIC MUSCLE TRACKING VIA EVOLUTIONARY TUNING

XIAOGUANG CHANG
Postdoctoral Fellow
Department of Electrical and Computer Engineering
University of Louisville
Louisville, Kentucky

JOHN H. LILLY
Department of Electrical and Computer Engineering
University of Louisville
Louisville, Kentucky

ABSTRACT—This paper studies the evolutionary design of a fuzzy P+ID controller for an actual pneumatic muscle actuator system. The control of pneumatic muscles is a challenging problem because of their high degree of nonlinearity, time-varying parameters, and uncertainty. A fuzzy P+ID controller is constructed using an incremental fuzzy logic controller in place of the proportional term in a conventional PID controller. Several controller parameters are optimized via an evolutionary algorithm. The optimization is performed using a recurrent neuro-fuzzy dynamic model of the muscle rather than the muscle itself. Control results are presented, where the control objective is to force muscle length to follow a reference signal under a load. After evolutionary design, excellent tracking performance is obtained with the real muscle without the need for further tuning of controller parameters. The tracking performance is compared to that of another fuzzy controller.

Key Words: Fuzzy control, evolutionary algorithms, neurofuzzy modeling, pneumatic actuators

1. INTRODUCTION

Pneumatic muscles (PMs) are a special type of pneumatic actuator made of an airtight rubber tube (or bladder) inside a strong braided plastic sheath with a helical weaving, as shown in Figure 1. A valve controls the flow of gas from a pressurized source into the bladder. When the valve opens, gas is delivered into the bladder and the bladder widens and shortens. The resulting contractile force is quite large in relation to the PM's size. If inflation and deflation can be accurately controlled, the PM length and force exerted can also be controlled.

In comparison with other actuators, pneumatic muscles possess some unique advantages. First, PMs have extremely high power/weight (1kW/kg) and power/volume ratio (1W/cm^3). They are easily and quickly constructed with few tools from cheap, commonly-available materials. Also, pressurized gas can be created as a byproduct of easily produced chemical reactions, obviating the need for heavy equipment such as compressors, batteries, generators or motors. If weight is not a concern, tanks filled with pressurized gas can be used to supply power to the PM. Finally, PMs are actuators which, if they fail, do so in a relatively safe manner. PM failure is the result of the bladder bursting under relatively low internal pressures, typically under 100 psi. Therefore, in contrast to hydraulic or electrical actuators, a PM failure is not likely to injure humans or damage equipment, making PMs good candidates for use in close proximity to humans and/or delicate equipment.

A PM can actuate linear motion within its active range. For PMs which are 8 - 12 inches long and 2 - 3 inches in diameter, the PM's active range is a few inches, which is roughly comparable to human

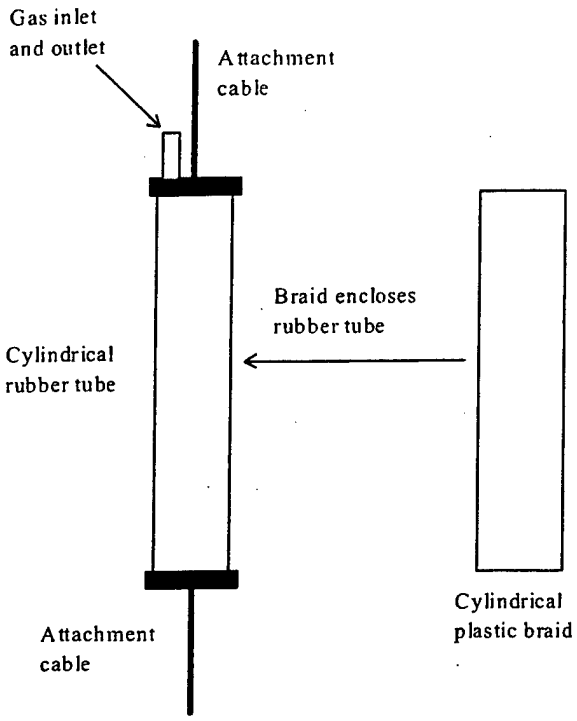


Figure 1. Pneumatic Muscle Construction.

muscles. Therefore, in addition to other configurations, PMs have been used in human-worn exoskeletons for strength augmentation of military personnel or rehabilitation for stroke victims and other injured or disabled patients. PMs are also finding applications in robotics systems in place of electrical actuators for driving joints [1 - 4]. They are excellent candidates for space, underwater, and other applications in which light weight, difficulty of supplying power, and obtaining replacement parts are concerns.

Nevertheless, PM use has been limited because they are very nonlinear and time-varying (due to differing inflation and deflation dynamics), making accurate position and force control challenging problems. Also, variations in load, the pressure of supply and other mechanical parameters cause uncertainty in the positional accuracy of the PM system.

So far, several advanced control strategies have been utilized to control pneumatic muscles, including adaptive control [5 - 7], variable structure control [8 - 10], predictive control [11], gain scheduling control [12] and neural network control [13]. These have met with varying degrees of success. Fuzzy logic-based control is known to excel with complex and highly nonlinear systems with parameter uncertainty. Moreover the design of fuzzy controllers is not model-based, making them well-suited for the control of pneumatic muscles.

Several fuzzy approaches to PM control have been tried [14, 15]. In [14], Chan et al. proposed a fuzzy model reference learning controller (FMRLC) and applied it to force the muscle length to follow a reference signal. In [15], Carbonell et al. proposed a nonlinear fuzzy back-stepping controller for set-point regulation of a pneumatic muscle actuator.

In general, fuzzy controllers have many parameters to be determined. The tuning of these parameters is typically done by a time-consuming trial and error process. Therefore, automatic tuning of design parameters is advantageous. Some studies [16 - 18] on evolutionary computation have shown that evolutionary algorithms (EAs) provide powerful ways to efficiently search in poorly understood, irregular spaces inspired by the mechanism of natural evolution. Following their successful applications in many difficult optimization problems, EAs are becoming increasingly used in the design of fuzzy systems. Work in EA-based design of fuzzy systems can be divided into two categories: rule-base construction and rule-base tuning. For rule-base construction, EAs are used for finding rules [19, 20] and determining the number of rules [21] from a data set. For rule-base tuning, EAs are used for optimizing membership functions in the fuzzy system according to some performance criterion [22].

In this paper, we design a fuzzy P+ID controller for an actual pneumatic muscle system located in the Human Sensory Feedback Laboratory at Wright-Patterson Air Force Base in Dayton, Ohio. For this purpose, a recurrent neuro-fuzzy model of the PM is developed, and on the basis of this model a modified evolutionary algorithm is used to tune the controller parameters. The resulting controller is then applied to the actual PM with excellent tracking results.

This paper is organized as follows. Section 2 studies PM dynamic modeling via a recurrent neuro-fuzzy network. Section 3 proposes a fuzzy P+ID control strategy for the PM and an evolutionary algorithm is developed for tuning controller parameters. Section 4 is devoted to the identification of a fuzzy model and the design of the fuzzy P+ID controller for the PM. In addition, some detailed control results are presented. Finally, some remarks and conclusions will be drawn in Section 5.

2. NEURO-FUZZY MODELING FOR PNEUMATIC MUSCLES

Several closed-form mathematical models have been developed for PMs in various configurations [23 - 26]. The physical configuration considered in our research is that of a PM hanging vertically actuating a mass (Figure 2). In [25], the dynamics of a pneumatic muscle system in the configuration of Figure 2 are described by a second-order nonlinear differential equation with differing inflation and deflation dynamics. Inflation and deflation dynamics are different because in deflation the PM system vents against a constant atmospheric pressure; during inflation, however, the pressure buildup is in a closed volume and the forcing function has to fight against itself as it inflates. Thus, deflation is easier than inflation.

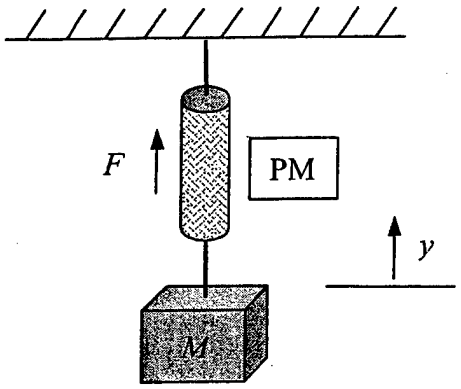


Figure 2. PM physical configuration.

In order to model this system via a recurrent neuro-fuzzy inference system (RNFIS) [27], we choose the general model form

$$\hat{y}(k) = f(\hat{y}(k-1), \hat{y}(k-2), x(k-1), x(k-2)) + e(k) \tag{1}$$

where k is the time step, $\hat{y}$ is the length of the muscle, x denotes the voltage input to the valve, f is some nonlinear function, and e denotes the modeling error. The neuro-fuzzy system used to realize (1) is shown in Figure 3 for the case of three fuzzy sets on each input universe.

The recurrent neuro-fuzzy inference system in Figure 3 is six-layered. Layer A is an input layer, and neurons in this layer represent fuzzy variables found in the fuzzy inference machine. These are volts applied to the valve and PM length in the past two time steps. Layer B is a term layer. Neurons in this layer are called term nodes. They represent fuzzy sets corresponding to linguistic values (small, medium, large, etc.) of the input variables. These fuzzy sets are characterized by asymmetrical triangular membership functions in the form:

$$\mu(v) = \begin{cases} \dfrac{v - m + \sigma_L}{\sigma_L} & m - \sigma_L \le v \le m \\ \dfrac{m + \sigma_R - v}{\sigma_R} & m \le v \le m + \sigma_R \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

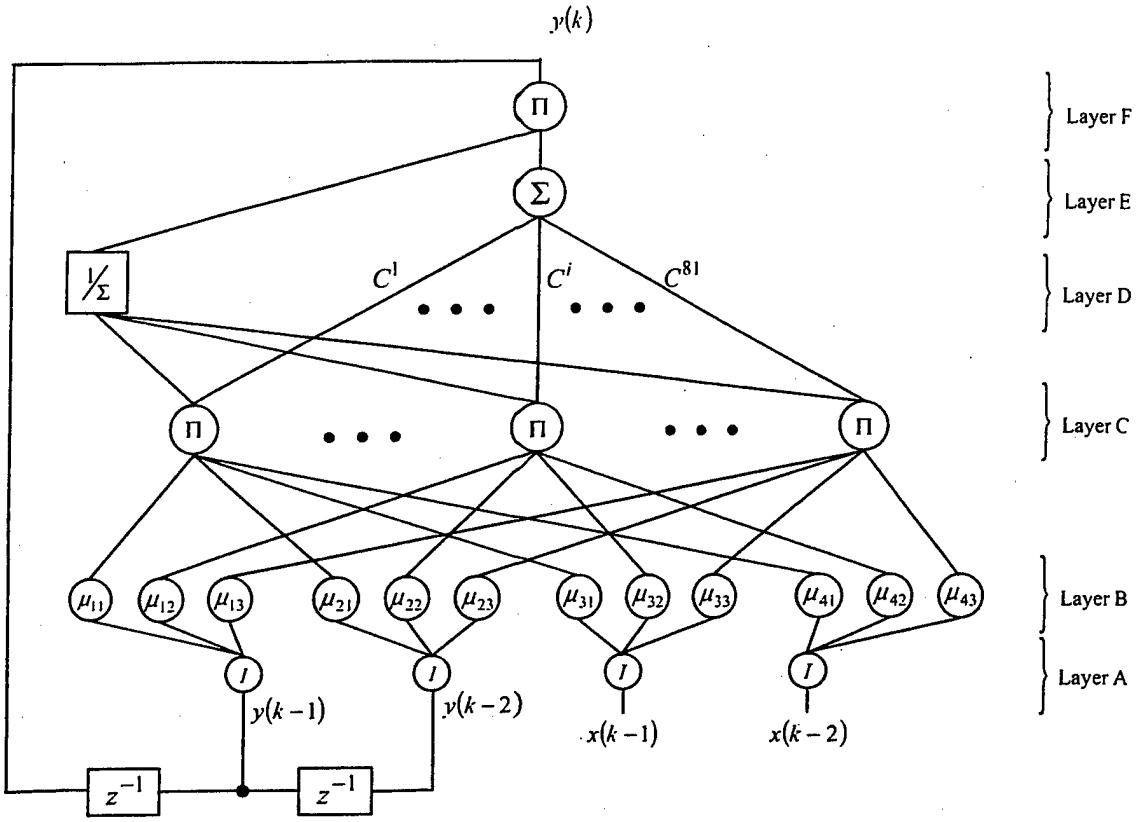


Figure 3. RNFIS used to model PM.

where v represents an input to the fuzzy system (i.e. $y(k\text{-}1)$, $y(k\text{-}2)$, $x(k\text{-}1)$, or $x(k\text{-}2)$), m is the value of v at which the membership function attains its maximum value of 1, and σ_L and σ_R are the left and right spreads of the triangles, i.e. the support of a particular membership $\mu(v)$ is $\{v : m-\sigma_L \le v \le m+\sigma_R\}$. Layer C is a rule layer. Neurons with label $\prod$ calculate the degree of match for parts of the premise of each rule, which evaluates the degree of activation of each rule. The label $\prod$ indicates that the product T-norm is used to implement conjunction in the premise of each rule. Layer D contains a neuron that calculates the reciprocal of sum of degrees of match of rules. Also in this layer are weights C^i appearing in the consequents of the rules. The neuron in layer E is an aggregation neuron, which sums the weighted premise values for each rule. Layer F is the defuzzification layer. Its neuron forms the product of the weighted premise values and inverse sum of premise values. The output of the neuron in this layer represents the estimate of the PM length at the current time instant.

