

**SEPTEMBER 2013**

**M.Sc. in Mechanical Engineering**

**ÜNAL HAYTA**

**UNIVERSITY OF GAZİANTEP  
GRADUATE SCHOOL OF  
NATURAL & APPLIED SCIENCES**

**APPLICATION OF SOFT COMPUTING TECHNIQUES  
TO CONTROL OF A FLEXIBLE MECHANICAL SYSTEM**

**M. Sc. THESIS  
in  
MECHANICAL ENGINEERING**

**BY  
ÜNAL HAYTA  
SEPTEMBER 2013**

**Application of Soft Computing Techniques  
To Control of a Flexible Mechanical System**

**M. Sc. Thesis**

**in**

**Mechanical Engineering**

**University of Gaziantep**

**Supervisor**

**Prof. Dr. Sadettin KAPUCU**

**By**

**Ünal HAYTA**

**September 2013**

© 2013 [Ünal HAYTA]

REPUBLIC OF TURKEY  
UNIVERSITY OF GAZİANTEP  
GRADUATE SCHOOL OF NATURAL & APPLIED SCIENCES  
MECHANICAL ENGINEERING

Name of the thesis: Application of Soft Computing Techniques to Control of a Flexible Mechanical System

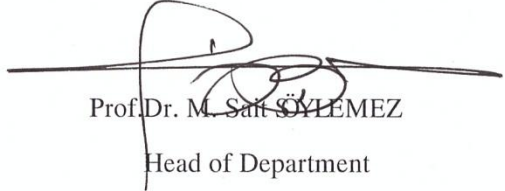
Name of student Ünal HAYTA

Exam Date : 10.09.2013

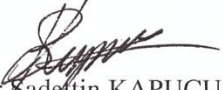
Approval of the Graduate School of Natural and Applied Sciences

  
Assoc. Prof. Dr. Metin BEDİR  
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

  
Prof. Dr. M. Salt ÖYLEMEZ  
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

  
Prof. Dr. Sadettin KAPUCU  
Supervisor

Examining Committee Members Signature

Prof. Dr. Sedat BAYŞİÇ

Prof. Dr. Sadettin KAPUCU

Assoc. Prof. Dr. Abdulkadir ÇEVİK

  
.....  
  
.....  
  
.....

**I hereby that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Ünal HAYTA

Signature

## **ABSTRACT**

### **APPLICATION OF SOFT COMPUTING TECHNIQUES TO CONTROL OF A FLEXIBLE MECHANICAL SYSTEM**

HAYTA, Ünal

M.Sc. in Mechanical Engineering

Supervisor: Prof. Dr. Sadettin KAPUCU

September 2013, 70 pages

Robotic manipulators play an important role in industry. Nowadays most company needs more productivity with less energy consumption. For reasons like these flexible link manipulators in other words flexible link systems are used more widely in several applications. Although flexible systems have several advantages, they have some disadvantages. The most vital disadvantage is residual vibration at the end of motion. This vibration is undesirable in industry. So different type of control schemes have been applied to eliminate or reduce this residual vibration. In the content of this thesis four different control techniques are applied to the single link flexible manipulator. Among these controllers PID controller (see chapter 4) and command shaping controller (see chapter 5) are classical type of controllers and they are designed for comparison of classical controller with soft computing types. The control techniques are proportional-integral-derivative control, command shaping control, fuzzy logic control and neural network control. These four controllers are applied separately to the system and results are discussed. Proposed controllers are designed by considering quick response and small overshoot.

**Key Words:** flexible manipulators, PID control, command shaping control, fuzzy logic control, neural network control

## ÖZET

### BİLİŞSEL HESAPLAMA TEKNİKLERİNİN ELASTİK MEKANİK BİR SİSTEMİN KONTROLÜNDE KULLANILMASI

HAYTA, Ünal

Yüksek Lisans, Makine Müh. Bölümü

Tez Yöneticisi: Prof. Dr. Sadettin KAPUCU

Eylül 2013, 70 sayfa

Robot manipülatörler endüstride önemli roller oynamaktadır. Günümüzde çoğu şirket az enerji tüketimiyle yüksek üretim gücü istemektedirler. Bunu gibi nedenlerden dolayı esnek uzuvlu manipülatörler başka bir deęişle esnek uzuvlu sistemler daha yaygın kullanılmaya başlamıştır. Esnek uzuvlu sistemlerin birçok avantajı olmasına rağmen önemli dezavantajları da bulunmaktadır. Bunlardan en önemlisi de hareket sonundaki artık titreşimlerdir. Bu titreşimler endüstride istenmeyen özelliklerdendir. Bu yüzden deęişik kontrol yöntemleri bu artık titreşimleri yok etmek ya da azaltmak için kullanılır. Bu çalışmada dört farklı kontrol teknięi tek uzuvlu esnek manipülatörün pozisyon kontrolünde kullanılmıştır. Bu kontrol yöntemleri ise şunlardır: PID kontrol, girdi şekillendirme ile kontrol, bulanık mantık ile kontrol ve yapay sinir aęları ile kontrol. Bu kontrolörler sisteme ayrı ayrı uygulanmış ve sonuçları tartışılmıştır. Önerilen kontrol yöntemleri hızlı cevap ve düşük sapma karakteristięi düşünülerek oluşturulmuştur.

**Anahtar Kelimeler:** esnek manipülatörler, PID kontrol, girdi şekillendirme ile kontrol, bulanık mantık ile kontrol, yapay sinir aęları ile kontrol

## **ACKNOWLEDGEMENTS**

I would like to express my sincere gratitude to my supervisor Prof. Dr. Sadettin KAPUCU for the continuous support of my M.Sc. study and research, for his patience, motivation, enthusiasm, and immense knowledge.

I would like to thank to Prof. Dr. Canan DÜLGER, Prof. Dr. İbrahim H. GÜZELBEY, Assoc. Prof. Dr. Abdulkadir ÇEVİK, Asst. Prof. Dr. Engin YEŞİL and Asst. Prof. Dr. Gülay ÖKE for their valuable comments and sharing their knowledge.

The deepest thankfulness is to my parents, for their endless support and interest.



## TABLE OF CONTENTS

	Page
ABSTRACT.....	V
ÖZET.....	VI
ACKNOWLEDGEMENTS.....	VII
CONTENTS.....	VIII
LIST OF TABLES.....	X
LIST OF FIGURES.....	XI
CHAPTER 1	
INTRODUCTION.....	1
1.1 Introduction.....	1
1.2 Structure of Thesis.....	2
CHAPTER 2	
LITERATURE SURVEY .....	3
2.1 Modeling of the System.....	3
2.2 Control of System.....	3
2.2.1 Command Shaping Control.....	3
2.2.2 Fuzzy Logic Control.....	5
2.2.3 Neural Network Control.....	5
CHAPTER 3	
MODELING OF SLFM.....	7
3.1 Introduction.....	7
3.2 Dynamical Equation of Motion of SLFM.....	7
3.2.1 Flexible Manipulator System.....	7
3.2.2 Derivation of Equation of Motion.....	9
3.2.2.1 Derivation of PDE.....	9
3.2.2.2 Mode Summation Method.....	11
3.2.3 State Space and Transfer Function of the Model.....	16

CHAPTER 4	
PID CONTROL OF SLFM.....	19
4.1 Introduction.....	19
4.2 Simulations and Results.....	20
CHAPTER 5	
COMMAND SHAPING CONTROL OF SLFM.....	26
5.1 Introduction.....	26
5.2 Generation of Shaped Command.....	26
5.3 Simulations and Results.....	31
CHAPTER 6	
FUZZY LOGIC CONTROL OF SLFM.....	36
6.1 Introduction.....	36
6.2 Fuzzy Logic Control.....	37
6.3 Simulations and Results.....	39
CHAPTER 7	
NEURAL NETWORK CONTROL OF SLFM .....	47
7.1 Introduction.....	47
7.2 Neural Network Control.....	49
7.3 Simulations and Results.....	51
CHAPTER 8	
CONCLUSION AND DISCUSSION .....	55
8.1 Conclusion and Discussion.....	55
8.2 Recommendations for the Future Works.....	59
REFERENCES.....	61
APPENDICES.....	65

## LIST OF TABLES

	Page
Table 3.1. Values of system characteristics.....	17
Table 6.1. ‘If... then....’ relations for controllers.....	38
Table 6.2. Values which are will be used in simulations.....	41

## LIST OF FIGURES

	Page
Figure 3.1. SLFM model.....	8
Figure 3.2. Euler-Bernoulli beam model.....	9
Figure 3.3. Cross-section of beam.....	10
Figure 3.4. Cantilever Euler-Bernoulli beam.....	12
Figure 3.5. First three mode shapes.....	14
Figure 4.1. PID controller block diagram representation.....	20
Figure 4.2. State space of SLFM with step input function.....	20
Figure 4.3. Result of step forcing function.....	21
Figure 4.4. Block diagram of the PID control of SLFM.....	21
Figure 4.5. Response of system at $K_p = 1$ .....	22
Figure 4.6. Response of system at $K_p = 10$ .....	22
Figure 4.7. Response of system at $K_p = 100$ .....	23
Figure 4.8. Response of system at $K_p = 100$ and $K_d = 1$ .....	24
Figure 4.9. Response of system at $K_p = 100$ and $K_d = 10$ .....	24
Figure 4.10. Response of system at $K_p = 100$ , $K_d = 10$ and $K_i = 100$ .....	25
Figure 5.1. Proposed motion.....	28
Figure 5.2. Traveling time is equal to eight times of system period.....	32

Figure 5.3. Traveling time is equal to three times of system period.....	33
Figure 5.4. Traveling time is equal to 1.25 times of system period.....	34
Figure 6.1. Triangular membership function.....	37
Figure 6.2. Output surface.....	39
Figure 6.3. Output membership function.....	39
Figure 6.4. Block diagram of fuzzy logic control of SLFM.....	42
Figure 6.5. Fuzzy logic controller of hub position.....	42
Figure 6.6. Fuzzy logic controller interface.....	43
Figure 6.7. Sugeno type fuzzy logic controller editor.....	43
Figure 6.8. Membership function editor.....	44
Figure 6.9. Rule editor.....	45
Figure 6.10. (a) hub position, (b) tip point oscillation.....	46
Figure 7.1. Simple neuron representation.....	48
Figure 7.2. Block diagram of the controller and plant [34].....	50
Figure 7.3. Tip point deflection of SLFM.....	52
Figure 7.4. Tip point position error.....	52
Figure 7.5. Change of (a) proportional, (b) integral, (c) derivative gains.....	54
Figure 8.1. Response of system at $K_p = 100$ and $K_d = 10$ .....	55
Figure 8.2. Traveling time is equal to three times of system period.....	57
Figure 8.3. (a) hub position, (b) tip point oscillation.....	58
Figure 8.4. Tip point deflection of SLFM.....	59

## CHAPTER 1

### INTRODUCTION

#### 1.1 Introduction

In many industrial applications robotic manipulators have key roles in different operations. Until last decades robotic manipulators are designed to have high stiffness due to get rid of vibrations. Because of this high stiffness, manipulators would become more heavy and bulky. Afterwards it is seen that these types of robotic manipulators were not satisfactory. Because of their heavy design manipulators were consuming high power and operating with a low speed with respect to the operating payload. Also there is dynamical deflection which is the source of residual vibration when the operation is completed. Besides in some industries high speed operation and high accuracy become vital requirement. Especially in space industry applications manipulators are designed as light weight and slender due to low energy and weight requirements. In order to improve industrial productivity and satisfy other necessities flexible mechanical systems (FMSs) undertake a crucial task. Main advantages of FMSs are lower cost, larger work volume, higher operational speed, greater payload-to-manipulator weight ratio, smaller actuators, lower energy consumption, better manoeuvrability, better transportability, and safer operation due to reduced inertia.

Besides these FMSs have important drawbacks like end point accuracy and residual vibration due to flexibility of the manipulator. For these reasons control system should be designed for eliminating vibrations.

In the cause of complex nonlinear partial differential equations for dynamical equation of motion, it is a real challenge to design a controller. For this reason there

are many type of controllers proposed like input/output linearization via state feedback, PD control, adaptive control, neural network control, fuzzy logic control, lead-lag controller, output redefinition, singular perturbation, sliding mode control, stable inversion in the frequency domain, stable inversion in the time domain, pole placement, input shaping, optimal and robust control.

## **1.2 Structure of Thesis**

In this study a single link flexible manipulator (SLFM) is chosen. This system will be mathematically modelled (see chapter 3). Afterwards different types of control techniques will be applied in order to eliminate or at least reduce the residual vibrations which exist at the end of motion and control the position of the arm. In this study main objective is designing controller using soft computing techniques. Four types of techniques will be applied in this thesis. Among these controllers PID controller (see chapter 4) and command shaping controller (see chapter 5) are classical type of controllers and they are designed for comparison of classical controller with soft computing types. From many soft computing controller types fuzzy logic (see chapter 6) and neural network (see chapter 7) controllers are chosen for controlling the system. In last chapter all of controller types, which are applied in this thesis, will be compared.

## CHAPTER 2

### LITERATURE SURVEY

#### 2.1 Modeling of System

There are different type of models for modelling of the system such as the basic spring-mass discrete models, linear Euler-Bernoulli partial differential equation, generalized Newton-Euler algorithms, Lagrangian equations, associated to a Rayleigh-Ritz elastic field decomposition method, finite element decomposition and modal decomposition method [1]. In this thesis linear Euler- Bernoulli PDE model is chosen for modelling the system. Application of mode summation method to this model and using Lagrange's equation, equation of motion of single link flexible manipulator is achieved [2]. In the most of studies about control of FMS, it is seen that this model is very useful and easy to understand [3-7]. For equation of motion, generally, first three natural modes of link is calculated [8].

#### 2.2 Control of Systems

In the following three sections, literature survey of control techniques, which are used in this thesis, are presented.

##### 2.2.1 Command Shaping Control

Mohamed and Tokhi [9] have proposed and compared input shaping, low-pass and band-stop filtered input techniques on the level of vibration reduction at resonance modes, speed of response, robustness, and computational complexity. Cuccio et al. [10] have discussed preshaped input laws using a limited number of acceleration steps of equal duration in order to reduce the residual vibration of point-to-point moving elastic systems. To obtain smother trajectory, the number of steps is increased. Aspinwall [11] has presented a shaping input function method for generating torque trajectories which involves adding harmonics of a sine series.



Meckl and Seering [12] have constructed input force functions from either ramped sinusoids or versine functions. This approach involves tuning of these template functions by adding up its harmonics with coefficients chosen to minimize the energy of the resulting function. However, main problem with these methods is that exact knowledge of the system frequencies is required. K. Alnefaie et al. [13] have considered a triangular velocity profile to minimize the residual vibration of the rotating flexible beam by selecting proper rise time. Residual vibration amplitudes are strongly depended on the ratio of rise time to beam vibration period. Shina and Brennan [14] have analyzed two acceleration shaping methods based on transient response of the system for controlling the residual vibrations of a translating or rotating Euler-Bernoulli cantilever beam. Identical result to the input shaping method has been obtained by their proposed method. Mentioned methods, which are in the case of K. Alnefaie et al. [13] study, require proper motion time selection. T. Önsay, A. Akay [15] proposed multi-switch bang–bang functions that produce a time-optimal motion. Depending on the dynamic model of the system, accurate selection of switching times is needed. A well-known command generation is based on filtering the desired trajectory by a finite impulse filter that obtains the command reference by convolving the desired trajectory with a sequence of impulses [16]. One or the main drawback of this method is to deal with system uncertainties. This problem is solved by using more impulses sequence to become the shaped signal more robust to uncertainties, but this will result in a longer delay in the system response [16-17]. Alici et al. [18] have suggested a robust motion design method which is based on convolution of a cycloid-plus-ramp function with two impulse sequences. The proposed hybrid input shaping technique is insensitive to the uncertainties of the system parameters and does not lengthen the time delay.

### **2.2.2 Fuzzy Logic Control**

Sreenatha et al. [19] have controlled position of the AC servo motor driven flexible link manipulator with fuzzy logic controller. They have compared results with respect to different reference inputs. In their results it is easily seen that there is overshoot when the motion is completed. Subudhi et al. [20] have suggested PD type fuzzy logic controller and presented that their method has better performance than conventional PD and LQR controllers. Also they suggested a neuro-fuzzy control for

payload variability. Renno [21] has provide a method which has a concept of dividing fuzzy logic controller two sub-controllers. One of these controllers for tip point vibration control and the other one is hub position of the link. Each controller produces a torque signal and these torque signals are summed. This summation is used as the control signal. Ranges of each controller are found from inverse dynamics of system. Hasan et al. [22] have used fuzzy logic controller to control single link flexible manipulator. In this study steady state error, settling time, overshoot and rise time are tried to estimate for different reference inputs. They compared the proposed method results with LQR controller results.

Tokhi et al. [23] have used fuzzy logic algorithm for control of flexible link manipulators. Different type of fuzzy logic controller types are applied to system and results are discussed. Jang et al. [24] have explained MATLAB fuzzy logic toolbox. Generally, fuzzy logic algorithm theory and program interface is shown. Also there are some applications and demos for understanding the usage of this toolbox. Nguyen et al. [25] have explained fuzzy logic theory and its control application very briefly. He gives an application of fuzzy logic controller to inverted pendulum. The theory is easily understandable and design procedures are explained clearly.

### **2.2.3 Neural Network Control**

There are different types of neural network control. Some of them are based on imitating of existing controller besides these others are based on creating controller from neural network own structure. In this thesis main focus of use of neural network is creating neural network controller.

Tang et al. [26] have proposed a neural network controller for controlling tip position of a flexible link manipulator. Error is used to construct neural network. For comparative study filtered and unfiltered tracking error are used for designing neural network controller. Talebi et al. [27] have designed neural network controller for tip position tracking of flexible link manipulators based on modified output re-definition. This approach needs prior knowledge about the system. There are four neural network structure proposed by Talebi et al. Two of them are created by using modified feedback-error-learning approach for learning the inverse dynamics of the

system. Third one is based on main criteria for designing the controller is hub position while eliminating tip vibration. The last one consists of two neural networks. First one is responsible for generating appropriate output for minimum phase behavior. The second one is used for implementing inverse dynamics of controller. Su et al. [28] have designed a controller based on inverse dynamics of rigid manipulators. The proposed neural network learns nonlinearities of the flexible link from the inverse dynamics. No prior knowledge is needed for nonlinearities. The adjustment of the network gains are based on the tracking error. Öke et al. [29] have proposed a method which uses neural network to compute incremental changes for the reference values of the joint angle to achieve successful tip tracking. To minimize the tip deflection neural network cost function is used. Joint angle control is achieved by PD controller.

Nguyen et al. [25] have explained neural network for control briefly. There are also some examples like PI control simulation with neural network, instantaneous linearization with neural network. Also Tokhi et al [23] have discussed modular neural network technique.

## CHAPTER 3

### MODELING OF SLFM

#### 3.1 Introduction

In the vibration point of view SLFMs are generally described as continuous systems. So it is beneficial to define continuous and discrete systems. Sometime a system can be thought as discrete and modeled as ordinary differential equations (ODEs), another time the same system can be thought as continuous and modeled by partial differential equations (PDEs). So it is easy to understand that discrete and continuous systems are related. The continuous systems can be modeled as infinite numbers of mass and springs. So there will be infinite numbers of natural frequencies. Besides, the response of discrete systems depends on time and initial condition. In continuous systems vibration depends on position and time. In discrete system ordinary differential equation is converted to matrix algebra with eigenvalue problem. The eigenvalues are natural frequencies of the system. In continuous systems, boundary and initial conditions are important because response depends on coordinate and time. There are two types of boundary conditions:

1. Geometric boundary conditions: Displacement and slope
2. Natural boundary conditions: Force and moment

#### 3.2 Dynamical equation of motion of the system

##### 3.2.1 The Flexible Mechanical System

In this thesis Euler- Bernoulli beam theory is chosen for the modeling single link flexible manipulator. Theory has two important assumptions [30]:

1. The material is linear elastic according to Hooke's law
2. Plane sections remain plane and perpendicular to the neutral axis.

