REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR- ANNABA UNIVERSITY
UNIVERSITE BADJI MOKHTAR ANNABA

جامعة باجي مختار- عنابـة

Année Universitaire : 2017/2018

Faculté: Sciences de l'Ingéniorat
Département: Electronique

Final year project
Presented to obtain a MSc Degree

Title : Control of a robot manipulator

Domaine : Sciences et Technologies
Filière : Electronique
Spécialité: Instrumentation

By CHIRATA BRENDA

Supervised by Pr N. Guersi

Jury :

President: Pr Abbassi. A Badji Mokhtar University of Annab

Examiners:
Pr Lakel.S Badji Mokhtar University of Annaba

Pr Khaldoun. Z Badji Mokhtar University of Annaba

Acknowledgements

I would like to thank God for giving me strength to pursue my studies in Algeria and most of all the gift of life because without it, i would never have made it thus far
I would also want to express my most sincere gratitude to my supervisor: Mr N Guerci for being always available throughout the period when i was working on this research paper.
I also want to thank the all the members of stuff in the department of electronics at University Badji Mokhtar of Annaba for doing all they did for me to get at this level.

Dedication

I also want to dedicate this work to my dad who passed away in November 2017, he was always my pillar of strength when school was hard for me in Algeria.
I won’t forget my mom   and all my siblings who helped me financially throughout my studies in Algeria

Summary

Abstract
Resumé
Table of figures
General Introduction

Chapter 1 Litterature review on robotics

Control of a robot manipulator

Abstract.
This research focuses on doing a comparative study between three ways to control a robot manipulator. These three ways of control are proportional-plus-derivative(PD), proportional-derivative-integrative (PID) and adaptive control. We look mainly at computed torque with proportional plus derivative(PD), computed torque control(CTC) with proportional-derivative-integrative(PID) and (CTC) with adaptive controllers. Given the parameters at each link of a robot manipulator, we derive a mathematical model of a robot arm manipulator with two degrees of freedom using the Lagrange equation of motion. We go on to discuss how this mathematical model is simulated using PD, PID and adaptive computed torque controllers. Finally the results of simulation in Matlab of these three types of control are shown and a comparison is made.

Resumé
Ce travail se concentre sur la réalisation d'une étude commandes de manipulateurs robotiques, soit le contrôle proportionnel plus dérivé (PD), le contrôle proportionnel-dérivé-intégratif (PID) et le contrôle adaptatif utilisant des contrôleurs de calculs de couple. Dans le premier chapitre, il y a une revue de la recherche bibliographie sur la robotique. Étant donné les paramètres à chaque articulation du manipulateur, nous dérivons un modèle mathématique d'un manipulateur de bras de robot avec deux degrés de liberté en utilisant l'équation de Langage de mouvement. Nous continuons à discuter de la façon dont ce modèle mathématique est simulé en utilisant trois différents types de contrôleurs de calculs de couple. Enfin les résultats de simulation dans Matlab de ces trois types de contrôle sont montrés et une comparaison est faite.

Key words :
Computed torque, PID, PD, controller, tracking, reference, control law, dynamics, stability, error, link, joints

Chapter 1: Litterature Review on Robotics

This research focuses on doing a comparative study between three robot manipulator control systems i.e proportional-plus-derivative(PD) control, proportional-derivative-integral(PID) control and adaptive control.

General Introduction

Robots are mechanical devices that can be programmed to follow a set of instructions. They have a processing unit, sensors to perceive their environment, and motors and actuators to help move their limbs or wheels. They may speak, make other sounds, or flash with lights and colours in response to the environment as per instructions. Robotics on the other hand is the study of robots and it is a vast domain that touches a lot of disciplines, some of which are not related to engineering**[1].** It combines various engineering disciplines like electronics, mechanics, computer sciences, etc. Robotic manipulators present a complex field of study related to kinematics, dynamics, computer vision and control. In this research, there will be a presentation of the analysis of the mathematical model and the simulation of a robotic manipulator with two degrees of freedom(2-DOF) . There will be use of the motion equations of Lagrange and Matlab code and Matlab Simulink to achieve this goal. The 2-DOF robotic arm is a dynamic nonlinear system that requires complex control methods. In the first chapter, we shall study the history and introduction of robots and their usage in the industry of today, the standard architecture of robots and different types of robots by different classification methods. In the second chapter we shall look at the dynamics of a two link robotic arm, how to derive the Lagrangian equations of motion using the potential and kinetic energies for each link. On the third chapter we discuss the 2-DOF robot arm control. We look at the importance of robot control, its applications and three types of robot arm manipulator control systems which are PD and PID and adaptative control. Finally on chapter four we compare the three types of robot control discussed in chapter three and compare the results and draw conclusions.

1.1 Introduction:

In this chapter there shall be a review of the history of robots. There shall be an in depth analysis of the general architecture of industrial robots and the different types of mobile robots that exist. There shall be a discussion of the different ways that can be used to classify robots.

1.2 History

Robots began as entertainment for royalty. Al-Jazari and Leonardo Da Vinci are people who were interested in inventing machines and they worked to build automatons for their benefactors. Al-Jazari built a floating band that resembled humans and performed a number of drum beats and songs that depended on the programming of a series of pegs. Da Vinci created an automaton based on the knight's armor. It could stand and move its arms, neck and mouth(it could open its mouth). In 1961 an inventor called George Devol installed his robot(its name was Unimate), into a General Motors factory in Trenton, New Jersey; that was the first attempt to use robots for modern industrial. Unimate would lift die-cut metal pieces and stack them for the human workers. This development changed the dynamics of robotics and brought them into the workplace, making them pivotal to a business**[1]**

Today robots have become highly reliable, dependable and technologically advanced factory equipment. Robots are now being used to undertake jobs that are dirty, dull or dangerous for humans. Robots perform tasks that can otherwise be impossible for humans or jobs that would require the effort of a lot of people or would have taken ages to complete. In the military, robots have changed the landscape of national defence. Robots have shown significance in decreasing human work load especially in industries. Industries are mostly used in manufacturing industries where the task performed are repetitive and monotonous and thus boring. Today the robotics industry have grown beyond comprehension and robots are being used in many places that we shall look at in the next section.

1.3 Robot Applications

Robots perform a lot of tasks in different fields and the amount of tasks delegated to them is rising everyday.

1.3.1 Industrial

These types of robots are mostly utilised in the manufacturing industries. In most cases they have arms particularly for usage in material handling, p, welding and others.

1.3.2 Domestic

These ones are used at home. They consists of numerous different gears that we can see in robotic sweepers, robotic sewer cleaners and other robots that can perform different household tasks.

1.3.3 Medical purposes

These are employed in medicine and medicinal institutes.

1.3.4 Service

These could be various data collecting robots, robots to exhibit technologies, robots used for research, etc.

1.3.5 Military

Robots that are used in the military and armed forces. They can be bomb discarding robots, shipping robots, exploration drones and so on.

1.3.6 Entertainment

Examples of these types of robots include robosapien , the running photo frames and articulated robot arms that are used as movement simulators.

1.3.7 Hobby and Competition

Robots that are created by students for fun or competition. They include Sumo-bots, Line followers, etc.

There are a lot of different types of robots. Some of them have two legged some are three legged, some robots can swim and others can fly. All these are interesting technological advancements in the are of robotics.In figure 1.1 we shall see a flying robot called drone. This is a robot of which a lot of research has been done. It is used mainly for surveillance.

Figure 1.1 Example of a flying robot

We shall look at the main components of all robots.

1.4 Components of industrial robots

Most industrial robots have five basic components:

1.4.1 Arm

The arm positions the sensors and end-effectors in a way that gives them the ability to do their pre-programmed tasks. Most robotic arms resemble human arms in that they have wrists, shoulders, elbows and fingers. The number of joints of a robot equals the number of its degrees of freedom. For example, if a robot has three degrees of freedom(3DOF) it means it can move in 3 ways; left and right, up and down and forward and backward. Most robots that are being used today have 6 DOF in order to reach all possible points in space within their work envelope.

1.4.2 Drive

The drive is the one which responsible to move the sections between the joints (links) into their desired position. Drives are usually powered by hydraulic, electric or pneumatic energy.

1.4.3 Controller

All robots are connected to computer controllers. Controllers regulate the components of the robotic arm and keep them working together. Controllers also allow the robot to be networked to other systems in order to work together with other machines. Most robots of today are pre-programmed but In the future, controllers with artificial intelligence (AI) should allow robots to think and program themselves.

1.4.4 Sensor

Sensors are responsible for sending information to the controller (information is sent in electrical signals). This information can be about the robot sorroundings where the controller is alerted of the exact position of the arm, or the state of the environment around the robot.

1.4.5 End Effector

The end-effector can be best imagined as the "hand" on the end of the robotic arm. End-effectors can be tools like tweezers, scalpel, blowtorch, welder, spray gun, or just about anything the arm to perform its job. Sometimes it is possible to change end-effectors and reprogrammed a robot for a completely different task. **[2]**

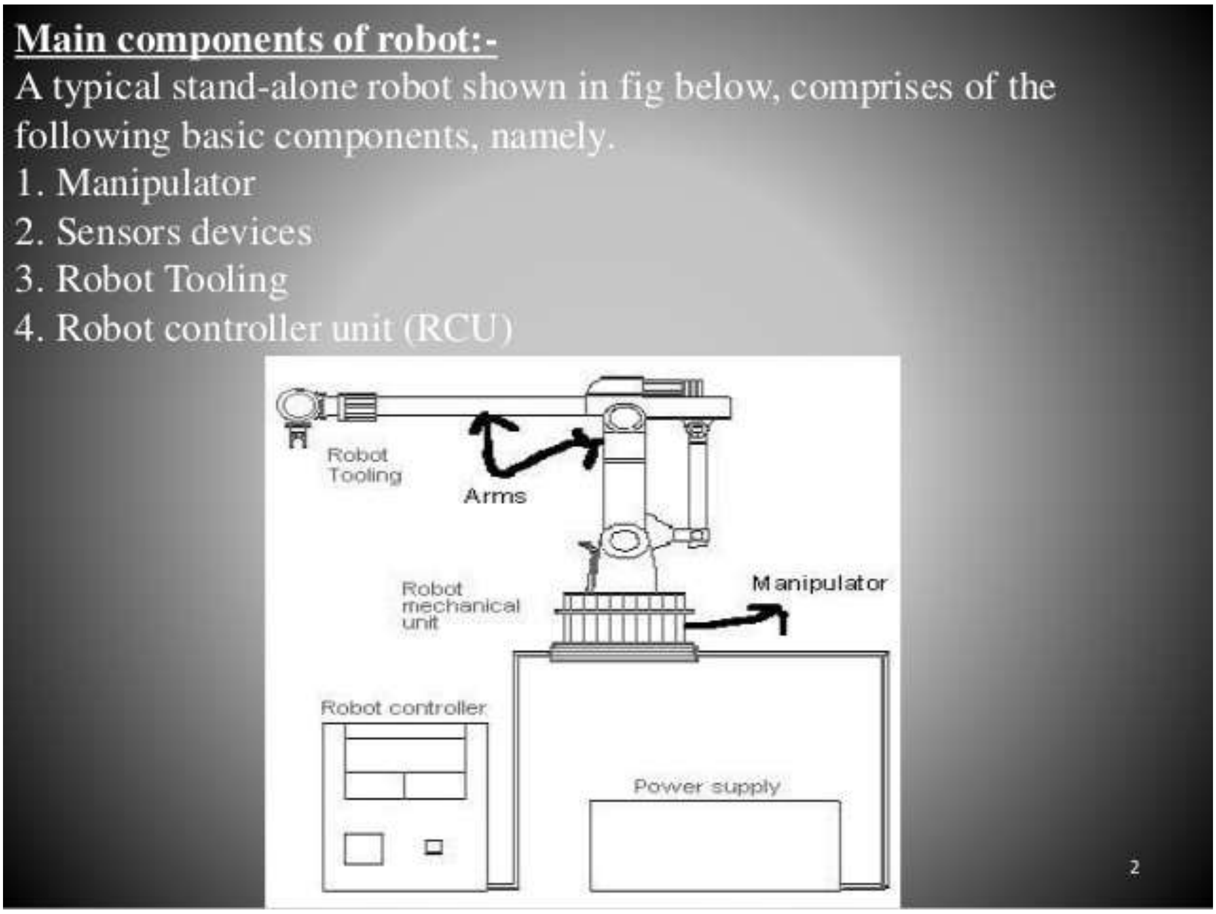


Figure 1.2: Main components of industrial robots [2]

1.5 The Mechanical System of a Robot

A system is a group of connected particles organised in a body of material or immaterial things **[Oxford dictionary]**. It can also be defined as an assembly of parts in formation of a unitary body. A system is said to be dynamic if it consist of; an input, output and a state. The state is a function of the input and the previous state **[3].** The state can only be defined at a specific time since the input and output at any given time is not constant. The state is a functional of the input, which is a property of dynamic systems. Meaning at any given time, the state of a dynamic system depends on the value of the input at that time and also on the past inputs(with the initial state excluded). From this we can deduce that dynamic systems have *memory.* Another aspect of dynamic systems is that the future inputs have no effect on the current state so we say that all dynamic systems are non anticipative or causal.
When a system is made up of electronic elements we say that its an electronic system and when its made up of mechanical elements its a mechanical system. In mechanical systems, inputs are the driving forces and moments that are exerted by the actuators and the environment. The outputs are the set of signals that are picked up by the sensors. **[2]**

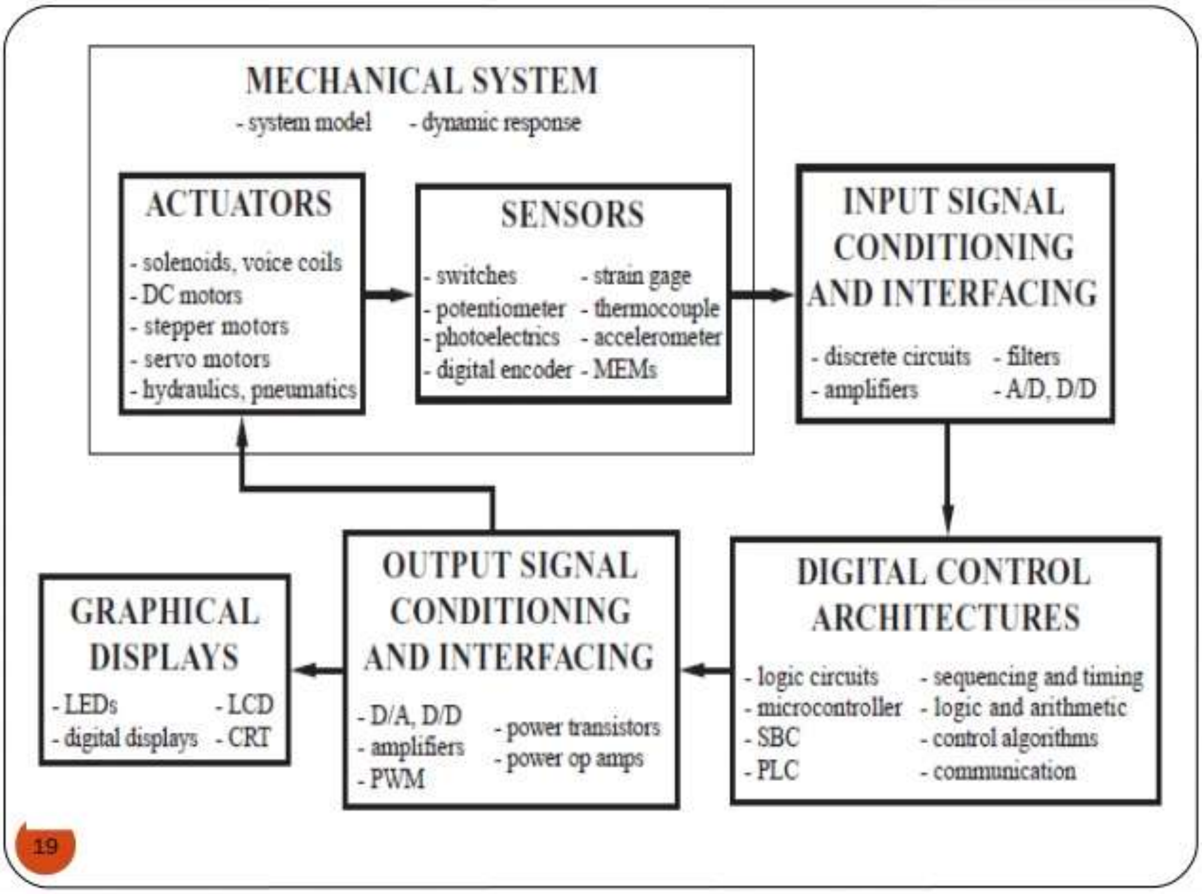


Figure 1.3: components of a robot mechanical structure

Throughout this research, we shall look at a robot as a mechanical system.

1.6 Robot classification

1.6.1 Types of Robots by Function

Robot classification is a controversial, by virtue of the intense activity displayed in the areas of robotics research, innovation, robot design and applications. A look at the Table of Contents of the Proceedings: 2005 IEEE International Conference on Robotics and Automation will reveal a vast spectrum of robots currently working on the shopfloor, in the operating room, in rehabilitation centers and in homes. In attempting a classification of robots, the most comprehensive criterion would be by function. We thus have a tentative, but by no means comprehensive, classification:**[4]**

- Manipulators: these are robotic hands and arms;
- motion generators: these are flight simulators
- mobile robots: legged and wheeled robots;
- swimming robots; and
- flying robots e.g drone. **[4]**

1.6.2 Types of Robots by Size

The most common type of robots under this criterion are macro-robots, or robots the dimensions of which are measured in meters. These are mainly used in the manipulation of heavy parts in automobile industries.

1.6.3 Types of Robots by Application

Robot applications have widespread as far as robot architectures. Current applications span the classical industrial robots for arc-welding, for example, on to material-handling, surveillance, surgical operations, rehabilitation and entertainment.
This research shall focus on robotic manipulator arms.

1.7 Robot Manipulators

1.7.1 History

These are mechanical systems that manipulate objects. To manipulate means to move something with one's hands, as the word comes from the Latin word manus, meaning hand **[5].** Manipulators are robotic mechanical systems that deserve attention for various reasons. One is that, in their simplest form, as robotic arms, they occur in industry. Another is that the architecture of robotic arms constitutes the simplest of all robotic architectures, and hence, appear as constituents of other complex robotic mechanical systems.. The basic idea behind the foregoing concept is that hands are among the organs that the human brain can control mechanically with the highest accuracy, as the work of an accomplished guitar player, or of a surgeon can attest**.[6]**
Manipulators are devices that help a human operator to perform a manipulating task. Although manipulators have existed ever since man created the first tool, only very recently, (end of World War II) have manipulators developed to the extent that they are now capable of actually mimicking motions of the human arm, and of the human hand. **[5]**
The difference between the early manipulator and the robotic manipulator of this day is the "robotic" qualifier, which came into the picture in the late sixties. A robotic manipulator can be distinguished from the early manipulator by its ability of lend itself to computer control. The early manipulator needed the presence of a human being to complete the work. Nowadays, in order to have a master manipulator perform a gesture, the robotic manipulator can be programmed once and for all to repeat the same task forever. These programmable manipulators have existed for close to 70 years, since the advent of the microprocessor.[6] It is the microprocessor which was introduced in 1976 by Intel which allowed a human master to teach the manipulator by actually driving the manipulator itself through a desired task, while recording all motions undergone by the master. That is how the manipulator would later repeat the identical task by mere playback.
The capabilities offered by robotic mechanical systems go well beyond the mere playback of pre-programmed tasks. Current research aims at providing robotic systems with software and hardware that will allow them to make decisions on the spot and learn while performing a task. **[7]**

1.7.2 General structure of a robot manipulator

The general structure of a robot manipulator, when considered in its working environment can be decomposed into five main components.

1.7.2.1 The mechanical structure

This part is made of rigid members or links articulated together through mechanical joints.

1.7.2.2. **The actuators**

These provide the mechanical power that helps to act on the mechanical structure against inertia, gravity and other external forces to modify the configuration and in turn, the geometric location of the tool. The actuators can be hydraulic, electric or pneumatic type and should be controlled in the appropriate manner.

1.7.2.3. The mechanical transmission devices

These devices connect and adapt the actuators to the mechanical structure. They transmit the mechanical efforts from the power sources to the mechanical joints and adapt the actuators to their load. Gear trains are a good example of mechanical transmission devices.