The task of constructing an accurate RNFIS for PM modeling is divided into two phases: structure learning and parameter tuning. The structure learning phase is aimed at finding the structure of the RNFIS, such as the membership functions and the rules in the rule base. Structure learning involves fuzzy identification from data. The parameter-tuning phase consists of optimization of all weights in the RNFIS, i.e. input and output membership function centers and spreads. In the parameter-tuning phase, we use the well-known error backpropagation algorithm (BP) [27] for tuning model parameters. We will not address BP as applied to the network of Figure 3 here. We now briefly discuss the structure learning phase.

2.1 Structure Learning via VISIT

At the beginning, the RNFIS is empty without any rules or fuzzy sets. A simple self-organizing algorithm known as the Variable Input Spread Inference Training algorithm (VISIT) [28] is used to perceive structure features from a sequence of training data. VISIT is a variation of the well-known Modified Learning From Examples (MLFE) algorithm for identification of fuzzy systems from data.

In VISIT, the membership functions can be any shape in general, as long as they are convex. In this paper, we will use asymmetrical triangular input membership functions and singleton output membership functions.

To begin the VISIT algorithm, training constants σ_0, λ, ε, and w are specified by the user. The constant σ_0 is the initial value for the spread of the first membership function on each universe. The constant λ determines when a new membership function is created on a universe of discourse. If a new training point is within a λ-cut of an existing membership function on a universe, a new membership function is not created on that universe. The constant ε is the maximum identification error which is tolerated before a new rule and new memberships are added to the fuzzy system. Finally, w determines the amount of overlap between adjacent memberships on a universe. We now give the VISIT algorithm.

2.2 VISIT Algorithm

1. Set $i=1$. Get the first training pair $tp^1=(x_1, x_2, \ldots, x_n, y)^1$. On each input universe of discourse form a fuzzy set A_j^1 characterized by a membership function $\mu_{A_j^1}(x_j^1)$ with center at $m_j^1=x_j^1$ and spread σ_0. On the output universe of discourse form a fuzzy set B^1 characterized by a singleton membership function with support y^1.
2. Add the rule

   If x_1 is A_1^1 and x_2 is A_2^1 and ... and x_n is A_n^1 then y is B^1.
3. If there are no more training pairs, the training cycle is completed. Otherwise, increment *i* and get the next training pair $tp^i=(x_1, x_2, \ldots, x_n, y)^i$.
4. If $\left|f(tp^i)-y^i\right|<\varepsilon$ where $f(tp^i)$ denotes the crisp output of the existing fuzzy system evaluated at the new training pair tp^i, discard tp^i and go to 3.
5. On the *j*th input universe, evaluate all membership functions at x_j^i. Call the fuzzy set whose membership function is maximum $A_j^{\max}$. If $x_j^i \notin \lambda-cut(A_j^{\max})$, form a new fuzzy set A_j^{new} on the *j*th universe characterized by a membership function with center $m_j^{new}=x_j^i$ and left and right spreads σ_j^L and σ_j^R as in step 6 below. On the output universe form a new fuzzy set B^{new} characterized by a singleton membership function with support y^i.
6. On each input universe on which a new membership function was added in 5, the spreads of each side of the new membership function and its nearest neighbors on the left and right are re-calculated as follows. The right spread of the new membership function and the left spread of the nearest right neighbor are re-calculated as

$$\sigma_j^R=\frac{1}{w}\left|m_j^{new}-m_j^{nRn}\right| \tag{3}$$

The left spread of the new membership function and the right spread of the nearest left neighbor are re-calculated as

$$\sigma_j^L=\frac{1}{w}\left|m_j^{new}-m_j^{nLn}\right| \tag{4}$$

where m^{nRn} denotes the nearest existing center to the right of m_j^{new} and m_j^{nLn} denotes the nearest existing center to the left of m_j^{new}.

7. Add the rule

$$\text{If } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } A_2 \text{ and } \ldots \text{ and } x_n \text{ is } A_n \text{ then } y \text{ is } B^{new}$$

if there is no other rule in the rule base that is inconsistent (i.e. same premise, different consequent) with this rule, where the fuzzy sets in the premise are the ones maximized by the corresponding inputs

8. If there is another rule (rule p generated from tp^k) in the rule base that is inconsistent with the rule formed in 7 (rule q generated from the present training pair tp^l), define fuzzy system f_p (f_q) to be the fuzzy system with rule p (q) included in the rule base but rule q (p) omitted from the rule base. Retain rule p and omit rule q if

$$\left|f_p\left(tp^k\right)-y^k\right|+\left|f_p\left(tp^l\right)-y^l\right|<\left|f_q\left(tp^k\right)-y^k\right|+\left|f_q\left(tp^l\right)-y^l\right| \tag{5}$$

Otherwise, include rule q and omit rule p.

9. Go to 3.

2.3 Comment

It is possible that the rule to be added in step 7 has the same premise but a different consequent than an existing rule, i.e. the newly created rule is *inconsistent* with an existing rule. In such a case, we need to determine which rule provides a better match to the data so we know which rule to retain in the rule base and which rule to omit. To do this, we create two fuzzy systems, each with only one of the inconsistent rules retained, but with the other inconsistent rule omitted. In (5), a comparison of the two fuzzy systems is made on both training pairs that generated the two inconsistent rules. The fuzzy system having less total identification error for both training pairs is retained and the other omitted. In this way, we determine which of the two inconsistent rules does a better job of describing the data and retain it in the rule base, omitting the other.

3 EVOLUTIONARY FUZZY P+ID CONTROL

Fuzzy controllers are gaining popularity in the control community due to their capabilities of dealing with uncertainty, but their systematic design is still an open problem. A typical fuzzy controller found in the literature often has many parameters, some of which are intercorrelated and co-dependent. It is often impossible to perceive the relationship between closed-loop performance and the parameters of the fuzzy controller. This increases the difficulty of design of fuzzy controllers. For these reasons, we propose an evolutionary fuzzy P+ID controller for PM control.

3.1 Fuzzy P+ID Controller

A fuzzy P+ID controller, shown in Figure 4, is constructed using an incremental fuzzy logic controller with output $\Delta x_f(k) = FLC(e,\dot{e})$ in place of the proportional term in a conventional PID controller. The PM input is a voltage $x(k)$ which is expressed in incremental form as

$$\begin{aligned}\Delta x(k) &= x(k)-x(k-1)\\ &= K_P S_{dx}\Delta x_f(k)+K_I Te(k)-K_D\frac{y(k)-2y(k-1)+y(k-2)}{T}\end{aligned} \tag{6}$$

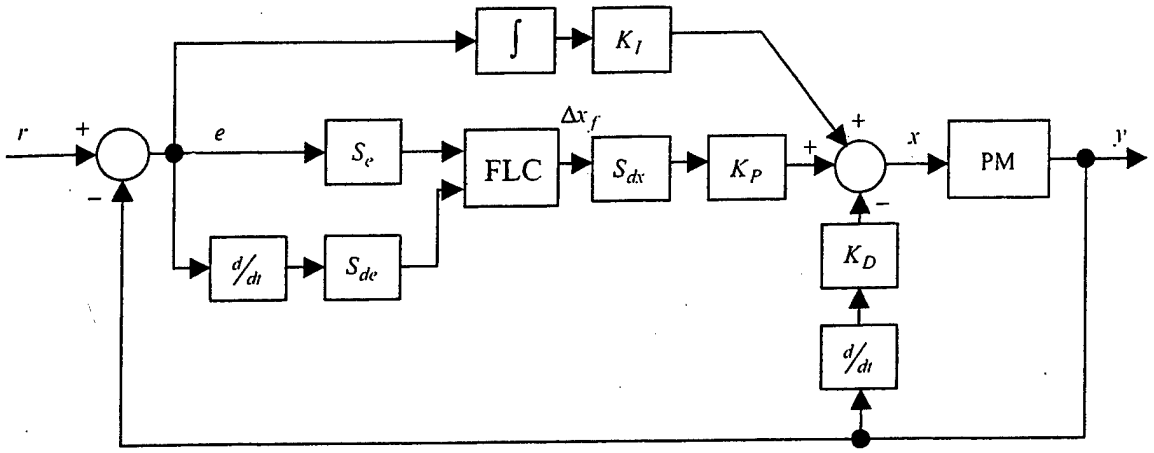


Figure 4. Fuzzy P+ID controller for PM.

In (6), K_P, K_I, and K_D are PID control gains, S_{dx} is the output scaling gain of the fuzzy-P subsystem, and T is the step size. In general, scaling gains S_e and S_{de} in Figure 4 are chosen so that values of $S_e e$ and $S_{de}\dot{e}$ lie in the interval [-1, 1], and S_{dx} is often chosen to be equal to the reciprocal of S_{de}. In our approach, these scaling gains will be determined by an evolutionary algorithm.

To reduce the number of control parameters, we design membership functions for the fuzzy-P controller with the following assumptions:

1. $e(k)$ and $\dot{e}(k)$ universes of discourse have five triangular membership functions, i.e. NB, NS, ZO, PS and PB. These attain their maximum value of unity at $-\beta$, $-\alpha$, 0, α, and β respectively and form a partition of unity. This number of memberships is chosen to keep the evolutionary optimization task relatively simple while maintaining good control performance.
2. $\Delta x_f(k)$ universe has five singleton membership functions located at $-\beta$(NB), $-\alpha$(NS), 0(ZO), α(PS) and β(PB).

A general rule in the fuzzy-P part of the P+ID controller can be written as:

$$RULE\ i\ (i=1,2,\ldots,25):$$
$$If\ e(k)\ is\ \mu_e^j(e(k))\ and\ \dot{e}(k)\ is\ \mu_{de}^k(\dot{e}(k))\ then\ \Delta x_f(k+1)=C^i$$

where μ_e^j and μ_{de}^k are one of the fuzzy sets NB, NS, ZO, PS and PB. The consequent C^i takes on values in the set $\{-\beta, -\alpha, 0, \alpha, \beta\}$. The rule base of the fuzzy-P part of the controller is determined from expert knowledge and shown in Table 1. The crisp output of the fuzzy-P system then can be calculated as:

$$\Delta x_f(k+1)=\frac{\sum_i C^i\mu_i}{\sum_i \mu_i} \tag{7}$$

where $i=1,\ldots 25$ and μ_i is the premise value of rule i.

Table I. Rule base for fuzzy P+ID controller.

Δv_i		e				
		NB	NS	ZO	PS	PB
$\dot{e}$	PB	ZO	PS	PB	PB	PB
	PS	NS	ZO	PS	PB	PB
	ZO	NB	NS	ZO	PS	PB
	NS	NB	NB	NS	ZO	PS
	NB	NB	NB	NB	NS	ZO

There are three kinds of parameters in the fuzzy P+ID controller: fuzzy control parameters, PID parameters, and scaling gains for the fuzzy-P subsystem. The quantities α and β are fuzzy control parameters. The proper tuning of α and β will greatly improve performance. In general, α is related to steady-state accuracy and β is related to the response speed. Small α implies high steady-state accuracy; large β speeds up the transient response [29, 30]. The PID parameters K_P, K_I, and K_D determine the control resolution and the stability of the controlled system. According to [29, 30], the proven sufficient conditions show that the system stability is not destroyed when a fuzzy P+ID controller takes the place of a conventional PID controller. This means that closed-loop stability is guaranteed regardless of the choice of α and β. The quantities S_e, S_{de}, and S_{dx} are scaling gains for the fuzzy-P subsystem. Control parameters and scaling gains will be determined by an evolutionary algorithm. The PID parameters will be determined using the Ziegler-Nichols tuning method.

3.2 Evolutionary Design of Fuzzy P+ID Controller

The design of fuzzy controllers is complicated by the fact that we do not know the relationship between membership functions and the control performance. To avoid a tedious trial and error tuning process, the following evolutionary algorithm is proposed to automatically tune the five parameters α, β, S_e, S_{de}, and S_{dx} of the fuzzy P +ID controller.

3.3 Evolution Algorithm:

1. Initialize randomly an even number of P individuals.
2. Evaluate the fitness of the P individuals using the fitness function f defined in (8) below.
3. Select the $P/2$ fittest individuals and copy them as the first $P/2$ individuals of the next generation. Discard the other individuals.
4. Apply the mutation operator defined in (9) below to the individuals selected in 3 and generate another $P/2$ individuals for the next generation.
5. Go to Step 2 and repeat until the stopping criterion is met.

A real vector $p(n)=[\alpha(n), \beta(n), S_e(n), S_{de}(n), S_{dx}(n)] \in \Re^5$ is used in the above algorithm to represent the nth individual in the population, which can be evolved by a set of mutation operations and the elitist selection operation. In each generation, the individuals with fitness in the top 50% are used to create the population for the next generation. For control applications, the well-known ITAE criterion is often used to form the fitness function:

$$f=\frac{1}{\sum_{k=1}^{K}\left|\frac{e(k)}{r(k)}\right|kT} \tag{8}$$

where K is the total number of time steps in the calculation, $e(k)$ is the tracking error at the kth time instant, $r(k)$ is the desired output at the kth time instant, and T is the step size. Note that the reference

signal $r(k)$ is always greater than zero. Thus, the larger the value of the fitness function, the better the tracking performance.