The length of link is assumed as constant and shear deformation effect of axial forces neglected [23].

There are some constants for modeling manipulator such as,  $\omega_n$  natural frequency of the SLFM,  $w$  cross sectional width,  $h$  cross sectional height,  $E$  modulus of elasticity,  $m$  weight per unit length,  $J$  moment of inertia of SLFM.

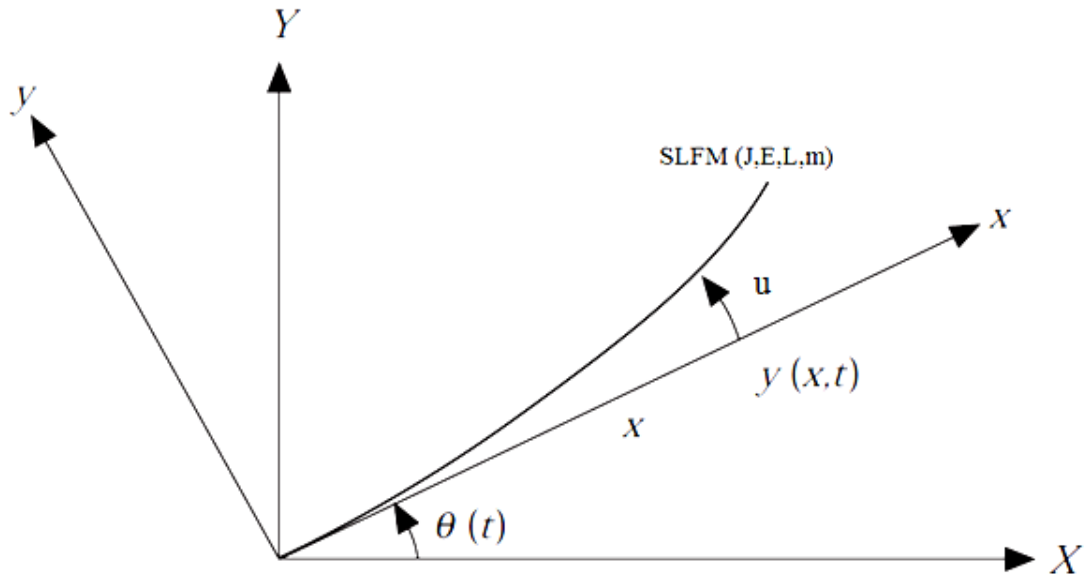


Figure 3.1: SLFM model [4]

For  $\theta$  angular position input,  $u$  is the elastic deformation,  $y(x,t)$  is the total displacement of a point on the manipulator at the distance  $x$  from the hub (base). So  $y(x,t)$  is the function of rigid body motion  $\theta$  and elastic deformation  $u$ .

$$y(x,t) = x\theta(t) + u(x,t) \quad (3.1)$$

In the next section derivation of PDE of SLFM is presented by using Hamilton's principle. With application of mode summation method to PDE equation of motion will be derived.

### 3.2.2 Derivation of Equation of Motion

#### 3.2.2.1 Derivation of PDE

For derivation of PDE of SLFM a link is subjected to purely bending moment as shown in figure 3.2.

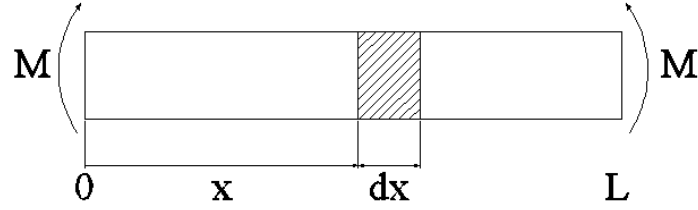


Figure 3.2: Euler-Bernoulli beam model

Considering the infinitesimal segment of the beam model, which has dimension  $dx$  at distance  $x$  from zero, has displacement  $u$ , the kinetic and potential energies becomes respectively:

$$T = \int_0^l \frac{1}{2} m dx \dot{u}^2 = \frac{1}{2} \int_0^l m \dot{u}^2 dx \quad (3.2)$$

$$V = \frac{1}{2} \int_0^l \sigma \varepsilon dv = \frac{1}{2} \int_0^l \frac{\sigma^2}{E} dv \quad (3.3)$$

where  $\dot{u}$  is the time derivative of the elastic displacement,  $\sigma$  is the bending stress,  $\varepsilon$  is strain,  $v$  is the volume.

The cross section of beam is depicted in figure 3.3. The beam cross section is in rectangular shape and uniform along the length. Potential energy of the beam can be written as:

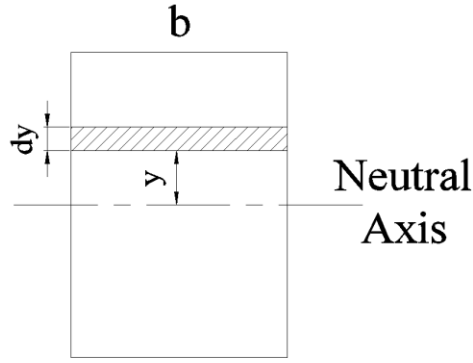


Figure 3.3: Cross-section of beam

$$V = \frac{1}{2} \iint \frac{\sigma^2}{E} b dx dy = \frac{1}{2} \iint \frac{M^2 y^2}{EI^2} b dx dy = \frac{1}{2} \int \frac{M^2}{EI^2} (y^2 dA) dx = \frac{1}{2} \int \frac{M^2}{EI} dx \quad (3.4)$$

where  $I$  is the inertia,  $y$  is the distance from the neutral axis,  $A$  is the cross sectional area.

Note that ratio of normal stress at a section to distance from neutral axis is given as

$$\frac{\sigma}{y} = \frac{M}{I} \quad (3.5)$$

and this is equal to;

$$\frac{\sigma}{y} = \frac{M}{I} = \frac{E}{R} \quad (3.6)$$

where  $R$  is the Radius of curvature. Simplified curvature of a line is given as;

$$\frac{1}{R} = \frac{d^2 u}{dx^2} \quad (3.7)$$

Substituting equations (3.6) and (3.7) into equation (3.4) potential energy becomes;

$$V = \frac{1}{2} \int_0^l EI \left( \frac{d^2 u}{dx^2} \right)^2 dx \quad (3.8)$$

While deriving the equation of motion of elastic bodies, use of Hamilton principle is more effective than the other methods. Hamilton principle is given as;

$$\int_{t_1}^{t_2} \delta L dt = 0 \quad (3.9)$$

where  $L$  is the difference from kinetic energy to potential energy. So it can be written as:

$$L = T - V \quad (3.10)$$

Substituting equations (3.2), (3.8) and (3.10) into equation (3.9) Hamilton principle equation becomes;

$$\int_{t_1}^{t_2} \delta \left( \frac{1}{2} \int_0^l m \left( \frac{du}{dt} \right)^2 dx - \frac{1}{2} \int_0^l EI \left( \frac{d^2u}{dx^2} \right)^2 dx \right) dt = 0 \quad (3.11)$$

The solution of equation (3.11) is;

$$EI \frac{d^4u(x,t)}{dx^4} + m \frac{d^2u(x,t)}{dt^2} = 0 \quad (3.12)$$

This equation is known as Euler-Bernoulli partial differential equation.

### 3.2.2.2 Mode Summation Method

When the mode summation method is used for solution of equation (3.12), displacement  $u(x,t)$  can be written as:

$$u(x,t) = \sum_i \varphi_i(x) q_i(t) \quad (3.13)$$

If equation (3.13) puts into equation (3.12):

$$EI \frac{d^4\varphi(x)}{dx^4} q(t) + m \frac{d^2q(t)}{dt^2} \varphi(x) = 0 \quad (3.14)$$

$$\frac{d^2q(t)}{dt^2} / q(t) = - \frac{EI}{m} \left( \frac{d^4\varphi(x)}{dx^4} / \varphi(x) \right)$$

In the case of simple harmonic motion, ratio of acceleration to displacement is equal to square of natural frequency.



$$\frac{d^2 q(t)}{dt^2} / q(t) = -\omega^2 \longrightarrow \frac{d^2 q(t)}{dt^2} + \omega^2 q(t) = 0 \quad (3.15)$$

If you want to solve equation (3.14), Laplace transform has to be done to equation (3.14). Generalized coordinate  $q_i(t)$  is:

$$q_i(t) = A \cos(\omega_i t) + B \sin(\omega_i t) \quad (3.16)$$

For the normal modes of vibration of such a cantilever beam  $\varphi_i(x)$  second part of equation (3.14) must be solved. In figure 3.4 cantilever Euler-Bernoulli beam is shown [8].

$$\frac{d^4 \varphi(x)}{dx^4} = \frac{m\omega^2}{EI} \varphi(x) \quad (3.17)$$

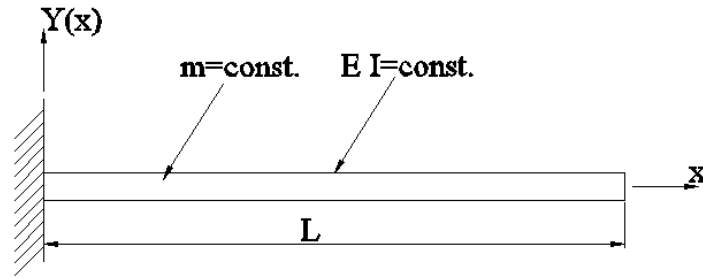


Figure 3.4: Cantilever Euler-Bernoulli beam

Let us take

$$\frac{m\omega^2}{EI} = \beta^4 \longrightarrow \omega = \beta^2 l^2 \sqrt{\frac{EI}{ml^4}} \quad (3.18)$$

So equation (3.17) becomes:

$$\frac{d^4 \varphi(x)}{dx^4} - \beta^4 \varphi(x) = 0 \quad (3.19)$$

In order to solve equation (3.19), Laplace transform should be taken. Result is:

$$\varphi_i(x) = A \sinh(\beta_i x) + B \cosh(\beta_i x) + C \sin(\beta_i x) + D \cos(\beta_i x) \quad (3.20)$$

In the case of cantilever beam, there are four boundary equations.

$$\text{Displacement at } x=0: u(0,t) = 0 \quad (3.21)$$

$$\text{Slope at } x=0: \left. \frac{du(x,t)}{dx} \right|_{x=0} = 0 \quad (3.22)$$

$$\text{Shear force at } x=L: EI \left. \frac{d^3u(x,t)}{dx^3} \right|_{x=L} = 0 \quad (3.23)$$

$$\text{Bending moment at } x=L: EI \left. \frac{d^2u(x,t)}{dx^2} \right|_{x=L} = 0 \quad (3.24)$$

By using these four boundary conditions we can obtain frequency equation:

$$\cosh(\beta L)\cos(\beta L) = -1 \quad (3.25)$$

When we solve this frequency equation numerically we can find infinite number of  $\beta_i$ . It means that SLFM has infinite numbers of natural frequencies.

So natural modes equation becomes:

$$\varphi_i(x) = A_i \left[ (\sin \beta_i L - \sinh \beta_i L)(\sin \beta_i x - \sinh \beta_i x) + (\cos \beta_i L + \cosh \beta_i L)(\cos \beta_i x - \cosh \beta_i x) \right] \quad (3.26)$$

$i=1,2,\dots$

For the first three modes,  $\beta L$  values are:

$$1^{\text{st}} \text{ mode: } \beta L = 1.875$$

$$2^{\text{nd}} \text{ mode: } \beta L = 4.694$$

$$3^{\text{rd}} \text{ mode: } \beta L = 7.855$$

First three natural modes are shown in figure 3.5:

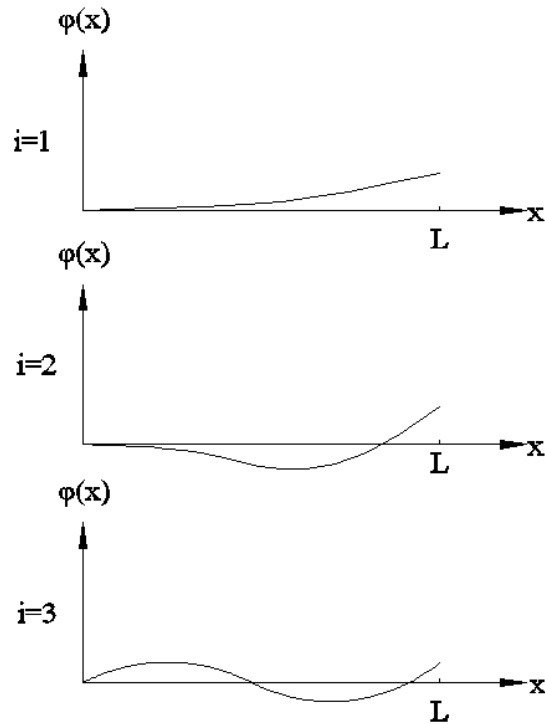


Figure 3.5: First three mode shapes

Also mode summation method can be used for converting partial differential equation to ordinary differential equation [2]. Let us take equation (3.12) and forced it with a torque  $\tau = -mx\ddot{\theta}(t)$ .

$$EI \frac{d^4 u(x,t)}{dx^4} + m \frac{d^2 u(x,t)}{dt^2} = -mx\ddot{\theta}(t) \quad (3.27)$$

If we want to write energy equation in terms of generalized co-ordinate, the kinetic energy is:

$$T = \frac{1}{2} \sum M_i \dot{q}_i^2 \quad (3.28)$$

where  $M_i$  is generalized mass:

$$M_i = \int_0^l \varphi_i(x)^2 m(x) dx \quad (3.29)$$

The potential energy is:

$$U = \frac{1}{2} \sum K_i q_i^2 \quad (3.30)$$

where  $K_i$  generalized stiffness matrix:

$$K_i = \int_0^l EI \left[ \varphi_i''(x) \right]^2 dx \quad (3.31)$$

Work done by applied force for virtual displacement  $\delta q_i$  is:

$$\delta W = \sum_i \delta q_i \int_0^l -m x \ddot{\theta}(t) \varphi(x) dx \quad (3.32)$$

From equation (3.32) generalized force can be easily found:

$$Q_i = -m \ddot{\theta}(t) \int_0^l x \varphi(x) dx \quad (3.33)$$

If we put T, U and Q into Lagrange equation (3.34) and solve it for  $q_i$ , ordinary differential equation (3.35) can be derived.

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_i} \right) - \frac{\partial T}{\partial q_i} + \frac{\partial U}{\partial q_i} = Q_i \quad (3.34)$$

$$\ddot{q}_i(t) + \omega_i^2 q_i(t) = (\Gamma_i / M_i) \ddot{\theta}(t) \quad (3.35)$$

where  $\Gamma_i$  is the mode participation factor for mode  $i$ :

$$\Gamma_i = -m \int_0^l x \varphi_i(x) dx \quad (3.36)$$

and  $M_i$  is generalized mass:

$$M_i = \int_0^l \varphi_i^2(x) m(x) dx \quad (3.37)$$

### 3.2.3 State Space and Transfer Function of the Model

In order to simulate the system, ordinary differential equations must be converted to state space or transfer function. In the simulations first mode of vibration is only taken into account because biggest amplitude of vibration occurs in first mode. The amplitude of the other mode is too small with respect to first mode.

The result of equation (3.35) is only vibration of tip point of SLFM. In order to add rigid body motion the following equations have to be solved at the same time.

$$\ddot{\theta} = \frac{1}{J} \tau(t)$$
$$\ddot{q}_i(t) + 2\delta\omega_i \dot{q}_i(t) + \omega_i^2 q_i(t) = \left( \frac{\Gamma_i}{M_i J} \right) \tau(t)$$

where torque  $\tau(t)$  is used input for both equations. Also damping coefficient  $\delta$  is added into equation (3.35).

In the case of state space, states of system are:

$$\begin{aligned} x_1 &= \theta \\ x_2 &= \dot{\theta} \\ x_3 &= q \\ x_4 &= \dot{q} \end{aligned} \tag{3.38}$$

So state space matrices are:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{1}{J} \tau(t) \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= \frac{\Gamma_1}{M_1 J} \tau(t) - 2\delta\omega_1 x_4 - \omega_1^2 x_3 \end{aligned}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_1^2 & -2\delta\omega_1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ \frac{1}{J} \\ 0 \\ \frac{\Gamma_1}{M_1 J} \end{bmatrix}, C = [L \ 0 \ 1 \ 0], D = [0] \quad (3.39)$$

$$\begin{aligned} \dot{X} &= AX + Bu \\ Y &= CX \end{aligned} \quad (3.40)$$

By substituting equation (3.39) to (3.40), state space form of system can be written:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_1^2 & -2\delta\omega_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{J} \\ 0 \\ \frac{\Gamma_1}{M_1 J} \end{bmatrix} \tau(t) \quad (3.41)$$

$$Y = [L \ 0 \ 1 \ 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

For simulation studies values, which are presented in table 3.1, will be used.

Table 3.1 Values of system characteristics

	Symbol	Value
<b>Length of SLFM</b>	$L(m)$	0.7
<b>Cross-sectional width</b>	$w(m)$	0.002
<b>Cross-sectional height</b>	$h(m)$	0.0255
<b>Modulus Elasticity</b>	$E(GPa)$	71
<b>Mass per unit length</b>	$m(kg/m)$	0.1382
<b>First mode frequency</b>	$\omega_1(rad/s)$	21
<b>Damping coefficient</b>	$\delta$	0.02

Using the equation (3.42) system will be represented by transfer function.

$$G(s) = C(sI - A)^{-1}B \quad (3.42)$$

$$G(s) = \frac{L}{Js^2} + \frac{\Gamma_1}{JM_1(s^2 + 2\delta\omega_1s + \omega_1^2)} \quad (3.43)$$

## CHAPTER 4

### PID CONTROL OF SLFM

#### 4.1 Introduction

PID controller is one of the mostly used controller types. Its popularity comes from its simplicity. Error, which is difference between desired set point and the real output, exists in all systems. Controller's main aim is minimizing this error. PID controller performs this function in three steps. First one is proportional controller action. It is denoted as P. Proportional controller action depends on present error. It takes this present error multiply a constant value  $K_p$ . In this way, steady state error is eliminated. Second one is integral action. Its abbreviation is I and it is depends on accumulation of past errors. In this step integral of error is multiplied by a constant  $K_i$ . Thus integral controller is also eliminates the steady state error more precisely, but it can be lead to decrease stability characteristics. The last controller action is derivative. It is a prediction of future errors. By taking derivative of error and multiplying it with constant  $K_d$  derivative control action is achieved. Derivative control action increases the stability of the system like a damper effect. If it is too big  $K_d$  value, derivative action can cause steady state error. Block diagram representation of controller is shown in figure (4.1).