1.7.2.4. The sensors

They give senses to the robot. They can take for example the form of tactile, optical or electrical devices. They may be classified in the following two groups according to function:

1.7.2.4.1 Proprioceptive sensors

These provide information about the mechanical configuration of the manipulator itself (such as velocity and position information);

1.7.2.4.2 Exteroceptive sensors

They provide information about the environment of the robot (such as distance from an obstacle, contact force...)

1.7.2.5. The control unit

This assumes three different roles simultaneously and these roles are:

1.7.2.5.1 Information role

This consists of collecting and processing the information that is provided by the sensors.

1.7.2.5.2 A decision role

This role consists of planning the geometric motion of the manipulator structure starting from the task definition provided by the human operator and from the status of both the system and its environment transmitted by the sensors;

1.7.2.5.3 A communication role

This consists of organising the flow of information between the control unit, the manipulator and its environment.

1.8 The robot manipulator control unit

In order to assume the functions just described, the control unit has softwares and knowledge bases such as

1.8.1 A model (kinematic and/or dynamic) of the robot

The model expresses the relationship between the input commands to the actuators and the resulting motion of the structure.

1.8.2 A model of the environment

This describes the geometric working environment of the robot. It gives information such as the occurrence of zones where collisions are likely to occur and allows to plan the path accordingly.

1.8.3 Control algorithms

These govern the robot motion at a lower level and are responsible for the mechanical response of the structure and its actuators (assuming thus position and velocity control with prescribed accuracy and stability characteristics).

1.8.4 A communication protocol

Which assumes the management of the messages (in shape, priority...) exchanged between the various components of the system.

The control unit may have either a centralized architecture where the same processor assumes all the functions described above, or a hierarchical organization where the system is organized around a master unit that assigns to each one of the slave units some of the functions to be performed.**[7]**

1.9 Robot Control

The control of robotic manipulators is a mature yet fruitful area for research, development, and manufacturing. A useful robot is one that is able to control its movement and the interactive forces and torques between the robot and its environment. This research is concerned with the control aspect of robotic manipulators. To control usually requires the availability of a mathematical model and of some sort of intelligence to act on the model. The mathematical model of a robot is obtained from the basic physical laws governing its movement. Intelligence, on the other hand, requires sensory capabilities and means for acting and reacting to the sensed variables. These actions and reactions of the robot are the result of controller design.

.

1.10 Conclusion.

In this chapter we have seen the evolution of robots, their applications, the different types of robots that exist and their mechanical architecture. All these things have helped us to understand general information about robots.

Chapter 2: Control of a robot manipulator

2.1 Introduction

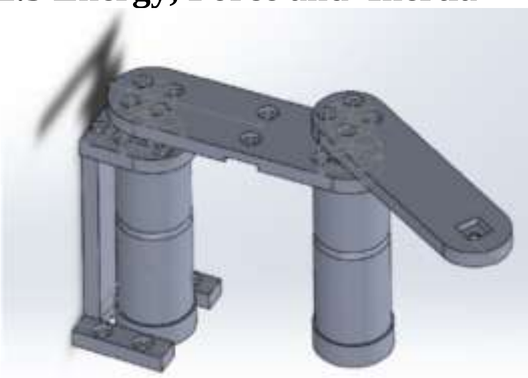
In this chapter we shall discuss how to come up with a dynamic model of a robot manipulator with two degrees of freedom. We shall use the potential and kinetic energies of the system and the equation of Lagrange to come up with this robot model. This dynamic model shall be derived from first principles.

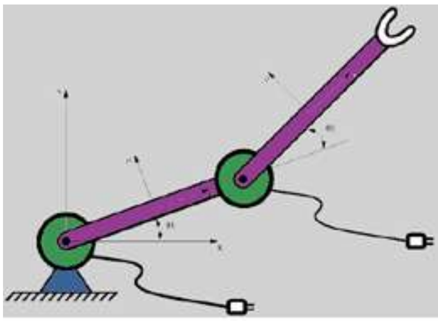
2.2 Overview of robot modelling

To better express the behaviour of a physical system, we should create an analytical model. Robot modelling and analysis essentially involve its kinematics. For any design to be rated as a good design, there must be solid proof that it is useful for the purpose it was made for. For control design purposes, it is mandatory to have a mathematical model that shows the dynamical behaviour of a system. In this section we shall derive the dynamic equations of motion for a robot manipulator. There is derivation of the the kinetic and potential energy of the manipulator and use of Lagrange's equations of motion.

In order for us understand the arm dynamics, we shall look at some basic physics concepts first.

2.3 Energy, Force and Inertia





Robot manipulator with 2 DC motors

Figure 2.1 robot manipulator with 2 degrees of freedom

2.4 The Centripetal and Coriolis forces

2.4.1 The centripetal force ($\boldsymbol{F_c}$)

When a mass m orbits a point at a radius *r* at an angular velocity **ω** , its centripetal force, $\boldsymbol{F_c}$ is given by **[4]**

$$\boldsymbol{F_c} = \frac{mv^2}{r} = m\boldsymbol{\omega}^2\mathbf{r} = m\dot{\theta}^2\mathbf{r} \qquad \textbf{(1)}$$

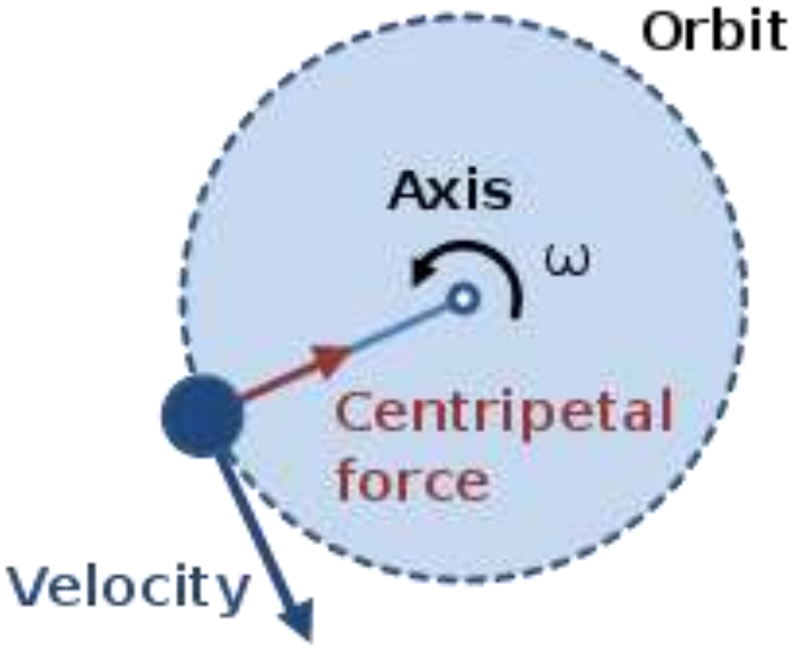


Figure 2.2: The Centripetal force

2.4.2 The Coriolis Force (F_{cor})

If a sphere is moving with an angular velocity ω_0 around its centre and a body of with a mass m is moving at a velocity of v on its surface then the coriolis force, F_{cor} exerted on this body is given by **[4]**

$$F_{cor} = -2m\omega_0 v \qquad (2)$$

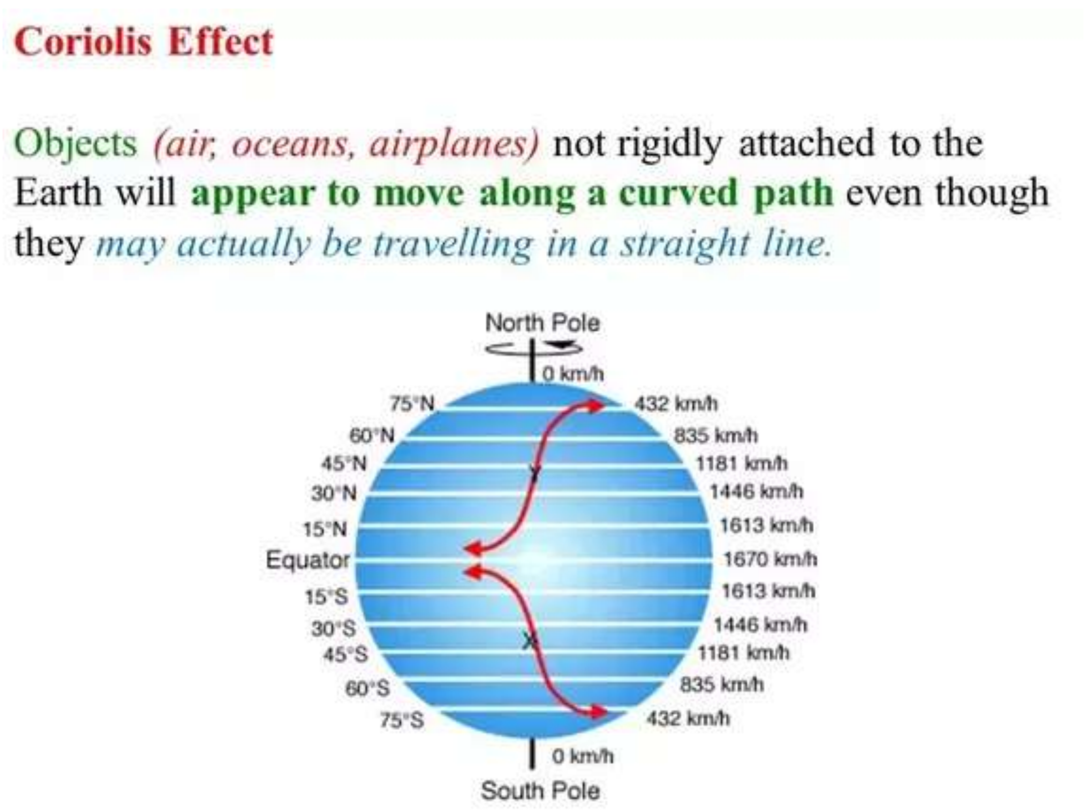


Figure 2.3: The Coriolis effect

2.5 Kinetic and Potential energy

2.5.1 Kinetic Energy

A mass, $\boldsymbol{m}$ that is moving with a linear velocity of $\boldsymbol{v}$ has kinetic energy, $\boldsymbol{K_E}$ given by the equation: **[9]**

$$K_E = \frac{1}{2} m v^2 \qquad (3)$$

If the body is rotating at an angular velocity, $\boldsymbol{\omega}$ and has a moment of inertia **I**, then the kinetic energy becomes: **[9]**

$$K_{Erot} = \frac{1}{2} I \omega^2 \qquad (4)$$

Where I is given by the equation

$$I = \int_{vol} \rho(r) r^2 dr \qquad (5)$$

Where $\rho(r)$ is the mass distribution at radius r in a volume. If r is a point mass, then I is given by

$$I = mr^2 \qquad (6)$$

And the kinetic energy of rotation becomes:

$$K_{Erot} = \frac{1}{2} m \omega^2 r^2 \qquad (7)$$

2.5.2 The potential energy

A mass, **m** at a height of **h** with a gravitational field **g** has a potential energy, $\boldsymbol{\rho}$ which is defined by: **[9]**

$$\rho = mgh \qquad (8)$$

2.6 Momentum, Angular momentum and Torque

The momentum (**p**) of a body with mass *m* which is moving at a velocity of *v* is given by **[9]**

$$\mathbf{p}=m\mathbf{v}. \qquad (9)$$

The angular momentum of a mass *m* that is moving around an origin from which the mass has distance *r* is

$$\boldsymbol{p}_{ang}= \mathbf{r}\times\mathbf{p}. \qquad (10)$$

The torque or moment (**N**) of a force , **F** with respect to the same origin is defined to be

$$\mathbf{N}=\mathbf{r}\times\mathbf{F}. \qquad (11)$$

2.7 Lagrange's Equations of Motion

In motion control, the dynamical model of robot manipulators is conveniently described by Lagrange dynamics.

Where $\boldsymbol{q}$ is an n×1 vector of generalized coordinates, $\boldsymbol{\tau}$ is an n×1 vector of generalized forces, and L the Lagrangian (the difference between the kinetic and potential energies).

The Lagrange's equation of motion is given by **[2]**

$$\frac{\boldsymbol{d}}{\boldsymbol{dt}}\frac{\boldsymbol{dL}}{\boldsymbol{d\dot{q}}}-\frac{\boldsymbol{dL}}{\boldsymbol{dq}}=\boldsymbol{\tau} \qquad \mathbf{(12)}$$

Where

$$L= K_E\text{-}\rho. \qquad (13)$$

2.7.1 Dynamics of a Polar Arm

The kinematics for a planar robotic arm with 2DOF are given below as a way to show use of physical principles given early in this chapter on a real physical system. Its dynamics is shown in figexamine Figure 6.

The joint-variable (q) and joint-velocity vector ($\dot{\boldsymbol{q}}$) are given by:

$$\mathbf{q}=\begin{bmatrix}\boldsymbol{\theta}\\ \boldsymbol{r}\end{bmatrix} \text{ and } \dot{\boldsymbol{q}} = \begin{bmatrix}\dot{\boldsymbol{\theta}}\\ \dot{\boldsymbol{r}}\end{bmatrix} \qquad (14)$$

While the general form of the corresponding force vector is given by

$$\boldsymbol{\tau} = \begin{bmatrix}\boldsymbol{n}\\ \boldsymbol{f}\end{bmatrix} \qquad (15)$$

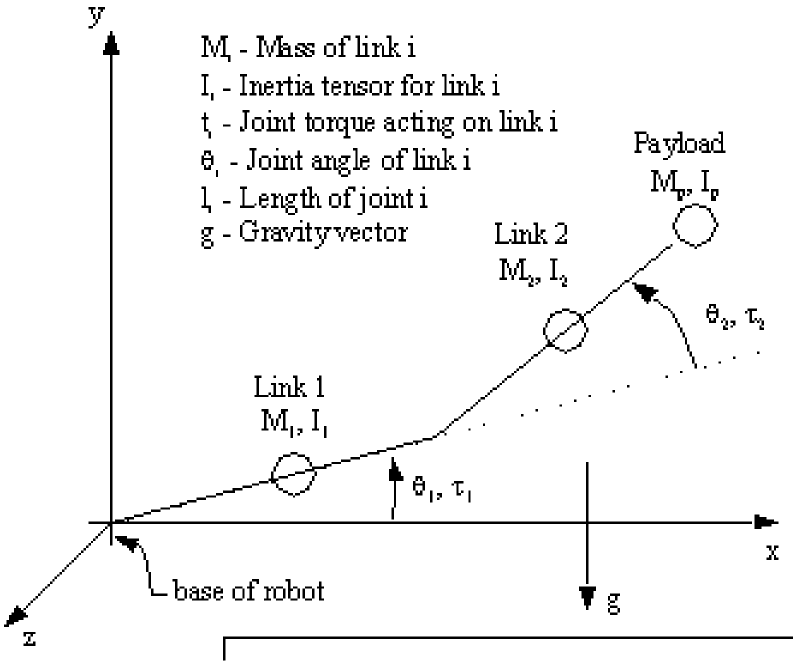


Figure 2.4: Dynamics of a planar robot arm

with *n* a torque and f force. The torque *n* and force *f* may be provided by motors or hydraulic actuators.

2.7.1.1 Potential and Kinetic Energy

The total kinetic energy for the system is given by:

$$K_{Etotal}= \frac{1}{2}m\dot{\theta}^2 r^2 + \frac{1}{2}mv^2$$

Where v=$\dot{r}$

The total potential energy of the system is;

$$\rho = mgh$$

Where **h= rsinθ**

2.7.1.2. Lagrange's Equation

The Lagrangian is given by L;

$$L= K_{Etotal} - \rho = \frac{1}{2}m\dot{\theta}^2 r^2 + \frac{1}{2}m\dot{r}^2 - mgrsin\theta$$

Now

$$\frac{dL}{d\dot{q}} = \begin{bmatrix} \frac{dL}{d\dot{\theta}} \\ \frac{dL}{d\dot{r}} \end{bmatrix} = \begin{bmatrix} m\dot{\theta}r^2 \\ m\dot{r} \end{bmatrix}$$

Implying

$$\frac{d}{dt}\frac{dL}{d\dot{q}} = \begin{bmatrix} m\ddot{\theta}r^2 + 2mr\dot{r}\dot{\theta} \\ m\ddot{r} \end{bmatrix}$$

$$\frac{dL}{dq} = \begin{bmatrix} -mgrcos\theta \\ mr\dot{\theta}^2 - mgsin\theta \end{bmatrix}$$

Thus the Lagrange equation of motion reduces to dynamic equations of the robotic arm that are given by;

$$m\ddot{\theta}r^2 + 2mr\dot{r}\dot{\theta} + mgrcos\theta = n$$

$$m\ddot{r} + mr\dot{\theta}^2 - mgsin\theta = f$$

From these 2 last equations of this example we can form a generalised form below

$$\boldsymbol{H}(\boldsymbol{q})\,\ddot{q} + \boldsymbol{C}(\boldsymbol{q}, \dot{q})\,\dot{q} + \tau_g(\boldsymbol{q}) = \tau \;, \qquad (16)$$

where $\boldsymbol{H}(\boldsymbol{q})$ represent the $(n \times n)$ inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{q})\,\dot{q}$ represent the $(n \times 1)$ vector of Coriolis and centrifugal forces, $\tau_g(\boldsymbol{q})$ is the $(n \times 1)$-vector of gravity force, and τ is the $(n \times 1)$-vector of joint control inputs to be designed.**[13]**

2.7.1.3 Manipulator Dynamics

Equation (16) can be written in vector form as follows

$$\begin{bmatrix} mr^2 & 0 \\ 0 & m \end{bmatrix}\begin{bmatrix} \ddot{\theta} \\ \ddot{r} \end{bmatrix} + \begin{bmatrix} 2mr\dot{r}\dot{\theta} \\ -mr\dot{\theta}^2 \end{bmatrix} + \begin{bmatrix} mgrsin\theta \\ mgcos\theta \end{bmatrix} = \begin{bmatrix} n \\ f \end{bmatrix} \qquad (25)$$

Remark

H(q), C(q,$\dot{q}$) and *G(q)* are all functions of *q*.