We define the following mutation operator for creation of new individuals from the fittest of the previous generation:

$$p_{j,m+1}(n+P/2) = p_{j,m}(n) + \sigma_j \rho_j(n) \exp\left(-\frac{f_n^m}{f_{\max}^m}\right) \tag{9}$$

where $j = 1, \ldots, 5$ indexes parameters in individuals, m is the generation number, $n = 1, \ldots, P/2$ is the index for individuals with the $P/2$ largest fitnesses in the current generation, P is the population size, σ_j is a learning coefficient which determines the amount of change of parameter j from generation to generation, ρ_j is a Gaussian random number $\in [-1, 1]$, f_n^m is the fitness of individual n in the mth generation, and $f_{\max}^m$ is the largest fitness in the mth generation.

In consideration of the constraints $0 \le \alpha < \beta \le 1$, the following techniques are used to deal with illegal individuals possibly generated by the mutation operators:

1. If $\alpha_{m+1}(n+P/2) < 0$, $\alpha_{m+1}(n+P/2) = \rho_\alpha(n)$, where ρ_α is a Gaussian random number $\in [0, 1]$.
2. If $\beta_{m+1}(n+P/2) > 1$, $\beta_{m+1}(n+P/2) = \rho_\beta(n)$, where ρ_β is a Gaussian random number $\in [0, 1]$.
3. If $\alpha_{m+1}(n+P/2) > \beta_{m+1}(n+P/2)$, exchange the values of $\alpha_{m+1}(n+P/2)$ and $\beta_{m+1}(n+P/2)$.

4. EXPERIMENTAL RESULTS AND DISCUSSION

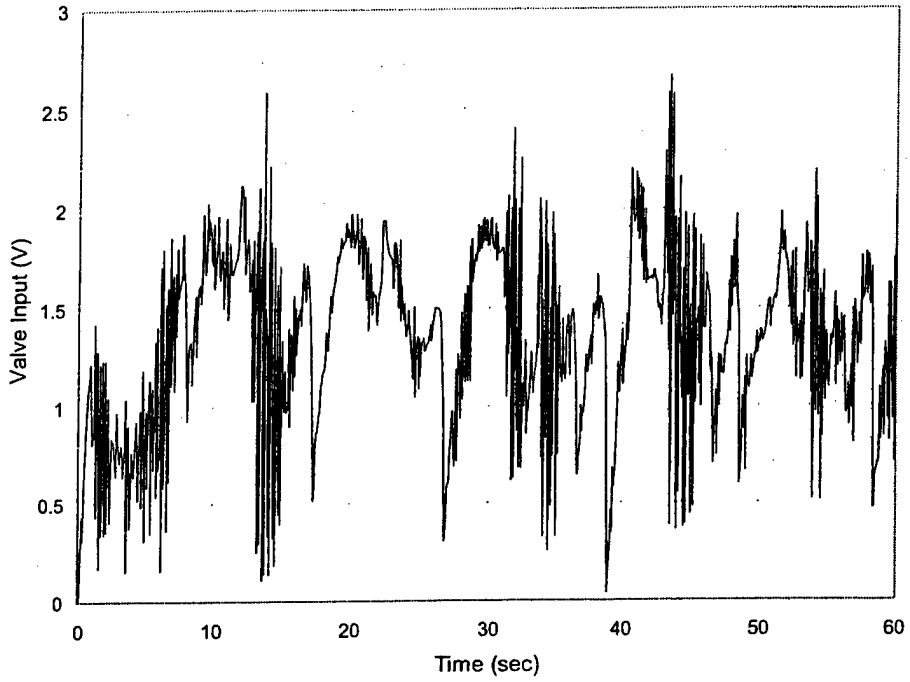
In this section, we design a fuzzy P+ID controller for a PM hanging vertically actuating a mass as shown in Figure 2. The system is located in the Human Sensory Feedback Laboratory at Wright-Patterson Air Force Base. The design procedure for the controller is as follows. First, a recurrent neuro-fuzzy model of the PM in the form of (1) is derived from experimental data taken from the physical system using the VISIT and BP algorithms as outlined in Section 2.

This RNFIS model is then used in place of the actual PM system in the tuning of the five parameters of the fuzzy controller ($\alpha, \beta, S_e, S_{de}, S_{dx}$) using the evolutionary method discussed in Section 3 and also to determine the PID gains K_P, K_I, and K_D. The PID gains are found via the well-known Ziegler-Nichols tuning method applied to the identified recurrent neuro-fuzzy system. We discuss each of these below.

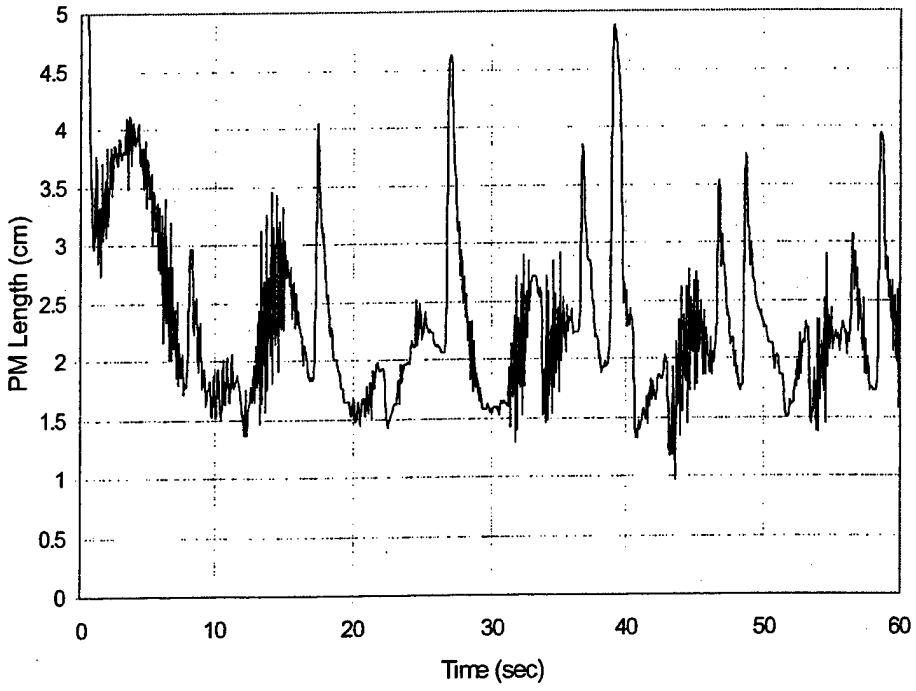
4.1 Dynamic Modeling of PM from Test Data

We collected several sets of input-output data from the PM and used these for training and test data for modeling. A typical data set is shown in Figure 5. Using these data, a RNFIS model as in Figure 3 was developed. In the structure learning phase, we used VISIT with $\varepsilon = 0.1$, $w = 0.85$, $\lambda = 0.27$, and $\sigma_0 = 1.17$. These parameters were chosen to yield a fuzzy system with a relatively small number of rules and fuzzy sets on each universe while also giving relatively small identification error. This provided a good initial system structure to be tuned further via BP.

In the parameter learning phase, we use a BP algorithm [27] for tuning model parameters (i.e. centers and spreads of the model membership functions identified by VISIT). After approximately 20,000 iterations, the BP training was ended at a mean square error of MSE=0.0011, where



(a)



(b)

Figure 5. Sample PM input-output data used for RNFIS identification (a) input, (b) output.

$$MSE = \frac{1}{K1}\sum_{k=1}^{K1}(\hat{y}(k) - y(k))^2 \tag{10}$$

In (10), $\hat{y}(k)$ is the output of the identified RNFIS model, $y(k)$ is the experimentally measured PM length (see Figure 5(b)), and $K1$ is the number of input-output pairs used. The model output is compared with that of the true PM in Figure 6. The sampling time in Figure 6 is $1/64$ second, yielding $K1 \approx 3600$. Obviously from Figure 6, the model well describes the PM dynamics. This process yielded the following 32 rules together with the input membership functions plotted in Figure 7.

4.2 Rule-base for RNFIS Model of PM

1. If $y(k\text{-}1)$ is mf_{11} and $y(k\text{-}2)$ is mf_{21} and $x(k\text{-}2)$ is mf_{31} and $x(k\text{-}1)$ is mf_{41} then $y(k) = 1.900$
2. If $y(k\text{-}1)$ is mf_{11} and $y(k\text{-}2)$ is mf_{21} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{41} then $y(k) = 2.154$
3. If $y(k\text{-}1)$ is mf_{11} and $y(k\text{-}2)$ is mf_{21} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{42} then $y(k) = 1.805$
4. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{21} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{42} then $y(k) = 0.160$
5. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{42} then $y(k) = 2.481$
6. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{41} then $y(k) = 2.478$
7. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{31} and $x(k\text{-}1)$ is mf_{42} then $y(k) = 0.668$
8. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{21} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{41} then $y(k) = 0.325$
9. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{31} and $x(k\text{-}1)$ is mf_{41} then $y(k) = 1.001$
10. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{43} then $y(k) = 0.425$
11. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{33} and $x(k\text{-}1)$ is mf_{41} then $y(k) = 2.179$
12. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{31} and $x(k\text{-}1)$ is mf_{43} then $y(k) = 1.163$
13. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{33} and $x(k\text{-}1)$ is mf_{42} then $y(k) = 0.461$
14. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{33} and $x(k\text{-}1)$ is mf_{43} then $y(k) = 0.876$
15. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{33} and $x(k\text{-}1)$ is mf_{43} then $y(k) = 0.573$
16. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{33} and $x(k\text{-}1)$ is mf_{41} then $y(k) = 0.829$
17. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{31} and $x(k\text{-}1)$ is mf_{43} then $y(k) = 3.594$
18. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{31} and $x(k\text{-}1)$ is mf_{44} then $y(k) = 0.611$
19. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{33} and $x(k\text{-}1)$ is mf_{42} then $y(k) = 1.071$
20. If $y(k\text{-}1)$ is mf_{11} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{31} and $x(k\text{-}1)$ is mf_{41} then $y(k) = 2.271$
21. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{23} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{44} then $y(k) = 1.154$
22. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{23} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{43} then $y(k) = 1.225$
23. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{23} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{42} then $y(k) = 1.169$
24. If $y(k\text{-}1)$ is mf_{12} and $y(k\text{-}2)$ is mf_{23} and $x(k\text{-}2)$ is mf_{31} and $x(k\text{-}1)$ is mf_{43} then $y(k) = 0.534$
25. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{22} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{42} then $y(k) = 0.685$
26. If $y(k\text{-}1)$ is mf_{11} and $y(k\text{-}2)$ is mf_{21} and $x(k\text{-}2)$ is mf_{31} and $x(k\text{-}1)$ is mf_{42} then $y(k) = 1.841$
27. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{23} and $x(k\text{-}2)$ is mf_{33} and $x(k\text{-}1)$ is mf_{43} then $y(k) = 0.587$
28. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{23} and $x(k\text{-}2)$ is mf_{33} and $x(k\text{-}1)$ is mf_{44} then $y(k) = 0.439$
29. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{23} and $x(k\text{-}2)$ is mf_{34} and $x(k\text{-}1)$ is mf_{42} then $y(k) = 0.607$
30. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{23} and $x(k\text{-}2)$ is mf_{32} and $x(k\text{-}1)$ is mf_{44} then $y(k) = 0.094$
31. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{23} and $x(k\text{-}2)$ is mf_{33} and $x(k\text{-}1)$ is mf_{42} then $y(k) = 0.893$
32. If $y(k\text{-}1)$ is mf_{13} and $y(k\text{-}2)$ is mf_{23} and $x(k\text{-}2)$ is mf_{33} and $x(k\text{-}1)$ is mf_{41} then $y(k) = 2.043$

4.3 Evolutionary Design of Fuzzy P+ID Controller Using Neuro-fuzzy Model

An optimal fuzzy P+ID controller is designed for the pneumatic muscle via the evolutionary algorithm of the previous section based on the obtained RNFIS model above. Note that parameters α, β, S_e, S_{de}, and S_{dx} were optimized, so we needn't specify scaling gains in advance according to the actual ranges of e and $\dot{e}$. PID parameters were set as $K_P = -0.7$, $K_I = -0.22$, and $K_D = -0.5$ using the Ziegler-Nichols tuning method based on open-loop tests with the identified RNFIS model.