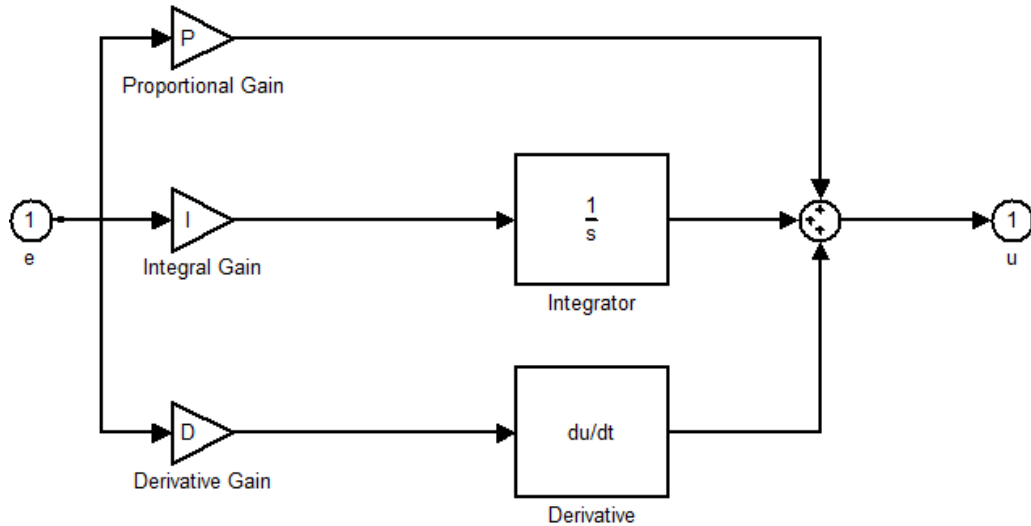


Figure 4.1: PID controller block diagram representation.

In this representation  $e$  is the error,  $u$  is the control signal. Relation between  $e$  and  $u$  is:

$$U(s) = K_p E(s) + \frac{K_p}{T_i s} E(s) + K_p T_d s E(s) = \left( K_p + \frac{K_i}{s} + K_d s \right) E(s) \quad (4.1)$$

For tuning gains Zeigler-Nichols suggested a method [31]. This tuning method gives us an educated guess for the parameter values and provides a starting point for the fine tuning. But in this section gain values will be found by trial and error.

## 4.2 Simulations and Results

In this section position control of SLFM is aimed with PID controller. For the simulation MATLAB, Simulink software will be used.

Firstly it is beneficial to see uncontrolled response of the system. In order to do this, system should be represented by state space Simulink block and this block should be connected step forcing function.

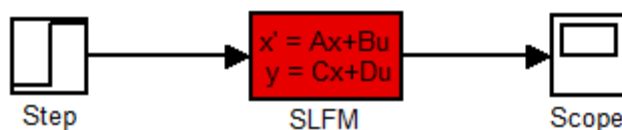


Figure 4.2: State space of SLFM with step input function.

The result is shown in figure (4.3):

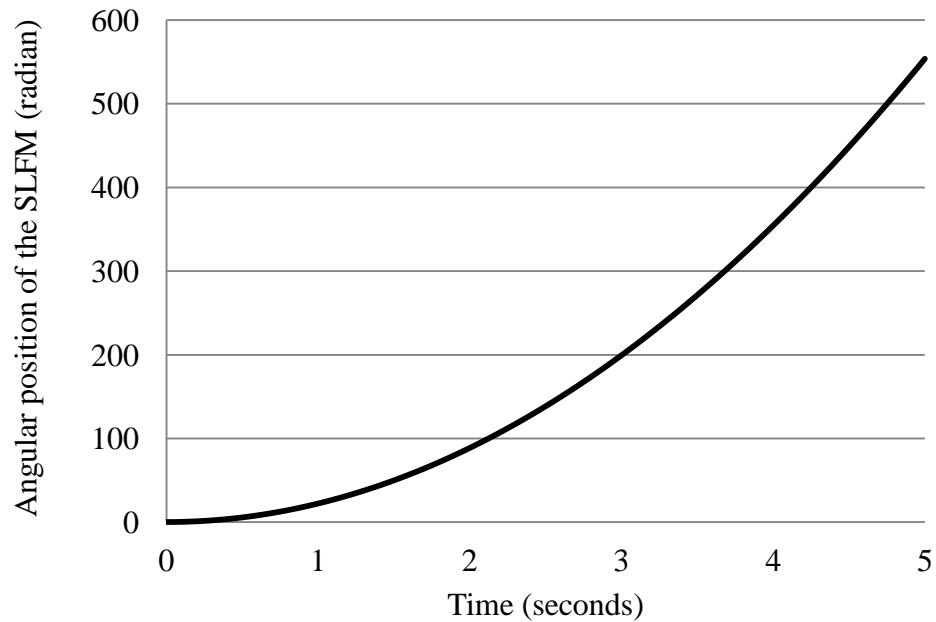


Figure 4.3: Result of step forcing function

It is easily seen that system is uncontrolled. Because of big scale, oscillation tip of the link is not seen from the figure 4.3. So it is necessary to use a controller to this system. From Simulink toolbox proportional-integral-derivative controller block is connected to the plant. Input of the controller is position error and output is control signal which will be used for input of the plant.

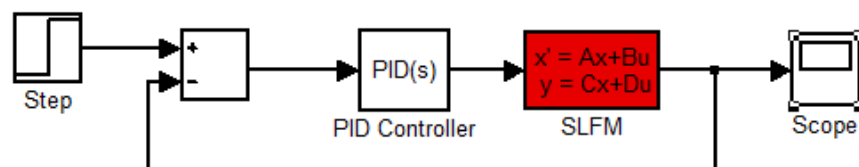


Figure 4.4: Block diagram of the PID control of SLFM

Firstly only proportional controller will be used and its gain will be tuned by trial and error. The other control action gains (derivative-integral gains) will be zero. For the beginning  $K_p$  is taken 1 and result is:

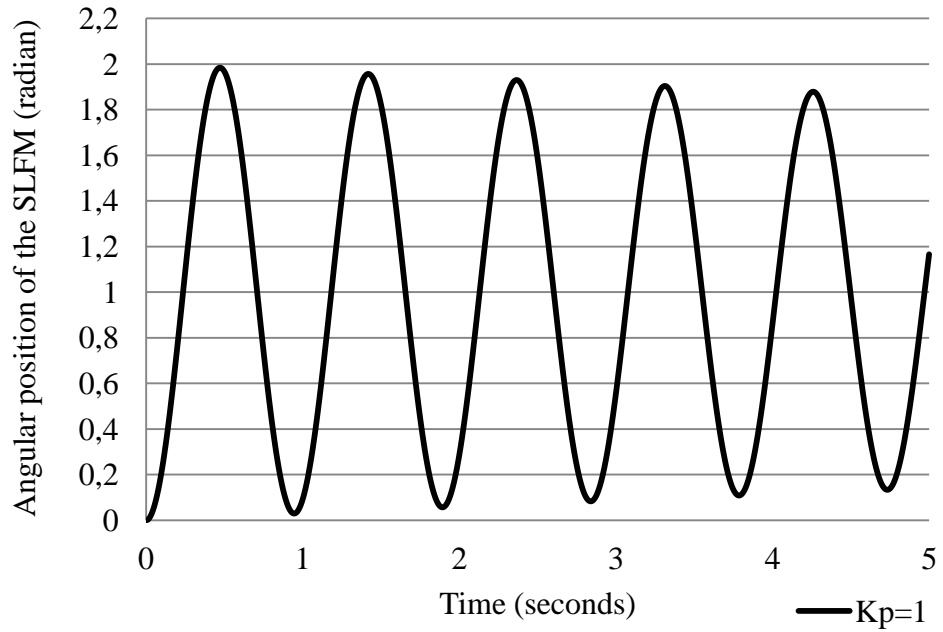


Figure 4.5: Response of system at  $K_p = 1$

It is seen that system output is oscillating around the 1. So  $K_p$  should be increased.

Let's take  $K_p = 10$ :

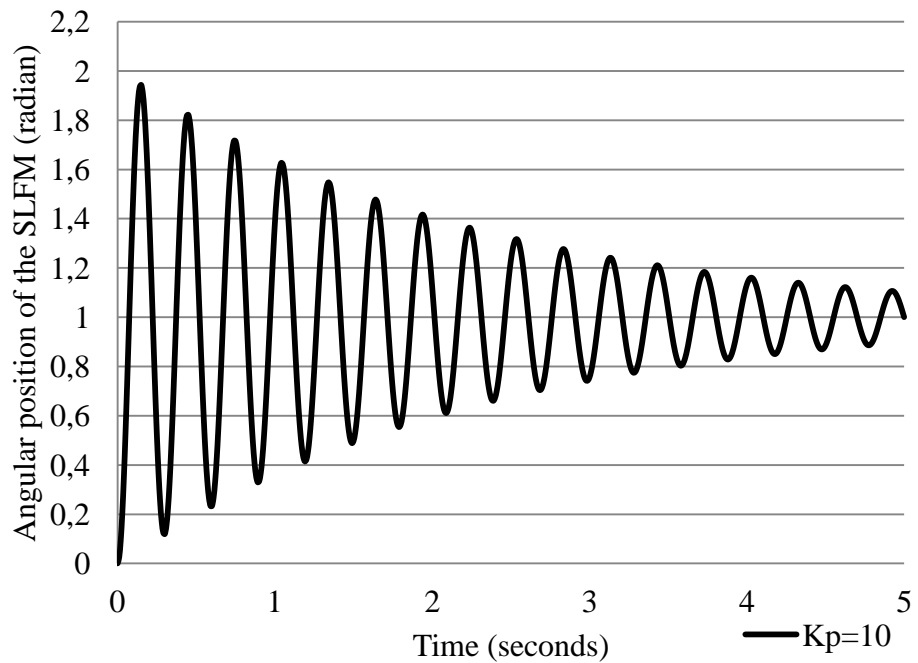


Figure 4.6: Response of system at  $K_p = 10$

Oscillation is still continuing, but it decreases with respect time. At last take  $K_p = 100$ :

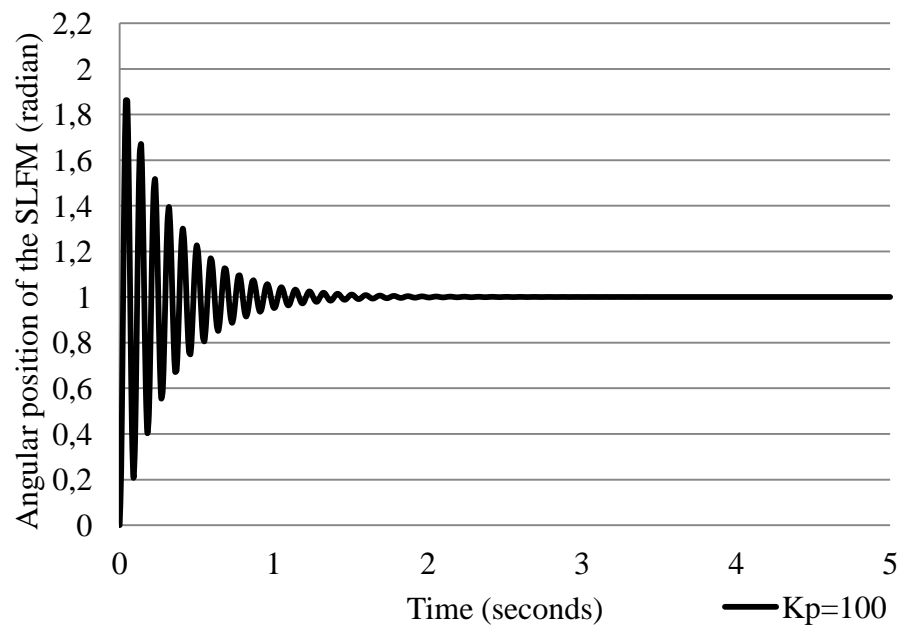


Figure 4.7: Response of system at  $K_p = 100$

It is seen that when proportional gain increases response time decreases. At the same time oscillation amplitude increases. So another control action is needed. This is derivative control action. Derivative control action eliminates the oscillations as previously said. So let's take  $K_d = 1$ :

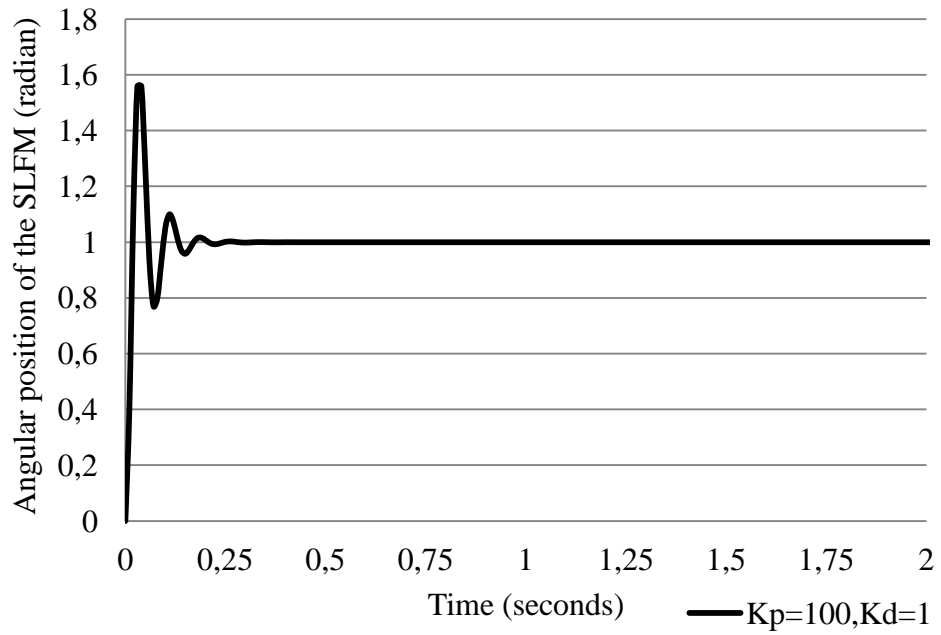


Figure 4.8: Response of system at  $K_p = 100$  and  $K_d = 1$

As a result of adding derivative gain as 1, system's response time is faster. Settling time is approximately is 0.25 seconds. Also peak oscillation value decreases to below of 1.6 radians.

By increasing the derivative gain settling time can decrease. Take  $K_d = 10$ :

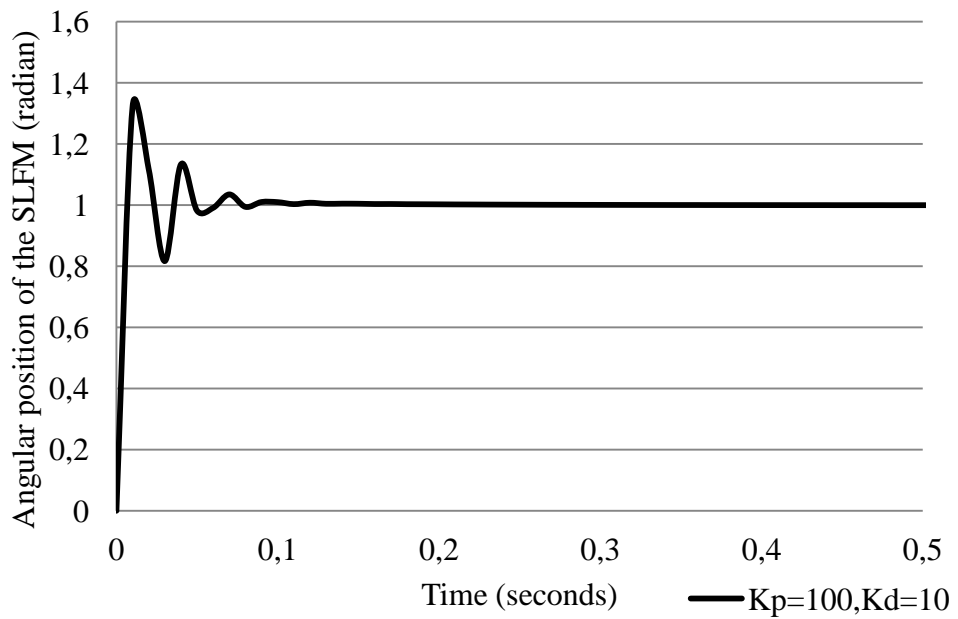


Figure 4.9: Response of system at  $K_p = 100$  and  $K_d = 10$

In this case response time decreased to 0.1 seconds. Also peak value is less than 1.4 radians. This result is sufficient in most cases. So there is no need to add integral gain to the controller. But if you insist on adding integral gain, it will be easily seen that there is no change in response.

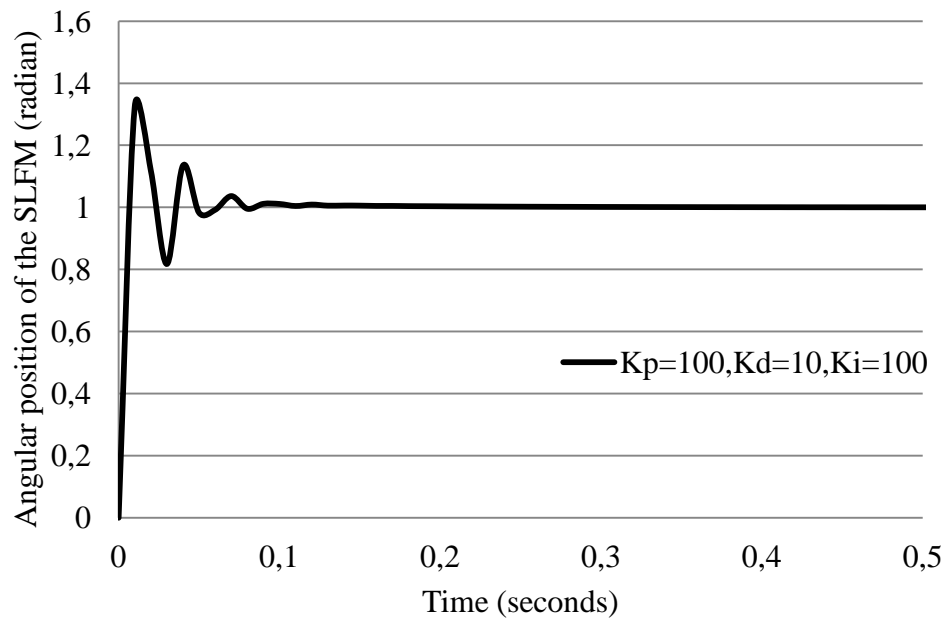


Figure 4.10: Response of system at  $K_p = 100$ ,  $K_d = 10$  and  $K_i = 100$

Adding integral gain has not given any advantage.

As a result for this plant only proportional-derivative controller is sufficient. By taking gains as  $K_p = 100$  and  $K_d = 10$ , response time is 0.1 and peak value is less than 1.4 radians. These controller's results will be compared other controller results in the following sections.

## CHAPTER 5

### COMMAND SHAPING CONTROL OF SLFM

#### 5.1 Introduction

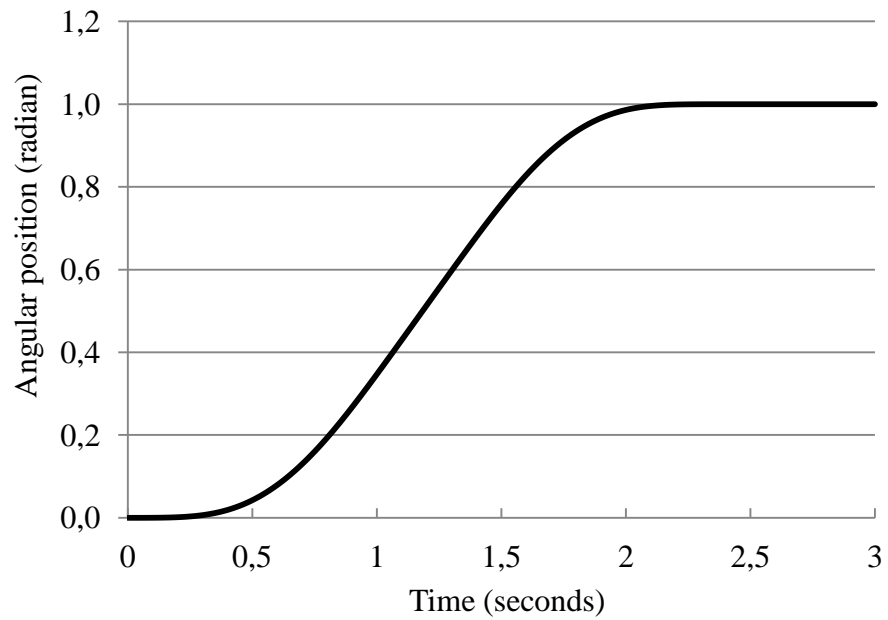
There are two main classes to control flexible link manipulators. These are open-loop control and closed loop control. In the case of open loop control system is controlled by generating a control input by considering characteristics of the system. Accordingly systems most effective vibrations can be eliminated. In open loop systems there is no need of any feedback of any state. So this control system does not require any sensor. In contrast with open loop, close loop controllers uses feedback of states which are required to control. Because of this reason, close loop controllers are also named as feedback controllers. They use measurements and estimations for controlling action. If there is any uncertainty, feedback controller must be used. But in the case of open loop control, there should not be any uncertainty in the system. Because controlled input is designed by using characteristics of system.