In our research we shall consider a 2 link robot arm shown in fig 2.4

2.8 Dynamics of a Two-Link Planar Elbow Arm

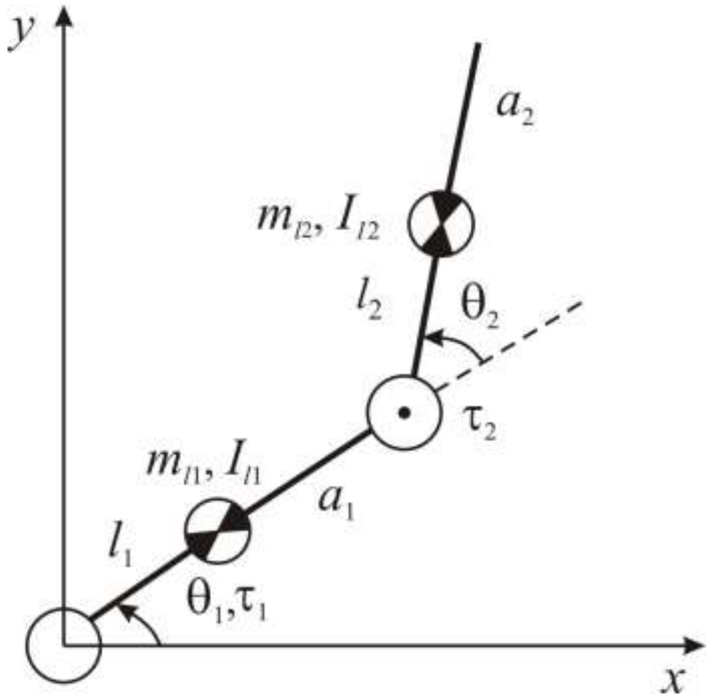


Figure 2.5: A 2 link planar elbow arm

The torques supplied by the actuators will be represented by τ_1 and τ_2
In the first link we have

$$x_1 = l_1 \cos\theta_1 \qquad \text{(II.1)}$$
$$y_1 = l_1 \sin\theta_1 ,$$

And in the second link we have

$$x_2 = (l_1 + a_1)\cos\theta_1 + l_2 \cos(\theta_1 + \theta_2)$$
$$y_2 = (l_1 + a_1)\sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) \qquad \text{(II.2)}$$

we are going to suppose that $l_1 + a_1 = l_{12}$
Of which

$$\begin{cases} \dot{x}_1 = -l_1 \dot{\theta}_1 \sin\theta_1, \\ \dot{y}_1 = l_1 \dot{\theta}_1 \cos\theta_1 , \end{cases} \quad \text{and} \quad \begin{cases} \dot{x}_1{}^2 = l_1{}^2 \dot{\theta}_1{}^2 \sin^2\theta_1, \\ \dot{y}_1{}^2 = l_1{}^2{}_1 \dot{\theta}_1{}^2 \cos^2\theta_1, \end{cases} \qquad \text{(II.3)}$$

In the same way;

$$\begin{cases} \dot{x}_2 = -(l_{12} \dot{\theta}_1 \sin\theta_1 + l_2 (\dot{\theta}_1 + \dot{\theta}_2) \sin(\theta_1 + \theta_2)), \\ \dot{y}_2 = l_{12} \dot{\theta}_1 \cos\theta_1 + l_2 (\dot{\theta}_1 + \dot{\theta}_2) \cos(\theta_1 + \theta_2)), \end{cases} \qquad \text{(II.4)}$$

Let’s suppose

$\theta_1 + \theta_2 = \emptyset$ and $\dot{\theta}_1 + \dot{\theta}_2 = \dot{\emptyset}$

That means

$$\begin{cases} \dot{x}_2 = -(l_{12}\,\dot{\theta}_1 \sin\theta_1 + l_2\,\dot{\emptyset}\sin\emptyset), \\ \dot{y}_2 = l_{12}\,\dot{\theta}_1 \cos\theta_1 + l_2\,\dot{\emptyset}\cos\emptyset), \end{cases} \quad \text{(II.5)}$$

$$\begin{cases} \dot{x}_2{}^2 = l_{12}{}^2\,\dot{\theta}_1{}^2 \sin^2\theta_1 + 2l_{12}\,l_2\,\dot{\emptyset}\dot{\theta}_1 \sin\emptyset \sin\theta_1 + l_2{}^2\dot{\emptyset}^2\sin^2\emptyset, \\ \dot{y}_2{}^2 = l_{12}{}^2\,\dot{\theta}_1{}^2 \cos^2\theta_1 + 2l_{12}\,l_2\,\dot{\emptyset}\dot{\theta}_1 \cos\emptyset cos\theta_1 + l_2{}^2\dot{\emptyset}^2\cos^2\emptyset, \end{cases} \quad \text{(II.6)}$$

2.9.1 Kinetic Energy, K_{E1}

For link 1 is given by

$$K_{E1} = \frac{1}{2} I_{l1}\dot{\theta}_1{}^2 + \frac{1}{2} m_{l1} v_1{}^2 \quad \text{(II.7)}$$

Where

$$v_1{}^2 = \dot{x}_1{}^2 + \dot{y}_1{}^2 = l_1{}^2\,\dot{\theta}_1{}^2$$

and for link 2 we have

$$K_{E2} = \frac{1}{2} I_{l2}\dot{\emptyset}^2 + \frac{1}{2} m_{l2} v_2{}^2 \quad \text{(II.8)}$$

where $v_2{}^2 = \dot{x}_2{}^2 + \dot{y}_2{}^2$

$$v_2{}^2 = l_{12}{}^2{}_1\,\dot{\theta}_1{}^2 + l_2{}^2\theta_1{}^2 + l_2{}^2\theta_2{}^2 + 2l_2{}^2\theta_1\theta_2 + 2l_{12}l_2\dot{\theta}_1{}^2\cos\theta_2 + 2l_{12}l_2\dot{\theta}_1\dot{\theta}_2\cos\theta_2 \quad \text{(II.9)}$$

The total kinetic energy for the system is therefore given by the sum of the two;

$$K_{Esys} = \frac{1}{2} I_{l1}\dot{\theta}_1{}^2 + \frac{1}{2} I_{l2}\dot{\emptyset}^2 + \frac{1}{2} m_{l1} v_1{}^2 + \frac{1}{2} m_{l2} v_2{}^2 \quad \text{(II.10)}$$

$$K_{Esys} = \frac{1}{2} I_{l1}\dot{\theta}_1{}^2 + \frac{1}{2} I_{l2}\dot{\theta}_1{}^2 + \frac{1}{2} I_{l2}\dot{\theta}_2{}^2 + I_{l2}\dot{\theta}_1\dot{\theta}_2 + \frac{1}{2} m_{l1} v_1{}^2 + \frac{1}{2} m_{l2} v_2{}^2 \quad \text{(II.11)}$$

By substitution velocities in (11) the total energy of the system becomes;

$$K_{Esys} = \frac{1}{2} I_{l1}\dot{\theta}_1{}^2 + \frac{1}{2} I_{l2}\dot{\theta}_1{}^2 + \frac{1}{2} I_{l2}\dot{\theta}_2{}^2 + I_{l2}\dot{\theta}_1\dot{\theta}_2 + \frac{1}{2} m_{l1} l_1{}^2\,\dot{\theta}_1{}^2 + \frac{1}{2} m_{l2}(l_{12}{}^2\,\dot{\theta}_1{}^2 + l_2{}^2\dot{\theta}_1{}^2 + l_2{}^2\dot{\theta}_2{}^2 + 2l_2{}^2\dot{\theta}_1\dot{\theta}_2 + 2l_{12}l_2\dot{\theta}_1{}^2\cos\theta_2 + 2l_{12}l_2\dot{\theta}_1\dot{\theta}_2\cos\theta_2) \quad \text{(II.12)}$$

We shall determine the potential energy as well

2.9.2 Potential Energy

Potential energy of the system is found from the equations below

$\boldsymbol{\rho_{sys}} = \boldsymbol{m_{l1}}g\boldsymbol{h_{l1}} + \boldsymbol{m_{l2}}g\boldsymbol{h_{l2}}$

Where

$$\begin{cases} \boldsymbol{h_{l1}} = \boldsymbol{y_1} = \boldsymbol{l_1}\mathbf{sin}\boldsymbol{\theta_1} \\ \boldsymbol{h_{l2}} = \boldsymbol{y_2} = \boldsymbol{l_{12}}\mathbf{sin}\boldsymbol{\theta_1} + \boldsymbol{l_2}\mathbf{sin}(\boldsymbol{\theta_1} + \boldsymbol{\theta_2}) \end{cases} \quad \text{(II.13)}$$

Thus

$$\boldsymbol{\rho_{sys}} = m_{l1}gl_1 sin\theta_1 + m_{l2}g(l_{12}sin\theta_1 + l_2 sin(\theta_1 + \theta_2)) \quad \textit{(II.14)}$$

Dermining the equation of Lagrange

$$L = K_{Esys} - \rho_{sys}$$

$$L = \frac{1}{2}I_{l1}\dot{\theta}_1^{\,2} + \frac{1}{2}I_{l2}\dot{\theta}_1^{\,2} + \frac{1}{2}I_{l2}\dot{\theta}_2^{\,2} + I_{l2}\dot{\theta}_1\dot{\theta}_2 + \frac{1}{2}m_{l1}l_1^2\,\dot{\theta}_1^{\,2} + \frac{1}{2}m_{l2}(l_{12}^2\,\dot{\theta}_1^{\,2} + l_2^2\dot{\theta}_1^{\,2}$$

$$+l_2^2\dot{\theta}_2^{\,2} + 2l_2^2\dot{\theta}_1\dot{\theta}_2 + 2l_{12}l_2\dot{\theta}_1^{\,2}\cos\theta_2 + 2l_{12}l_2\dot{\theta}_1\dot{\theta}_2\cos\theta_2)$$

$$- m_{l1}gl_1\sin\theta_1 - m_{l2}g(l_{12}sin\theta_1 + l_2 sin(\theta_1 + \theta_2)) \quad \text{(II.15)}$$

2.10 Determining the robot model

Having found the Lagrangian of the system, we now use the Euler- Lagrange equation to determine the robot model for a two link robot arm

$$\frac{d}{dt}\frac{dL}{d\dot{\theta}_i} - \frac{dL}{d\boldsymbol{\theta}_i} = \tau_i \quad \text{where } i = 1,2$$

$$\frac{dL}{d\boldsymbol{\theta_1}} = -m_{l1}g(l_1\cos\boldsymbol{\theta_1}) - m_{l2}g(l_{12}\cos\boldsymbol{\theta_1} + l_2\cos(\boldsymbol{\theta_1} + \boldsymbol{\theta_1})) \quad \text{(II.16)}$$

$$\frac{dL}{d\dot{\theta}_1} = I_{l1}\dot{\theta}_1 + I_{l2}\dot{\theta}_1 + I_{l2}\dot{\theta}_2 + m_{l1}l_1^2\dot{\theta}_1 + m_{l2}(l_{12}^2\dot{\theta}_1 + l_2^2\dot{\theta}_1 + l_2^2\dot{\theta}_2 + 2l_2l_{12}\dot{\theta}_1\cos\theta_2 + l_2l_{12}\dot{\theta}_2\cos\theta_2)$$

(II.17)

$$\frac{d}{dt}\frac{dL}{d\dot{\theta}_1} = I_{l1}\ddot{\theta}_1 + I_{l2}\ddot{\theta}_1 + I_{l2}\ddot{\theta}_2 + m_{l1}l_1^2\ddot{\theta}_1 + m_2(l_{12}^2\,\ddot{\theta}_1 + l_2^2\,\ddot{\theta}_1 + l_2^2\,\ddot{\theta}_2 + 2l_2l_{12}\,\ddot{\theta}_1\cos\theta_2$$

$$- 2l_2l_{12}\dot{\theta}_1\dot{\theta}_2\sin\theta_2 + l_{12}l_2\ddot{\theta}_2\cos\theta_2 - l_{12}l_2\dot{\theta}_2^{\,2}\sin\theta_2) \quad \text{(II.18)}$$

Which gives us

$$\frac{d}{dt}\left(\frac{dL}{d\dot{\theta}_1}\right) - \frac{dL}{d\theta_1} = (I_{l1} + I_{l2} + m_{l1}l_1^2 + m_{l2}(l_{12}^2 + l_2^2 + 2l_2l_{12}\cos\theta_2)\,\ddot{\theta}_1$$

$$+(I_{l2} + m_{l2}(l_2^2 + l_2l_{12}\cos\theta_2)\,\ddot{\theta}_2 + (-m_{l2}l_2l_{12}\sin\theta_2)\dot{\theta}_2^{\,2}$$

$$+2(-m_{l2}l_2l_{12}\sin\theta_2)\,\dot{\theta}_1\dot{\theta}_2 + m_{l1}gl_1\cos\theta_1 + m_{l2}g(l_{12}\cos\theta_1 + l_2\cos(\theta_1 + \theta_2)) \quad \text{(II.19)}$$

In the same way, we have

$$\frac{dL}{d\theta_2} = -m_{l2}(l_2l_{12}\dot{\theta}_2^{\,2}\sin\theta_2 + l_2l_{12}\dot{\theta}_1\dot{\theta}_2\sin\theta_2) + m_{l2}gl_2\cos(\theta_1 + \theta_2) \quad \text{(II.20)}$$

$$\frac{dL}{d\dot{\theta}_2} = I_{l2}\dot{\theta}_2 + I_{l2}\dot{\theta}_1 + m_{l2}(l_2^2\dot{\theta}_2 + l_2^2\dot{\theta}_1 + l_{12}\,l_2\dot{\theta}_1\cos\theta_2) \quad \text{(II.21)}$$

$\frac{d}{dt}\left(\frac{dL}{d\dot{\theta}_2}\right) = I_{l2}\ddot{\theta}_2 + I_{l2}\ddot{\theta}_1 + m_{l2}\,({l_2}^2\ddot{\theta}_2 + {l_2}^2\ddot{\theta}_1 + l_{12}l_2\ddot{\theta}_1 cos\theta_2 - l_{12}l_2\dot{\theta}_1\dot{\theta}_2 sin\theta_2)$ (II.22)

$\frac{d}{dt}\left(\frac{dL}{d\dot{\theta}_2}\right) - \frac{dL}{d\theta_2} = \left(I_{l2} + m_{l2}({l_2}^2 + l_{12}\, l_2\cos\theta_2\,)\right)\ddot{\theta}_1 + (I_{l2} + m_{l2}{l_2}^2)\,\ddot{\theta}_2 +$
$(\,m_{l2}l_{12}l_2 \sin\theta_2)\,{\dot{\theta}_1}^2$
$+\; m_{l2}gl_2\cos(\theta_1 + \theta_2\,)$
(II.23)

From the Lagrange equation, we can deduce τ_i by replacing it with its equivalent from the equations above;

$\tau_1 = \frac{d}{dt}\left(\frac{dL}{d\dot{\theta}_1}\right) - \frac{dL}{d\theta_1}$ and $\tau_2 = \frac{d}{dt}\left(\frac{dL}{d\dot{\theta}_2}\right) - \frac{dL}{d\theta_2}$ (II.24)

In robotics if the movement of the robot manipulator is circular, the torque or force can be expressed in matrix form (see below); this is the dynamic model of the robot.

$\tau_i = \mathrm{M}(\theta)\,\ddot{\theta} + \mathrm{C}(\theta, \dot{\theta}\,)\,\dot{\theta} + \mathrm{G}(\theta)$ (i = 1,2) (II.25)

For a system with 2DOF, this standard form can be developed to:

$$\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix} = \begin{bmatrix}M_{11} & M_{12}\\ M_{21} & M_{22}\end{bmatrix}\begin{bmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{bmatrix} + \begin{bmatrix}2h_{112}\dot{\theta}_2 & h_{122}\dot{\theta}_2\\ h_{211}\dot{\theta}_1 & 0\end{bmatrix}\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix} + \begin{bmatrix}g_1\\ g_2\end{bmatrix} \quad \text{(II.26)}$$

Of which

$$\begin{cases}\tau_1 = M_{11}\ddot{\theta}_1 + M_{12}\ddot{\theta}_2 + 2h_{112}\dot{\theta}_2\dot{\theta}_1 + h_{122}{\dot{\theta}_2}^2 + g_1\\ \tau_2 = M_{21}\ddot{\theta}_1 + M_{22}\ddot{\theta}_2 + h_{211}{\dot{\theta}_1}^2 + g_2\end{cases} \quad \text{(II.27)}$$

Equating this standard form to the robot dynamic model found(b), we can show that

$$M_{11} = I_{l1} + I_{l2} + m_{l1}{l_1}^2 + m_{l2}({l_{12}}^2 + {l_2}^2 + 2l_{12}l_2 cos\theta_2)$$
$$M_{12} = M_{21} = I_{l2} + m_{l2}({l_2}^2 + l_{12}l_2 cos\theta_2)$$
$$M_{22} = I_{l2} + m_{l2}{l_2}^2$$
$$h_{122} = h_{112} = -h_{211} = -m_{l2}l_{12}l_2 sin\theta_2$$
$$g_1 = m_{l1}g(l_1 cos\theta_1) + m_{l2}g(l_{12}cos\theta_1 + l_2\cos(\theta_1 + \theta_2))$$
$$g_2 = m_{l2}gl_2\cos(\theta_1 + \theta_2)$$

Thus

$$\mathrm{M(q)} = \begin{bmatrix} I_1 + I_2 + m_{l1}{l_1}^2 + m_{l2}({l_{12}}^2 + {l_2}^2 + 2l_{12}l_2 cos\theta_2) & I_2 + m_{l2}({l_2}^2 + l_{12}l_2 cos\theta_2)\\ I_2 + m_{l2}({l_2}^2 + l_{12}l_2 cos\theta_2) & I_2 + m_{l2}{l_2}^2\end{bmatrix}$$

The vector of Coriolis and centrifuge is given by;

$$\mathrm{C}(\theta, \dot{\theta})\,\dot{\theta} = \begin{bmatrix} -2m_{l2}l_{12}l_2\dot{\theta}_1\dot{\theta}_2 sin\theta_2 - m_{l2}l_{12}l_2{\dot{\theta}_2}^2 sin\theta_2\\ m_{l2}l_{12}l_2{\dot{\theta}_1}^2 sin\theta_2\end{bmatrix}$$

The vector of gravitational forces is given by

$$\mathrm{G}= \begin{bmatrix} m_{l1}gl_1 cos\theta_1 + m_{l2}gl_{12}cos\theta_1 + m_{l2}gl_2 cos(\theta_1 + \theta_2)\\ m_{l2}gl_2 cos(\theta_1 + \theta_2)\end{bmatrix}$$

It is worth noting that, in our derivation of the robot model we were not considering any disturbances that may occur to our robot system(e.g friction). However in reality, we have these disturbances and our robot model has to take these into consideration. This is why we have this equation below.

The dynamics in joint space of a serial-chain *n*-link robot manipulator considering the presence of friction at the robot joints can be written as:

$\tau = M(q)\ddot{q} + \mathrm{C}(\mathrm{q},\dot{q})\,\dot{q} + \mathrm{F}(\dot{q}) + G(q) + \tau_d$

where $M(q)$ is the $n \times n$ symmetric positive definite inertia matrix, $C(q,\dot{q})\,\dot{q}$ is the $n \times n$ vector of centripetal and Coriolis torques, $g(q)$ is the $n \times 1$ vector of gravitational torques $F(\dot{q}) = \mathrm{diag}\{ f_{v1}, \ldots, f_{vn}\}$ is the $n \times n$ positive definite diagonal matrix which contains the viscous friction coefficients of the robot joints, and τ is the $n \times 1$ vector of applied torque inputs.

2.11 Conclusion

To summarise; the general form of the robot arm dynamical equation is obtained by determining the arm kinetic and potential energies, followed by finding the Lagrangian, and then finally substituting into Lagrange's equation of motion to obtain the final equation which is referred to as the robot dynamical equation of motion. We have derived the robot equation of motion from first principles. This equation however assumes that we will be working in an ideal environment , which is impractical. We have finally shown the mathematical model of robot that takes errors of the system and from the outside environment into consideration.

Chapter 3 Control and Simulation

3.1 Introduction:

In this chapter we are going to look at a classic way of robot control called computed torque control. We will use two controllers, proportional plus derivative (PD) controller and proportional-derivative-integral (PID) controller. We shall also look at adaptive control and see how to do simulation using these three types of control.

3.2 Overview of Robot control

The control systems theory is a very interesting subject in robotics. In general all robots require some sort of control systems theory. Over the years, different methods for control have been developed. These different methods of control can be divided into two categories namely Classical Control (CC) and Intelligent Control (IC).

In the development of a CC system to control a plant, the designer constructs a mathematical model of the system. The model should contain all the dynamics of the plant that affects controlling it. This control type is called the Mathematicians Approach, since it involves use of a mathematical model in order to attain the control.

In developing an IC system to control for a plant, the designer inputs the system behaviour and the IC system abstractly models the system. This type of control is called the Lazy man's Approach because the designer doesn't need to know the internal dynamics of the plant to be controlled. IC is usually used in cases where construction of a system model is too complex to be modelled**. [11]**

In this chapter we shall discuss classical control since we have already derived our mathematical model in the chapter 2. We shall look at different types of classical control first.

3.3 The control system theory

Control is the process of forcing a system output variable to conform to some desired value, called reference value. It is achieved through use of controllers.
In systems control, a controller tries to manipulate the inputs of the system to realize desired behaviour at the output of the system**.[13]**
In control systems engineering, the control theory deals with the control of continuously operating dynamical systems in engineered processes and machines. The objective is to develop a control model for controlling such systems through use of a control action in an optimum manner without delay or overshoot and ensuring control stability.[14]
To achieve this, a controller with the requisite corrective behaviour is required. The controller monitors a controlled process variable (PV), and compares it with the reference (set point) (Ref). The difference between actual and desired value of the PV is called the *error* signal., or Ref-PV error. This error is applied as feedback to generate a control action to bring the controlled PV to the same value as the set point. **[15]**

Controlling a robot or other complex intelligent machine requires that it have a plan or model of what it expects to accomplish. Like all such plans, it is quite likely to require modification within moments of the beginning of its execution, but it is still essential. The purpose of a control system is to compare the plan to reality, and to issue commands to the servos or other output devices to make reality follow the plan.