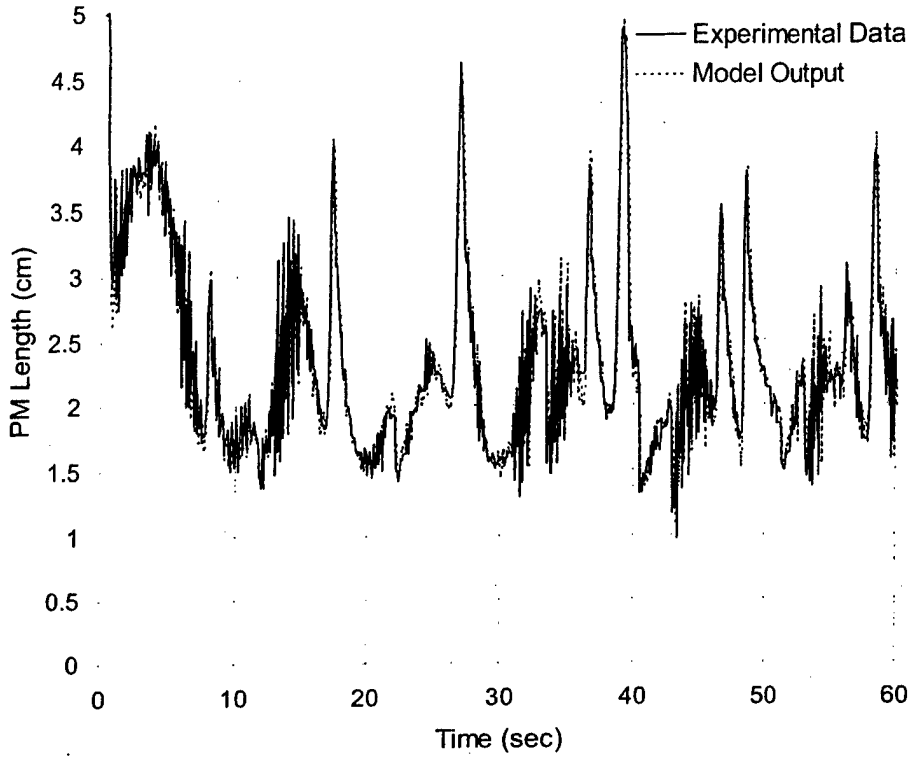


Figure 6. PM identification results.

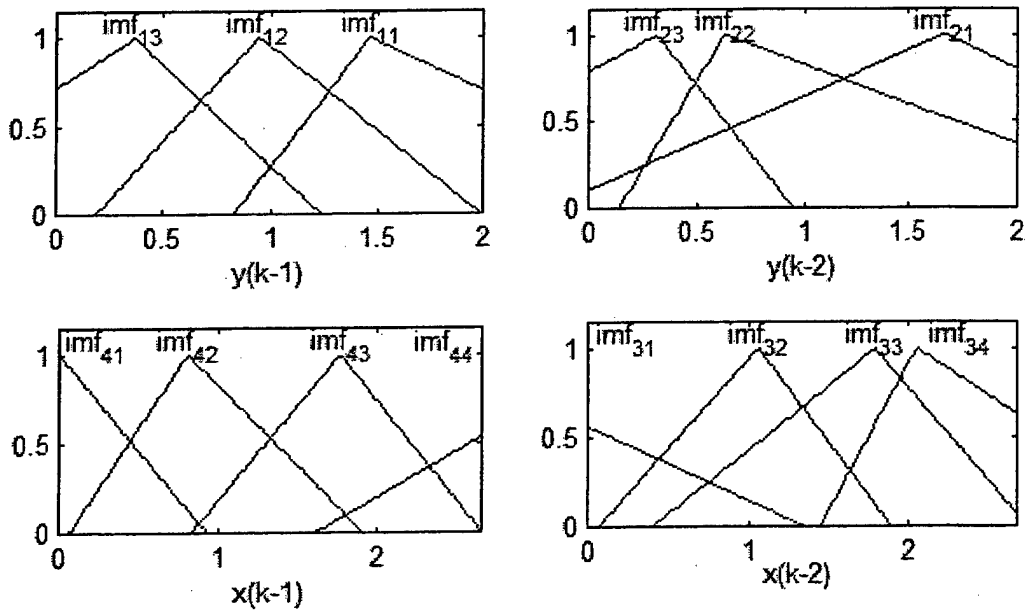


Figure 7. Input membership functions for the RNFIS model of PM.

After 400 generations with a population of $P = 120$, the following optimal parameters are obtained: $\alpha^* = 0.203$, $\beta^* = 0.632$, $S_e^* = 0.9982$, $S_{de}^* = 0.2232$, and $S_{dx}^* = 1.002$. Corresponding

membership functions of the fuzzy controller are plotted in Figure 8, and its control surface is plotted in Figure 9.

We tested the controller's ability to force the PM to track a reference signal. For these experiments the pneumatic muscle was loaded with a mass of 20 kg. The initial length of the muscle is 5 cm, corresponding to the PM fully deflated and extended. For comparison, the P+ID controller performance is compared to that of a fuzzy model reference learning controller (FMRLC) which was tuned by trial and error for best performance [14]. The tracking performance of both controllers is plotted in Figure 10, and a comparison of the tracking errors is shown in Figure 11. The reference signal is a combination sinusoid/triangle function.

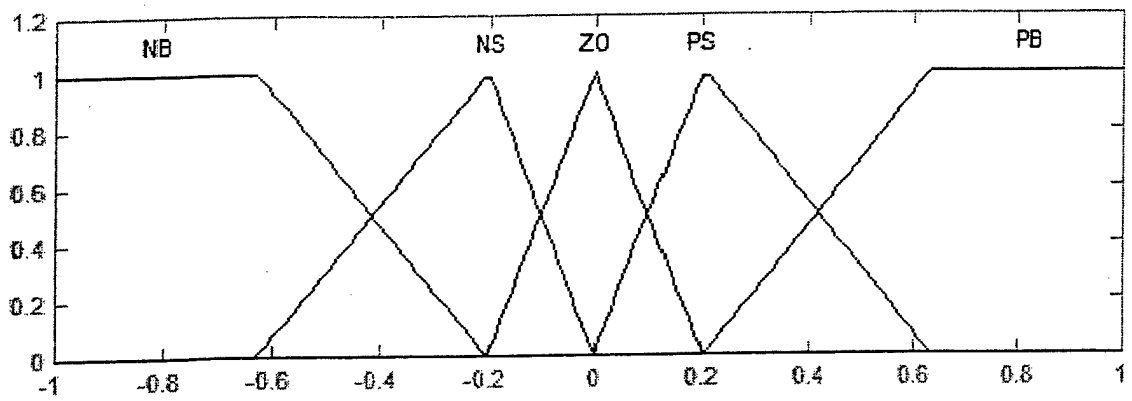


Figure 8. Membership functions for *e* and $\dot{e}$.

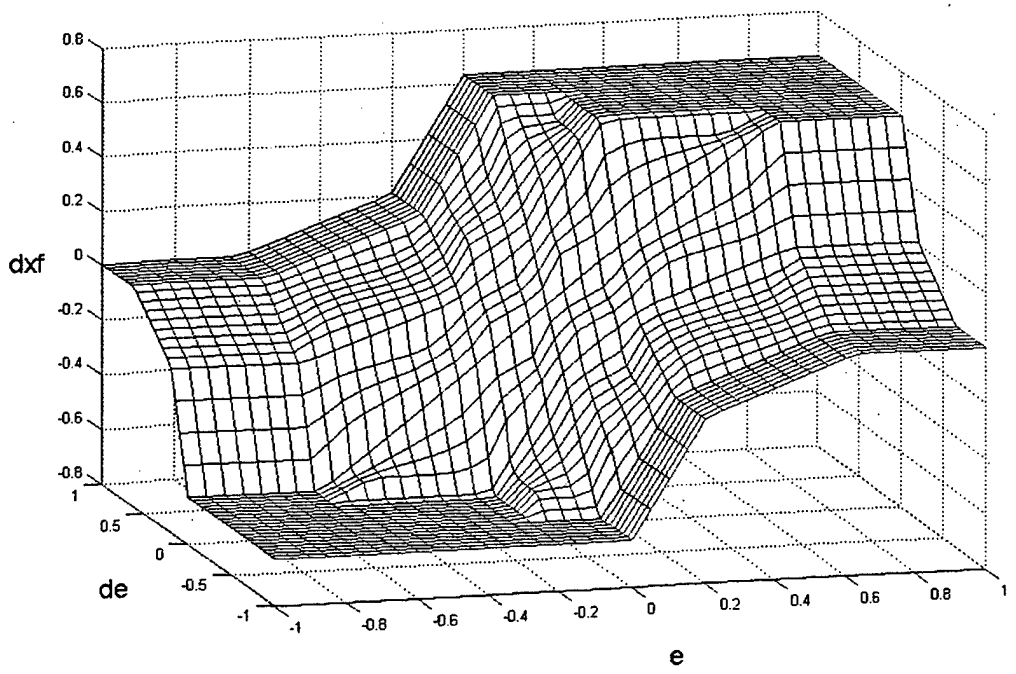
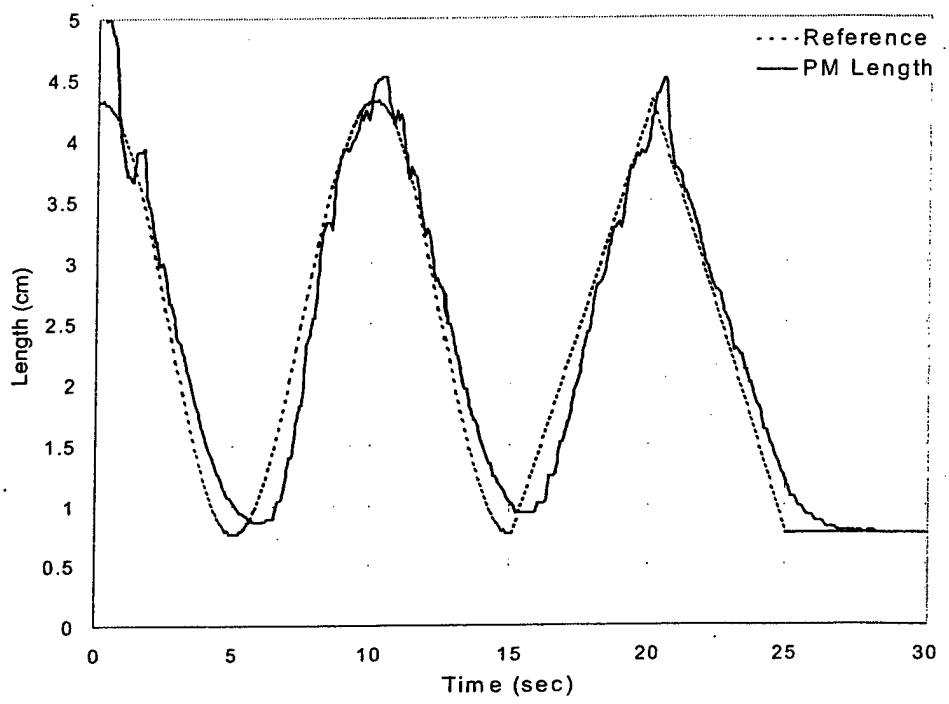
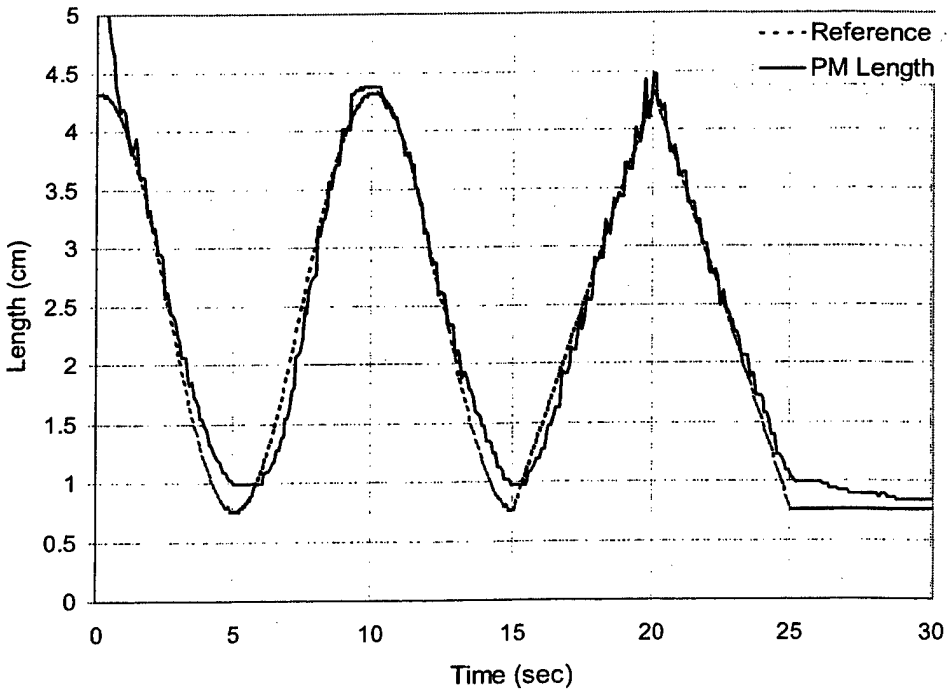


Figure 9. Control surface for fuzzy-P part of controller.



(a)



(b)

Figure 10. Tracking performance with (a) FMRLC controller, (b) fuzzy P+ID controller.