As mentioned in chapter two, there are various open loop control strategies like input shaping, Fourier expansion of forcing function, computed torque control and etc. In the next section a four piece acceleration motion profile is proposed. This acceleration profile will be used as input function to the system.

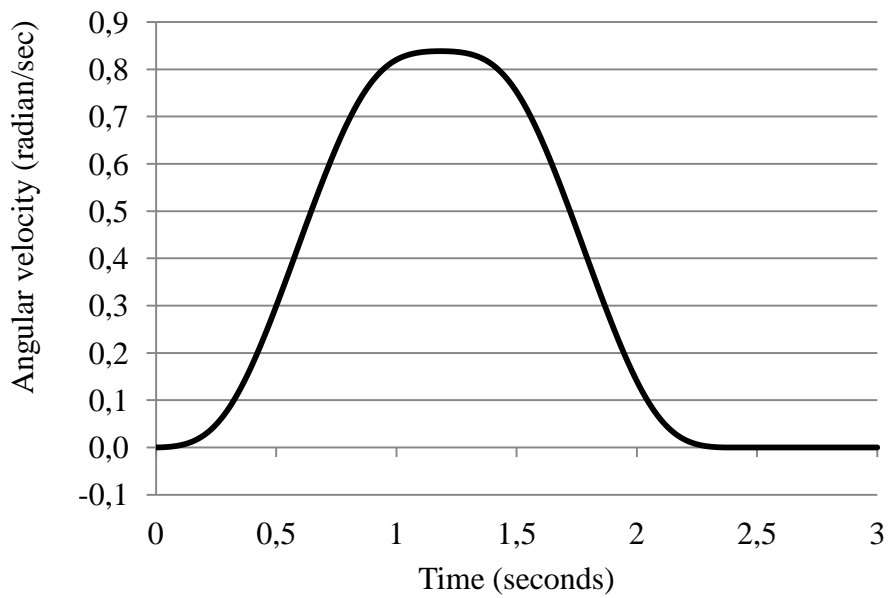
#### 5.2 Generation of Shaped Command

In order to eliminate residual vibration of links which are rotated from point to point, acceleration and deceleration periods are used as motion profile. In this section an acceleration profile is generated by using another study which is presented in reference [32]. The motion profile is formed by superimposing of three functions, a cycloid, a ramped versine and a ramp. By using total displacement, transportation time and system parameters the amplitudes of the three functions are calculable.

A schematic drawing of the proposed profile of displacement, velocity, acceleration and jerk is depicted in figure 5.1.

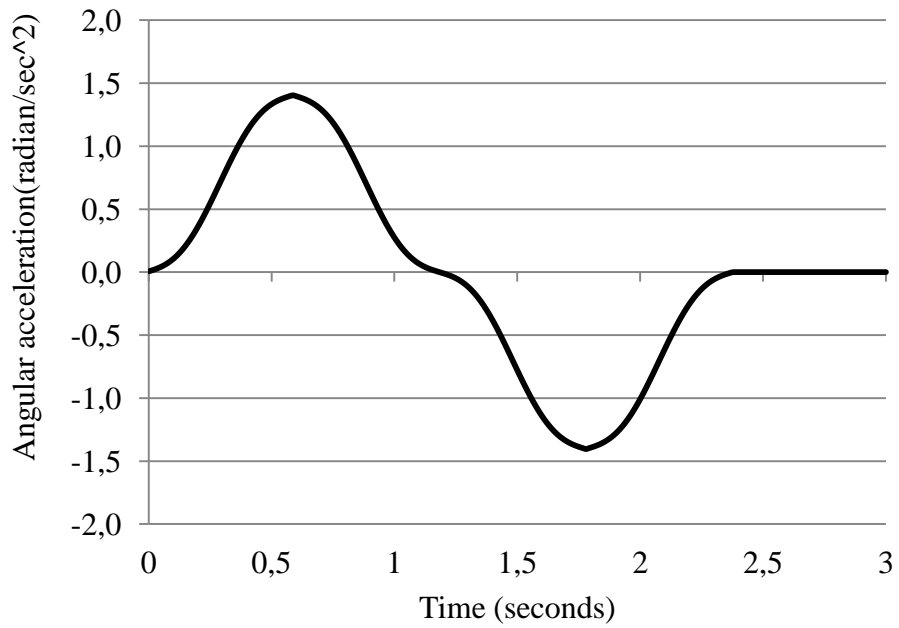


(a) Position

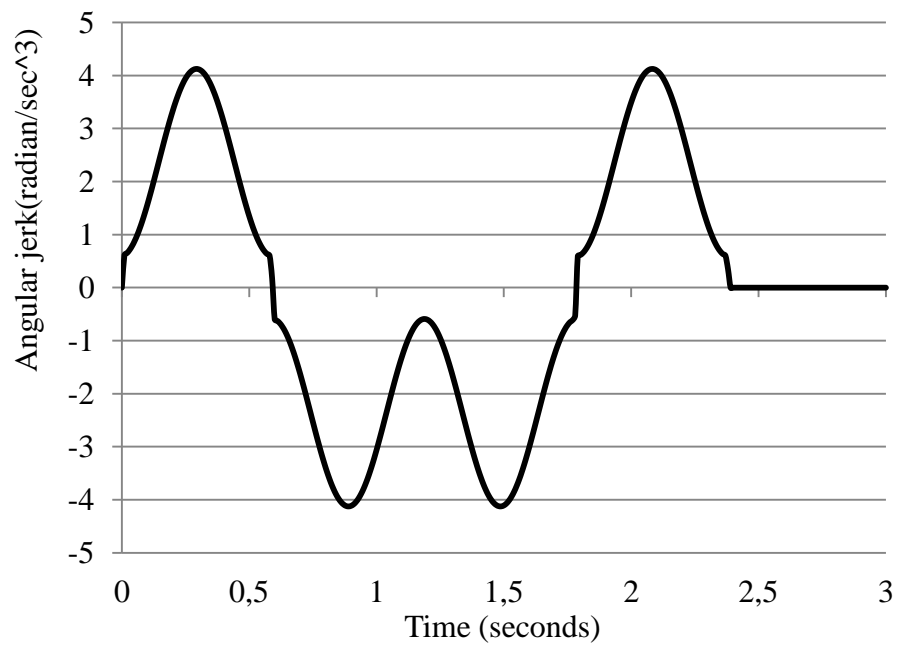


(b) Velocity





(c) Acceleration



(d) Jerk

Figure 5.1: Proposed motion profile's (a) displacement, (b) velocity, (c) acceleration and (d) jerk

For a given total travel time  $t_r$ , the motion profile consists two phase; acceleration phase during  $t_{ra} = t_r/2$  and deceleration phase during  $t_{rd} = t_r/2$ . The acceleration

sector further divided into two sub sector; acceleration rise at half of acceleration period  $t_{rar} = t_{ra}/2$ , and acceleration return at half of acceleration period  $t_{rare} = t_{ra}/2$ . Similarly, the deceleration phase also divided into two sub phases; deceleration rise at half of deceleration period  $t_{rdr} = t_{rd}/2$ , and deceleration return at half of deceleration period  $t_{rdre} = t_{rd}/2$ . The velocity and displacement curves are obtained by integrating. Here the following conditions should be held for shaping and combining the curves: the sum of the duration times of the phases shall be equal to total  $t = t_{ra}/2 + t_{rd}/2$ , the velocities at each sector junctions shall be equal, and the jerk at each sector junctions shall have no infinite jerk value. The motion profile is derived from a cycloid, a ramped versine, and a ramp curve for each sector to obtain smooth acceleration.

General form of the acceleration profile is expressed as:

$$a = \frac{A_1 R t}{2\pi} + \frac{A_2}{2\pi} [R t - \sin(R t)] + \frac{A_3 R t}{2\pi} + \frac{A_3}{2\pi} [1 - \cos(R t)] \quad (5.1)$$

where  $A_1$  is the amplitude of a ramp motion profile,  $A_2$  is the amplitude of a ramped cycloid motion profile,  $A_3$  is the amplitude of a ramped versine motion profile,  $t$  is time into motion,  $\tau$  is the section time,  $R = 2\pi/\tau$ . Furthermore, maximum acceleration can be written as  $A = A_1 + A_2 + A_3$ , and arranging equations above becomes:

$$a = \frac{ARt}{2\pi} - \frac{A_2}{2\pi} \sin(Rt) + \frac{A_3}{2\pi} (1 - \cos(Rt))$$

In this equation  $A_1, A_2, A_3$  is equal to:

$$A_1 = \frac{AR(R - 2\xi\omega_n)}{\omega_n^2} = \frac{A\tau_n(\tau_n - 2\xi\tau)}{\tau^2} \quad (5.1a)$$

$$A_2 = A \left( 1 - \frac{R^2}{\omega_n^2} \right) = A \left( 1 - \frac{\tau_n^2}{\tau^2} \right) \quad (5.1b)$$

$$A_3 = \frac{A2\xi R}{\omega_n} = \frac{A2\xi\tau_n}{\tau} \quad (5.1c)$$

where  $\omega_n$  is the first mode frequency  $\omega_1$  because the vibration amplitude contribution of the modes other than first are negligible small as said previously and  $\xi$  is the damping ratio. Note that theoretically there is no traveling time  $t_r$  restriction on the system and this is the main advantages of this new reference command.

As mentioned previously acceleration profile is consisted of four subsectors. These are:

For the acceleration rise at half of acceleration period  $t \in [0, t_{rar} = t_r/4]$ :

$$a = \frac{ARt}{2\pi} - \frac{A_2}{2\pi} \sin(Rt) + \frac{A_3}{2\pi} (1 - \cos(Rt)) \quad (5.2a)$$

For the acceleration return at half of acceleration period  $t \in [t_{rar} = t_r/4, 2t_{rar} = t_{ra} = t_r/2]$ :

$$a = A - \left[ \frac{AR(t - t_{rar})}{2\pi} - \frac{A_2}{2\pi} \sin(R(t - t_{rar})) + \frac{A_3}{2\pi} (1 - \cos(R(t - t_{rar}))) \right] \quad (5.2b)$$

For the deceleration rise at half of deceleration period  $t \in [t_{ra} = t_r/2, 3t_r/4]$ :

$$a = - \left[ \frac{AR(t - t_{ra})}{2\pi} - \frac{A_2}{2\pi} \sin(R(t - t_{ra})) + \frac{A_3}{2\pi} (1 - \cos(R(t - t_{ra}))) \right] \quad (5.2c)$$

For the deceleration return at half of deceleration period  $t \in [3t_r/4, t_r]$ :

$$a = -A + \left[ \frac{AR(t - 3t_r/4)}{2\pi} - \frac{A_2}{2\pi} \sin(R(t - 3t_r/4)) + \frac{A_3}{2\pi} (1 - \cos(R(t - 3t_r/4))) \right] \quad (5.2d)$$

Since the acceleration and deceleration phase and their subsections durations are selected as one fourth of total travelling time therefore  $A_1$ ,  $A_2$  and  $A_3$  for all section becomes;

$$A_1 = \frac{AR(R - 2\xi\omega_1)}{\omega_1^2} = \frac{A\tau_1(\tau_1 - 2\xi\tau)}{\tau^2}$$

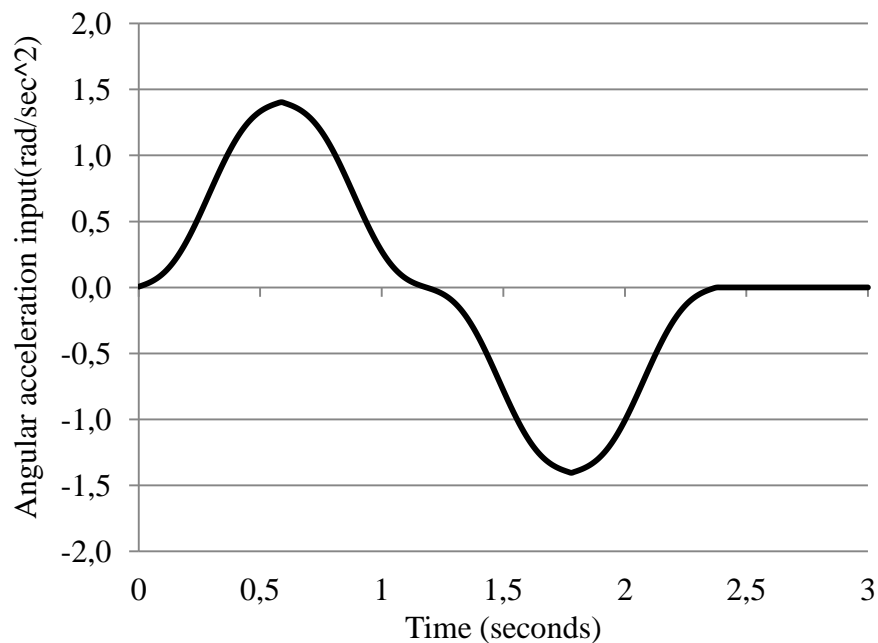
$$A_2 = A \left( 1 - \frac{R^2}{\omega_1^2} \right) = A \left( 1 - \frac{\tau_1^2}{\tau^2} \right)$$

$$A_3 = \frac{A_2 \xi R}{\omega_1} = \frac{A_2 \xi \tau_1}{\tau}$$

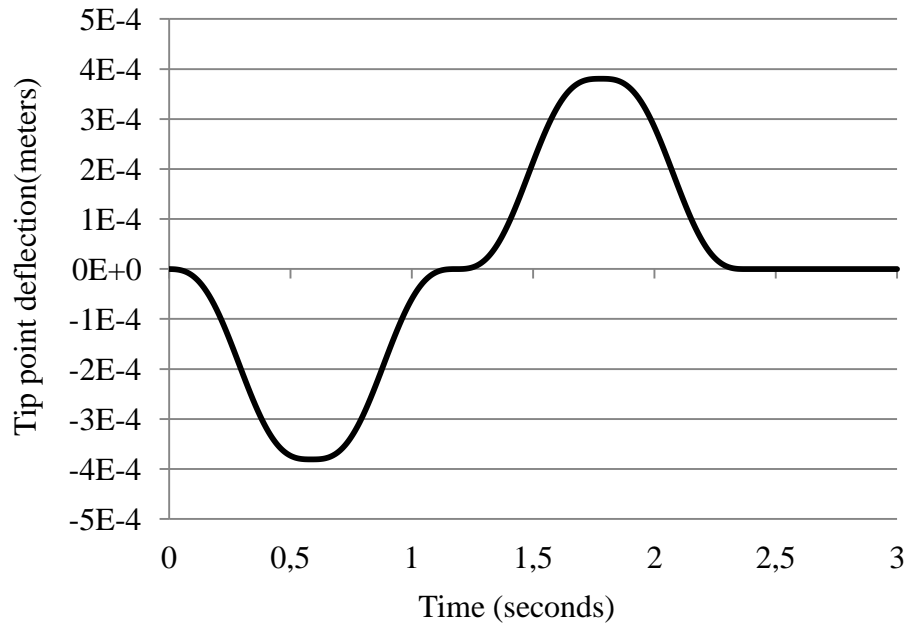
### 5.3 Simulations and Results

Simulations are performed to verify validity and effectiveness of the proposed motion design to eliminate or reduce residual vibration of the flexible system. The flexible link manipulator parameters are given in table 3.1. For simulations, MATLAB software is used. In this program plant and forcing function are modeled as M-file (see appendix A). In order to solve ordinary differential equation ode45 command is used. After solving the equation of motion, oscillation of tip point of SLFM with respect to acceleration input which is applied by motor shaft, is given in figure 5.2. In order to obtain displacement and velocity profiles, acceleration input integrals are taken.

The proposed method results of residual vibration elimination are presented in figure 5.2 and 5.3 for arbitrary selected two travelling times; eight times and three times system period.

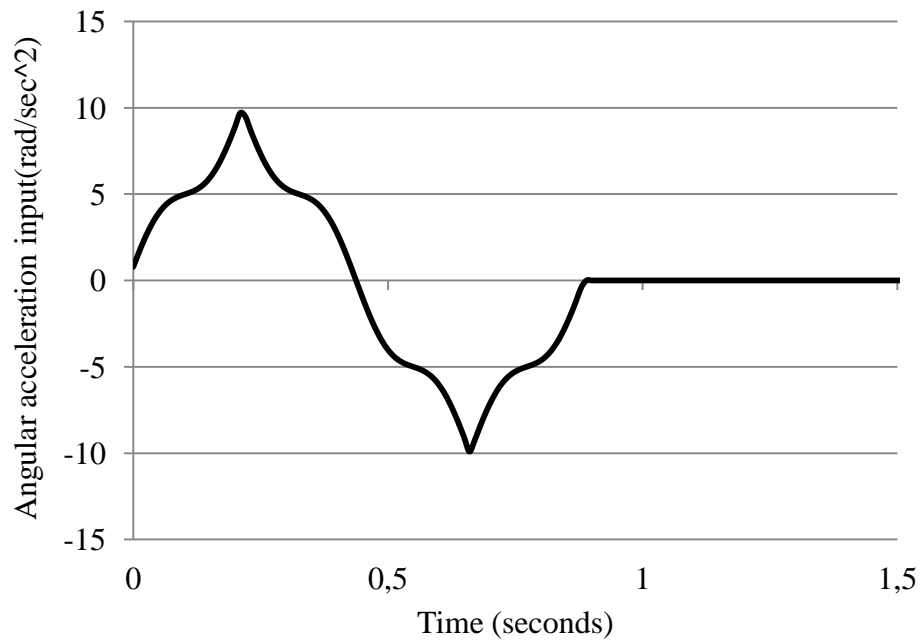


(a)

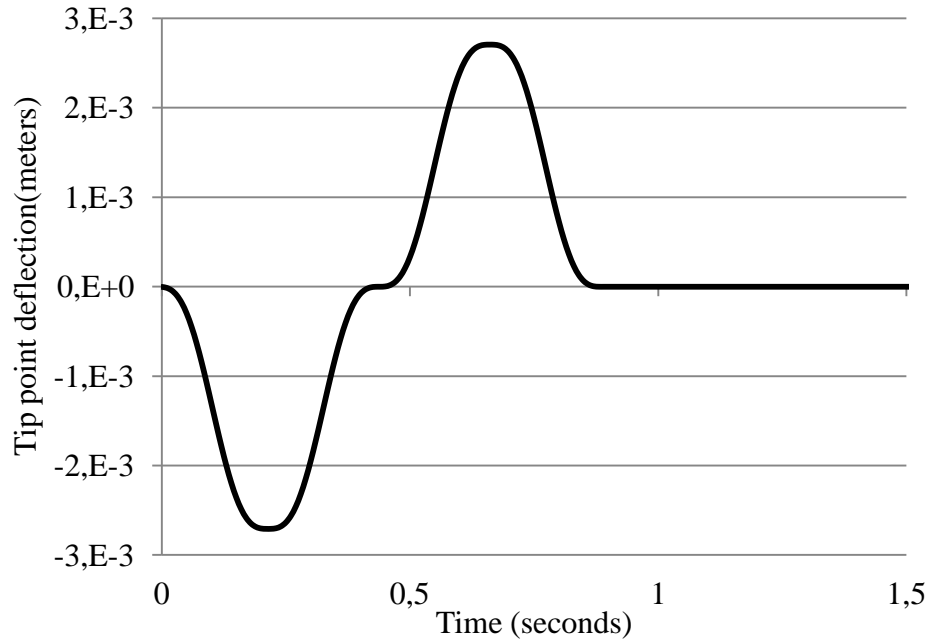


(b)

Figure 5.2: Traveling time is equal to eight times of system period. (a) Proposed acceleration profile, (b) tip point deflection of SLFM with respect to input in (a).