Controller is a device which can sense information from linear or nonlinear system (e.g., robot manipulator) to improve the systems performance [3]. The main targets in designing control systems are stability, good disturbance rejection, and small tracking error[5]. Several industrial robot manipulators are controlled by linear methodologies (e.g., Proportional-Derivative (PD) controller, Proportional- Integral (PI) controller or Proportional- Integral-Derivative (PID) controller), but when robot manipulator works with various payloads and have uncertainty in dynamic models this technique has limitations. From the control point of view, uncertainty is divided into two main groups: uncertainty in unstructured inputs (e.g., noise, disturbance) and uncertainty in structure dynamics (e.g., payload, parameter variations). In some applications robot manipulators are used in an unknown and unstructured environment, therefore strong mathematical tools used in new control methodologies to design nonlinear robust controller with an acceptable performance (e.g., minimum error, good trajectory, disturbance rejection). Joint space and operational space control are closed loop controllers which they have been used to, provide robustness and rejection of disturbance effect. The main target in joint space controller is design a feedback controller that allows the actual motion tracking of the desired motion. This control problem is classified into two main groups. Firstly, transformation the desired motion to joint variable by inverse kinematics of robot manipulators [6].

3.4 Types of robot control

3.4.1 Passive Control

This type of control is characterised by the following properties;

- It either has no actuation or is under-actuated
- it structurally modifies the plant (robot) dynamics
- it is used when viable: cheap, robust

3.4.2 Open Loop Control

Open loop control has the following characteristics

- It has actuation but has no sensing
- It exploits the knowledge of a system
- dynamics to compute appropriate inputs
- It requires very accurate model of plant(robot) dynamics

3.4.3 Active (Feedback) Control

- Is mostly exploited for autonomous robots
- It uses sensors and actuators that are connected by a computer to modify dynamics

- It allows for uncertainty and noise modelling**.[13]**

The types of controllers that we shall use for our simulation are of modern control type. This means they have the plant, controller and feedback. While there are a lot of control types again in this category, this work shall only concentrate on computed torque control (PD and PID).

3.5 Computed Torque control

Computed torque controller (CTC) is a powerful nonlinear controller. It is widely used in control of robot manipulator. This controller works very well when all dynamic and physical parameters are known but when the robot manipulator has variation in dynamic parameters, in this situation the controller has no acceptable performance **[14].** Many types of robot control can be considered as special cases of the computed-torque controllers. Computed torque, is a special application of feedback linearization of nonlinear systems. As a matter of fact, one way to classify robot control schemes is to divide them as “computed-torque-like” or “non computed-torque-like.”**[3].** In practice, most physical systems (e.g., robot manipulators) parameters are unknown or time variant, therefore, computed torque like controller used to compensate dynamic equation of robot manipulator**[4, 6].** Computed-torque control allows us to conveniently derive very effective robot controllers, and some modern design techniques. Many digital robot controllers are also computed-torque-like controllers

3.6 Control: Computed torque analysis

A basic problem in controlling robots is to make the manipulator follow a pre-planned desired trajectory. We must position our robot in the right place at the right instances before the robot starts to do any useful work

In this chapter we shall discuss computed-torque control (CTC). This is a family of easy-to-understand control schemes that often work well in practice. These schemes involve the decomposition of the controls design problem into an inner-loop design and an outer-loop design. Before we can control a robot arm, we must to know the desired path for performing a task. We shall assume that our main objective is to move the robot along a prescribed desired trajectory. We shall show how to reconstruct a continuous desired path from a given table of desired points the end effector should pass through. we demonstrate this by showing how to simulate robot controllers on a computer using matlab. This should be done to verify the effectiveness of any proposed control scheme prior to actual implementation on a real robot manipulator. We assume that there is given a prescribed path q_d(t) that the robot arm is supposed to follow. We design control schemes that make the manipulator follow this desired path or trajectory.

3.7 The inner and outer loop for computed torque controller

3.7.1 The inner loop

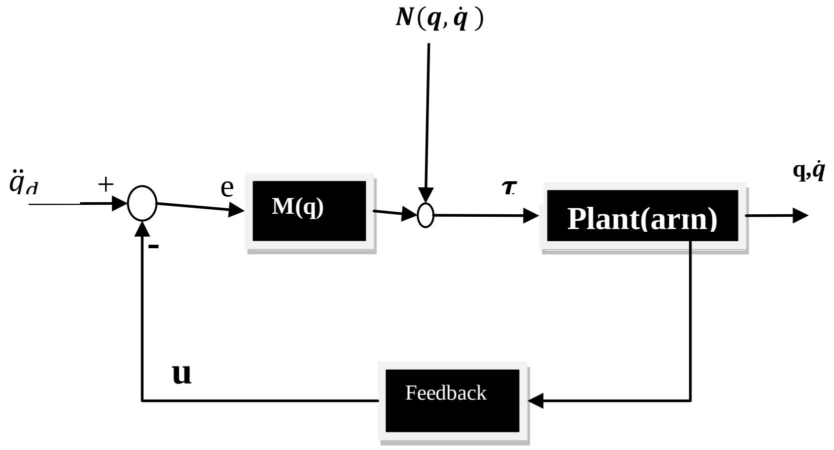


Figure 3.2: CTC inner and outer loop

The dynamics of the robot joints can be written as:

$\tau = M(q)\ddot{q} + C(q,\dot{q})\,\dot{q} + F(\dot{q}) + G(q) + \tau_d$ (III.1)

Or $\tau = M(q)\ddot{q} + N(q,\dot{q}) + \tau_d$ (III.2)

where *M(q)* is the n×n symmetric positive definite inertia matrix, $C(q,\dot{q})\,\dot{q}$ is the n×1 vector of centripetal and Coriolis torques, g(q) is the $n \times 1$ vector of gravitational torques, $Fv = \text{diag}\{fv1, \ldots, fvn\}$ is the n × 1 positive definite diagonal matrix which contains the viscous friction coefficients of the robot joints, and τ is the n × 1 vector of applied torque inputs.**[15]**
The tracking error can be defined as e;

$$e = q_d - q \quad \text{(III.3)}$$

where q_d(t) is the desired arm trajectory and q(t) the actual arm trajectory.
Use of (III.2) above and finding second derivative of (III.3) proves that $\ddot{e}$ is:

$$\ddot{e} = \ddot{q}_d + M^{-1}(N + \tau_d - \tau) \quad \text{(III.4)}$$

This acceleration error can be split to get the control input function and the disturbance function which are given by u and ω respectively;

$$u = \ddot{q}_d + M^{-1}(N - \tau) \quad \text{(III.5)}$$

$$\text{and } \omega = \ddot{q}_d + M^{-1}(\tau_d) \quad \text{(III.6)}$$

implying $\ddot{e} = u + \omega$

a state x(t) is defined by ;

$$x(t) = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \quad \text{(III.7)}$$

and use this matrix together with u and ω to define tracking error dynamics;

$$\frac{d}{dt}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u + \begin{bmatrix} 0 \\ I \end{bmatrix} \omega \quad \text{(III.8)}$$

Making τ *in equation* (III.5) *the subject of the formula* gives **the computed torque control law**

$\tau = M(\ddot{q}_d - u) + N$ (III.9)

If we select u(t) that stabilizes (III.8) so that e(t) goes to zero, then the nonlinear control input given by $\tau(t)$ in (III.9) will cause trajectory following in the robot arm (III.1) **[3]**. In fact, substituting (III.9) into (III.2) yields

$$M(q)\,\ddot{q} + N + \tau_d = M(\ddot{q}_d - u) + N$$

or

$$\ddot{e} = u + M^{-1}\tau_d \quad \text{(III.10)}$$

(8) can be easily stabilised. This is achieved by nonlinear transformation of (III.5) into a simple design problem for a linear system consisting of n decoupled subsystems, each one of which obey Newton's laws. We shall see several ways for selecting u(t). Since u(t) depends on q(t) and $\dot{q}(t)$, then the outer loop is a feedback loop. In general, we may select a dynamic compensator H(s) so that

$$U(s)=H(s)E(s) \quad \text{(III.11)}$$

H(s) can be selected for good closed-loop behaviour. According to (III.10), the closed-loop error system then has transfer function

$$T(s)=S^2 I - H(s) \quad \text{(III.12)}$$

NB: (III.9) shows that $\tau(t)$ is computed by substituting $\ddot{q}_d - u$ for $\ddot{q}$ in (III.2), meaning by solving the robot Newton-Euler inverse dynamics.
The outer-loop signal u(t) can be chosen by robust and adaptive control techniques. In the next section, we shall see ways to choose u(t) and some variations on computed-torque control.

3.7.2 PD Outer-Loop Design

We can use the proportional-plus derivative (PD) feedback as a way to select u(t)

$$u= -K_v\dot{e} - K_p e \quad \text{(III.15)}$$

Which means the computed torque control law , (III.9) becomes

$$\tau = Mq(\ddot{q}_d + K_v\dot{e} + K_p e) + N(q,\dot{q}) \quad \text{(III.14)}$$

The closed loop error dynamics becomes ;

$$K_v\dot{e} + K_p e + \ddot{e} = \omega \quad \text{(III.15)}$$

Which can be written in state form as ;

$$\frac{d}{dt}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix}\omega \quad \text{(III.16)}$$

We can then get the characteristic polynomial of the closed loop from

$$Cp(s) = \| s^2 I + K_v s + K_p \| \quad \text{(III.17)}$$

Choosing PD Gains
Gain matrices should have the dimensions of the number of degree of liberty of our robot model (in our case gains should have 2 dimensions). And these matrices should be diagonal matrices;

$$K_v = diag\{K_{V_i}\} \text{ and } K_p = diag\{K_{p_i}\} \quad \text{(III.18)}$$

Implying

$$Cp(s) = \prod (s^2 + K_{V_i}s + K_{p_i}) \quad \text{(III.19)}$$

If we keep all the values of K_{V_i} *and* K_{p_i} positive, then the system error will be asymptotically stable. Also, if $\omega(t)$ *is kept within a certain range* (if it is bounded), e(t) will also be bounded.
Note: boundedness of $\omega(t)$ is due to boundedness of $\tau_d(t)$
The characteristic polynomial can be written in standard form as :

$$p(s) = s^2 + 2\xi\omega_n s + \omega_n^2 \quad \text{(III.20)}$$

where
ω_n is the natural frequency and ξ is the damping ratio. Now to achieve desired performance in error, e(t), the gain must be:

$$K_{p_i} = \omega_n^2 \quad \text{and} \quad K_{V_i} = 2\xi\omega_n \quad \text{(III.21)}$$

Where ξ is the desired damping ratio and ω_n is the natural frequency of joint error i
These gains are selected for ξ = 1, implying

$$K_{p_i} = K_{v_i}{}^2/4 \quad \text{and} \quad K_{v_i} = 2\sqrt{K_{p_i}} \qquad \text{(III.22)}$$

ω_n should be large in order to have fast responses and should be selected by taking into consideration performance objectives.
The detailed discription of the outer loop design is described by Frank Lewis[3]. Here we shall use our experiment to show how its done.

3.8 Simulation

3.8.1 PD Computed-Torque Control law

In this section we shall discuss the detailed mechanics of simulating a PD computed-torque controller using MatLab.
The dynamic model of our robot is given by;

$$\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix} = \begin{bmatrix}(m_{l1}+m_{l2})\,l_2{}^2 + m_{l2}l_2{}^2 + 2m_{l2}l_{12}l_2 cos\theta_2 & m_{l2}l_2{}^2 + m_{l2}l_{12}l_2cos\theta_2\\ m_{l2}l_2{}^2 + m_{l2}l_{12}l_2cos\theta_2 & m_{l2}l_2{}^2\end{bmatrix}\begin{bmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{bmatrix} + \begin{bmatrix}-m_{l2}l_{12}l_2\left(2\,\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2{}^2\right)\sin\theta_2\\ m_{l2}l_{12}l_2\dot{\theta}_1{}^2 sin\theta_2\end{bmatrix} + \begin{bmatrix}(m_{l1}+m_{l2})gl_{12}cos\theta_1 + m_{l2}gl_2\cos(\theta_1+\theta_1)\\ m_{l2}l_2l_2\cos(\theta_1+\theta_1)\end{bmatrix} \qquad \text{(III.23)}$$

The standard form of which is ;

$$\tau = M(q)\,\ddot{q} + C(q,\dot{q})\dot{q} + G(q) \qquad \text{(III.24)}$$

The PD computed torque control law is given by;

$$\tau = M(q)(\ddot{q}_d + K_v\dot{e} + K_p e) + N(q,\dot{q}) \qquad \text{(III.25)}$$

Figure 3.2 below shows computed torque control with proportional plus derivative.

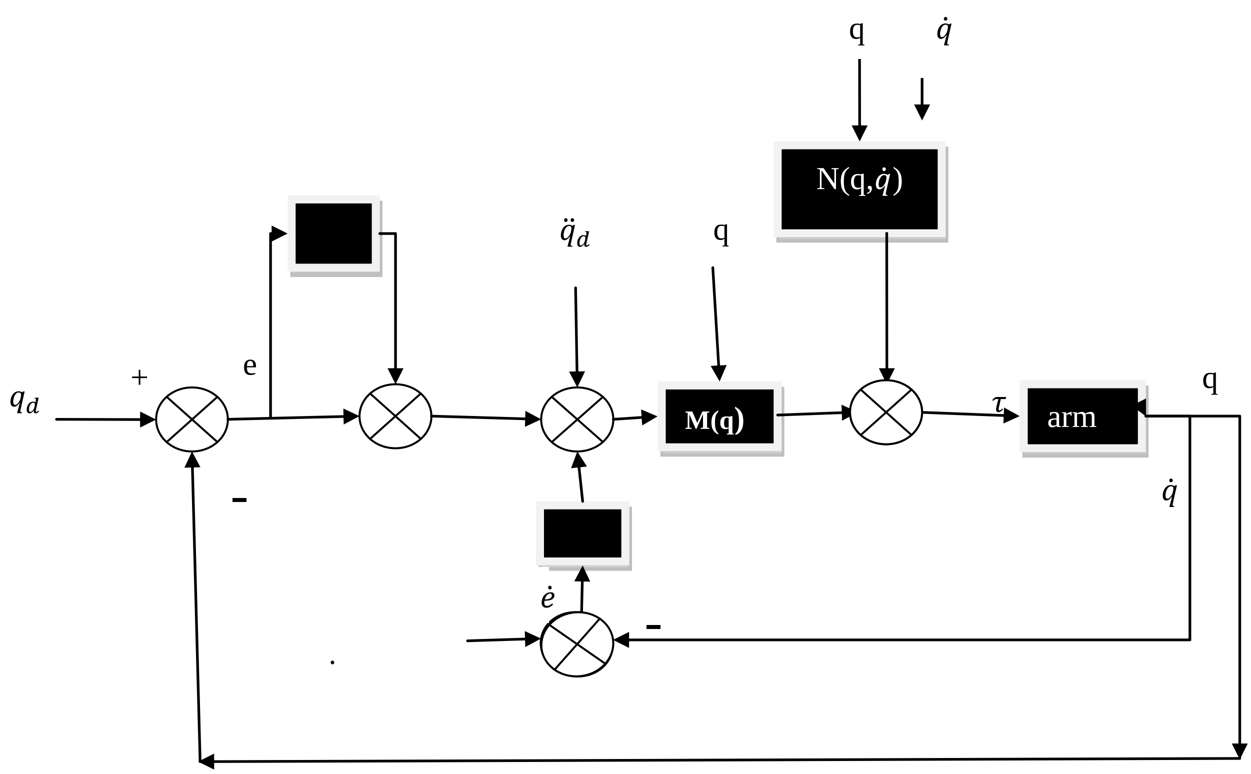


Figure 3.2: CTC with PD

Let θ_{1d} *and* θ_{2d} be the trajectories that we desire link 1 and link 2 to follow respectively

$$\theta_{1d} = \sin(t) \text{ and } \theta_{2d} = \cos(t) \qquad \text{(III.26)}$$

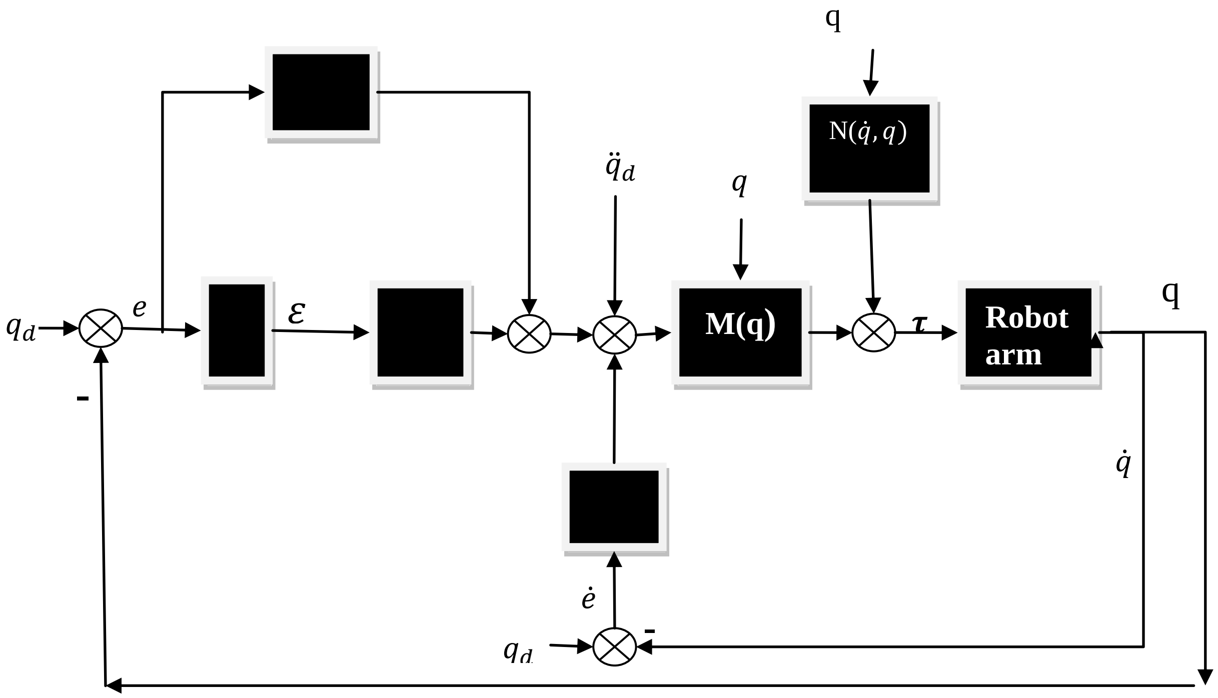
For the best tracking we should have the time constant of the closed-loop system as 0.1 s. For critical damping, this means that *Kv*=diag*{kv}*, K_p=diag*{kp}*, where

ω_n= 1/1.0=10 and

K_p = 2000, K_v = 20 (III.27)

Note : selecting of controller parameters(like PD gains) depends on the performance objectives which is the period of the desired trajectory in our case.

3.8.2 PID computed torque controller



To have PID computed torque control we should include an integrator in the feed-forward

Figure 3.3:PID CTC

$$e = \dot{\boldsymbol{\varepsilon}} \qquad \text{(III.28)}$$

u becomes;

$$u = -K_v\dot{e} - K_p e - K_i\varepsilon$$

(III.29)

implying the computed torque control law for the PID controller is:

$$\tau = M(q)(\ddot{q}_d + K_v\dot{e} + K_p e + K_i\varepsilon) + N(\dot{q},q) + \tau_d \qquad \text{(III.30)}$$

The state x becomes

$$x= [\varepsilon^T \; e^T \; \dot{e}^T]^T \qquad \text{(III.31)}$$

the error dynamics for the system becomes

$$\frac{d}{dt}\begin{bmatrix}\varepsilon\\ e\\ \dot{e}\end{bmatrix} = \begin{bmatrix}0 & I & 0\\ 0 & 0 & I\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\varepsilon\\ e\\ \dot{e}\end{bmatrix} + \begin{bmatrix}0\\ 0\\ I\end{bmatrix}u + \begin{bmatrix}0\\ 0\\ I\end{bmatrix}\omega \qquad \text{(III.32)}$$

Fig3.3 shows the PID Computed torque controller.
The error dynamics for the closed loop system is:

$$\frac{d}{dt}\begin{bmatrix}\varepsilon\\ e\\ \dot{e}\end{bmatrix} = \begin{bmatrix}0 & I & 0\\ 0 & 0 & I\\ -K_i & -K_p & K_v\end{bmatrix}\begin{bmatrix}\varepsilon\\ e\\ \dot{e}\end{bmatrix} + \begin{bmatrix}0\\ 0\\ I\end{bmatrix}\omega \qquad \text{(III.33)}$$

Diagonal selection of gains **[3]**
The characteristic polynomial for the closed loop becomes

$$\text{Cp(s)} = \text{I}s^3 I + K_v s^2 + K_p s + K_i \text{ I} \qquad \text{(III.34)}$$

The control gains;

K_v = diag$\{K_{v_i}\}$, $K_p = diag\{K_{p_i}\}$, $K_i = \{K_{i_i}\}$

The Routh law of stability requires that K_{i_i} be small, ie smaller than the product of K_{p_i} and K_{v_i}. **[3]**

τ_d (a disturbance to the arm dynamics) wass given a value of 1Nm by so doing we will be modelling it like friction. We can make a comparison between PD and PID computed torque control.