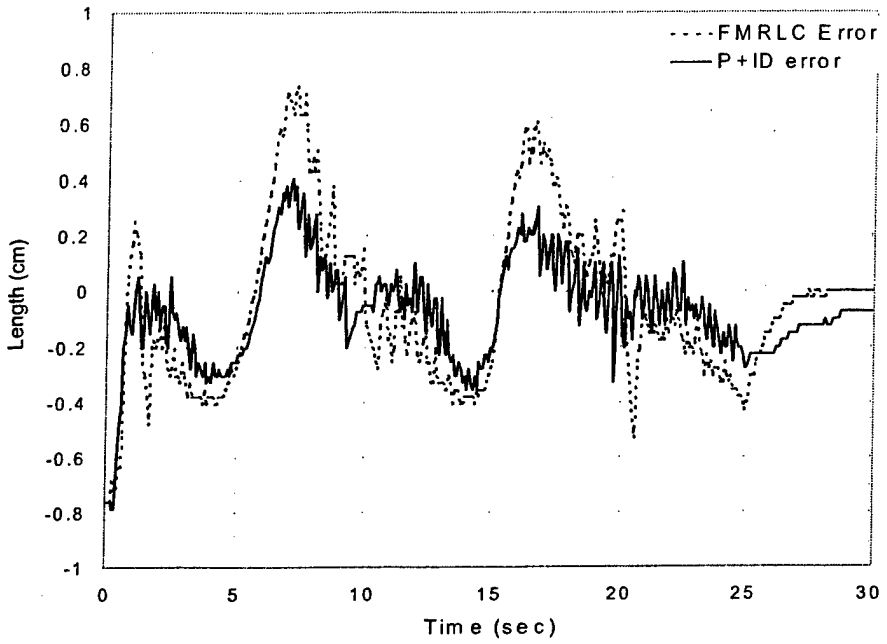


Figure 11. Comparison of tracking errors.

From Figure 11, we see that the evolutionary fuzzy P+ID controller performance is superior to that of the FMRLC i.e. the tracking error is less. To compare the tracking performances rigorously, we use the maximal deviation, defined as

$$MD = \min_k \left(\left| r(k) - y(k) \right| \right) \tag{11}$$

and the average deviation, defined as

$$AD = \frac{1}{K2} \left(\sum_k \left| r(k) - y(k) \right| \right) \tag{12}$$

where k ranges over all time samples, $r(k)$ is the reference signal, $y(k)$ is the PM length, and $K2$ is the total number of time steps. For the FMRLC, the maximal deviation is MD = 0.41 and the average deviation is AD = 0.1173. For the evolutionary fuzzy P+ID controller, we have MD = 0.16 and AD = 0.0558, demonstrating the superiority of the evolutionary fuzzy P+ID controller.

The control actions exerted by the two controllers in the above tracking performances are plotted in Figure 12. The control input from the evolutionary fuzzy P+ID controller is seen to be less than that from the FMRLC. Therefore, the evolutionary fuzzy P+ID controller attains better tracking performance using smaller control effort than the FMRLC. Because the control signal in the PM system corresponds to the volts supplied to the valve, this means that the fuzzy P+ID controller is more efficient, i.e. can use less energy and get better results. This can be valuable in applications where energy is limited, such as use in remote locations, underwater or spacecraft applications.

It should be noted that the FMRLC used for these experiments was tuned over many trials to yield good tracking. To be fair, however, these FMRLC results may not be the best attainable; better tracking may result from further tuning. In contrast, the excellent tracking results shown in Figure 10(b) for the evolutionary fuzzy P+ID controller were obtained upon first application of the controller, i.e. no tuning was required. Most fuzzy systems require extensive tuning before they perform satisfactorily. Therefore, the combination of neuro-fuzzy modeling and evolutionary design based on this model is a powerful design tool for fuzzy control.

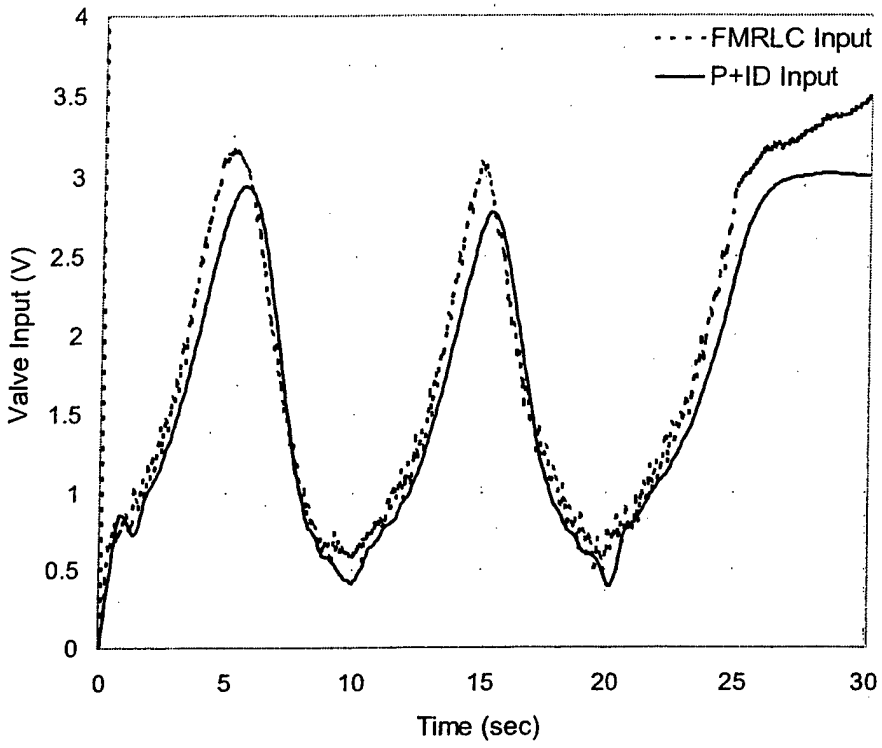


Figure 12. Comparison of control actions.

5. CONCLUSIONS

The control of pneumatic muscles (PM) is difficult due to their complex dynamic characteristics such as nonlinearity, uncertainty and some time-varying features such as differing inflation and deflation dynamics. This paper studies the tracking control of a pneumatic muscle suspended vertically actuating a mass. The controller for this research is an evolutionary fuzzy P+ID controller that was implemented on the Pneumatic Muscle Test Station at the Human Sensory Feedback Laboratory at Wright-Patterson Air Force Base. The controller was designed using a recurrent neuro-fuzzy model of the PM obtained from test data taken in the lab. This controller is compared with a fuzzy model reference learning controller (FMRLC) also designed for this task. Tracking performance of the evolutionary fuzzy P+ID controller is seen to be superior to the FMRLC performance. Also, tracking performance is immediately excellent upon first application of the controller with no tuning of the controller necessary.

ACKNOWLEDGEMENTS

This research was supported by AFOSR grant #F49620-00-1-0300. We also wish to thank Dr. Daniel Repperger and Jim Berlin at Wright Patterson Air Force Base for their valuable assistance in performing the experiments reported in this paper.

REFERENCES

[1] Q. Richard, "Design, Analysis, and Control of a Low Power Joint for Walking Robot by Phasic Activation of McKibben Muscle," *IEEE Transactions on Robotics and Automation*, Vol. 15, No. 4, pp. 599-604, 1999.

[2] S. D. Prior, P. R. Warner, A. S. White, J. T. Parsons, and R. Gill, "Actuator for Rehabilitation Robot," *Mechatronics*, Vol. 3, No. 3, 1993.

[3] Bridgestone Corp., *Soft Arm ACFAS Robot System*, Tokyo, Japan, 1987.

[4] N. Toshiro and T. Toshihiro, "Application of Rubber Artificial Muscle Manipulator as a Rehabilitation Robot," *IEEE/ASME Transactions on Mechatronics*, Vol. 2, No. 4, pp. 259-267, 1997.

[5] J. Lilly, "Adaptive Tracking for Pneumatic Muscle Actuators in Bicep and Tricep Configurations," *IEEE Transactions on Neural Systems and Rehabilitation Engineering* (to appear).

[6] G. C. Darwin, A. Gustavo, and M. Goodwin, "Control of Pneumatic Muscle Actuators," *IEEE Control Systems Magazine*, Vol. 15, No. 1, pp. 40-48, 1995.

[7] A. Gustavo, J. B. Colin, and G. C. Darwin, "Adaptive Position Control of Antagonistic Pneumatic Muscle Actuators," *Proc. IEEE International Conference on Intelligent Robots and Systems*, Pittsburgh, PA, USA, August 1995.

[8] B. Tondu and P. Lopez, "Modeling and Control of McKibben Pneumatic Artificial Muscle Robot Actuators," *IEEE Control Systems Magazine*, Vol. 20, No. 2, pp. 15-38, 2000.

[9] D. W. Repperger, K. R. Johnson, and C. A. Phillips, "VSC Position Tracking System Involving a Large Scale Pneumatic Muscle Actuator," *Proc. IEEE Conference on Decision and Control*, Tampa, FL, USA, December 1998.

[10] M. Hamerlain, "Anthropomorphic Robot Arm Driven by Artificial Muscles using a Variable Structure Control," *Proc. IEEE International Conference on Intelligent Robots and Systems*, Pittsburgh, PA, USA, August 1995.

[11] T. Nagaoka, Y. Konishi, and H. Ishigaki, "Nonlinear Optimal Predictive Control of Rubber Artificial Muscle," *Proc.of SPIE - The International Society for Optical Engineering*, Vol. 2595, pp. 54-61, Machine Tool, In-Line, and Robot Sensors and Controls, Philadelphia, PA, USA, October 1995.

[12] D. W. Repperger, C. A. Phillips, and M. Krier, "Controller Design Involving Gain Scheduling for a Large Scale Pneumatic Muscle Actuator," *Proc. IEEE Conference on Control Applications*, Kohala Coast, HI, USA, August 1999.

[13] T. Hesselroth, K. Sarkar, P. Van der Smagt, and K. Schulten, "Neural Network Control of a Pneumatic Robot Arm," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 24, No. 1, pp. 28-38, 1994.

[14] S. W. Chan, J. Lilly, D. W. Repperger and J. E. Berlin, "Fuzzy PD+I Learning Control for a Pneumatic Muscle," *2003 IEEE International Conference on Fuzzy Systems* (submitted).

[15] P. Carbonell, Z. P. Jiang, and D. W. Repperger, "A Fuzzy Backstepping Controller for a Pneumatic Muscle Actuator System," *Proc. IEEE International Symposium on Intelligent Control*, Mexico City, September 2001.

[16] David E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.

[17] H. P. Schewefel, *Evolution and Optimum Seeking*, John Wiley & Sons, New York, 1995.

[18] L. J. Fogel, *Intelligence Through Simulated Evolution - Forty Years of Evolutionary Programming*, John Wiley & Sons, New York, 1999.

[19] S. J. Kang, C. H. Woo, H. S. Hwang, and K. B. Woo, "Evolutionary Design of Fuzzy Rule Base for Nonlinear System Modeling and Control," *IEEE Transactions on Fuzzy Systems*, Vol. 8, No. 1, pp. 37-45, 2000.

[20] Y. Shi, R. Eberhart, and Y. Chen, "Implementation of Evolutionary Fuzzy System," *IEEE Transactions on Fuzzy Systems*, Vol. 7, No. 2, pp. 109-11, April 1999.

[21] O. Cordon and F. Herrera, "A Two Stage Evolutionary Process for Designing TSK Fuzzy Rule-based Systems," *IEEE Transactions on Systems, Man and Cybernetics: Part B*, Vol. 29, pp. 703-715, Dec.1999.

[22] D. Spiegel and T. Sudkmp, "Employment Locality in the Evolutionary Generation of Fuzzy Rule Base," *IEEE Transactions on Systems, Man and Cybernetics- Part B*, Vol. 32, No. 3, pp. 296-305, 2002.

[23] C. P. Chou and B. Hannaford, "Measurement and Modeling of McKibben Pneumatic Artificial Muscles," *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 1, pp. 90-102, 1996.

[24] C. P. Chou and B. Hannaford, "Static and Dynamic Characteristics of McKibben Pneumatic Artificial Muscles," *Proc. IEEE International Conference on Robotics and Automation*, San Diego, CA, USA, May 1994.

[25] D. B. Reynolds, D. W. Repperger, C. A. Philips, and G. Bandry, "Dynamic Characteristics of Pneumatic Muscle," *Computers in Biology and Medicine* (submitted).

[26] N. Tsagarakis and G. C. Darwin, "Improved Modeling and Assessment of Pneumatic Muscle Actuators," *Proc. IEEE International Conference on Robotics and Automation*, San Francisco, CA, USA, April 2000.

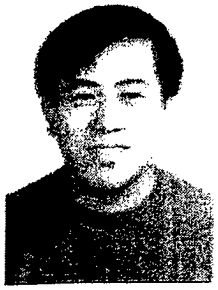
[27] X. Chang and J. H. Lilly, "Neuro-Fuzzy Modeling and Evolutionary Fuzzy Control for Pneumatic Muscles", *IEEE Transactions on Fuzzy Systems* (submitted).

[28] J. S. Branson and J. H. Lilly, "Obtaining Fuzzy Systems from Data – the VISIT Algorithm," IEEE Southeastern Symposium on System Theory, Auburn AL, March 1999.

[29] X. Chang, "Fuzzy P+ID Controller and Behavior Modeling for Complex Systems," Ph.D. Dissertation, Sept. 2000.

[30] W. Li, X. Chang, Jay Farrell and F. Wahl, "Design of an Enhanced Hybrid Fuzzy P+ID Controller for a Mechanical Manipulator," *IEEE Transactions on Systems, Man and Cybernetics – Part B* (to appear).

ABOUT THE AUTHORS



Xiaoguang Chang received the Ph.D degree in computer science from Tsinghua University, P. R. China in 2001. Now, he is a research associate at the Department of Electrical and Computer Engineering, University of Louisville, Louisville, KY. His main research interests include control, modeling using soft computing techniques, pattern recognition and robotics.