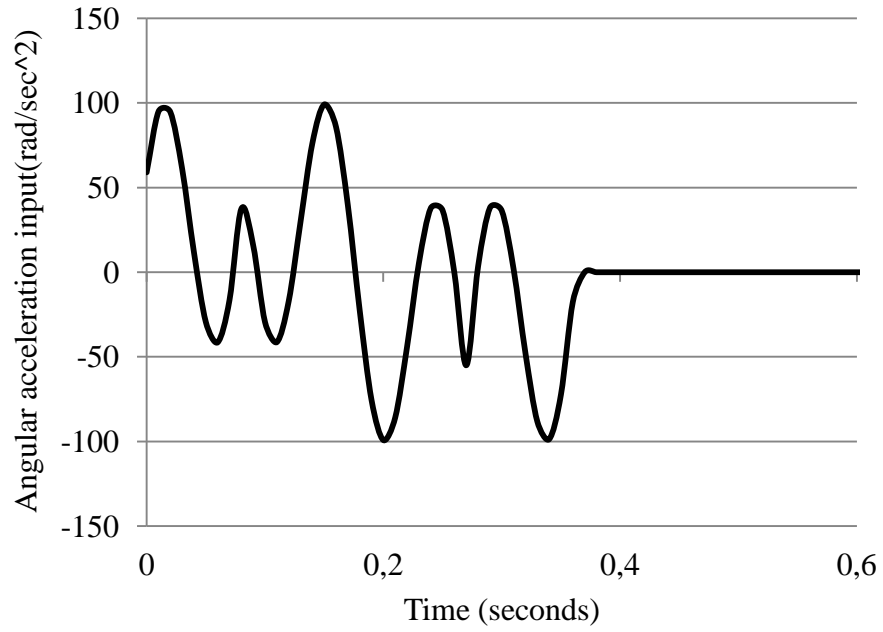
(a)



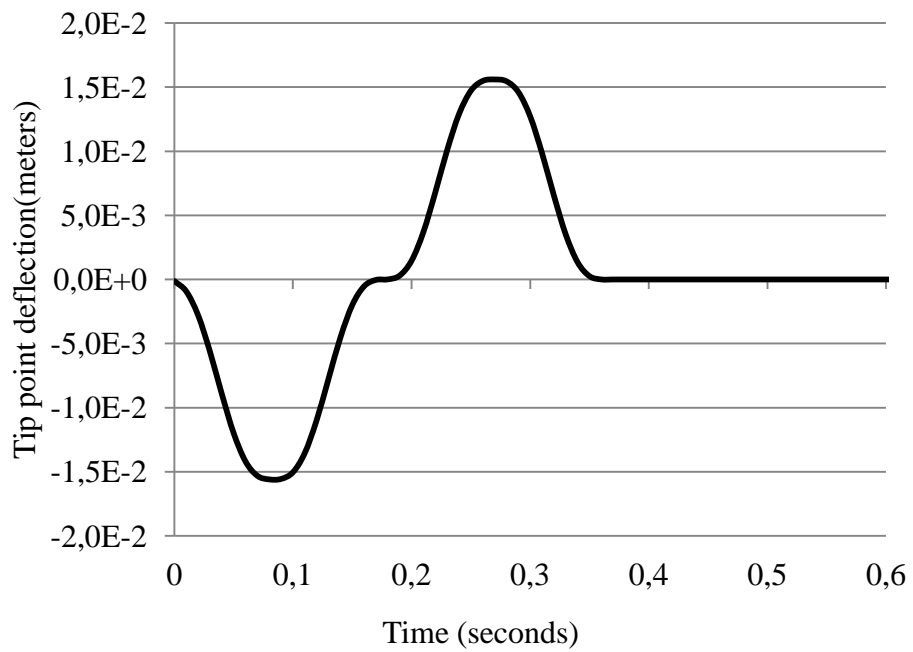
(b)

Figure 5.3: Traveling time is equal to three times of system period. (a) Proposed acceleration profile, (b) tip point deflection of SLFM with respect to input in (a).

As it is seen that from figures 5.2 and 5.3, amplitude of point to point acceleration profile increases when travelling time decreases. Besides at the end of the motion residual vibration is eliminated. In elastic systems if traveling time of input function is equal to period or multiplies of period of system, it is obvious and well known that there will be no residual vibration when motion is completed. In order to show effectiveness of proposed method in any travelling times, 1.25 times system period as travelling time. The result is shown in figure 5.4:



(a)



(b)

Figure 5.4: Traveling time is equal to 1.25 times of system period. (a) Proposed acceleration profile, (b) tip point deflection of SLFM with respect to input in (a).

Amplitude of acceleration becomes bigger when compared with long travelling time duration as in figure 5.2 and 5.3. But it is easily seen that at the end of the motion there is no residual vibration.

Consequently in this section an open loop control strategy is proposed. In this method acceleration profile is generated as input function which eliminates the residual vibrations. It is seen that for any travelling time, the proposed method can eliminate residual vibrations of tip point without imposing any travelling time increase. This method of control will be used in the following chapters to make comparison between the controllers.



## CHAPTER 6

### FUZZY LOGIC CONTROL OF SLFM

#### 6.1 Introduction

In this section question of ‘what is fuzzy logic?’ will be answered and its applications will be explained.

In real world facts are not described sharply like ‘she is tall’ or ‘he is fat’. Always there will be an uncertainty. So natural language should not use with sharp boundaries. In order to represent these imprecise statements fuzzy logic sets is used all around the world.

Fuzzy logic gives us a chance to describe input and output relations of the statements by using human linguistic rules. These linguistic rules consist of human experiences. Fuzzy logic theory generates a ‘If ..... then....’ relation between input and output by using these human experiences. Also linguistic variables like big, small, medium will be used to generate this relationship. Usage of these relationships will be explained in the following section.

There are a lot of application areas of fuzzy logic. Some of them are: control, classification, optimization, and etc. In this chapter application of fuzzy logic controller to SLFM will be discussed.

## 6.2 Fuzzy Logic Control

The use of fuzzy logic to control a system can be explained as describing a controller with linguistic rules. Four main modules exist in fuzzy logic controller. These are fuzzification, knowledge base, inference engine and defuzzification module. Firstly input and output variables must be defined. For plant which is used in this thesis has one input which is torque and two outputs which are hub position and tip point oscillation of the SLFM. So inputs to fuzzy logic controller will be hub position error, its time derivative, tip point position error and its time derivative and output will be torque as control signal. But in this chapter these two degrees of freedom will be controlled by separately. So two controllers will be used. One of them will control the hub position and the other one will control the tip point. After defining the variables the next step is fuzzification of input variables. The fuzzification process is achieved by membership functions. Membership functions are characteristic functions which are used for mapping input variables between  $[0, 1]$ . In this thesis for each variable three triangular membership functions will be used. These are negative, zero and positive as shown in figure (6.1). The variables can take values in  $x$  axis. The interval on  $x$  axis is named as universe of discourse.

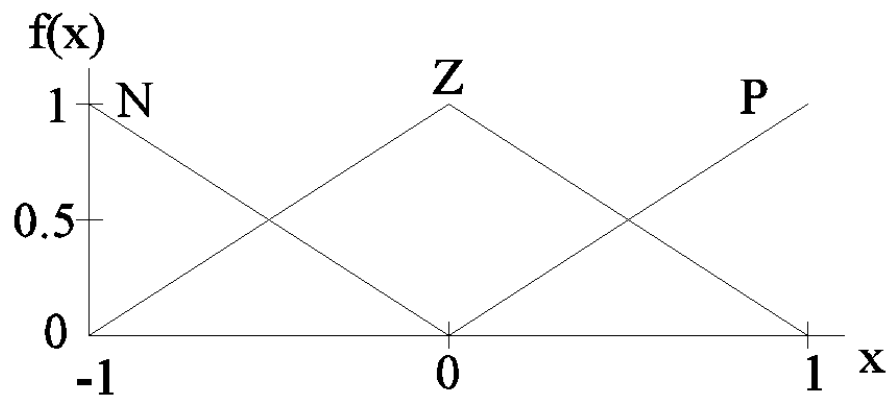


Figure 6.1: Triangular membership function

After fuzzification process knowledge base module will be created. In knowledge module 'If... then....' relations exist. Number of relations can be calculated by number of variables and number of membership functions. For first controller there are two variables which are position of hub and its derivative and both of them have three membership functions. So totally nine 'If... then....' relations exist. For the first controller relations are given in table 6.1.

Table 6.1: 'If... then....' relations for controllers

	error (e)			
	N	Z	P	
Time derivative of error ( $\dot{e}$ )	N	NB	NS	Z
	Z	NS	Z	PS
	P	Z	PS	PB

The third module is inference engine. There are different types of fuzzy inference systems. Some of them are Mamdani method, Sugeno method and simplified method. For this application Sugeno type fuzzy inference system will be used. Biggest characteristic of this inference system, output membership functions are linear or constant. Also in 'If... then....' relations 'and' operator is used. If table 6.1 is written as sentences, the result will be:

If input 1 is N and input 2 is N then output is NB

If input 1 is N and input 2 is Z then output is NS

If input 1 is N and input 2 is P then output is Z

If input 1 is Z and input 2 is N then output is NS

If input 1 is Z and input 2 is Z then output is Z

If input 1 is Z and input 2 is P then output is PS

If input 1 is P and input 2 is N then output is Z

If input 1 is P and input 2 is Z then output is PS

If input 1 is P and input 2 is P then output is PB

So the output surface is plotted in figure 6.2:

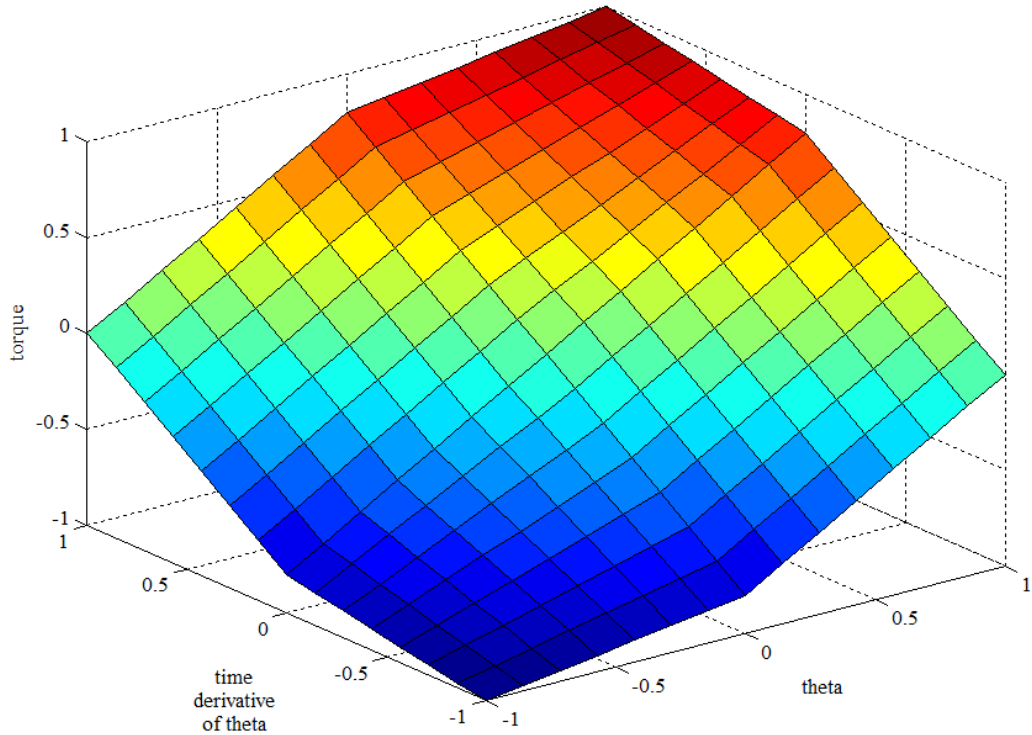


Figure 6.2: Output surface

The last step will be defuzzification module. This module transforms inference engine's results to crisp value. In this project weighted average defuzzification function is used. It is valid for symmetrical membership functions. It calculates the final crisp value by taking weighted average of maximum output membership function values. The output membership function is:

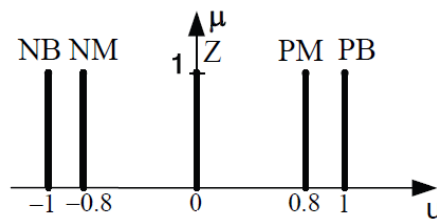


Figure 6.3: Output membership function

### 6.3 Simulations and Results

In this section application fuzzy logic controller to SLFM, which is coupled to a DC motor, will be studied. So equations of motion of SLFM and motor must be coupled. Differential equations of DC motor and SLFM are:

$$L \frac{di}{dt} + Ri + K_e \dot{\theta} = V(t) \quad (6.1)$$

$$J\ddot{\theta} + B\dot{\theta} = K_t i \quad (6.2)$$

$$\ddot{q} + 2\delta\omega_i \dot{q} + \omega_i^2 q = \frac{\Gamma_i}{M_i} \ddot{\theta} \quad (6.3)$$

Where  $L$  inductance constant,  $B$  is damping coefficient,  $K = K_e = K_t$  is electromotor force constant,  $R$  is resistance,  $V(t)$  is input voltage and  $\theta$  is hub position of the SLFM. The parameters in equation (6.3) are mentioned previously in chapter 3.

It is previously said that there are two degrees of freedom. These are controlled by two fuzzy logic controllers. For simulations state space form of the system will be created. The outputs of this state space are hub position and tip point deflection. The input is voltage.

In order to represent these equations in state space format, firstly the state variables must be specified. The states are:

$$x_1 = i$$

$$x_2 = \theta$$

$$x_3 = \dot{\theta}$$

$$x_4 = q$$

$$x_5 = \dot{q}$$

The differential equations become:

$$\begin{aligned} \dot{x}_1 &= -\frac{R}{L} x_1 - K_e \frac{x_3}{L} + \frac{V(t)}{L} \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= -\frac{B}{J} x_3 + \frac{K}{J} x_1 \\ \dot{x}_4 &= x_5 \\ \dot{x}_5 &= -\omega_1^2 x_4 - 2\delta\omega_1 x_5 + \frac{\Gamma}{M} \left( -\frac{B}{J} x_3 + \frac{K}{J} x_1 \right) \end{aligned} \quad (6.4)$$

The state space matrices with respect to these states:

$$A = \begin{bmatrix} -R/J & 0 & -K/J & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ K/J & 0 & -B/J & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -\Gamma K/MJ & 0 & -\Gamma B/MJ & -\omega_1^2 & -2\delta\omega_1 \end{bmatrix},$$

$$B = \begin{bmatrix} 1/L \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, C = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

In these equations values of the parameters are given in table 6.2:

Table 6.2: Values which are will be used in simulations

	Symbol	Value
<b>Length of manipulator</b>	$L(m)$	0.7
<b>Cross-sectional width</b>	$w(m)$	0.002
<b>Cross-sectional height</b>	$h(m)$	0.0255
<b>Modulus of elasticity</b>	$E(GPa)$	71
<b>Weight per unit length</b>	$m(kg/m)$	0.1382
<b>First mode frequency</b>	$\omega_1(rad/s)$	21
<b>Mass moment of inertia of motor</b>	$J(kgm^2)$	3.2284e-6
<b>Damping coefficient of motor</b>	$B(Nms)$	3.5077e-6
<b>Electromotor force constant</b>	$K(Nm/Amp)$	0.0274
<b>Resistant</b>	$R(ohm)$	4
<b>Inductance</b>	$L(H)$	2.75e-6

The control application will be simulated in SIMULINK software and its block diagram is shown in figure 6.4:

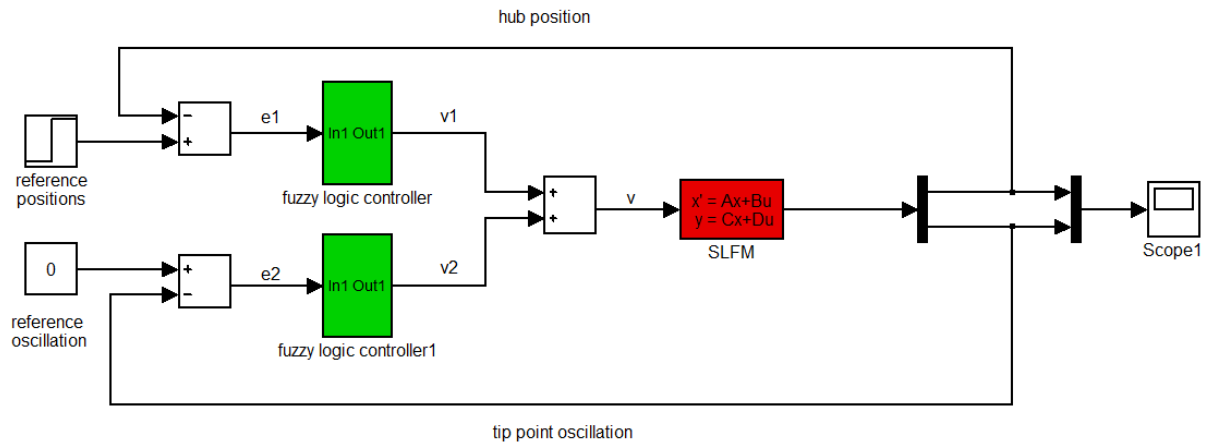


Figure 6.4: Block diagram of fuzzy logic control of SLFM

Inputs of controllers are errors and outputs are voltages. Error 1(e1) is equal to difference between reference position value and actual position value. Error 2 (e2) is equal to difference between zero and actual tip point deflection value. V1 and V2 are outputs of the controllers and create the final voltage value which is control signal.

In this step fuzzy logic controllers will be designed. Block diagram of fuzzy logic controller of theta is shown in figure 6.5:

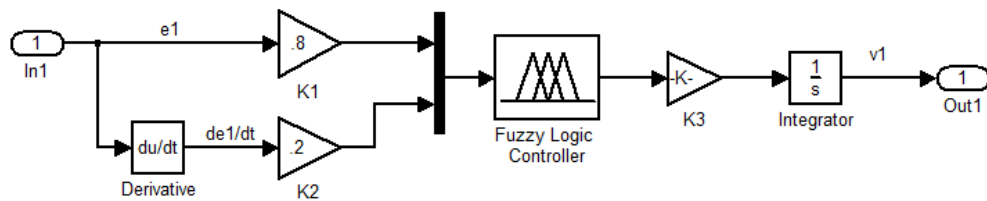


Figure 6.5: Fuzzy logic controller of hub position

K1, K2 and K3 gains are scaling constants. In this controller structure time error and its time derivative enter the fuzzy logic controller and output of fuzzy logic controller is integrated. So PI-type fuzzy logic controller can be generated. PI-type fuzzy logic controller means that fuzzy logic controller behaves like conventional PI controller.