3.8.3 Adaptive control by computed torque approach

Adaptive Computed-Torque Controller

Due to many problems associated with model formulation, we **never** get to have the exact robot model in real life. Two common uncertainties are unknown link masses and unknown friction coefficients**[3].** We have discussed one way of dealing with these types of parametric uncertainties early in this chapter where computed-torque controllers were being used. These controllers were using fixed estimates of the unknown parameters in place of the actual parameters.

The adaptive control strategy can, heuristically, be motivated by reasoning that one could expect better tracking performance if the parameter estimate was adjusted as the robot manipulator moves instead of always being a fixed quantity. That is, it seems reasonable to attempt to change our parameter estimates based on an adaptive update rule that would be a function of the robot configuration and the tracking error.[F. Lewis, 2004]

The adaptive update rule is formulated from the stability analysis of the tracking error system. That is, we ensure stability of the tracking error system by formulating the adaptive update rule and by analyzing the stability of the tracking error system at the same time. The first adaptive control strategy that we will examine is the method outlined in [F. Lewis]. This adaptive controller is based on the fact that the parameters appear linearly in the robot model. That is, the robot dynamics (1) can be written in the form

$$\text{M(q)}\ddot{q} + \text{C(q,}\dot{q}\text{)}\,\dot{q} + \text{F(}\dot{q}\text{)} + \text{G(q)} = \text{Y(q,}\dot{q}, \ddot{q}\text{)}\varphi \qquad \text{(III.35)}$$

where Y(q,$\dot{q}$, $\ddot{q}$) is an $n\times r$ matrix of known time functions and φ is an $r\times 1$ vector of unknown constant parameters.

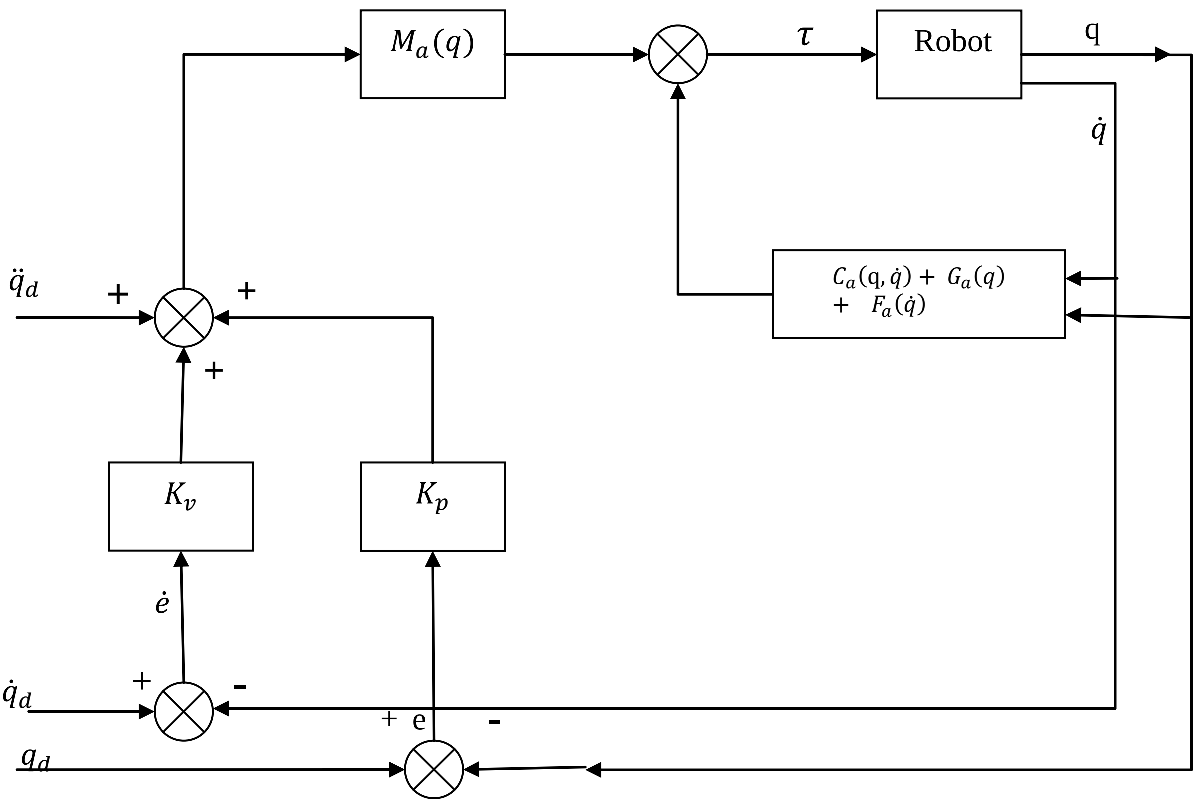


Figure 3.3: adaptive computed torque controller

The fact that the parameters appear linear in the robot model is crucial for the type of adaptive control that was formulated by Craig because it illustrates the separation of unknown parameters and the known time functions. The reason why the robot dynamics can be separated in this form is that the robot dynamics are linear in the parameters expressed in the vector form . This separation of unknown parameters and known time functions will be used in the formulation of the adaptive update rule and also in the stability analysis of the tracking error system.

Forming the tracking error system is the first step in the study of the adaptive computed torque controller. If we use (III.35) we may write the robot dynamics as follows;

$$\tau = \mathrm{Y}(\mathrm{q},\dot{q},\ddot{q}\ )\varphi \qquad \text{(III.36)}$$

From [F. Lewis] the adaptive computed-torque control law is given by

$$\tau = M_a(\mathrm{q})\big(\ddot{q}_d + K_v\dot{e} + K_p e\big) + C_a(\mathrm{q},\dot{q})\dot{q} + G_a(q) + F_a(\dot{q}) \qquad \text{(III.37)}$$

From the definition of the tracking error, (III.37) can be written as

$$\tau = M_a(\mathrm{q})\big(\ddot{e} + K_v\dot{e} + K_p e\big) + M_a(\mathrm{q})\ddot{q} + C_a(\mathrm{q},\dot{q})\dot{q} + G_a(q) + F_a(\dot{q}) \qquad \text{(III.38)}$$

If we use (III.35), (III.38), can be

$$\tau = M_a(\mathrm{q})\big(\ddot{e} + K_v\dot{e} + K_p e\big) + \mathrm{Y}(\mathrm{q},\dot{q},\ddot{q}\ )\varphi_a \qquad \text{(III.39)}$$

Of which φ_a is a vector of an n×1 dimensions which represents a time-varying estimate of the unknown constant parameters. Substitution of (III.39) into (III.36), gives us the tracking error system in (III.35):

$$\ddot{e} + K_v\dot{e} + K_p e = {M_a}^{-1}(q)\mathrm{Y}(q,\dot{q},\ddot{q}).\varphi_a \qquad \text{(III.40)}$$

Where the parameter error is

$$p_e = \varphi - \varphi_a \qquad \text{(III.41)}$$

(III.40) in state-space form gives us ;

$$\dot{e} = Ae + BM^{-1}(\mathrm{q})\, \mathrm{Y}(q, \dot{q}, \ddot{q}).\varphi_a \qquad \text{(III.42)}$$

Where the tracking error, e is given by:

$$\mathrm{e} = \begin{bmatrix} e \\ \dot{e} \end{bmatrix}$$

and

$$\begin{cases} \mathrm{B} = \begin{bmatrix} O_n \\ I_n \end{bmatrix}, \\ \mathrm{A} = \begin{bmatrix} O_n & I_n \\ -K_p & -K_p \end{bmatrix} \end{cases}$$

Where I_n is an n×n identity matrix and O_n an n×n zero matrix.

Lyapunov stability analysis is used to show that the tracking error , **e** is asymptotically stable if we choose the right choice of adaptive update law. We first select the positive-definite Lyapunov-like function

$$\mathrm{V} = e^T Pe + \varphi^{-T}\Gamma^{-1}\varphi_a \qquad \text{(III.43)}$$

where P is a $2n \times 2n$ positive-definite, constant, symmetric matrix, and Γ is a diagonal, positive-definite $r \times r$ matrix. Which means Γ can be written as

$$\Gamma = diag(\gamma_1, \gamma_2, \ \ldots, \ \gamma_n),$$

where the γis are positive scalar constants.

Differentiation of (III.43) with respect to time yields

$$\dot{V} = e^T P\dot{e} + \dot{e}^T \mathrm{Pe} + 2\, {\varphi_a}^T \Gamma^{-1}\dot{\varphi}_a \qquad \text{(III.44)}$$

To obtain (III.44) we use the fact that

$$[{\varphi_a}^T \Gamma^{-1}\varphi_a]^T = {\varphi_a}^T\, \Gamma^{-1}\, \dot{\varphi}_a \qquad \text{(III.45)}$$

since $\Gamma = \Gamma^T$(transposing a scalar does not change anything) Now substituting for e from (III.42) into (III.44) yields

$$\begin{aligned} \dot{V} = \ & e^T P(Ae + BM^{-1}(\mathrm{q})\, \mathrm{Y}(.)\varphi_a) \\ & + \left(Ae + B{M_a}^{-1}(\mathrm{q})\, \mathrm{Y}(.)\varphi_a \right)^T \mathrm{Pe} \\ & +2{\varphi_a}^T\, \Gamma^{-1}\, \dot{\varphi}_a \end{aligned} \qquad \text{(III.46)}$$

Combining terms in (III.46) and using the scalar transportation property gives

$$\dot{V} = -e^T \mathrm{Qe} + 2{\varphi_a}^T(\Gamma^{-1}\, \dot{\varphi}_a \ + \ Y^T\, (.)\, {M_a}^{-1}(\mathrm{q})\, B^T \mathrm{Pe}) \qquad \text{(III.47)}$$

where Q is a positive-definite, symmetric matrix which satisfies the Lyapunov equation

$$A^T \mathrm{P} + \mathrm{PA} = \text{-Q} \qquad \text{(III.48)}$$

To attain stability, it is always desirable to have $\dot{V}$ at least negative semi definite; thus, the choice of adaptation update rule becomes obvious. Specifically, by substitution

$$\dot{\varphi}_a = -\,\Gamma^{-1} Y^T(.){M_a}^{-1}(\mathrm{q})\, B^T \mathrm{Pe} \qquad \text{(III.49)}$$

Also, (III.47) becomes

$$\dot{V} = -e^T \mathrm{Qe} \qquad \text{(III.50)}$$

To determine explicitly the type of stability, we must do further analysis; however, note first that (III.49) gives the adaptive update rule for the parameter estimate vector φ_a since $\dot{\varphi}_a$is equal to zero. That is, by recalling that the actual unknown parameters are *constant,* we can substitute (III.41) into (III.49) to obtain the *adaptive update rule* ***[3]***

$$\dot{\varphi}_a = \Gamma Y^T(.){M_a}^{-1}(\mathrm{q})\, B^T \mathrm{Pe} \qquad \text{(III.51)}$$

Now, from the robot Equation (1), we can deduce that

$$\ddot{q} = M^{-1}(q)(\tau - C(q,\dot{q})\dot{q} - G(q) - F(\dot{q})) \qquad \text{(III.52)}$$

therefore, $\ddot{q}$ is bounded since $\ddot{q}$and τ depend only on the bounded quantities $q, \dot{q}$ and $\boldsymbol{\varphi_a}$. If $\ddot{q}$ is bounded, (10) shows that $\dot{e}$ is bounded. Since is bounded, we can state from (18) that$\dot{V}$ is bounded. Therefore, since V is lower bounded by zero, is negative semi definite, and is bounded, then by Barbalat's lemma ,[3]

$$\lim_{t\to\infty} V = 0$$

In summary;

i torque controller

$\tau = M_a(q)(\ddot{q}_d + K_v\dot{e} + K_p e) + C_a(q,\dot{q})\,\dot{q} + G_a(q) + F_a(\dot{q})$

Update rule

$\dot{\varphi}_a = \Gamma Y^T(q,\dot{q},\ddot{q})\, {M_a}^{-1}(q)\, B^T P e$

Where $e=\begin{bmatrix} e \\ \dot{e} \end{bmatrix}$, $B = \begin{bmatrix} I_n \\ O_n \end{bmatrix}$, $A= \begin{bmatrix} O_n & I_n \\ -K_p & -K_v \end{bmatrix}$

$Y(q,\dot{q},\ddot{q})\varphi_a = M_a(q)\ddot{q} + C_a(q,\dot{q})\,\dot{q} + G_a(q) + F_a(\dot{q})$

$A^T P + PA = -Q$

For some positive definite, symmetric matrices P and Q

Stability

Tracking error vector e is asymptotically stable.

Restrictions

<u>Parameter resetting method is required , measurement of $\ddot{q}$ is required</u>

We shall not go on to show it as proof but it should be known that the tracking error vector e is asymptotically stable. And the parameter error remains bounded if ${M_a}^{-1}(q)$ exists. This will definitely place a restriction on the parameter update law given in (III.51). That means it is a must to use the parameter resetting method discussed earlier to ensure that poor parameter estimates do not cause the inverse of (q) to explode.

The adaptive computed-torque controller is summarized in Table (1) above and depicted in Figure (3.4).

3.9 Conclusion

In this chapter we have described and explained the objective of control in systems engineering. We used that knowledge of control to describe the control of a 2DOF robotic system using proportional plus derivative, proportional-derivative-integral and adaptive computed torque control.

Chapter 4 : Simulation results

4.1 Introduction

In this chapter we shall discuss the details of the simulation of PD , PID computed torque control. We shall make a comparison between these two types of control and draw conclusions from observation made during simulation. The results of simulation shall also be shown, explained and compared to the results of related work. We shall start with looking at simulation and results for the PD control.

4.2 PD control

During PD CTC , we used the control law:

$$\tau = Mq(\ddot{q}_d + K_v\dot{e} + K_p\mathrm{e}) + \mathrm{N}(\mathrm{q},\dot{q})$$

and signal references ;