Dr. John H. Lilly is with the Electrical and Computer Engineering Department at the University of Louisville. His areas of research are adaptive identification and control, fuzzy control, adaptive noise cancellation, and control of large-scale and distributed parameter systems. He recently completed a sabbatical in the Human Sensory Feedback Laboratory at Wright-Patterson Air Force Base in Dayton, Ohio investigating various control methods, including several fuzzy controllers, for pneumatic muscles and implemented them on the Pneumatic Muscle Test Station in the HSF Lab. He is a reviewer for several IEEE Transactions and several other related journals. Dr. Lilly is a member of Eta Kappa Nu and Sigma Xi. He received the UL President's Young Investigator Award for his research. He a senior member of the IEEE and is a registered professional engineer.

Adaptive Tracking for Pneumatic Muscle Actuators in Bicep and Tricep Configurations

John H. Lilly, *Senior Member, IEEE*

***Abstract*—Adaptive tracking techniques are applied to pneumatic muscle actuators arranged in bicep and tricep configurations. The control objective is to force the joint angle to track a specified reference path. Mathematical models are derived for the bicep and tricep configurations. The models are nonlinear and in general time-varying, making adaptive control desirable. Stability results are derived, and the results of simulation studies are presented, contrasting the nonlinear adaptive control to a nonadaptive PID control approach.**

***Index Terms*—Exoskeletons, nonlinear adaptive control, pneumatic muscle actuators (PMAs).**

I. Introduction

BRAIDED pneumatic muscle actuators (PMAs) possess all the advantages of traditional pneumatic actuators (i.e. cheapness, quickness of response, high power/weight and power/volume ratios) without the main drawback (i.e. compliance or sponginess). For this reason, they are finding increased use in robotic systems. PMA technology is currently under study for use in exoskeleton suits to be worn by humans for force and/or mobility assistance. A difficulty inherent in PM technology for use in precision and/or force applications is the difficulty in controlling them precisely. This is because they are nonlinear and time varying (i.e., since they are made of flexible rubber or plastic, their characteristics vary with temperature and PM temperature varies with use). For this reason, adaptive control approaches are currently under investigation for precise control of PMs.

A PMA consists of a cylindrical, flexible rubber or plastic airtight tube inside a braided plastic sheath (see Fig. 1). When the tube is inflated, it widens and due to the braided sheath, the entire assembly shortens. The force exerted when the PM shortens is quite large in proportion to the PM's weight. PMs have the highest power/weight ratio (1 kW/kg [1]) and power/volume ratio (1 W/cm^3 [2]) of any actuator. PMs have been used for years in robotics to perform manipulation or precision tasks [3]–[10]. Another advantage of PM actuators is the ability to make them autonomous. They are extremely lightweight and can be made independent from other power sources. They may be energized from a small canister of gas that can rapidly create, from a chemical reaction, large pressures for inflation of the muscle.

Manuscript received May 15, 2001; revised October 15, 2001. This work was supported by the Air Force Office of Scientific Research under Grant F49620-00-1-0300.

The author is with the Department of Electrical and Computer Engineering, University of Louisville, Louisville, KY 40292 USA (e-mail: jlilly@louisville.edu).

Digital Object Identifier 10.1109/TNSRE.2003.816870

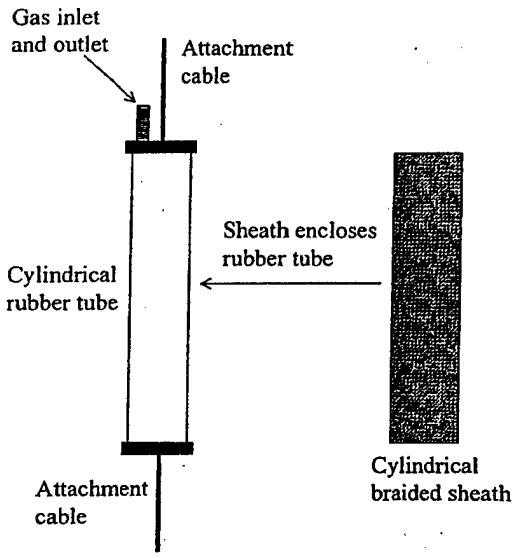


Fig. 1. Construction of pneumatic muscle actuator.

The main drawback of PMs is that they are very nonlinear and time varying, making them difficult to control precisely. This paper studies the closed-loop control of PM systems when used within the context of an accurate position-tracking scenario. The PM system is inherently a passive device and is classified as a "soft actuator" (due to its compliance), which demonstrates the ability to fail gracefully and in a safe manner. Since such systems are being designed to be used in contact with humans, safety is a strong requisite which is fulfilled by the fact that it is both passive and performs as a soft actuator (see [7]).

PM systems can be used to actuate an exoskeleton frame worn by humans to enhance strength and/or mobility assistance for humans. Concepts developed from this research can be used to help the disabled obtain better mobility enhancement. Such people have suffered from stoke, accidents, or other problems to reduce their mobility capability. The main result will be to augment mobility for the veteran via the construction of devices that provide strength or mobility assistance.

There have been several investigations into applications of PMs and their properties. PM research is ongoing at the Human Sensory Feedback (HSF) Laboratory, Wright Patterson Air Force Base [6], [7]. The HSF Lab contains a PM test station that consists of several PMs, sensors, actuators, and instrumentation to control the PM's operation. Related work is underway at the University of Salford, U.K. [1], [9], [10]. This work has the aim of providing a chemo-pneumatic power source to drive PMs. A project involving rehabilitation robotics is centered on a wheelchair-mounted robotic manipulator for

use by the physically disabled and the elderly [11]. Efforts have been directed toward the use of low-cost pneumatic actuators in the robotic manipulator.

The *Autopod* project [12] involves design and testing of a six-legged walking robot. The robot is actuated by PMs because they enable the robot to be low cost, small and lightweight, robust and untethered. The BioRobotics Lab at the University of Washington has several research projects that utilize PMs. The Powered Prosthetics Project addresses the problem of amputee walking via a PM-powered prosthesis. The Anthroform Arm Project seeks to synthesize a robotic arm and controller based completely on known experimental data from human biomechanics and neurophysiology. Both these projects use the McKibben Artificial Muscle [3]. The lab is also concerned with issues such as finite-element modeling [13] and fatigue properties [14] of the muscle.

The Intelligent Robotics Lab at Vanderbilt has developed mobile robots for inspection. The ROBotic Inspector (ROBIN) [15], is intended to be used for inspection of many types of man-made structures including bridges, buildings, ships, and planes. Its motions are powered by PMs. Another system, the Intelligent Soft Arm Control (ISAC) [16] is an intelligent robotic-aid system for the service sector such as hospitals and home. ISAC's main robot arm is called the Soft Arm, which uses PMs in a manner resembling the movements of the human muscle.

This paper is arranged as follows. Section II contains derivations of mathematical models of PMs in bicep and tricep configurations. Section III addresses nonlinear adaptive tracking of limbs actuated by PMs and presents stability results for PM adaptive tracking. Section IV presents simulations of the closed-loop adaptive tracking behavior of limbs with PMs in bicep and tricep configurations. Section V presents a discussion of the results, and Section VI contains conclusions.

II. Dynamic Modeling of Limbs With PM in Bicep and Tricep Positions

The dynamic behavior of PMs has been modeled as a combination of a nonlinear viscous friction and a nonlinear spring [6], [7]. The PM is inflated (hence, shortened) by opening a solenoid which controls the flow of pressurized gas into the rubber bladder. It is deflated by opening another solenoid venting the contents of the bladder to the atmosphere. Both solenoids are linear, i.e. their degree of opening can be controlled precisely for accurate control of muscle length. When inflated, the PM shortens, exerting a force which is equal to the product of pressure inside the bladder and the surface area of the bladder. Since the pressure inside the bladder is always nonnegative, the PM can only exert force in the direction of shortening, never in the lengthening direction. This is to say that, like human skeletal muscles, the PM exerts force when it shortens, but not when it lengthens. The coefficient of viscous friction and spring coefficient, in addition to being nonlinear, also depend on whether the PM is being inflated or deflated. The latter property is because deflation is easier than inflation due to the air pressure inside the bladder.

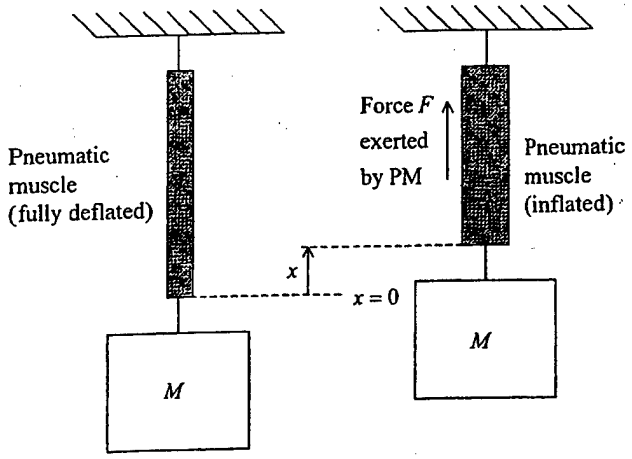


Fig. 2. PM driving a mass.

Fig. 2 shows a pneumatic muscle being inflated and lifting a mass. Let the uninflated (hence, maximum) length of the PM be defined as $x = 0$. If $x(t)$ is the change in length of the PM from its uninflated length (hence, $x(t)$ is nonpositive for all t), the dynamical equation describing the system of Fig. 2 is

$$F + B(\dot{x})\dot{x} + K(x)x = -M\ddot{x} \tag{2.1}$$

where the coefficients $B(\dot{x})$ and $K(x)$ depend on whether the PM is being inflated or deflated and are defined as (see [6] and [7])

$$\text{Inflation}: \begin{cases} B_i(\dot{x}) = 0.04\dot{x}^2 + 1.3\dot{x} + 12.6 \\ K_i(x) = 1.6x^2 + 10.9x + 27.1 \end{cases} \tag{2.2}$$

$$\text{Deflation}: \begin{cases} B_d(\dot{x}) = 0.12\dot{x}^2 + 2.49\dot{x} + 14.48 \\ K_d(x) = 3.6x^2 + 20.7x + 47.23 \end{cases} \tag{2.3}$$

where $x(t)$ is the PM change in length in centimeters.

The left-hand side of (2.1) gives the sum of all external forces acting on the mass, as follows. The system input F is the upward force exerted on the mass by the PM. It is an independent control variable which can be externally commanded by adjusting the inflation and deflation solenoids. The force exerted by the viscous friction action of the PM is given by $+B(\dot{x})\dot{x}$ (since upward motion corresponds to negative $\dot{x}$). The force due to the spring action of the PM is given by $+K(x)x$. The right-hand side of (2.1) gives the force due to the upward acceleration of the mass (upward acceleration corresponds to negative $\ddot{x}$).

The two basic configurations in which the PM can be arranged for use in exoskeletons are the bicep-type (Fig. 3) and tricep-type (Fig. 4) configurations. In this paper, the control problem for both configurations is precise control of the joint angle of a limb which is holding a mass. Specifically, we wish to actuate the PM by inflating and deflating it in such a way that the joint angle follows a reference function of time while the limb holds a mass.

A. PM in Bicep Configuration

Consider the limb configuration shown in Fig. 3, which depicts an arm lifting a mass, with the PM in the position of a

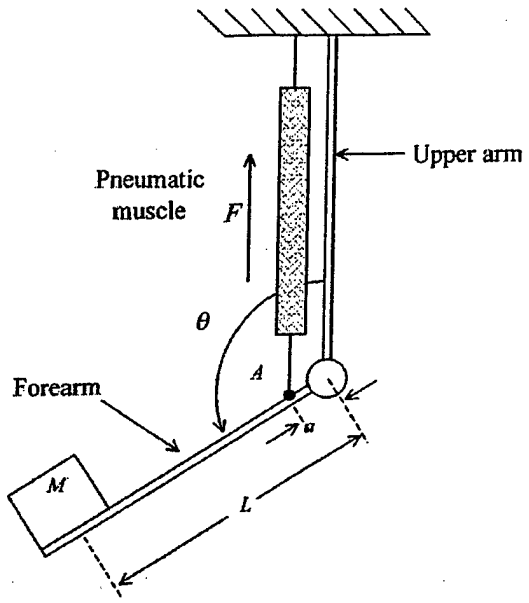


Fig. 3. PM in bicep configuration.

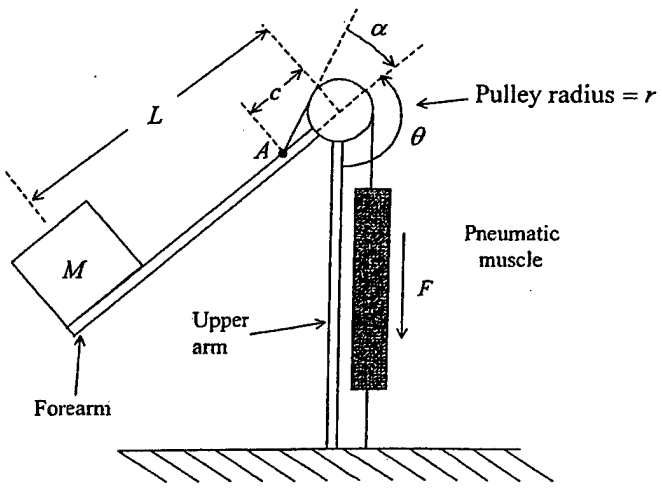


Fig. 4. PM in tricep configuration.

bicep. The upper arm remains stationary as the PM expands and contracts, moving the forearm. The upper end of the PM and upper arm are attached to a motionless reference point. The mass M is held at the end of the forearm. The forearm, which is considered massless, is attached to the upper arm by a frictionless joint. The PM is attached to the forearm at point A, which is a distance a from the joint. The distance from the center of mass of the load to the joint is L. The forearm is free to rotate through an angle θ, where $\theta = 0°$ corresponds to the arm being fully bent, i.e. the mass in the extreme upward position, and $\theta = 180°$ corresponds to the arm being fully straightened, i.e. the mass in the extreme downward position. For simplicity, we will assume the PM force always acts parallel to the forearm. This assumption is valid so long as θ is not close to either of its extremes.