Fuzzy logic controller block is taken from SIMULINK library. When double click on it, an interface will open. In this interface number of inputs and outputs, inference

engine and ‘If.....then.....’ rules can be written. For this problem there are two inputs. The view of interface is shown in figure 6.6.

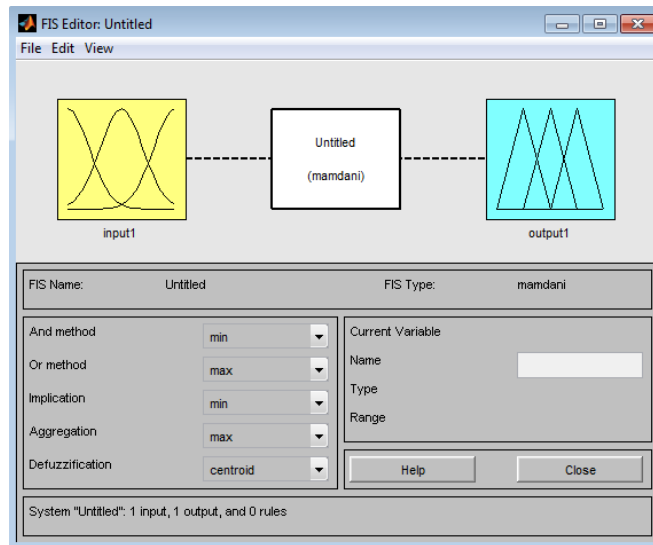


Figure 6.6: Fuzzy logic controller interface

First step is opening a new FIS editor. For this application Sugeno type inference system is chosen. Second step is defining the number of inputs.

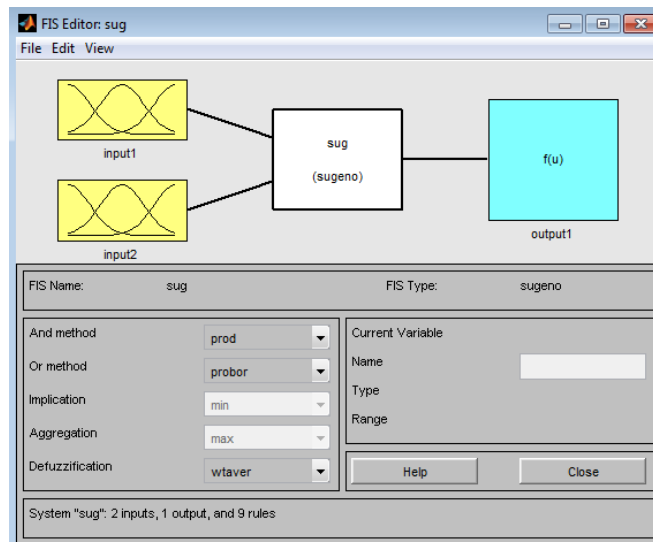


Figure 6.7: Sugeno type fuzzy logic controller editor

Another step is clicking on input block. An editor will open and membership functions for this variable can be chosen. For input1 the membership functions are shown figure 6.8.



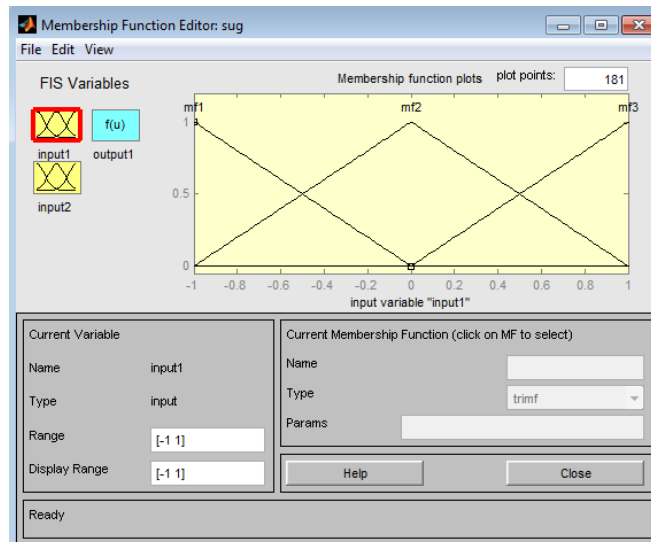


Figure 6.8: Membership function editor

As seen from figure 6.8 there are three membership functions for input 1. The same membership functions are assigned for input 2. The universe of discourse for both inputs is assigned between -1 and 1. Another important thing is tuning the shape of membership function. Generally triangular membership function and standard shapes are preferred. But it can be changed depends on requirements. If sharp rise is needed, width of mf2 can be decreased. So there will be rapid response but more overshoot. For this application standard membership functions are used.

Also there are five output membership functions. These are negative big (-1), negative small (-0.8), zero (0), positive small (0.8) and positive big (1).

After defining the membership functions, 'If.....then.....' relation will be written. These relations are previously given in this chapter. The editor of these rules is shown in figure 6.9.

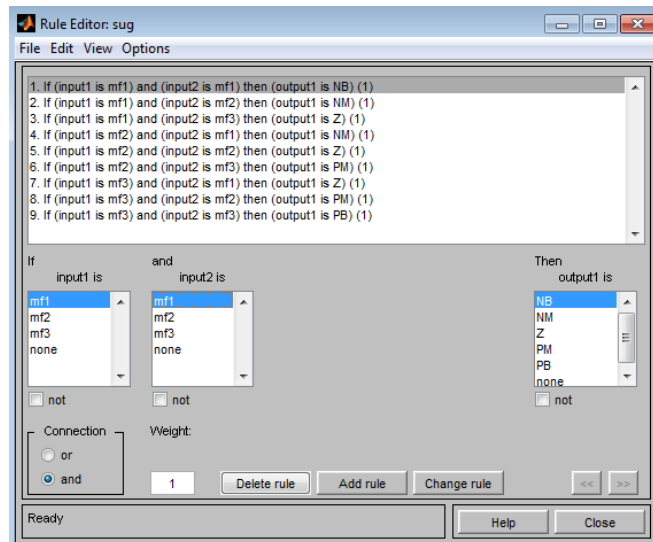
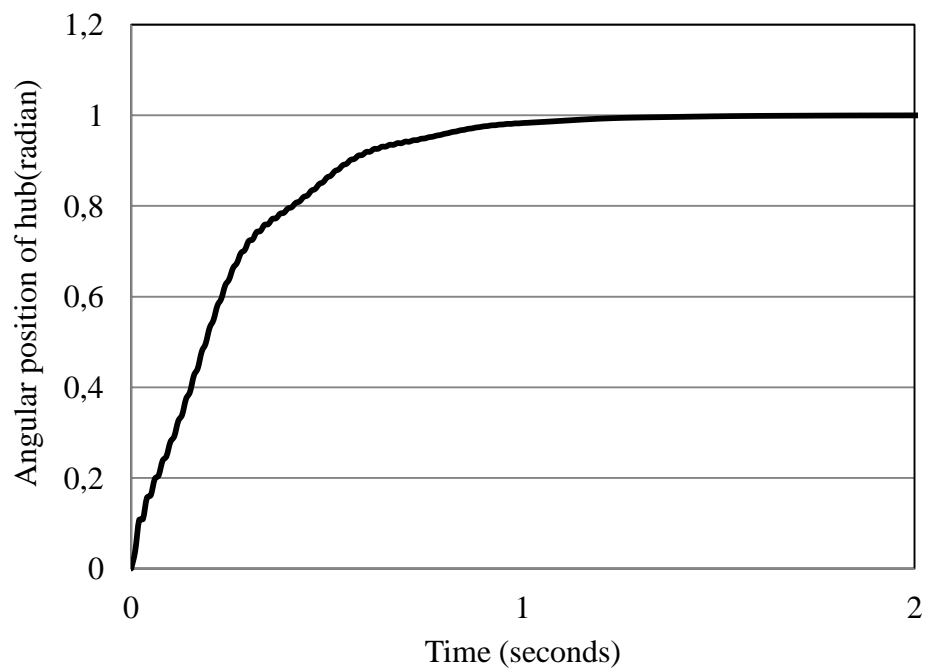
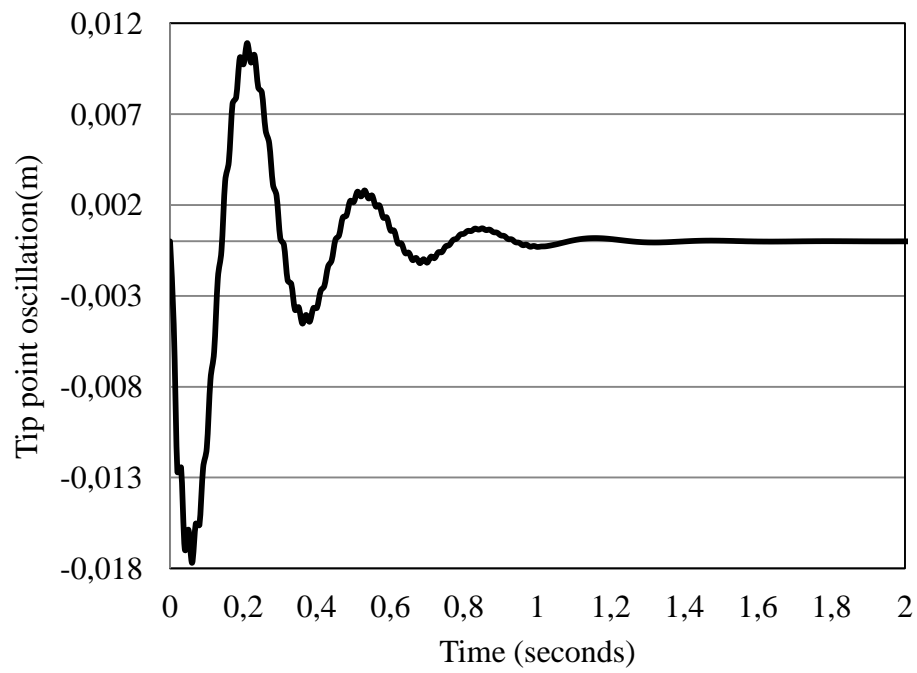


Figure 6.9: Rule editor

After defining the rules, simulation can be done. When system, which is represented in figure 6.4, is run, results can be taken from the scope. The output of the scope is shown in figure 6.10.



(a)



(b)

Figure 6.10: (a) hub position, (b) tip point oscillation

With respect to these results, hub position comes to reference value in 1 second. Besides this tip point has an oscillation. But this oscillation is eliminated in 1 second by the controller. Maximum overshoot is approximately 0.018 meters. This results show that when no overshoot is necessary in hub position, this controller is very effective.

## CHAPTER 7

### NEURAL NETWORK CONTROL OF SLFM

#### 7.1 Introduction

It is known that human brain has learning ability over time. By using this learning ability, humans can overcome complex situations. Also these properties of nervous systems can be used as a model which is work on complex data sets [25].

Neural network is a mathematical model which is inspired from human brain. Neural networks creates a mathematical relationship between inputs and outputs of a system. Neural networks composed of artificial neurons. So it is beneficial that to answer the question of ‘what is an artificial neuron?’ An artificial neuron is basic component of nervous system. In order to make it easy it understand, there are some statements [33]. These are:

1. Neurons are basic components of the whole nervous system.
2. Connection between the neurons are made by connection links and the signals are transferred by this connections.
3. All connection links have own weights and signals are multiplied by this weights.
4. Each neuron has own mathematical equation which depends on inputs, weights and activation function.

If inputs are  $x_1, x_2, \dots, x_n$  and its connection weights are  $w_1, w_2, \dots, w_n$ , the net input

will be  $\sum_{i=1}^n x_i w_i$ . So the output of the neuron is:

$$y = f\left(\sum_{i=1}^n x_i w_i - b\right) \quad (7.1)$$

The block representation of this neuron is presented in figure (7.1).

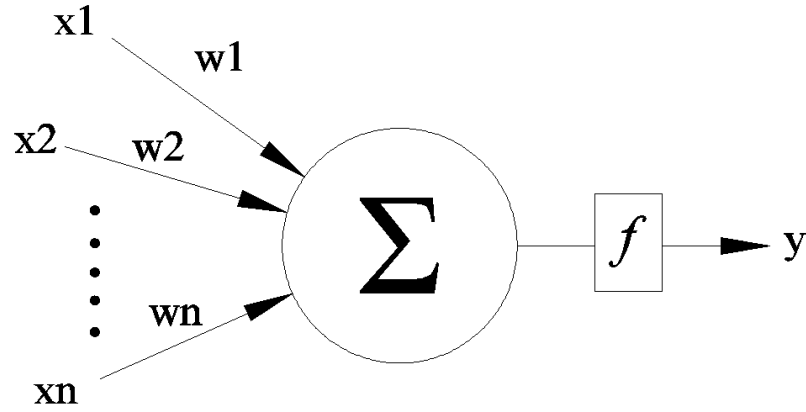


Figure 7.1: Simple neuron representation

In this thesis only feed-forward neural network will be used. Feed-forward means computation is only forward as indicated by arrows in figure 7.1. In this neuron structure there are an input layer which has input nodes as  $x_1, x_2, \dots, x_n$  and an output layer which is indicated as  $y$ . This structure of neural networks are called as single-layer neural networks. This single layer structure is also called as perceptron.

Learning process can be explained as tuning of weights and biases with respect to an error function. So another question arises that ‘how the error information can be achieved?’. In order to get this information training sets must be created. This sets are based on input and output relationship. Learning type from this type of information is named as supervised learning. There are inputs and outputs which are taken from system previously. The main aim is train the neural network to behave the same as the system. So same inputs are given to real system and neural network. The difference between outputs of neural network and real system is named as error. Error is used by a performance function. The main aim is the minimizing the performance function. In order to minimizing the error function delta rule is developed.

The delta rule is a learning algorithm. By using the delta rule, output of the neural network,  $o^q$  will be same as the output of the real system,  $y^q$ . So the performance function, which is given equation 7.2, will be minimized.

$$E^q = \frac{1}{2} \sum_{i=1}^m (y_i^q - o_i^q)^2 \quad (7.2)$$

where  $y$  is real output,  $o$  is neural network output.  $q$  number of outputs.

The gradient descent method will be used for optimizing performance function. In order to use this method,  $w_{ij}$  must be written in differential form. The output of the neural network,  $o_i$  the function of  $w_{ij}$  as given in equation 7.3.

$$o_i^q = f_i \left( \sum_{j=0}^n w_{ij} x_j^q \right) \quad (7.3)$$

where  $w_{ij}$  is weight between  $j^{th}$  input node and  $i^{th}$  output neuron.  $f_i$  is activation function at  $i^{th}$  output neuron. This activation function can be different type functions. The most commonly used activation function is sigmoid function is given in equation 7.4.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (7.4)$$

Wight  $w_{jk}$  update algorithm is:

$$w_{jk\_new} = w_{jk\_old} + \Delta w_{jk} \quad (7.5)$$

Where  $\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{ij}}$  and  $\eta$  is learning rate. In this equation  $\frac{\partial E}{\partial w_{ij}}$  can be written as:

$$\frac{\partial E}{\partial w_{ij}} = x_k (o_j - y_j) f_j' \left( \sum_{i=0}^n w_{ji} x_i \right) \quad (7.6)$$

This equation is written for the  $j^{th}$  neuron.

## 7.2 Neural Network Control

In reference [34] PID controller is built by neural networks. The block diagram representation of control structure is shown in figure 7.2.

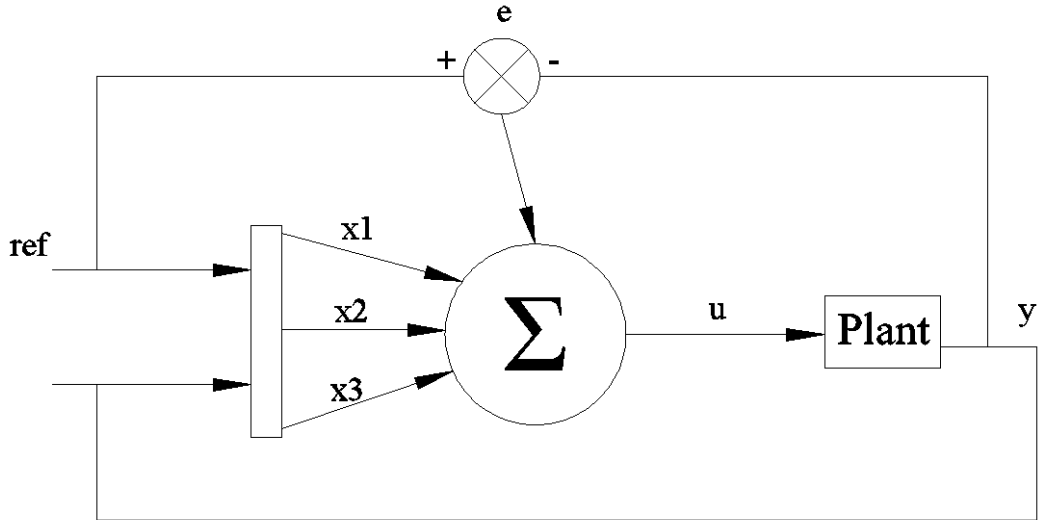


Figure 7.2: Block diagram of the controller and plant [34]

In this system neural network structure behaves like a PID controller. Where  $x_1, x_2, x_3$  are:

$$\begin{aligned}
 x_1(k) &= e(k) \\
 x_2(k) &= e(k) - e(k-1) \\
 x_3(k) &= e(k) - 2e(k-1) + e(k-2)
 \end{aligned}
 \quad (k = 1, 2, 3, \dots) \quad (7.7)$$

These inputs multiplied by weights which are updated by change of error. Weight are defined as:

$$\begin{aligned}
 w_1(k) &= w_1(k-1) + \eta_i e(k) u(k) x_1(k) \\
 w_2(k) &= w_2(k-1) + \eta_p e(k) u(k) x_2(k) \\
 w_3(k) &= w_3(k-1) + \eta_d e(k) u(k) x_3(k)
 \end{aligned}
 \quad (7.8)$$

where  $\eta_i$  is integral learning rate,  $\eta_p$  is proportional learning rate,  $\eta_d$  is derivative learning rate.

So control signal is described as:

$$u(k) = u(k-1) + K \sum_{i=1}^3 w_i(k) x_i(k) \quad (7.9)$$

K is neuron proportional gain.

In this section plant must be described as a discrete model. Because the controller has discrete structure. The system, which is shown equation 3.35, is:

$$\ddot{q}_i(t) + 2\delta\omega_i\dot{q}_i(t) + \omega_i^2 q_i(t) = \left( \frac{\Gamma_i}{M_i J} \right) \tau(t) \quad (3.35)$$

The state space matrices are:

$$A = \begin{bmatrix} 0 & 1 \\ -441 & -0.84 \end{bmatrix}, B = \begin{bmatrix} 0 \\ -7.6 \end{bmatrix}, C = [1 \quad 0], D = [0] \quad (7.10)$$

By using MATLAB 'c2d' code system can be converted to discrete system. Discrete system is:

$$G(z) = \frac{-3.8 \times 10^{-6} z - 3.78 \times 10^{-6}}{z^2 - 2z + 1} \quad (7.11)$$

### 7.3 Simulations and Results

In order to simulate the control action MATLAB m-file is generated. It is given in appendix B.

An initial tip point displacement is given to the system. The expected behavior from the controller is generating of a control signal which is eliminates the deflection on the beam. Weights in controller are updated with respect to error between tip point and reference value. Also control signal depends on error and weights.