θ_1 *and* θ_2 as sin(t) and cos(t) for link 1 and link 2 respectively

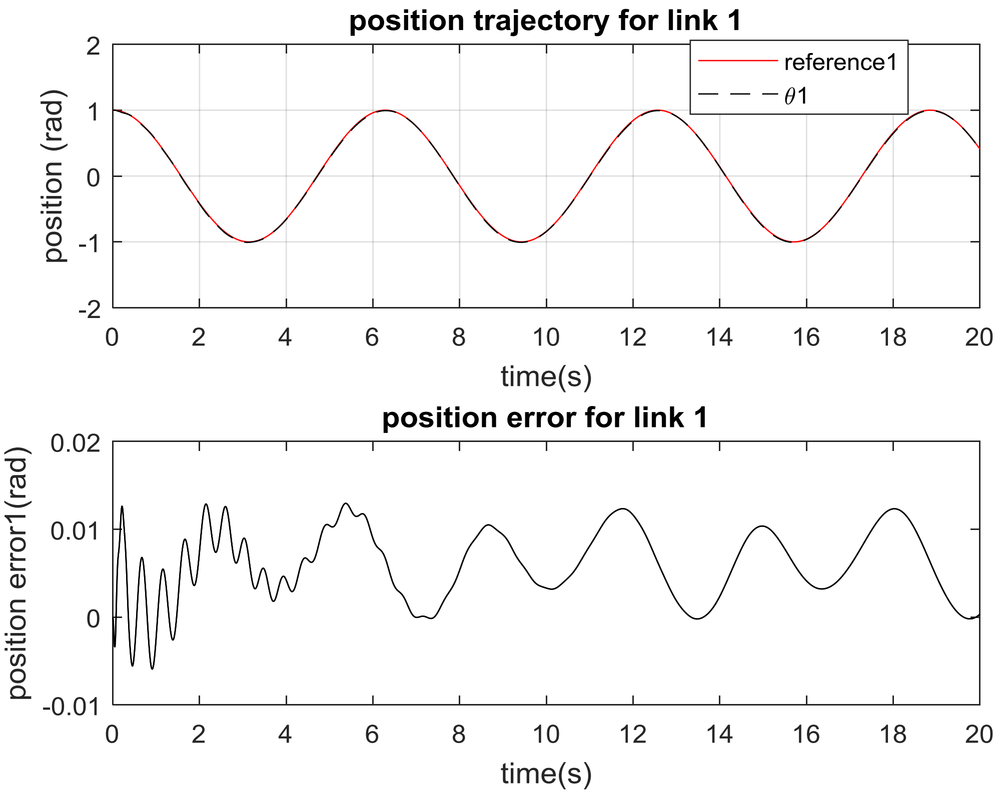


Figure 4.1: position trajectory and position error for link 1

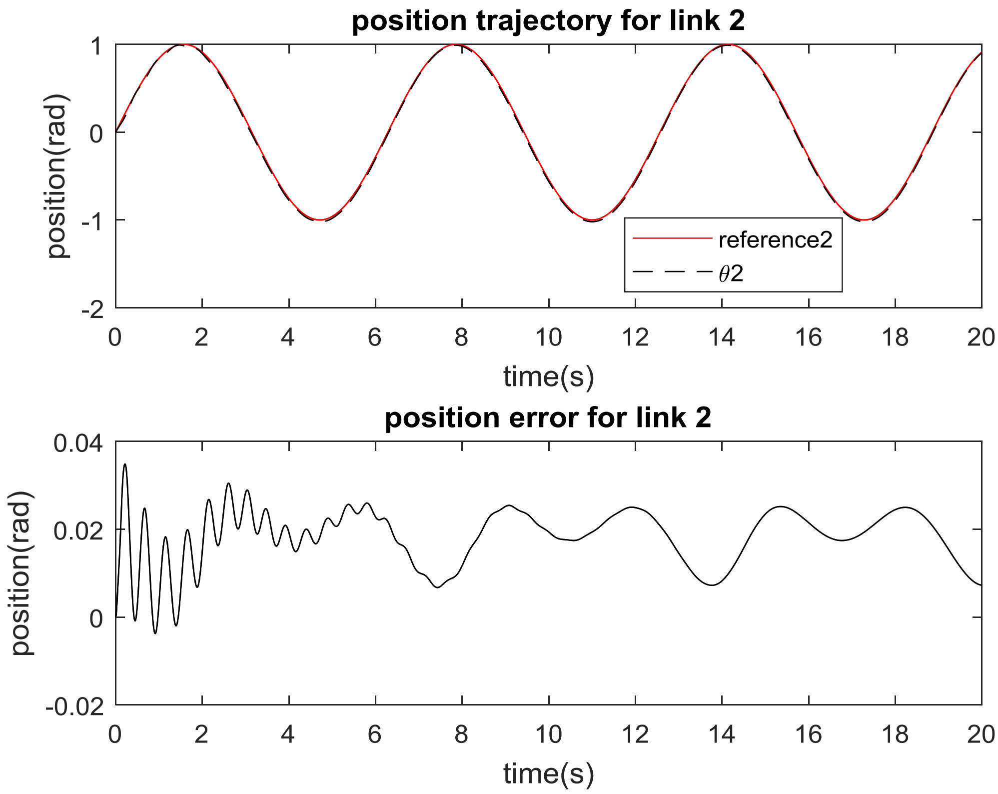


Figure4.2: position trajectory and position error for link 2

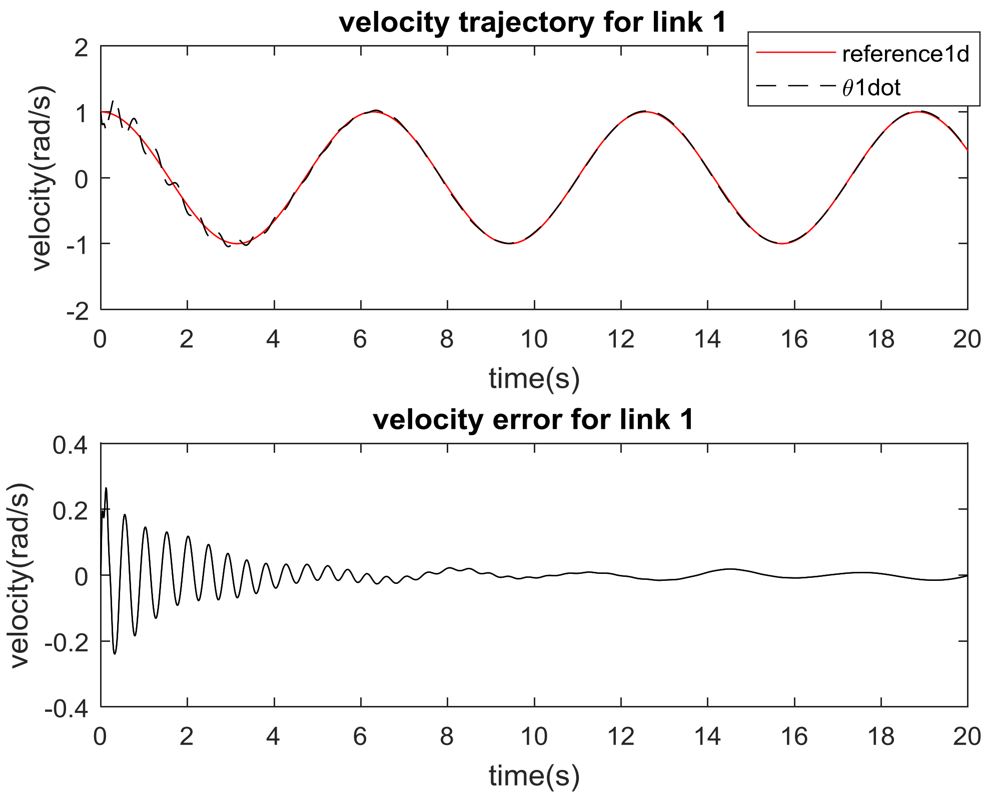


Figure 4.3; velocity trajectory and velocity error for link 1

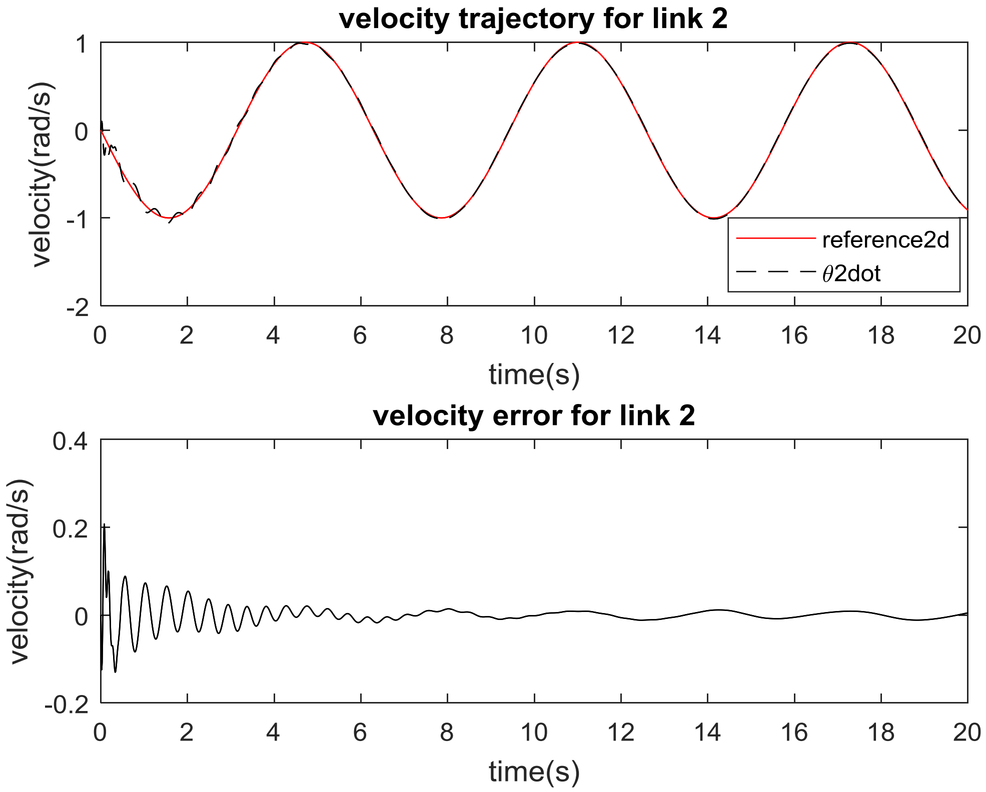


Figure 5.4: velocity trajectory and velocity error for link 2

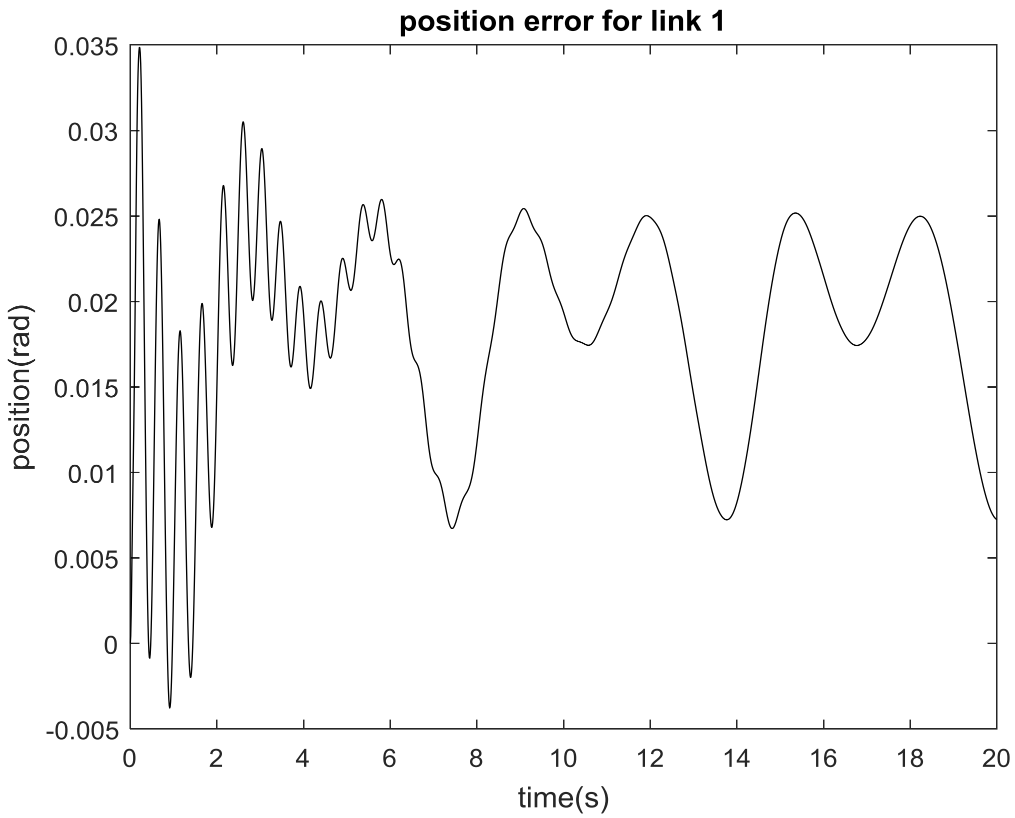


Figure 4.3; position error for link 1

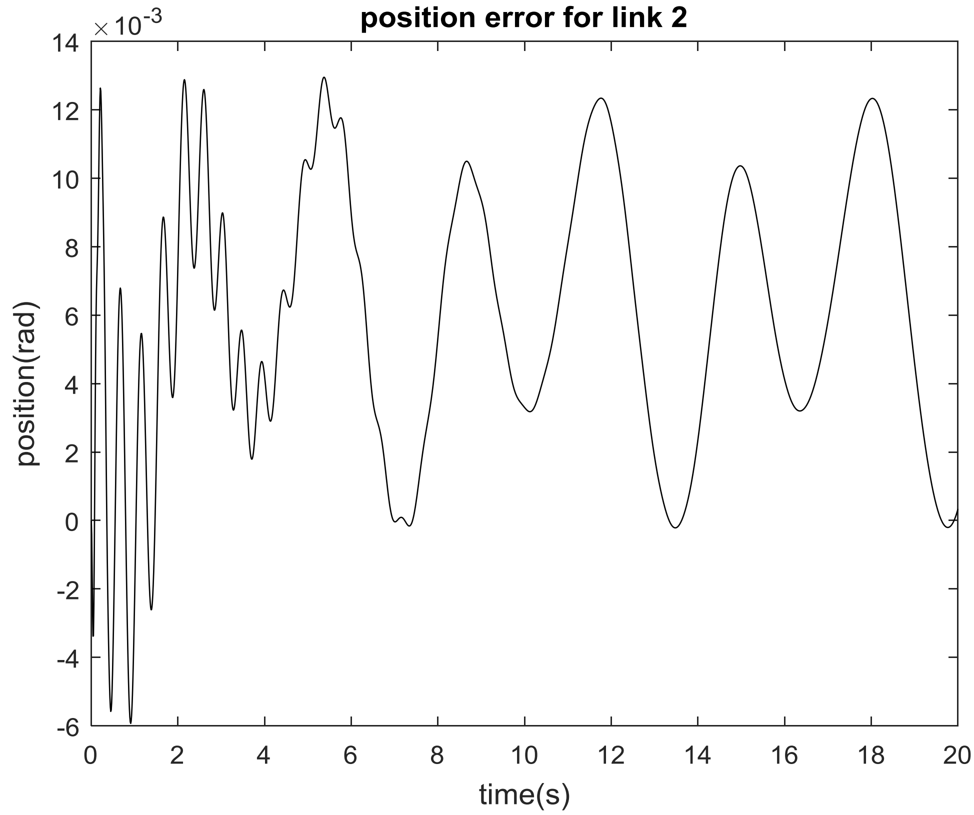


Figure 4.6: Position error for link 2

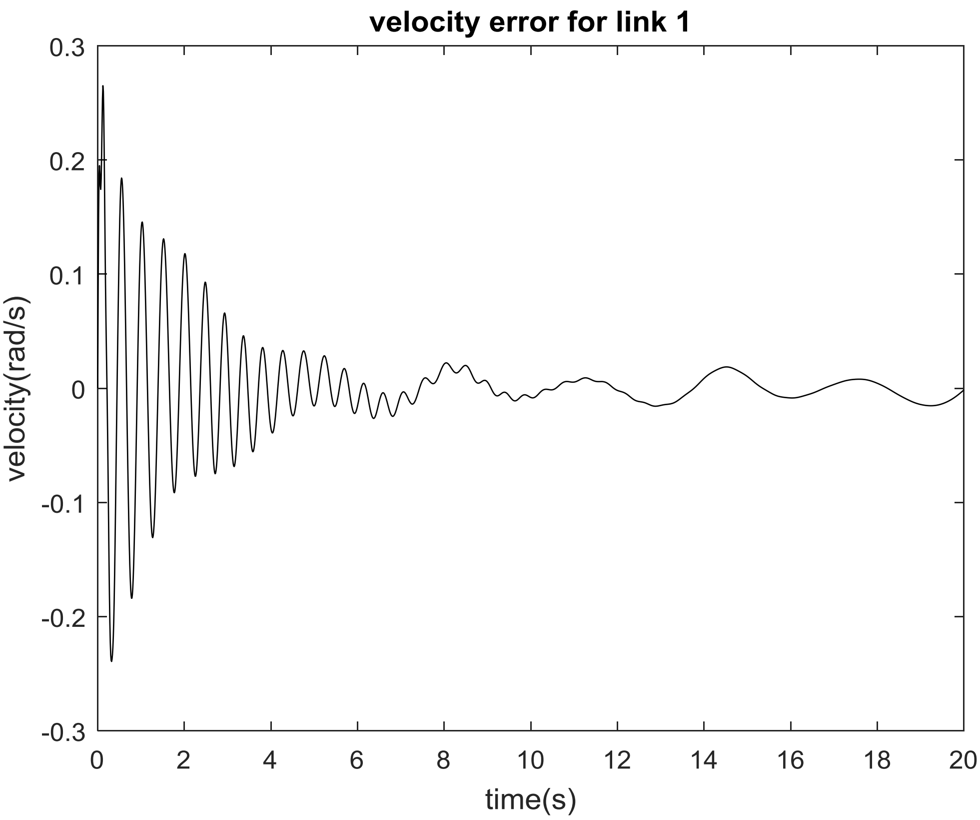


Figure 4.7: velocity error for link 1

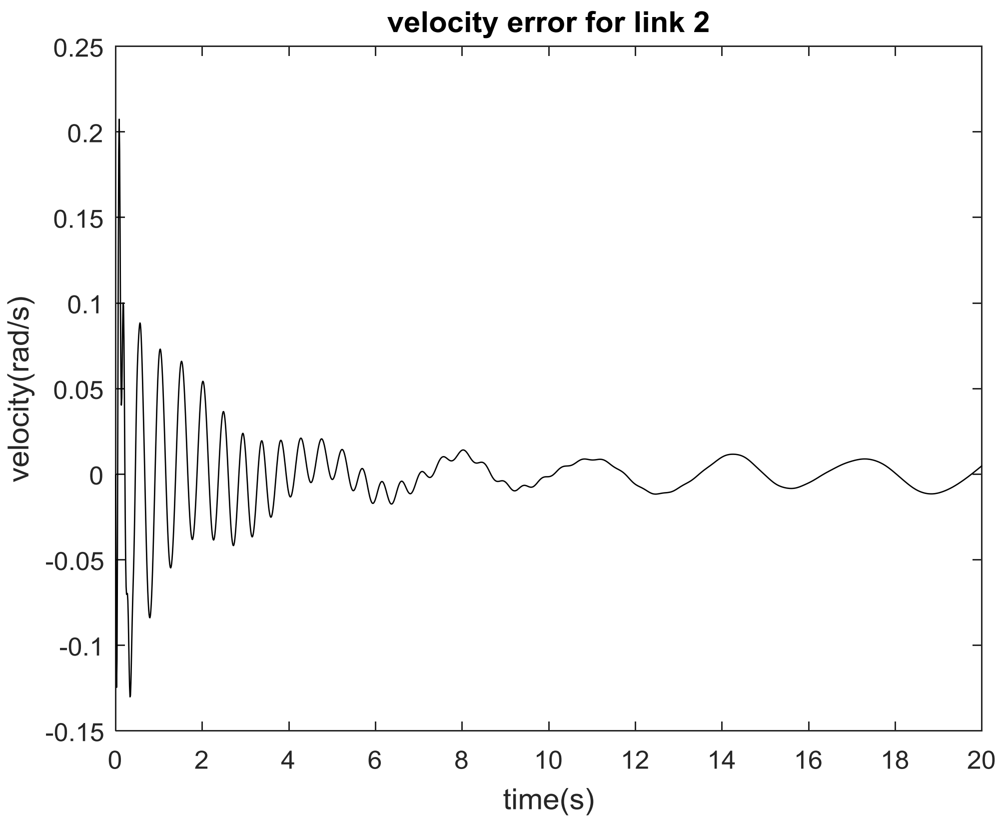


Figure 4.8: velocity error for link 2

4.3 Adaptive Control

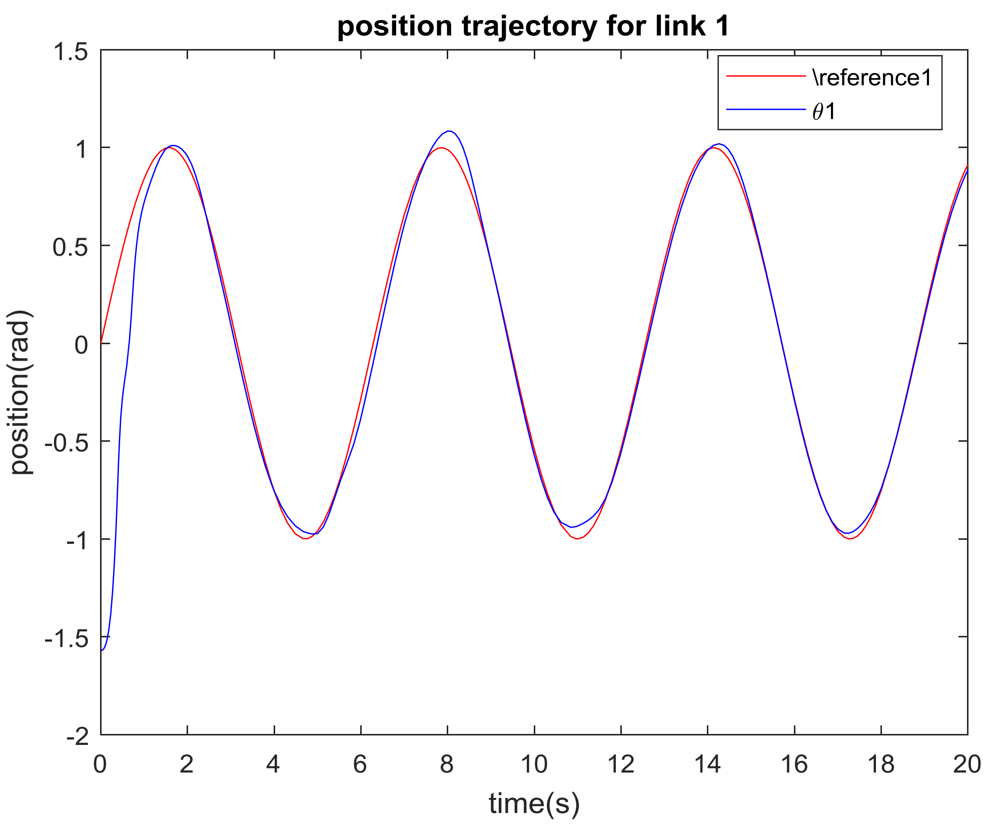


Figure 4.6: adaptive trajectory for link 1