Since the upward force exerted by the PM on the forearm at point A is $F + B(\dot{x})\dot{x} + K(x)x$, the clockwise torque imparted to the forearm by the PM is $(F+B(\dot{x})\dot{x}+K(x)x)a\sin\theta$. Therefore, the system dynamics are described by

$$-I\ddot{\theta} = (F + B(\dot{x})\dot{x} + K(x)x)\, a\sin\theta - MgL\sin\theta \quad (2.4)$$

where $I = ML^2$ is the moment of inertia of the mass about the joint, g is the acceleration of gravity, and $MgL\sin\theta$ is the counterclockwise torque imparted to the forearm by gravity. Then, using $x = -a(1+\cos\theta)$ and $\dot{x} = a\dot{\theta}\sin\theta$, we can rewrite (2.4) entirely in terms of θ as follows:

$$I\ddot{\theta} = -Ba^2\dot{\theta}\sin^2\theta + Ka^2\sin\theta(1+\cos\theta) + MgL\sin\theta - Fa\sin\theta \quad (2.5)$$

where B and K are now expressed in terms of θ and $\dot{\theta}$.

The external input to the system is F, which is determined by how much the PM is inflated. Note that since F is multiplied by $\sin\theta$ in (2.5), the system becomes uncontrollable at $\theta = 0°$ and at $\theta = 180°$. For this reason, joint angles should not approach these limits. We will see that the tricep configuration does not have this restriction.

B. PM in Tricep Configuration

Fig. 4 depicts an arm lifting a mass with the PM in the position of a tricep. The upper arm remains stationary as the PM expands and contracts, moving the forearm. The lower end of the PM and upper arm are attached to a motionless reference point. The mass M is held at the end of the forearm. The forearm, which is considered massless, is attached to the upper arm by a frictionless joint. Also at the joint is a frictionless pulley of radius r, over which a cable connecting the PM to the forearm passes. The PM is attached to the forearm at point A, which is a distance c from the joint. The cable makes an angle $\alpha = \sin^{-1}(r/c)$ with the forearm. The distance from the center of mass of the load to the joint is L. The forearm is free to rotate through an angle θ, where $\theta = 0°$ corresponds to the arm being fully straightened, i.e. the mass in the extreme upward position, and $\theta = 180°$ corresponds to the arm being fully bent, i.e. the mass in the extreme downward position.

Since the downward force exerted by the PM is $F + B(\dot{x})\dot{x} + K(x)x$, the clockwise torque imparted to the forearm by the PM is $(F+B(\dot{x})\dot{x}+K(x)x)c\sin\alpha$. Therefore, the system dynamics are described by

$$\begin{aligned} -I\ddot{\theta} &= (F + B(\dot{x})\dot{x} + K(x)x)\, c\sin\alpha - MgL\sin\theta \\ &= (F + B(\dot{x})\dot{x} + K(x)x)\, r - MgL\sin\theta \end{aligned} \quad (2.6)$$

where $I = ML^2$ is the moment of inertia of the mass about the joint, g is the acceleration of gravity, and $MgL\sin\theta$ is the counterclockwise torque imparted to the forearm by gravity. Then, using $x = -(1/2)\pi r(1+\cos\theta)$ and $\dot{x} = (1/2)\pi r\dot{\theta}\sin\theta$, we can rewrite (2.6) entirely in terms of θ as follows:

$$I\ddot{\theta} = -\frac{B\pi r^2}{2}\dot{\theta}\sin\theta + \frac{K\pi r^2}{2}(1+\cos\theta) + MgL\sin\theta - rF \quad (2.7)$$

where B and K are now expressed in terms of θ and $\dot{\theta}$. Note that the system with PM in tricep position is controllable for all θ because the force exerted by the PM always acts at an angle α to the forearm regardless of joint angle.

III. Adaptive Tracking for Limbs With PM in Bicep and Tricep Position

The mass M manipulated by the PM can be expected to vary significantly from use to use. In addition, the coefficients B and K will vary with PM temperature, and from unit to unit. Also, the physical distances r, a, and L may vary from unit to unit. Therefore, the bicep (2.5) and tricep (2.7) models are poorly known and time-varying, making nonadaptive control methods vulnerable to failure when used for tracking performance of the PM. Since the nonlinear functions of θ are known in (2.5) and (2.7) and only their coefficients are uncertain, we utilize a method of nonlinear adaptive tracking based on sliding control [19], [21]. It uses a well-known result from model reference adaptive control, which we give without proof (see, e.g., [19]).

Lemma: Consider two signals e and ϕ related by the following dynamic equation:

$$e(t) = H(p)\left[k\phi^T(t)v(t)\right] \tag{3.1}$$

where $e(t)$ is a scalar output signal, $H(p)$ is a strictly positive real transfer function, k is an unknown constant with a known sign, $\phi(t)$ is a $m \times 1$ vector function of time, and $v(t)$ is a measurable $m \times 1$ vector. If the vector ϕ varies according to

$$\dot{\phi}(t) = -\mathrm{sgn}(k)\gamma e v(t) \tag{3.2}$$

with γ being a positive constant, then $e(t)$ and $\phi(t)$ are globally bounded. Furthermore, if v is bounded, then $e(t) \to 0$ as $t \to \infty$.

A. Bicep Adaptive Tracking

Consider the problem of the arm lifting a mass with PM in bicep position as in Fig. 3. If we substitute $x = -a(1+\cos\theta)$ and $\dot{x} = a\dot{\theta}\sin\theta$ in (2.5), we get an equation in the form

$$h\ddot{\theta}+a_1\dot{\theta}^3\sin^4\theta+a_2\dot{\theta}^2\sin^3\theta+a_3\dot{\theta}\sin^2\theta+a_4\sin\theta(1+\cos\theta)^3 +a_5\sin\theta(1+\cos\theta)^2+a_6\sin\theta(1+\cos\theta)+a_7\sin\theta = -F\sin\theta$$

or

$$h\ddot{\theta} + \sum_{i=1}^{7} a_i f_i(\theta,\dot{\theta}) = -F\sin\theta \tag{3.3}$$

where $h = I/a$, $a_1,\ldots,a_7$ are parameters which depend on the physical properties of the system (i.e. M, a, and L, the coefficients in (2.2), etc.), and $f_1,\ldots,f_7$ are known functions of θ and $\dot{\theta}$.

Assume that $h, a_1,\ldots,a_7$ are unknown and it is desired that the PM angle $\theta(t)$ track a known desired angle $\theta_d(t)$. Define the error $e(t) = \theta(t) - \theta_d(t)$. Also, define the signal

$$y_r(t) = y_d(t) - \lambda_0 e(t) \tag{3.4}$$

where λ_0 is a positive constant and the combined error

$$s = \dot{e} + \lambda_0 e. \tag{3.5}$$

Consider a control F such that

$$-F\sin\theta = \hat{h}\ddot{y}_r - ks + \sum_{i=1}^{7}\hat{a}_i f_i \tag{3.6}$$

where k is a positive constant and $\hat{h}, \hat{a}_1,\ldots,\hat{a}_7$ are estimates of the unknown parameters $h, a_1,\ldots,a_7$. With this control law, we have the following result concerning the stability of asymptotic tracking of the arm with PM in bicep position.

Theorem 1: Consider the PM in bicep position moving a mass (Fig. 3). Assume the PM spring and viscous friction coefficients are as in (2.2). If the force F delivered by the PM satisfies (3.6), then all signals of the adaptive system are bounded with $\lim_{t\to\infty} e(t) = 0$ provided the parameter estimates are adjusted according to

$$\dot{\hat{h}} = -\gamma s\ddot{y}_r \tag{3.7a}$$

$$\dot{\hat{a}}_i = -\gamma s f_i, \quad i = 1,\ldots,7 \tag{3.7b}$$

where γ is a positive constant.

Proof: It can be shown that the tracking error from control law (3.6) is

$$h\dot{s} + ks = \tilde{h}\ddot{y}_r + \sum_{i=1}^{7}\tilde{a}_i f_i \tag{3.8}$$

where $\tilde{h} = \hat{h} - h$ and $\tilde{a}_i = \hat{a}_i - a_i$. This can be rewritten as

$$s = \frac{\frac{1}{h}}{p+\frac{k}{h}}\left[\tilde{h}\ddot{y}_r + \sum_{i=1}^{7}\tilde{a}_i f_i\right]. \tag{3.9}$$

This is in the form of (3.1) with the transfer function obviously being strictly positive real. Therefore, we have from the lemma that all signals of the adaptive system are bounded.

Consider the Lyapunov function candidate

$$V = hs^2 + \gamma^{-1}\left[\tilde{h}^2 + \sum_{i=1}^{7}\tilde{a}_i^2\right] \tag{3.10}$$

where $\tilde{h} = \hat{h} - h$ and $\tilde{a}_i = \hat{a}_i - a_i$. It is straightforward to show that the derivative of V along the trajectories of the closed-loop system is given by

$$\dot{V} = -2ks^2. \tag{3.11}$$

Therefore, we have $s \to 0$ as $t \to \infty$. It follows that $\lim_{t\to\infty} e(t) = 0$. □

B. Tricep Adaptive Tracking

Consider the problem of the arm moving a mass with PM in tricep position as in Fig. 4. If we substitute $x = -(1/2)\pi r(1+\cos\theta)$ and $\dot{x} = (1/2)\pi r\dot{\theta}\sin\theta$ in (2.7), we get an equation in the form

$$h\ddot{\theta} + a_1(\dot{\theta}\sin\theta)^3 + a_2(\dot{\theta}\sin\theta)^2 + a_3\dot{\theta}\sin\theta + a_4(1+\cos\theta)^3 +a_5(1+\cos\theta)^2 + a_6(1+\cos\theta) + a_7\sin\theta = -F \tag{3.12}$$

or

$$h\ddot{\theta} + \sum_{i=1}^{7} a_i f_i(\theta,\dot{\theta}) = -F \tag{3.13}$$

where $h = I/r$, $a_1,\ldots,a_7$ are parameters which depend on the physical properties of the system (i.e. M, a, L, the coefficients in (2.2), etc.), and $f_1,\ldots,f_7$ are known functions of θ and $\dot{\theta}$.

Assume that $h, a_1, \ldots, a_7$ are unknown and it is desired that the PM angle $\theta(t)$ track a known desired angle $\theta_d(t)$. Define the quantities $e(t) = \theta(t) - \theta_d(t)$, $y_r(t)$, and s, as above.

Consider a control F such that

$$-F = \hat{h}\ddot{y}_r - ks + \sum_{i=1}^{7} \hat{a}_i f_i \tag{3.14}$$

where k is a positive constant and $\hat{h}, \hat{a}_1, \ldots, \hat{a}_7$ are estimates of the unknown parameters $h, a_1, \ldots, a_7$. Then we have the following result concerning the stability of asymptotic tracking of the arm with PM in tricep position using the control (3.14):

Theorem 2: Consider the PM in tricep position lifting a mass (Fig. 4). Assume the PM spring and viscous friction coefficients are as in (2.2). If the force F delivered by the PM satisfies (3.14), then all signals of the closed-loop system are bounded with $\lim_{t\to\infty} e(t) = 0$ provided the parameter estimates are adjusted according to

$$\dot{\hat{h}} = -\gamma s\ddot{y}_r \tag{3.15a}$$

$$\dot{\hat{a}}_i = -\gamma s f_i, \quad i = 1, \ldots, 7 \tag{3.15b}$$

where γ is a positive constant.

Proof: The proof is similar to that of Theorem 1.

Comment 1: To implement the control laws (3.6) and (3.14), it is necessary to measure θ and $\dot{\theta}$. This should be no problem in PM applications, because these are the joint angle and its rate of change, respectively, and are easily measured.

Comment 2: As stated above, the bicep control F is multiplied by $\sin\theta$, which vanishes as the arm approaches either the vertical-up or vertical-down position. Thus, the arm cannot be controlled in the vicinity of these positions. For this reason, care should be taken to avoid arm angles close to vertical-up or vertical-down for bicep control. Tricep configuration does not have this limitation due to the fact that the force is always applied at an angle α to the forearm, regardless of the joint angle.

Comment 3: The assumption of coefficients (2.2) are necessary so that the plant parameters are constants. This assumption is equivalent to stipulating that the PM is not allowed to deflate. This may be the case if, e.g., the task is to lift a mass. If the PM were inflating and deflating, the PM spring and viscous friction coefficients would switch between (2.2) and (2.3). Therefore, the plant parameters would be time varying and the adaptive tracking problem would be considerably more difficult. The assumptions are technically necessary to prove the theorems, but do not appear to be necessary for asymptotic tracking in some simulations in which both inflation and deflation are involved. Of course, the theorems also apply to the case where only coefficients (2.3) are assumed. In this case, the PM is not allowed to inflate but only deflate, which might be the case when the task is to lower the mass.