By simulating the system, tip point oscillation output, which is shown in figure 7.3, can be achieved.



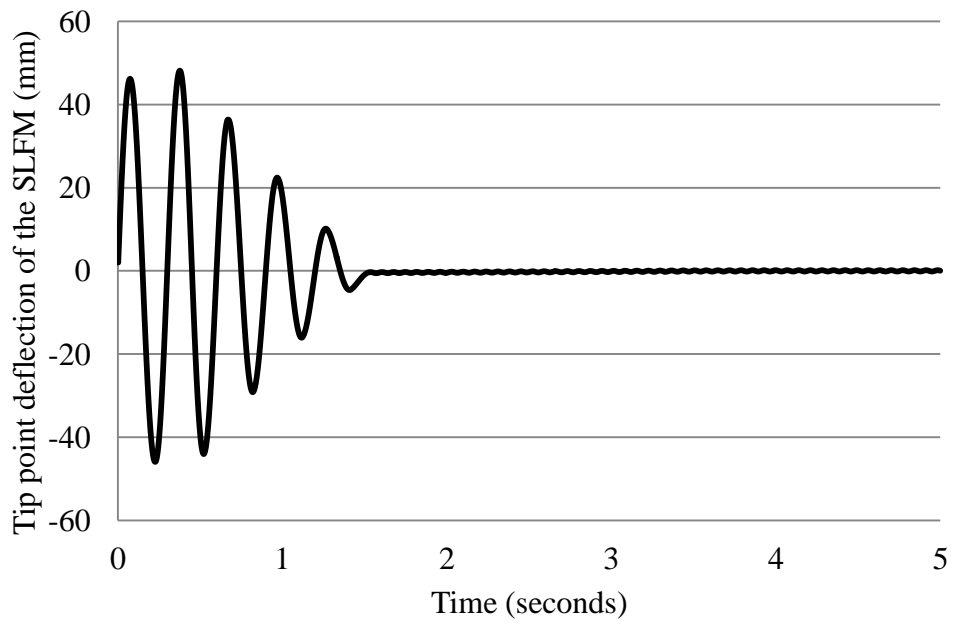


Figure 7.3: Tip point deflection of SLFM

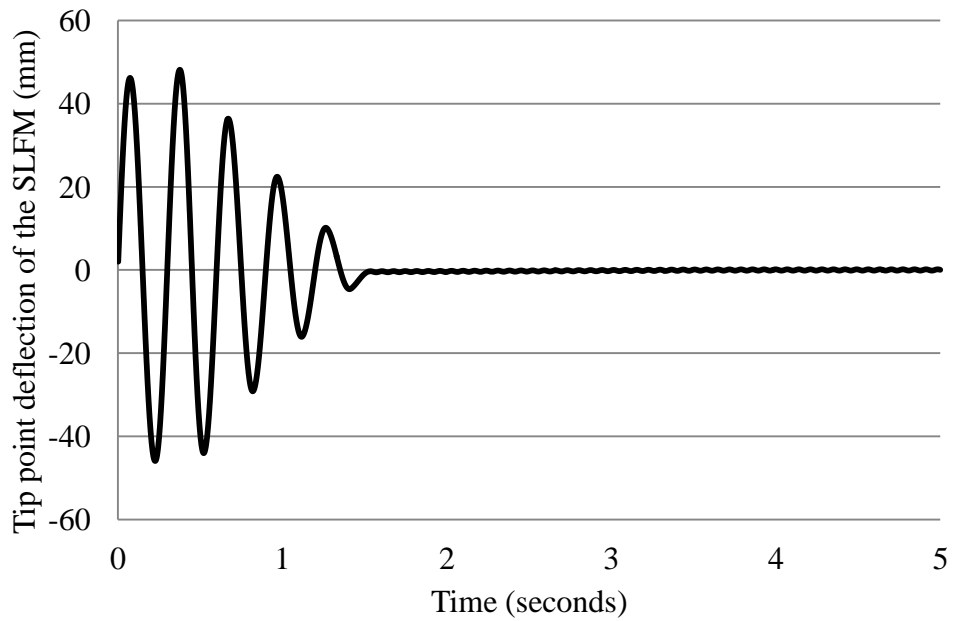
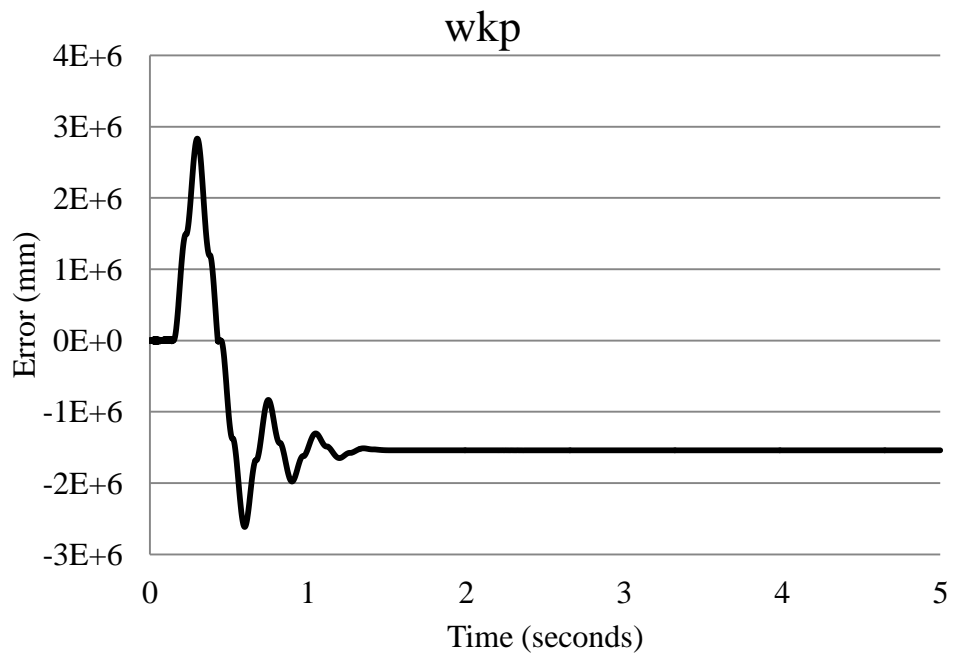


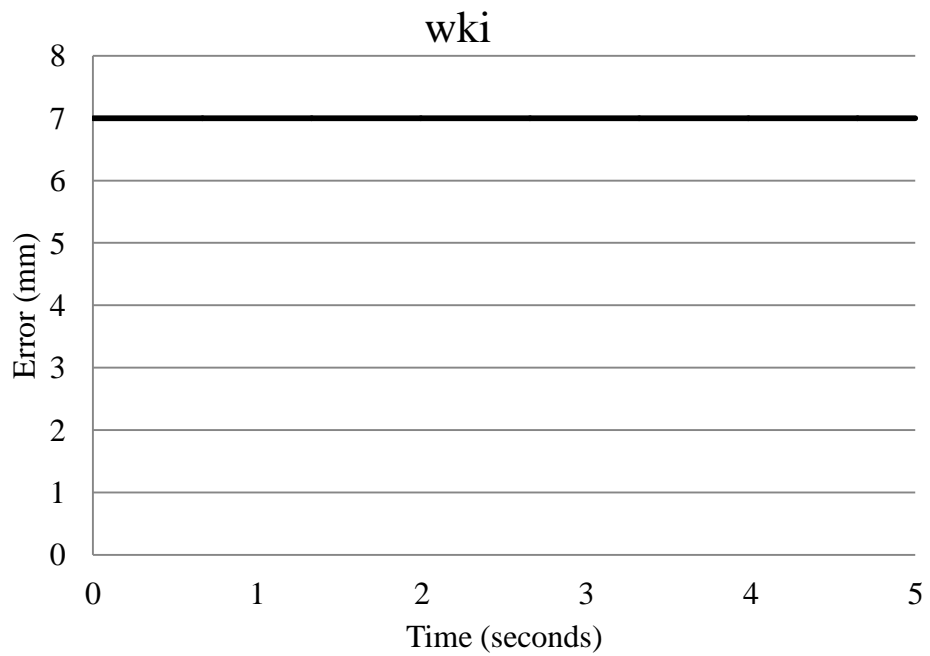
Figure 7.4: Tip point position error

It is easily seen that from the figures 7.3 and 7.4, at the beginning of the motion error increases. After 1 second past, error decreases to zero. This behavior can be tuned by changing parameters of the controller such as learning rates, initial gain values and

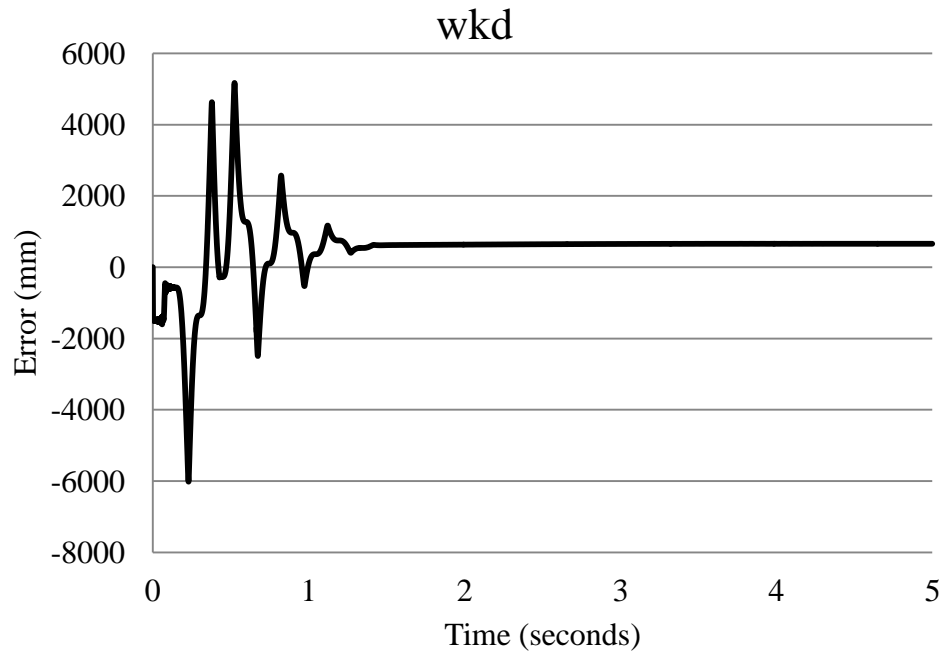
proportional neuron constant. The change of proportional, integral and derivative gains are given in figure 7.5.



(a)



(b)



(c)

Figure 7.5: Change of (a) proportional, (b) integral, (c) derivative gains

## CHAPTER 8

### CONCLUSION AND DISCUSSION

#### 8.1 Conclusion and Discussion

In the content of this thesis, first single link flexible manipulator has been mathematically modeled. After that Simulink and MATLAB code format of this model is designed. The next step has been designing the controller which controls the position of the manipulator. This controller should also eliminate or at least reduce the residual vibrations on the link.

With this idea firstly proportional-integral-derivative controller has been designed. In that stage it has been seen that proportional and derivative controller is sufficient.

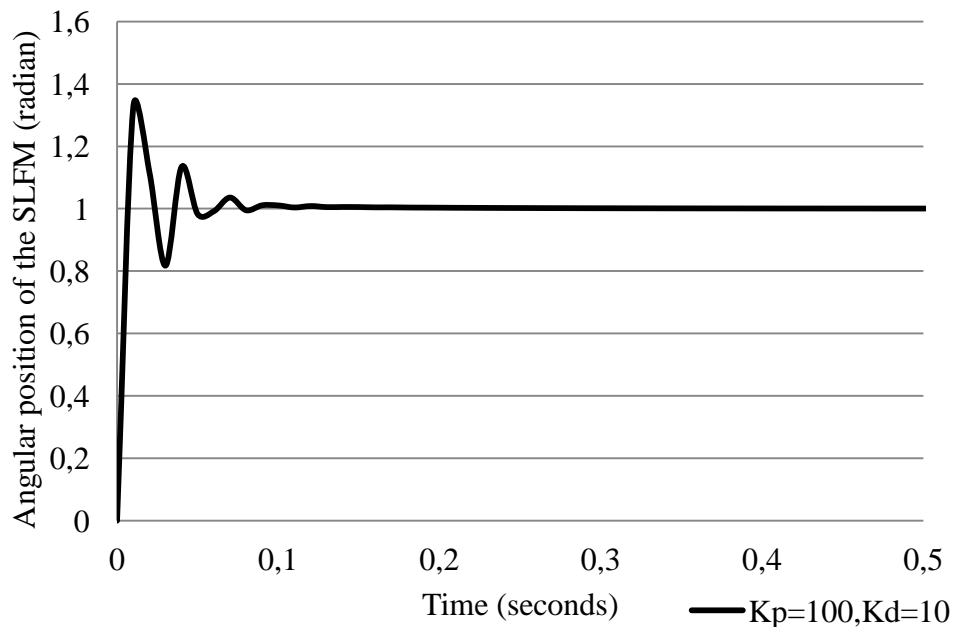
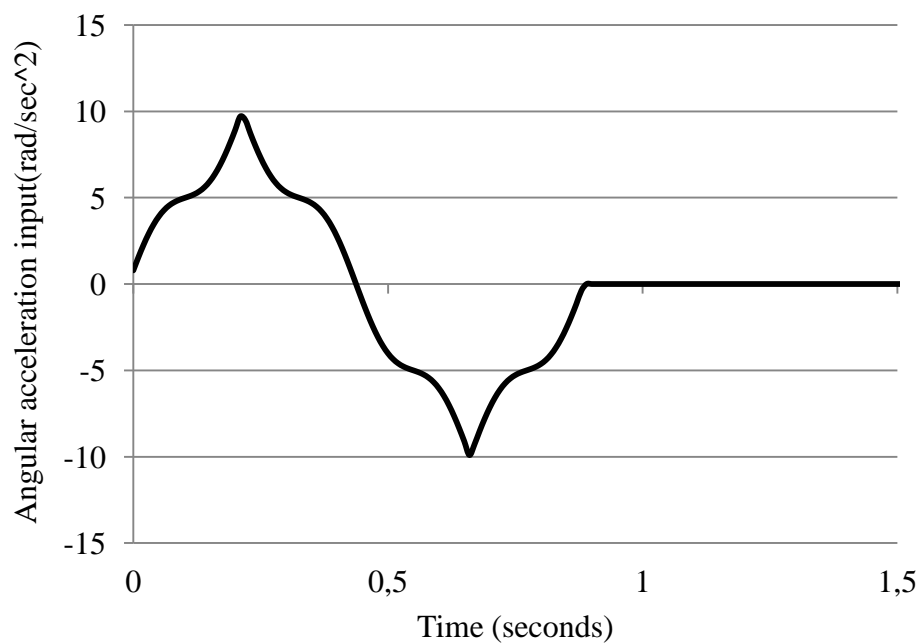


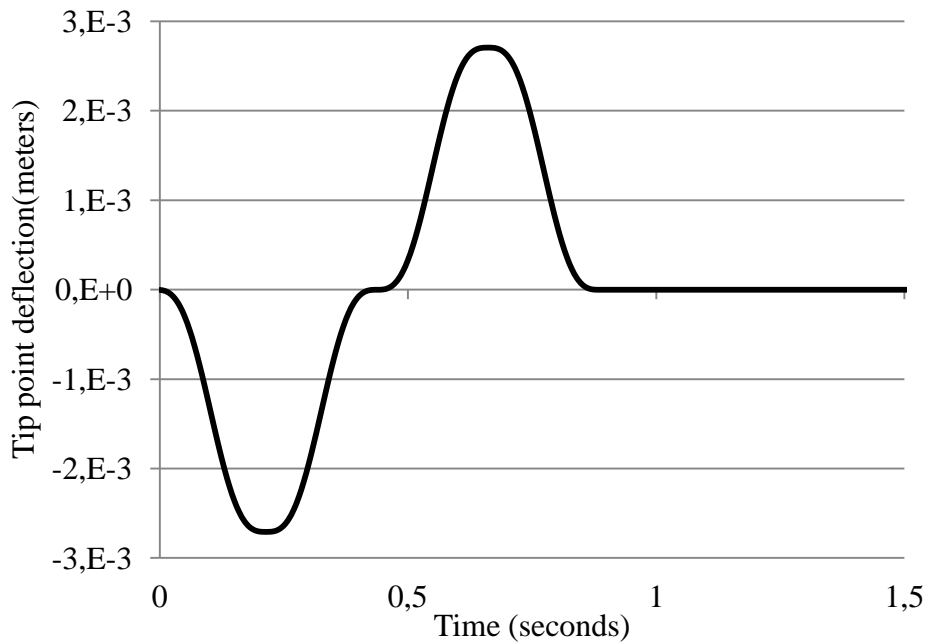
Figure 8.1: Response of system at  $K_p = 100$  and  $K_d = 10$

Systems response time is smaller than 0.1 seconds but overshoot value is bigger than 1.3 radians. If quick response requirement is high, this type of controller will be sufficient. But if there is no overshoot requirement, this controller will not be useful. Controller can be modeled as no overshoot controller but at this time response time will be longer. Besides these, if there are some uncertainties or some change of properties of manipulator, controller may not be do its duty correctly. So more robust controller must be used for this kind of situations.

The second controller is designed by command shaping techniques. In order to constitute this controller all characteristic properties of the system must be specified. Because command, which will be given to the system as input, is shaped by these characteristic properties. Total traveling time of the shaped command must be multiplications of the system period. For proposed command function, controller technique works efficiently for all multiplies of natural frequency. But in reality system can give an effective response between some limits. These limits are mainly depends on the actuator. The results which are shaped for 3 times system period for traveling time are shown in figure 8.2:



(a)

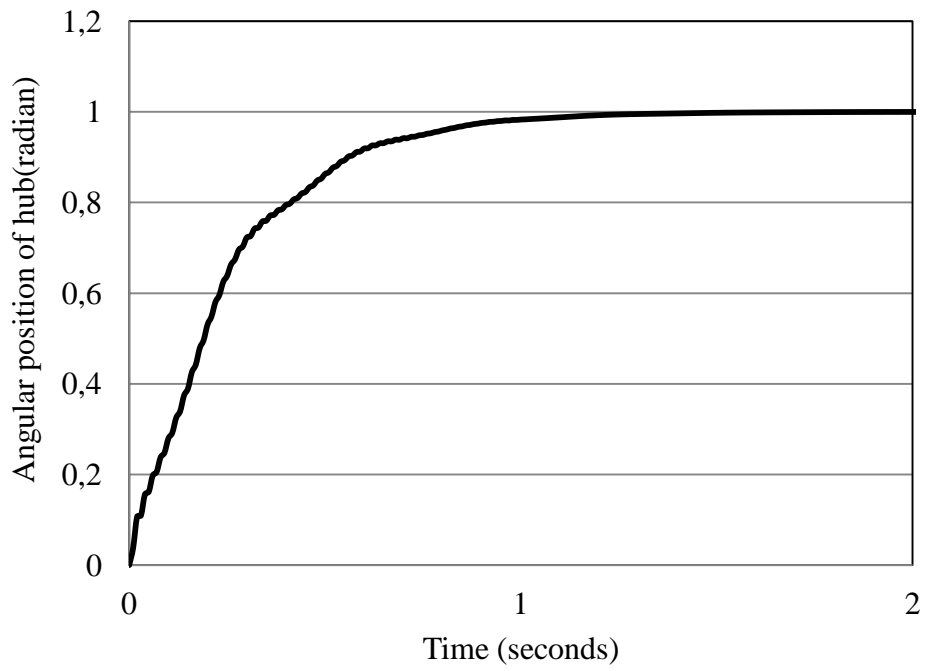


(b)

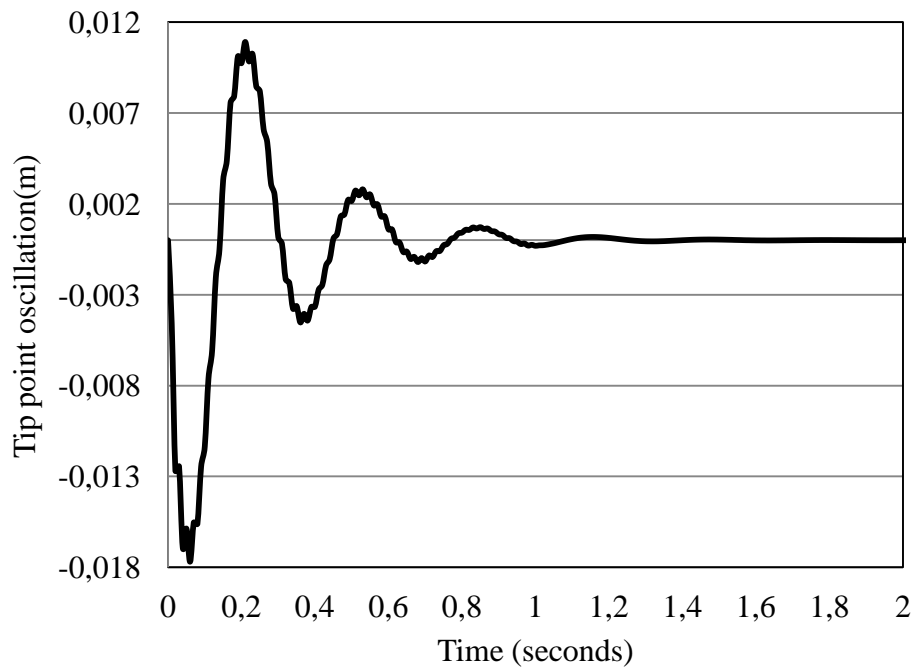
Figure 8.2: Traveling time is equal to three times of system period. (a) Proposed acceleration profile, (b) tip point deflection of SLFM with respect to input in (a).