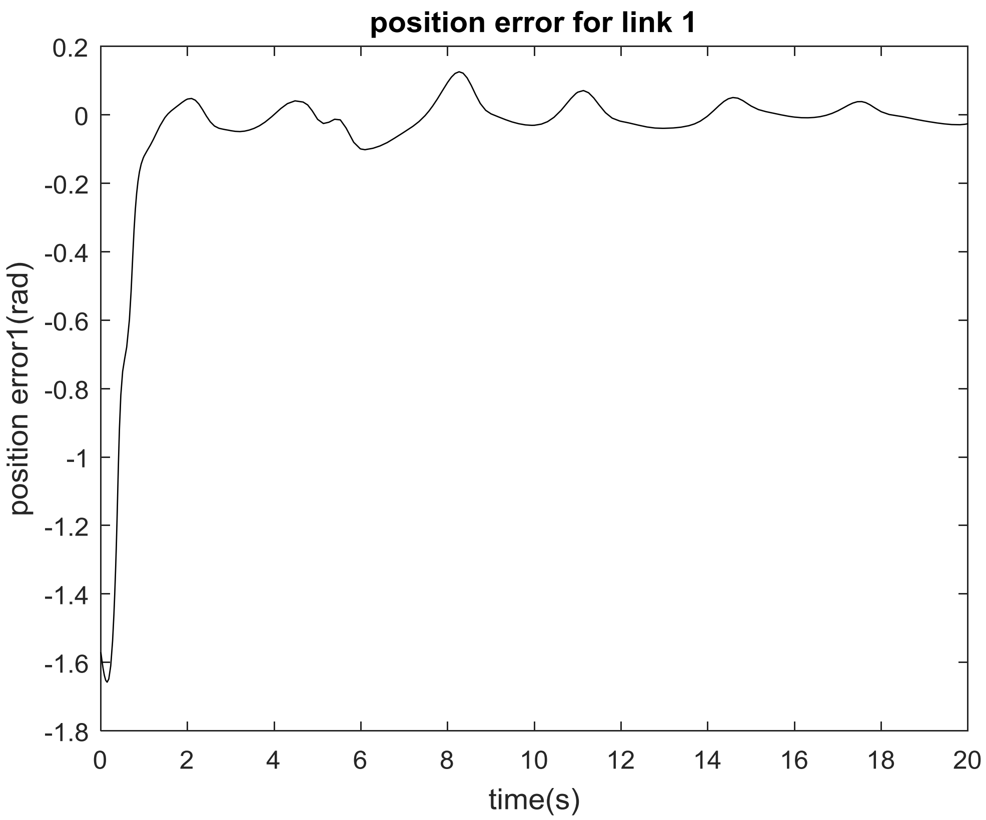


Figure 4. 7: Adaptive position error for link 1

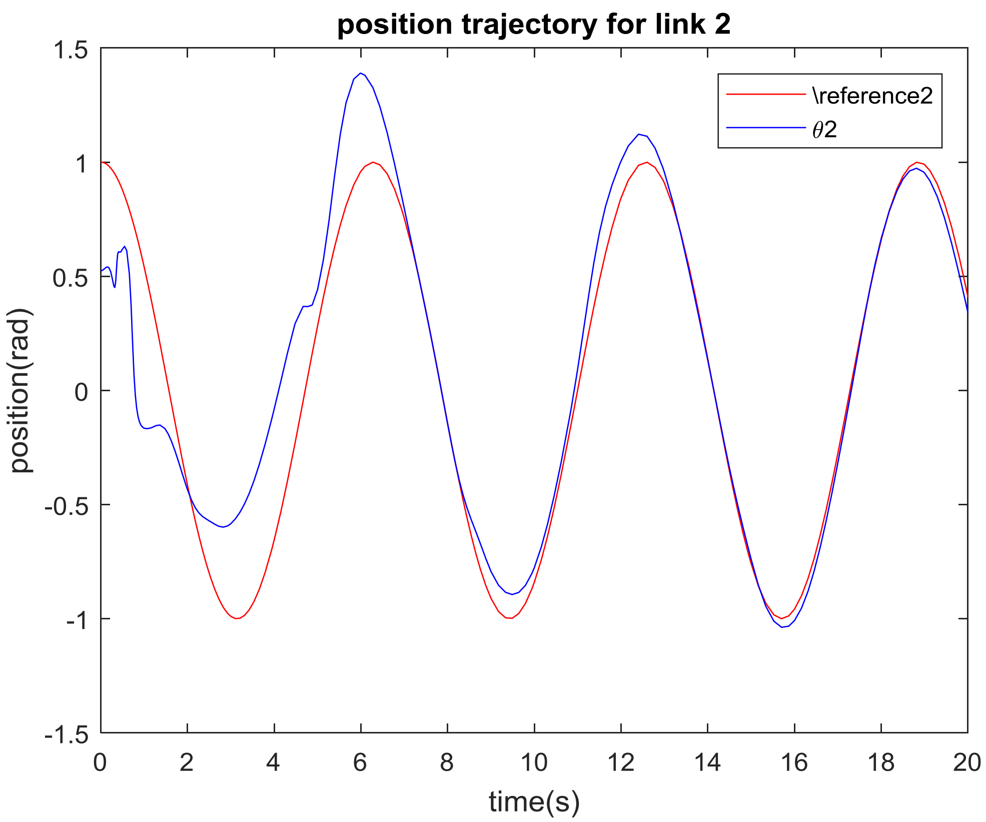


Figure 4.8: Adaptive position trajectory for link2

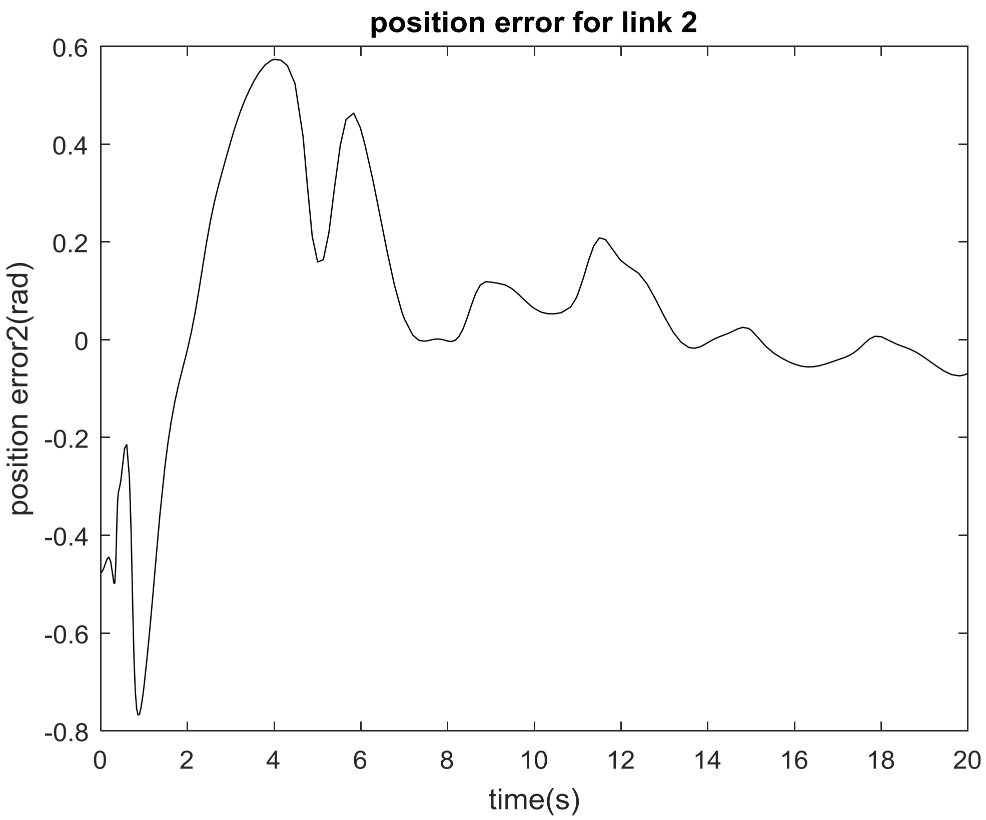


Figure 4. 9: Adaptive position error for link 2

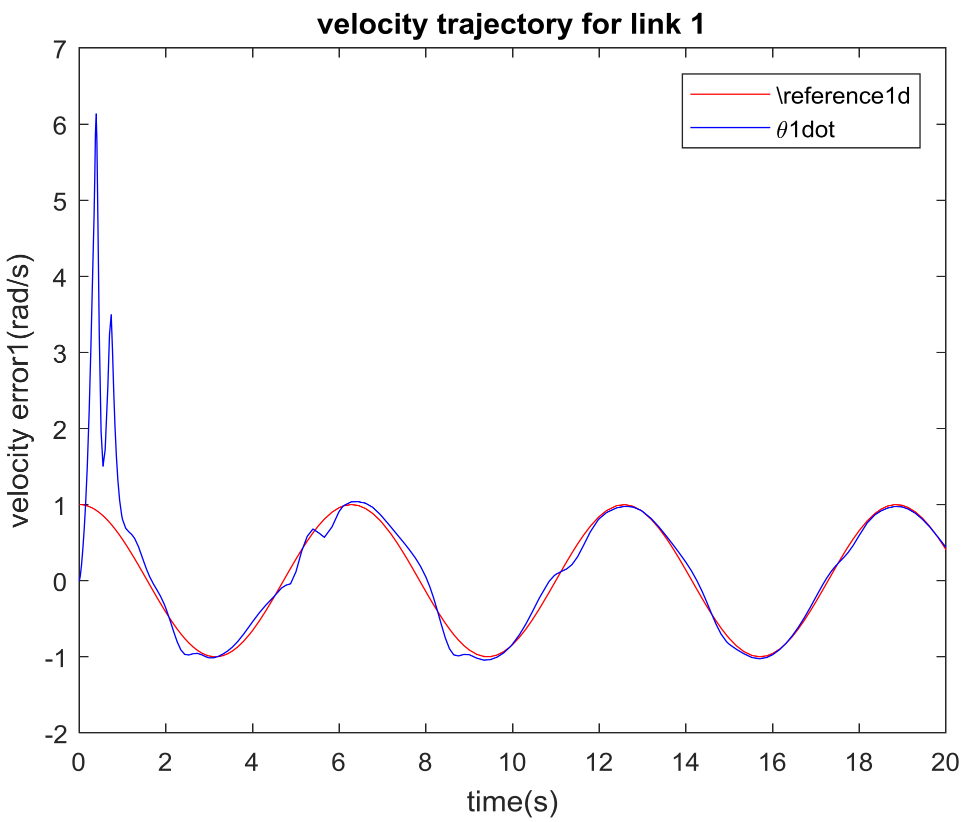


Figure 4.10: Adaptive velocity trajectory for link 1

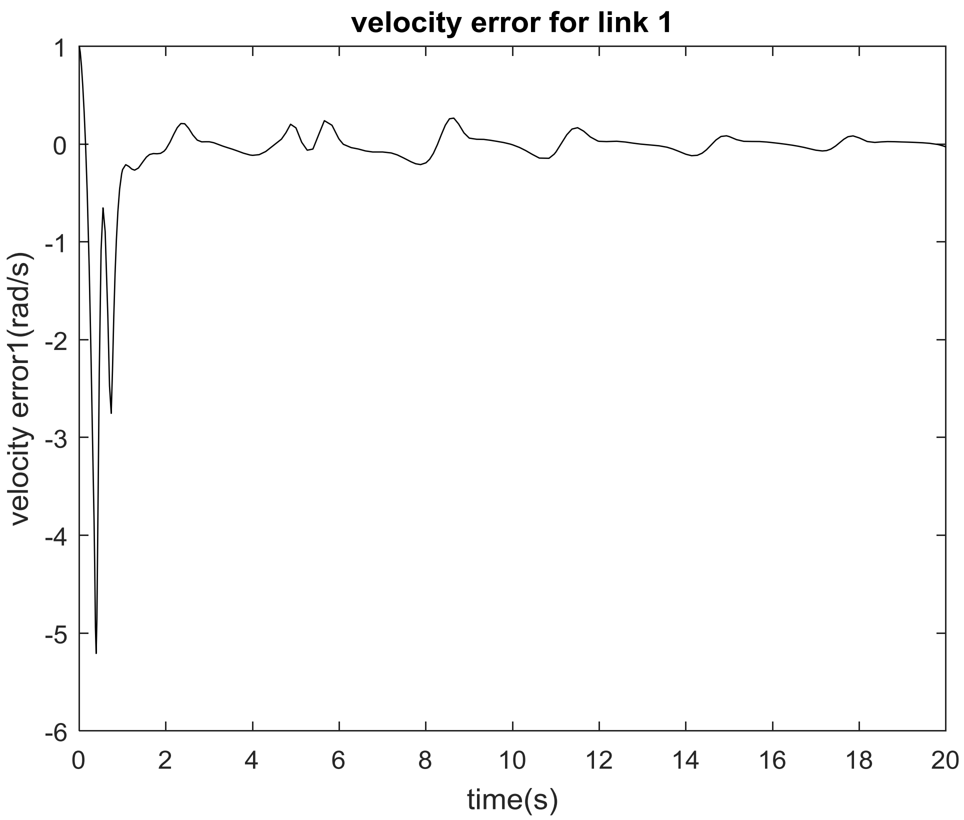


Figure4. 11: Adaptive velocity error for link 1

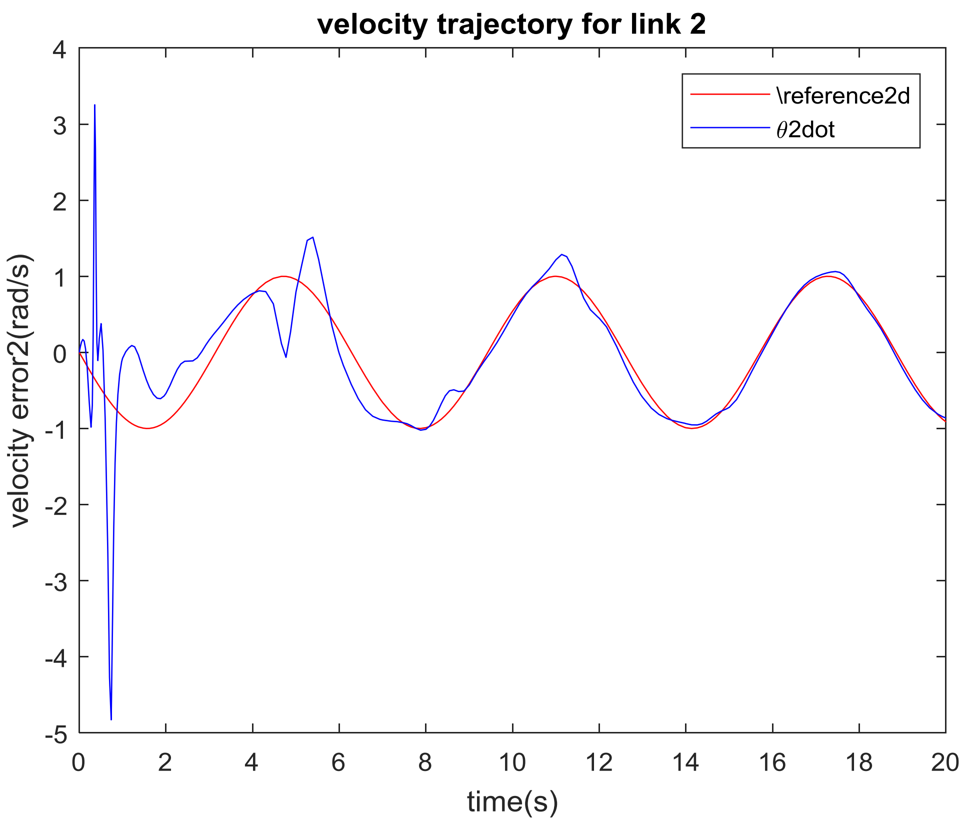


Figure 4.12: Adaptive velocity trajectory for link 2

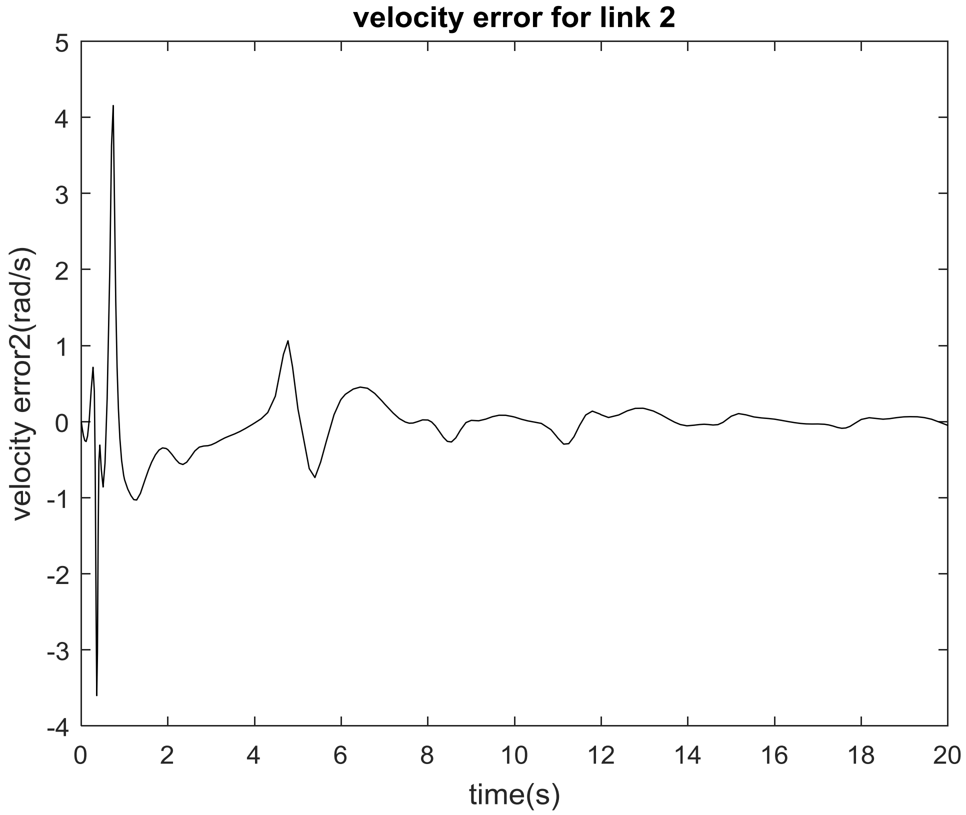


Figure 4. 13: Adaptive velocity error for link 2

4.4 PID Adaptive

For PID CTC we use the control law;

$$\tau = M(q)(\ddot{q}_d + K_v\dot{e} + K_p e + K_i\varepsilon) + N(\dot{q},q) + \tau_d$$

With selection of gains

$$K_v = \mathrm{diag}\{K_{v_i}\}, \; K_p = diag\{K_{p_i}\}, \quad K_i = \{K_{i_i}\}$$