IV. Simulation Results

The systems of (2.5) and (2.7) were simulated using a fourth-order Runge–Kutta algorithm with a step size of 0.01 s. The control for the bicep configuration is given by (3.6) and (3.7). The control for the tricep configuration is given by (3.14) and (3.15). The results of these simulations are given in the following.

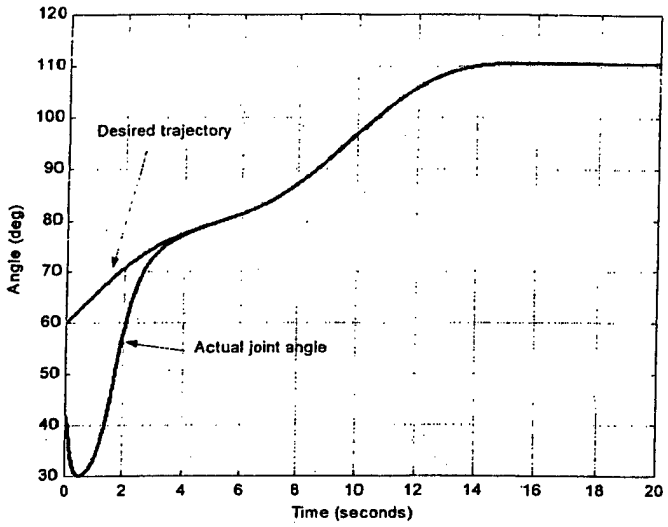


Fig. 5. Bicep adaptive tracking—$\gamma = 3 \times 10^4$ and $\lambda = k = 1$.

A. Bicep Simulation

Assume a configuration as in Fig. 3. Let $L = 0.5$ m, $a = 0.025$ m, $M = 50$ kg, and $g = 9.807$ m/s^2. Since $a = 0.025$ m, the full travel of the forearm from $\theta = 180°$ (arm fully straightened) to $\theta = 0°$ (arm fully bent) corresponds to a maximum change in length of the PM of 0.05 m. This corresponds approximately to the actual capability of the PM.

The desired trajectory for the joint angle between 0 and 15 s is

$$\theta_d(t) = [60° + 62.5°\,(\sin(2\pi f_1 t) + .05\sin(2\pi f_2 t))]\,\frac{\pi}{180} \tag{4.1}$$

with $f_1 = 0.01$ and $f_2 = 0.1$ Hz. Therefore, $\theta_d(t)$ is a sum of two sinusoids with initial condition $\theta_d(0) = 60°$. This trajectory spans monotonically increasing joint angles from 60° to approximately 110° and corresponds to the arm lifting a mass from a lower level to a higher level along the prescribed trajectory.

The input to the PM is given by (3.6) and (3.7) with $\gamma = 3 \times 10^4$, $\lambda_0 = 1$, and $k = 1$. The initial parameter guesses are zero, and the initial joint angle is 42°. The desired and actual joint angles are shown in Fig. 5. It is seen that the arm asymptotically tracks $\theta_d(t)$ after the initial adaptation stage (approximately 4 s).

B. Tricep Simulation

Assume a configuration as in Fig. 4. Let $L = 0.5$ m, $r = (0.05/\pi)$ m, $M = 50$ kg, and $g = 9.087$ m/s^2. Since $r = (0.05/\pi)$ m, the full travel of the forearm from $\theta = 180°$ (arm fully bent) to $\theta = 0°$ (arm fully straightened) corresponds to a maximum change in length of the PM of 0.05 m.

The desired trajectory for the joint angle is again as in (4.1). The input to the PM is given by (3.14) and (3.15) with $\gamma = 1 \times 10^5$, $\lambda_0 = 1$, and $k = 1$. The initial parameter guesses are zero, and the initial joint angle is 42°. The desired and actual

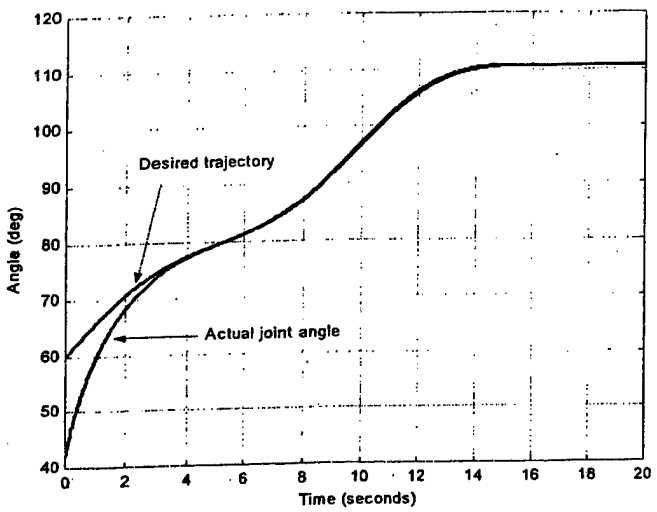


Fig. 6. Tricep adaptive tracking—$\gamma = 1 \times 10^5$ and $\lambda = k = 1$.

joint angles are shown in Fig. 6. Again, we have asymptotic tracking except in the initial adaptation stage.

V. Discussion

The simulations of PM in bicep and tricep configurations have been designed to closely conform to use in PM-actuated exoskeletons. In an exoskeleton, there are no rigid rods for forearm and upper arm, but the exoskeleton may possess some form of rigidity, i.e., a rigid enclosure for a human limb. Exoskeleton PMs are arranged in configurations very similar to human skeletal muscles, i.e. agonist/antagonist or bicep/tricep pairs. The bicep and tricep results in this paper apply to PMs used anywhere in an exoskeleton (arms, legs, etc.), as long as they are arranged in bicep or tricep configurations. An exoskeleton PM in the tricep configuration must have a path over which the PM cable passes to attach to the limb past the joint. This track has been modeled as a frictionless pulley in this study.

In typical exoskeleton applications, the mass actuated by the PM, or the moment of inertia of the moving joint, will vary significantly due to changing joint angles. For instance, when moving a mass from one point to another, the arm may bend, changing the load to the PM, which nevertheless must actuate the limb to follow a desired reference trajectory. This situation arises in robotics as well. Also, the nonlinear spring and nonlinear viscous damping coefficients are poorly known and change with time. This is because with use, the PM heats up, changing these coefficients. In addition, physical properties of the exoskeleton, i.e., arm lengths, distances to attach points, etc., may be poorly known.

Therefore, adaptive control methods have been applied to this problem, since fixed controllers are less robust to parameter changes than adaptive ones. The simulations in Section IV were also carried out with a PID controller designed to give good performance with $M = 50$ kg. If the mass remains in the vicinity of this value, the PID gives good results. However, if the mass changes significantly, the fixed PID cannot stabilize the system. With the adaptive controllers given in Section III, M can undergo a threefold change while retaining adequate tracking. However, the fixed PID is much less tolerant to changes in M, failing to stabilize the system for $M > 80$ kg.

Finally, we note that in real applications of PMs, they will most probably be arranged in agonist/antagonist pairs, as in [4]. Therefore, there will be a bicep/tricep pair rather than a single bicep or tricep acting alone. This would increase joint impedance and result in a more stable joint angle control problem. The present paper is intended to study the action of individual muscles only, without introducing agonist/antagonist interaction.

VI. Conclusion

Dynamic models for pneumatic muscles in bicep and tricep configurations actuating a mass have been derived. These configurations are very similar to exoskeleton applications in which PM's are used to increase strength and mobility in humans. The models are second-order and nonlinear in the joint angle. Their form makes them amenable to nonlinear adaptive control techniques, since the nonlinear functions of the joint angle are known, with only physical constants of the system being unknown. Simulations of closed-loop adaptive tracking of limbs moving masses show that adaptive control techniques are superior to fixed methods, i.e., fixed PID controllers.

References

[1] D. G. Caldwell, G. A. Medrano-Cerda, and M. Goodwin, "Control of pneumatic muscle actuators," *IEEE Control Syst. Mag.*, pp. 40–48, Feb. 1995.

[2] C.-P. Chou and B. Hannaford, "Static and dynamic characteristics of McKibben pneumatic artificial muscles," in *Proc. 1994 IEEE Robotics and Automation Conf.*, May 8–13, 1994, pp. 281–286.

[3] H. F. Schulte, Jr., "The characteristics of the McKibben artificial muscle," in *The Application of External Power in Prosthetics and Orthotics*. Washington, DC: Nat. Acad. Sci., Nat. Res. Council, 1961.

[4] K. Inoue, "Rubber actuators and applications for robotics," in *Robotics Research: The 4th International Symposium*, R. Bolles and B. Roth, Eds. Cambridge, MA: MIT Press, 1988.

[5] B. Hannaford and J. M. Winters, "Actuator properties and movement control: biological and technological muscles," in *Multiple Muscle Systems*, J. Winters and S. Woo, Eds. New York: Springer-Verlag, 1990.

[6] D. W. Repperger, C. A. Phillips, D. C. Johnson, R. D. Harmon, and K. Johnson, "A study of pneumatic muscle technology for possible assistance in mobility," in *Proc. 19th Annu. Int. Conf. IEEE Engineering in Medicine and Biology Soc.*, Chicago, IL, Nov. 1997, pp. 1884–1887.

[7] D. W. Repperger, K. R. Johnson, and C. A. Phillips, "Nonlinear feedback controller design of a pneumatic muscle actuator system," in *Proc. 1999 American Control Conf.*, San Diego, CA, June 1999.

[8] T. Noritsugu and T. Tanaka, "Application of rubber artificial muscle manipulator as a rehabilitation robot," *IEEE/ASME Trans. Mechatron.*, vol. 2, pp. 259–267, Dec. 1997.

[9] D. G. Caldwell, G. A. Medrano-Cerda, and M. Goodwin, "Braided pneumatic muscle actuator control of a multi-jointed manipulator," in *IEEE SMC Conf.*, vol. 1, Le Touquet, France, 1993.

[10] D. G. Caldwell, A. Razak, and M. J. Goodwin, "Braided pneumatic muscle actuators," in *Proc. IFAC Conf. Intelligent Autonomous Vehicles*, Southampton, U.K., Apr. 1993.

[11] S. D. Prior, P. R. Warner, A. S. White, J. T. Parsons, and R. Gill, "Actuators for rehabilitation robots," *Mechatron.—Special Issue Robot Actuators*, vol. 3, no. 3, pp. 285–294, June 1993.

[12] J. H. Cocatre-Zilgien, F. Delcomyn, and J. M. Hart, "Performance of a muscle-like 'leaky' pneumatic actuator powered by modulated air pulses," *J. Robot. Syst.*, vol. 13, no. 6, pp. 379–390, 1996.

[13] G. J. Klute, J. M. Czerniecki, and B. Hannaford, "McKibben artificial muscles: pneumatic actuators with biomechanical intelligence," in *Proc. IEEE/ASME 1999 Int. Conf. Advanced Intelligent Mechatronics (AIM '99)*, Atlanta, GA, Sept. 1999, pp. 221–226.

[14] G. J. Klute and B. Hannaford, "Fatigue characteristics of McKibben artificial muscle actuators," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Victoria, BC, Canada, Oct. 1998, pp. 1776–1781.

[15] R. T. Pack, J. L. Christopher, Jr., and K. Kawamura, "A rubbertuator-based structure-climbing inspection robot," in *Proc. IEEE 1997 Int. Conf. Robotics and Automation*, Albuquerque, NM, Apr. 1997, pp. 1869–1874.

[16] M. Bishay, M. E. Cambron, K. Negishi, R. A. Peters II, and K. Kawamura, "Visual servoing in ISAC, a decentralized robot system for feeding the disabled," in *Proc. 1995 IEEE Int. Symp. Computer Vision*, Nov. 1995, pp. 335–340.

[17] C. M. Close and D. K. Frederick, *Modeling and Analysis of Dynamic Systems*. Boston, MA: Houghton Mifflin, 1978.

[18] S. Sastry and M. Bodson. (1989) *Adaptive Control: Stability, Convergence, and Robustness*

[19] J.-J. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[20] C. Byrnes and A. Isidori, "Applications to stabilization and adaptive control," in *Proc. 23rd IEEE Conf. Decision and Control*, Las Vegas, NV, 1984.

[21] J.-J. Slotine and J. A. Coetsee, "Adaptive sliding controller synthesis for nonlinear systems," *Int. J. Control*, vol. 43, no. 6, pp. 1631–1651, 1986.

[22] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

John H. Lilly (S'81–M'82–SM'89) is with the Electrical and Computer Engineering Department, University of Louisville, Louisville, KY. He recently completed a sabbatical in the Human Sensory Feedback Laboratory, Wright-Patterson Air Force Base, Dayton, OH, where he investigated various control methods, including several fuzzy controllers, for pneumatic muscles and implemented them on the pneumatic muscle test station in the HSF lab. He is a reviewer for several IEEE Transactions and several other related journals. His research interests include adaptive identification and control, fuzzy control, adaptive noise cancellation, and control of large-scale and distributed parameter systems.

Dr. Lilly is a member of Eta Kappa Nu and Sigma XI. He received the UL President's Young Investigator Award for his research. He is a registered professional engineer.