It is seen from results response time is approximately 1 second. This result is taken for 1 radian angular displacement. There is approximately 0.0025 meters overshoot. Main disadvantage of this control type is its robustness. If there are some uncertain properties, this technique will not give a proper response. Because input function is created with respect to system properties.

The third controller is generated by using fuzzy logic. Fuzzy logic gives a great ease to use it in different applications because fuzzy logic uses human linguistic rules in order to describe relations between variables. By using these linguistic rules PI-type fuzzy logic controller has been designed in the content of this thesis. The results have shown that fuzzy logic is very effective to control the position of the link.



(a)



(b)

Figure 8.3: (a) hub position, (b) tip point oscillation

Response time is approximately 1 second and maximum overshoot in tip point is 0.018 m. This means that fuzzy logic controller is better than other controllers.

The last controller type is neural network controller. Neural network is a model which is originated from human brain. In this thesis neural network structure has been used for constructing a PID-type neural network controller. In this controller gains are adapted themselves with respect to error between actual output and reference input. Delta rule has been used to minimizing this error. The result is shown in figure 8.4.

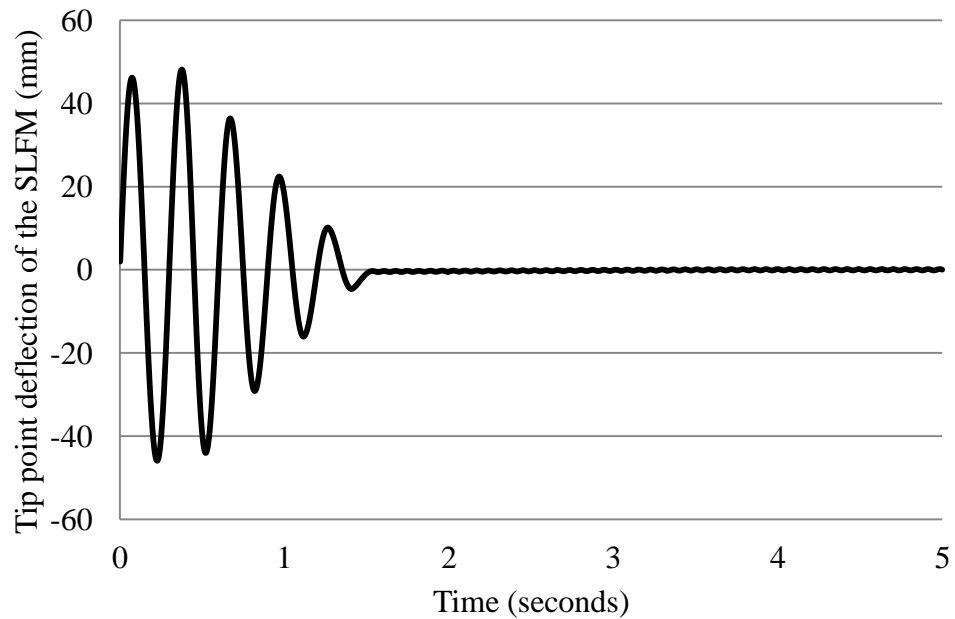


Figure 8.4: Tip point deflection of SLFM

It is shown from figure 8.4; response time is slightly more than 1 second. Also there is oscillation until 1 second. By tuning learning rate, proportional network constant, this result can be enhanced. Also this control structure can adapt itself for all uncertainties because of its learning capability.

Control applications shows that, system response time is one second for all controller types. Generally classical controllers cannot work efficiently, if system parameters are not exactly known. But controllers which are designed by using soft computing techniques are more capable for varying system parameters. They can be adapted themselves for different situations.



## **8.2 Recommendations for the Future Works**

Even though there are a lot of applications of flexible mechanical systems, they can be more popular with further studies. Currently studies are generally about single link systems. Besides there is a shift to multi-link flexible systems. By this way use of these systems will be more widespread.

Another opinion is about robustness of the control techniques. These kinds of systems can have some uncertainties. Also payloads on the links can be variable. So different kind of self-tuning control schemes like fuzzy logic or neural network should be discussed.

## REFERENCES

- [1] M. Benosman and G. Le Vey (2003). Control of flexible manipulators: A survey, *Robotica* vol. 22, pp. 533-545.
- [2] William T. Thomson (1972). Theory of vibration with applications, Prentice-Hall Inc.
- [3] H. Diken, A. Alghamdi (2003). Residual vibration response spectra for a servo-driven flexible beam, *Proceedings of the Institution of Mechanical Engineers*, 217,5, pp.577.
- [4] A. Ankaralı, H. Diken (1997). Vibration control of an elastic manipulator link, *Journal of sound and vibration*, 204, pp.162-170.
- [5] K. Alnefaie, H. Diken, A. Alghamdi (2009). Residual vibration of a rotating flexible beam subject to prescribed motion, *JKAU: Eng. Sci.*, vol.20, no.2, pp.97-107.
- [6] T. Mansour, A. Konno (2010). Vibration based control for flexible link manipulator, *Robot Manipulators New Achievements*, chapter: 24.
- [7] W. Bernzen (1999). Active vibration control of flexible robots using virtual spring-damper systems, *Journal of intelligent and robotic systems*, vol.24, pp.69-88.
- [8] L. Meirovitch (1967). Analytical methods in vibrations, The Macmillan Company.
- [9] Z. Mohamed, M.O. Tokhi (2004). Command shaping techniques for vibration control of a flexible robot manipulator, *Mechatronics*, vol. 14, pp. 69-90.
- [10] A. Cuccio, R. Garziera, P. Righettini, (1998). Vibration control input-laws in point-to-point motion: theory and experiments, *Mech. Machines Theory*, vol.33 (4), pp. 341–349.

- [11] D.M. Aspinwall, (1980). Acceleration Profiles for Minimizing Residual Response, *ASME Journal of Dynamic Systems: Measurement and Control*, vol. 102 pp. 3-6.
- [12] P.H. Meckl, W.P. Seering, (1988). Controlling velocity-limited systems to reduce residual vibration, *Robotics and Automation Proceedings IEEE International Conference*, vol.3, pp.1428-1433.
- [13] K. Alnefaie, H. Diken and A. Alghamdi, (2009). Residual vibration of a rotating flexible beam subject to prescribed motion, *JKAU: Eng. Sci.*, vol.20, no.2, pp.97-107.
- [14] K. Shin, M.J. Brennan, (2008). Two simple methods to suppress the residual vibrations of a translating or rotating flexible cantilever beam, *Journal of Sound and Vibration*, vol.312 pp.140-150.
- [15] T. Önsay, A. Akay, (1991). Vibration reduction of a flexible arm by time-optimal open-loop control, *Journal of Sound and Vibration*, vol.147, pp.283-300.
- [16] N.C. Singer, W.C. Seering (1990). Preshaping command inputs to reduce system vibration, *ASME Journal of Dynamic System, Measurement and Control*, vol.112, no.1, pp.76-82.
- [17] Neil C. Singer, (1989). Residual vibration reduction in computer controlled machines, *MIT Artificial Intelligence Laboratory, PhD thesis*.
- [18] G. Alici, S. Kapucu, S. Bayseç (2000). On preshaped reference inputs to reduce swing of suspended objects transported with robot manipulators, *Mechatronics*, vol.10, pp.609-626 .
- [19] A. G. Sreenatha, M. Pradhan (2002). Fuzzy logic controller for position control of flexible structures, *Acta Astronautica*, vol. 50, no. 11, pp. 665–671.
- [20] B Subudhi, A S Morris, (2003). Fuzzy and neuro-fuzzy approaches to control a flexible single-link manipulator, *Proc. Instn Mech. Engrs.*, vol.217, pp.387-399.
- [21] J. M. Renno, (2007). Inverse Dynamics Based Tuning of a Fuzzy Logic Controller for a Single-Link Flexible Manipulator, *Journal of Vibration and Control*, vol.13, pp.1741-1759.

- [22] M.A. Hasan, M.I. Ibrahimy, M.B.I. Reaz, (2009). A single link flexible manipulator control using fuzzy logic, *International journal of electronics*, vol.1, pp.13-21.
- [23] M.O. Tokhi, A.K.M. Azad, (2008). Flexible robot manipulators: Modeling, simulation and control, The Institution of Engineering Technology.
- [24] J.S.R. Jang, N. Gulley (1997). Matlab: Fuzzy Logic Toolbox, user's guide, The MathWorks Inc.
- [25] H.T. Nguyen, N. R. Prasad, C.L. Walker, E.A. Walker (2002). A first course in fuzzy logic and neural network control, Chapman & Hall/CRC.
- [26] Y.G. Tang, F.C. Sun, Z.Q. Sun, T.L. Hu, (2006). Tip position control of a flexible link manipulator with neural networks, *International journal of control, Automation, and Systems*, vol.4, no.3, pp.308-317.
- [27] H.A. Talebi, K. Khorasani, R.V. Patel, (1998). Neural network based control schemes for flexible-link manipulators: simulations and experiments, *Pergamon, Neural Networks*, vol.11, pp.1357-1377.
- [28] Z. Su, K. Khorasani, (2001). A neural network based controller for a single link flexible manipulator using the inverse dynamics approach, *IEEE Transactions on Industrial Electronics*, vol.48, no.6.
- [29] G. Öke, Y. Istefanopulas, (2006). End-effector trajectory control in a two link flexible manipulator through reference joint angle values modification by neural networks, *Journal of vibration and control*, vol.12, no.2, pp.101-117.
- [30] T. Haukaas, Euler-Bernoulli Beams, University of British Columbia, lecture notes
- [31] K. Ogata, (2002). Modern control engineering, 4<sup>th</sup> edition, Prentice Hall, Pearson Education International.
- [32] S. Kapucu, M. Kaplan, S. Bayseç, (2005).Vibration reduction of a lightly damped system using a hybrid input shaping method, *Proceedings of the Institution of Mechanical Engineers Part I, Journal of System and Control Engineering*, vol.219, no.5, pp.335-342.

[33] W.S. McCulloch, W. Pitts, (1943). A logical calculus of the ideas immanent in nervous activity, *Bull. of Math. Biophysics*, vol.5, pp. 115-133.

[34] L. J. KUN, (2011). Advanced PID control MATLAB, Electronics Industry Pub., 3<sup>rd</sup> edition .

## APPENDICES

### Appendix A

MATLAB codes which are used in chapter 5:

Main program codes:

```
close all
clear all
clc

global delta tp wi wp gama M input aa2
l=0.7;
m=0.1382;
E=71e9;
w=0.002;
h=0.0255;
I=(h*w^3)/12;

syms x

A=0.4251;      %%% wrt mode
beta=2.67;    %%% wrt mode

phi=A*((sin(beta*l)-sinh(beta*l))*(sin(beta*x)-
sinh(beta*x))+cos(beta*l)-cosh(beta*l))*(cos(beta*x)-
cosh(beta*x)));
gamma=-m*int(x*phi,0,l);
gama=eval(gamma);
wi=(beta*l)^2*sqrt(E*I/(m*l^4));      %%% wrt mode
```

```

M=m*int(phi^2,0,1);
M=eval(M);
%%
%%% INPUTS %%%

ti=2*pi/wi;
tp=ti*1.25;
tpp=tp/4;
delta=6/(12*(tpp^2)); %%% angular path

x0=[0 0];
dt=0.01; %time increment
t0=0;
tf=6; % total time
%%
for N=(1:tf/dt)
    %% solve the systems of odes until next frame
    [t,x]=ode45(@four_cyc_ramp,[t0,t0+dt],x0);
    x0=x(size(x,1),:);
    xs(N,:)=x0;
    aa2s(N,:)=aa2;
    clear x
    clear t
    inputs(N,:)=input;
    t0=t0+dt;
end
%%
tplot=0:dt:tf-dt;
figure
plot(tplot,inputs(1:N,1));grid on;
title('input')
figure
plot(tplot,xs(1:N,1));grid on;
title('output')

```

```

inputs=inputs';
input3(1,:)=tplot(1,:);
input3(2,:)=inputs(1,:);

```

**Function (four\_cyc\_ramp) codes:**

```

%Four piece ramp & cycloid for flexible beam
function xdot=four_cyc_ramp(t,x)
global delta tp wi wp gama M input aa2
tpp=tp/4;
wp=2*pi/tpp;
Y1=delta*(1-(wp/wi)^2);
Y2=delta*((wp/wi)^2);
Y=(Y1+Y2);
aa1=x(2);
if t<=tpp
    input=((Y1/(2*pi))*(wp*t-sin(wp*t))+Y2*t*wp/(2*pi));
    aa2=(-wi^2)*x(1)+(gama/M)*input;
elseif tpp<t && t<=2*tpp
    t=(t-tpp);
    input=(Y-((Y1/(2*pi))*(wp*t-
sin(wp*t))+Y2*t*wp/(2*pi)));
    aa2=(-wi^2)*x(1)+(gama/M)*input;
elseif 2*tpp<t && t<=3*tpp
    t=(t-2*tpp);
    input=-((Y1/(2*pi))*(wp*t-sin(wp*t))+Y2*t*wp/(2*pi));
    aa2=(-wi^2)*x(1)+(gama/M)*input;
elseif 3*tpp<t && t<=4*tpp
    t=(t-3*tpp);
    input=(-Y+((Y1/(2*pi))*(wp*t-
sin(wp*t))+Y2*t*wp/(2*pi)));

    aa2=(-wi^2)*x(1)+(gama/M)*input;
else

```



```

        input=0;
        aa2=(-wi^2)*x(1)+(gama/M)*input;
end
        xdot=[aa1 aa2]';

```

## Appendix B

MATLAB codes used in chapter 7:

```
%Single Neural Adaptive PID Controller
```

```
clear all;
```

```
close all;
```

```
x=[0,0,0]';
```

```
xiteP=9;
```

```
xiteI=0;
```

```
xiteD=1;
```

```
%Initilizing kp,ki and kd
```

```
wkp_1=100;
```

```
wki_1=7;
```

```
wkd_1=1;
```

```
%wkp_1=rand;
```

```
%wki_1=rand;
```

```
%wkd_1=rand;
```

```
error_1=0;
```

```
error_2=0;
```

```
y_1=1;y_2=0;y_3=0;
```

```
u_1=0;u_2=0;u_3=0;
```

```
ts=0.001;
```

```
A=[0 1 ; -441 -0.84]
```

```
B=[0;-7.6]
```

```
C=[1 0]
```

```

D=[0]
[b,a] = ss2tf(A,B,C,D);
sys=tf(b,a)
dsys=c2d(sys,ts,'z')
[num,den]=tfdata(dsys,'v')
for k=1:1:5000
    time(k)=k*ts;
    rin(k)=0;

    yout(k)=(-den(2)*y_1-den(3)*y_2+num(2)*u_1+num(3)*u_2);
    error(k)=rin(k)-yout(k);

%Adjusting Weight Value by hebb learning algorithm
M=3;
if M==1                %No Supervised Heb learning
algorithm
    wkp(k)=wkp_1+xiteP*u_1*x(1); %P
    wki(k)=wki_1+xiteI*u_1*x(2); %I
    wkd(k)=wkd_1+xiteD*u_1*x(3); %D
    K=100;
elseif M==2            %Supervised Delta learning algorithm
    wkp(k)=wkp_1+xiteP*error(k)*u_1; %P
    wki(k)=wki_1+xiteI*error(k)*u_1; %I
    wkd(k)=wkd_1+xiteD*error(k)*u_1; %D
    K=100;
elseif M==3            %Supervised Heb learning algorithm
    wkp(k)=wkp_1+xiteP*error(k)*u_1*x(1); %P
    wki(k)=wki_1+xiteI*error(k)*u_1*x(2); %I
    wkd(k)=wkd_1+xiteD*error(k)*u_1*x(3); %D
    K=1000;
elseif M==4            %Improved Heb learning algorithm
    wkp(k)=wkp_1+xiteP*error(k)*u_1*(2*error(k)-error_1);
    wki(k)=wki_1+xiteI*error(k)*u_1*(2*error(k)-error_1);
    wkd(k)=wkd_1+xiteD*error(k)*u_1*(2*error(k)-error_1);

```

```

K=1000;
end
x(1)=error(k)-error_1;           %P
x(2)=error(k);                 %I
x(3)=error(k)-2*error_1+error_2; %D

wadd(k)=abs(wkp(k))+abs(wki(k))+abs(wkd(k));
w11(k)=wkp(k)/wadd(k);
w22(k)=wki(k)/wadd(k);
w33(k)=wkd(k)/wadd(k);
w=[w11(k),w22(k),w33(k)];
    u(k)=u_1+K*w*x;           %Control law
if u(k)>150
    u(k)=150;
end
if u(k)<-150
    u(k)=-150;
end
error_2=error_1;
error_1=error(k);
    u_3=u_2;u_2=u_1;u_1=u(k);
y_3=y_2;y_2=y_1;y_1=yout(k);
    wkp_1=wkp(k);
wkd_1=wkd(k);
wki_1=wki(k);
end
figure(1);
plot(time,rin,'b',time,yout,'r');
xlabel('time(s)');ylabel('rin,yout');
%%
figure(2);
plot(time,error,'r');
xlabel('time(s)');ylabel('error');
figure(3);

```

```
plot(time,u,'r');
xlabel('time(s)');ylabel('u');
figure(4);
subplot(311);
plot(time,wkp,'r');
xlabel('time(s)');ylabel('kp');
subplot(312);
plot(time,wki,'r');
xlabel('time(s)');ylabel('ki');
subplot(313);
plot(time,wkd,'r');
xlabel('time(s)');ylabel('kd');
```