$$K_{i_i} < K_{p_i} * K_{v_i}.$$

We obtain the following results

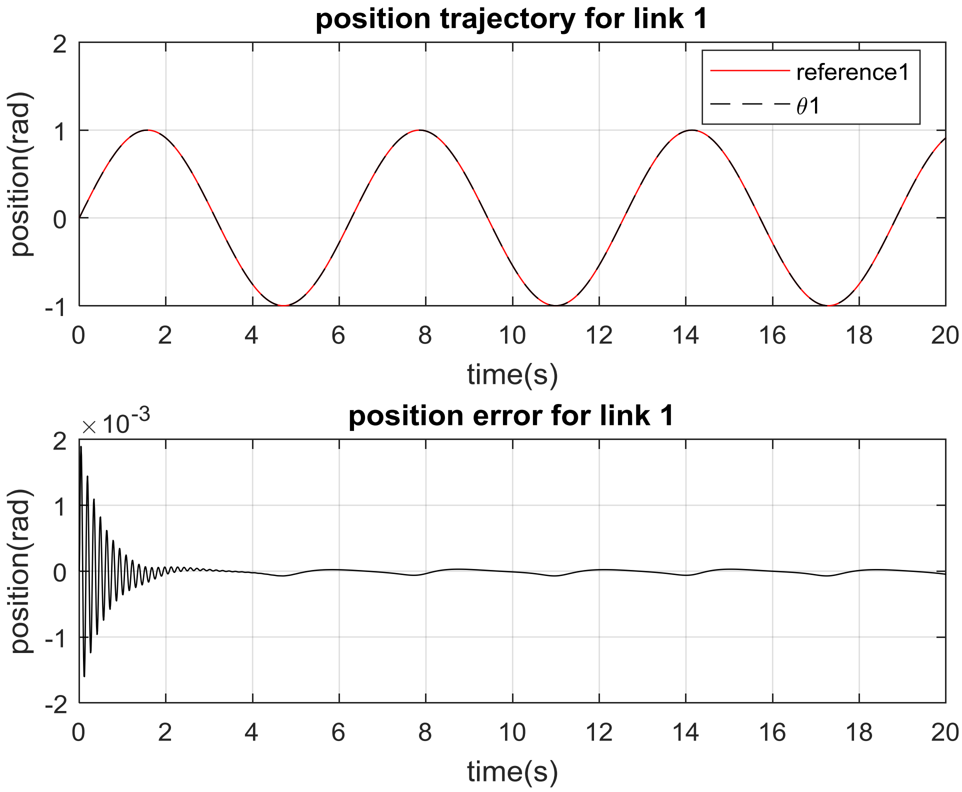


Figure 4. 14: control PID, position trajectory and position error for link 1

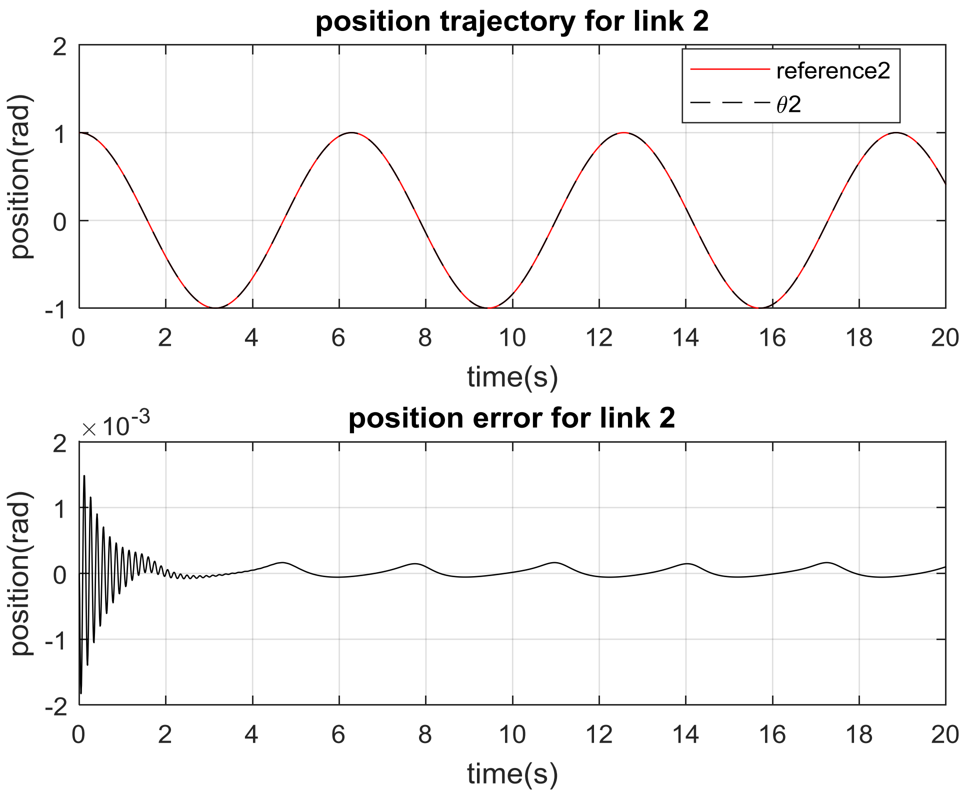


Figure 4.15: PID position trajectory and position error for link 2

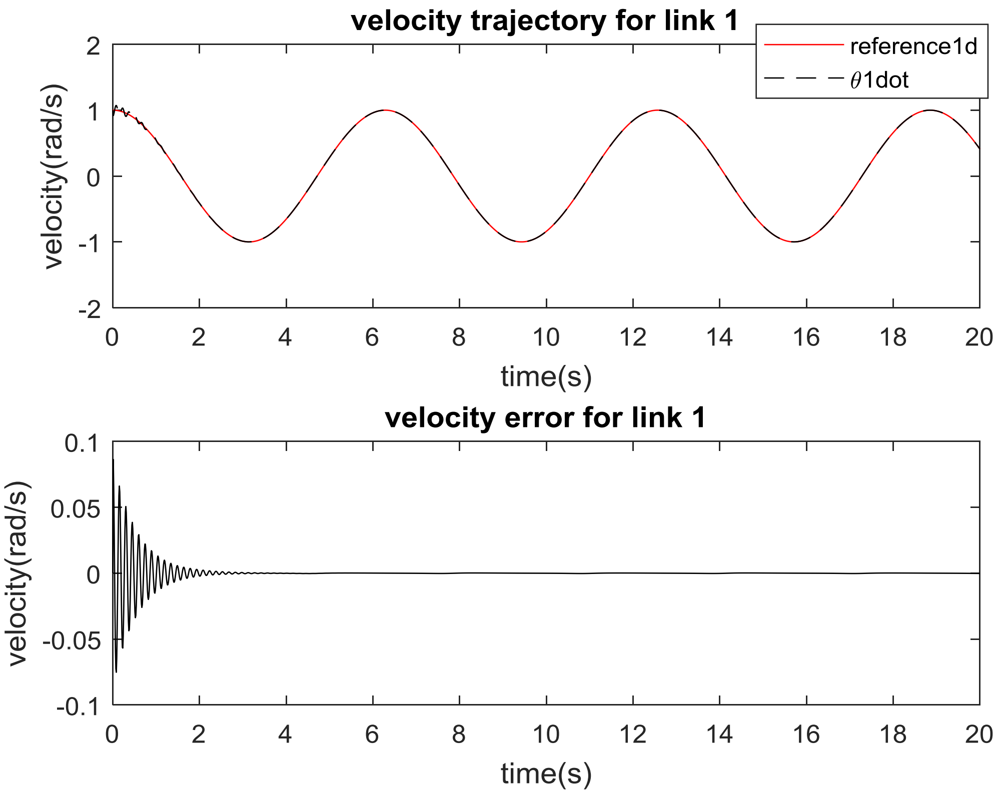


Figure 4. 16: PID velocity trajectory and velocity error for link 1

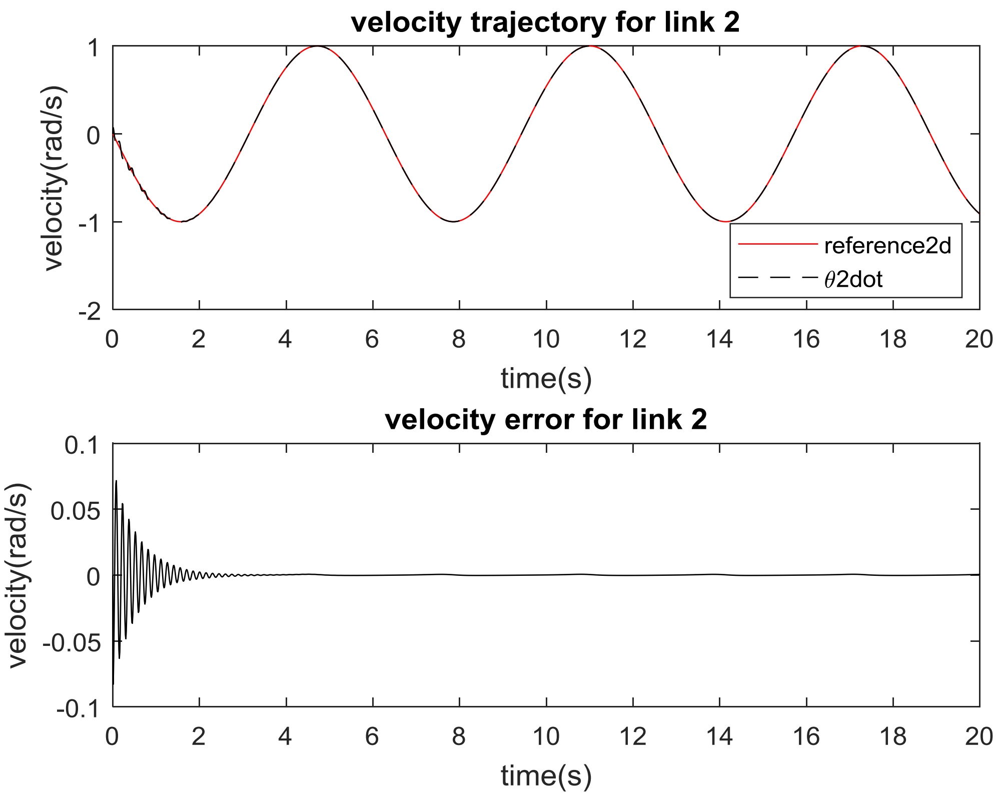


Figure 4. 17: PID velocity trajectory and velocity arror at link 2

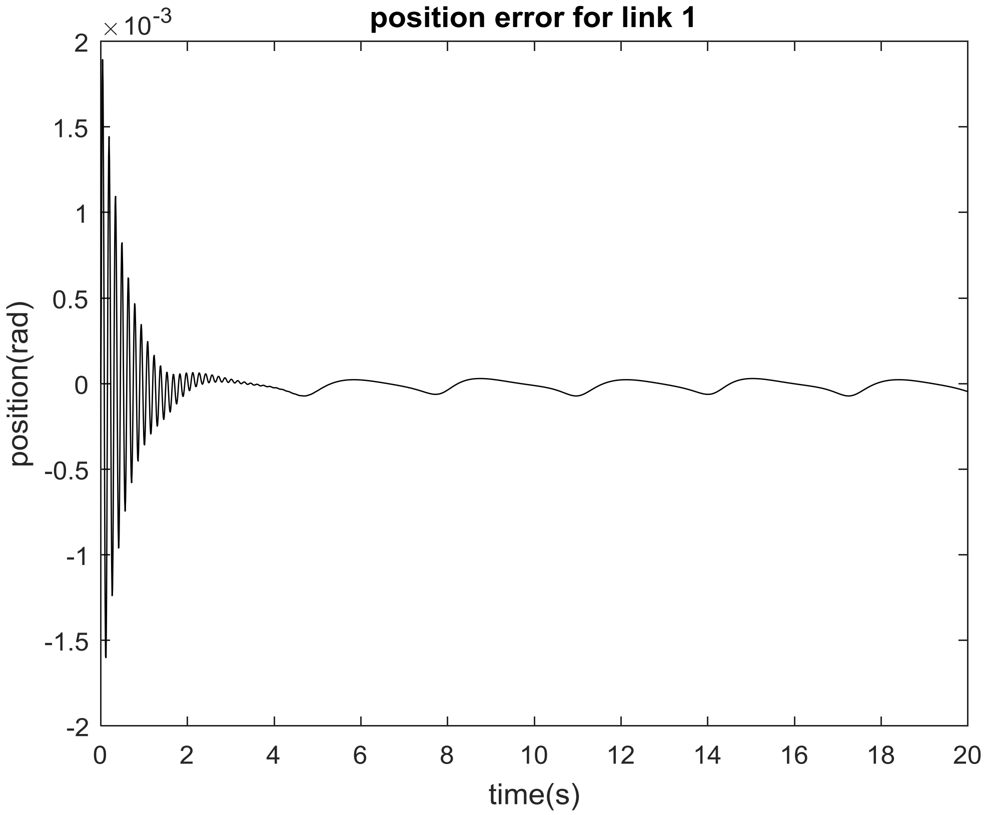


Figure 4.18: PID position error for link 1

In the first 1s, there is an error which starts from 0 to 0.001 to -0.001 rad . this error gradually decreases and later stabilises to oscillate to an error of 0.00001 rad. However at t= 0.45s there is a drop in error, (error=-1 rad) which later stabilises again to 0.00001rad

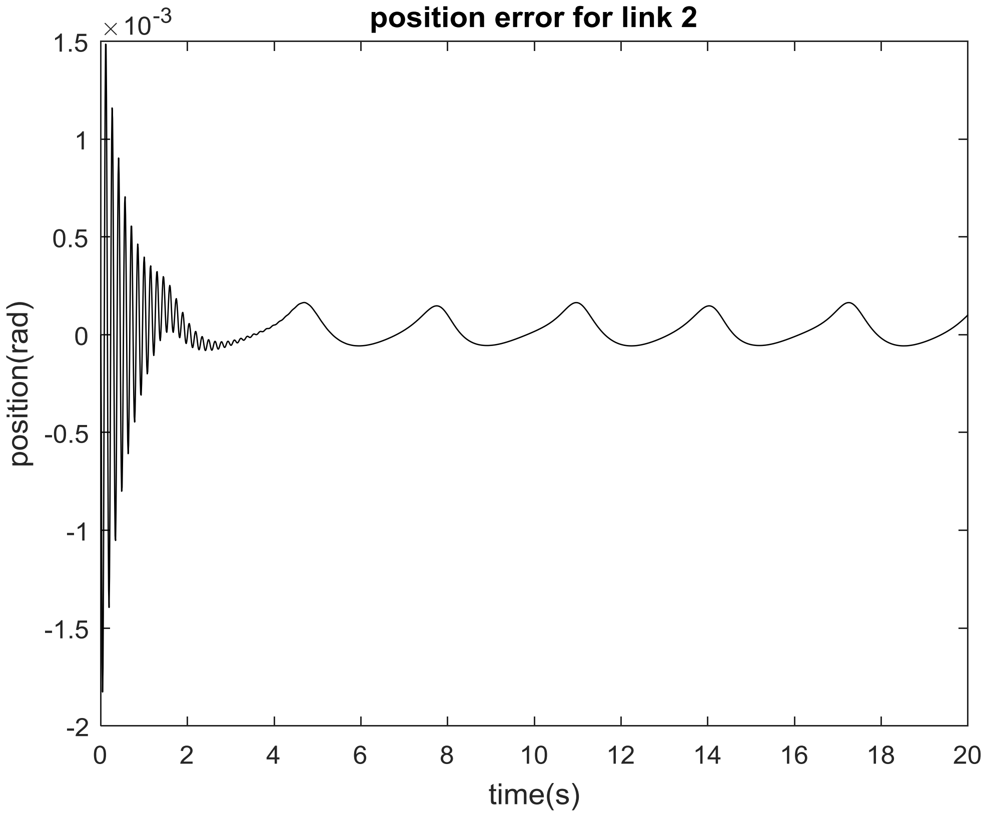


Figure 4.19: PID position error for link 2

In the first 1 second, error oscillates from -10^{-3} to 10^{-3} rad then stabilises to an error that oscillates around 5. 10^{-5} from t=4s to t=20s.

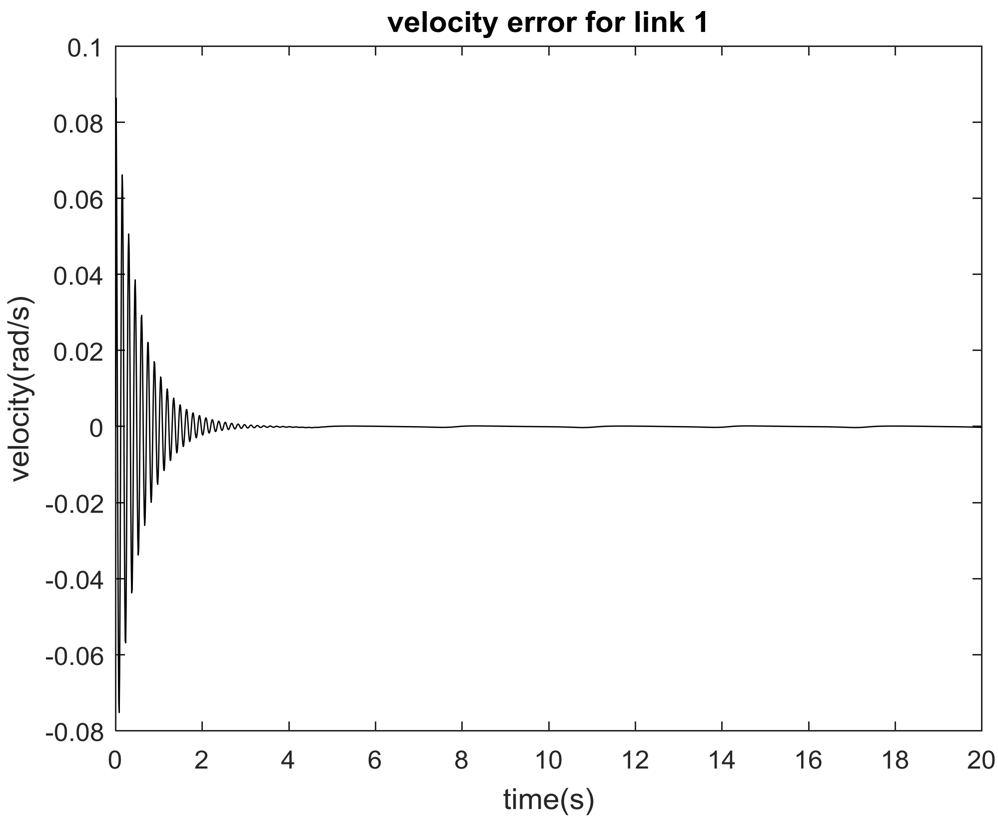


Figure 4.20: PID velocity error for link 1

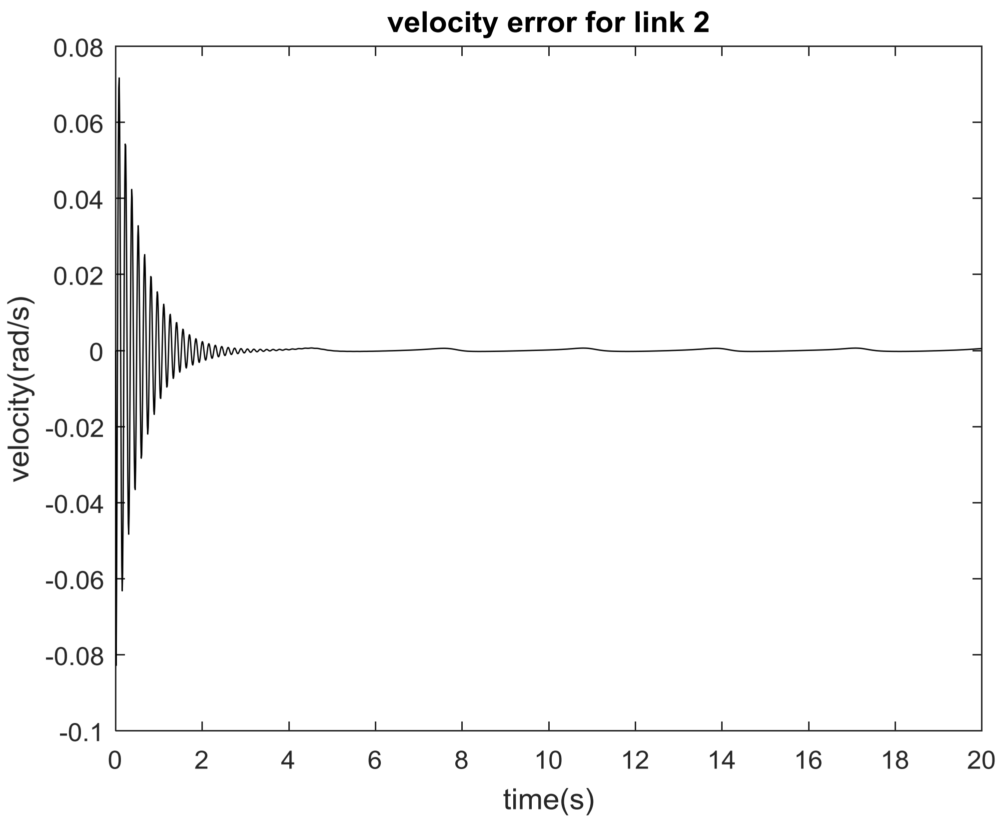


Figure 4.21:PID velocity error for link 2

4.5 Comparison of error

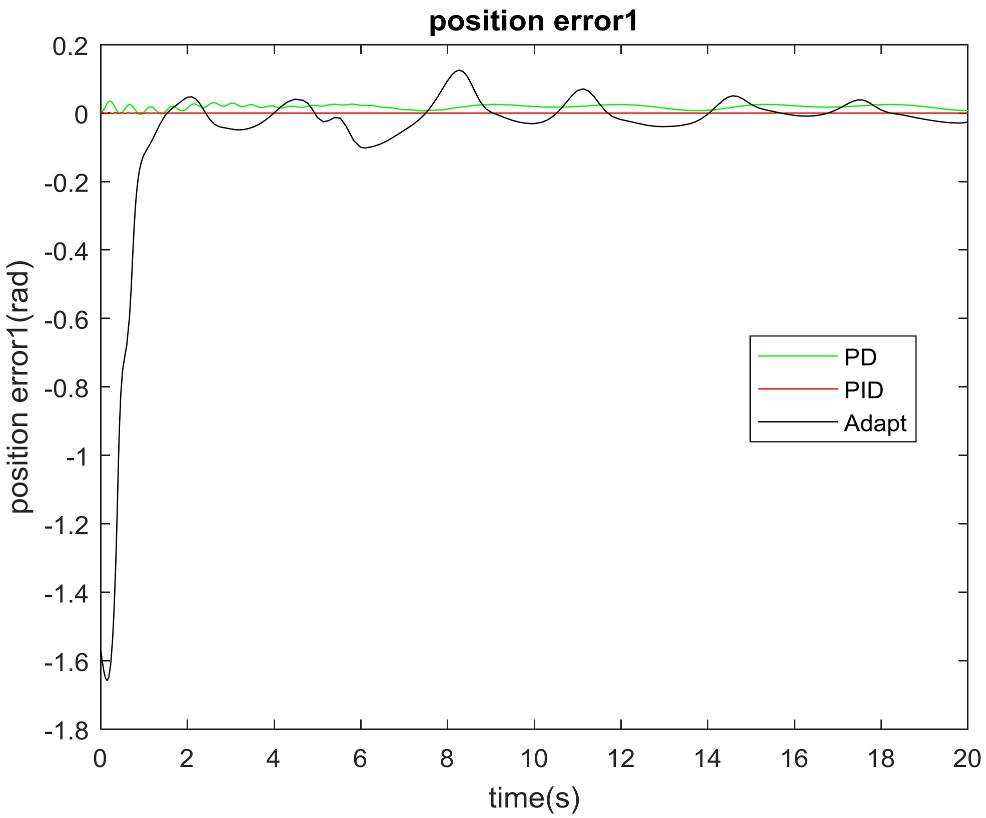


Figure 4.22 :PD, PID and adaptive position errors for link 1

PID control shows an extremely stable error from throughout the time of observation.
Adaptive control shows a shoot from -1.65 rad to 0.5 rad in the first 3 s and and keeps oscillating around zero with a very negligible amplitude.
PD control is steady throughout the time of observation but is less stable than the PID control error

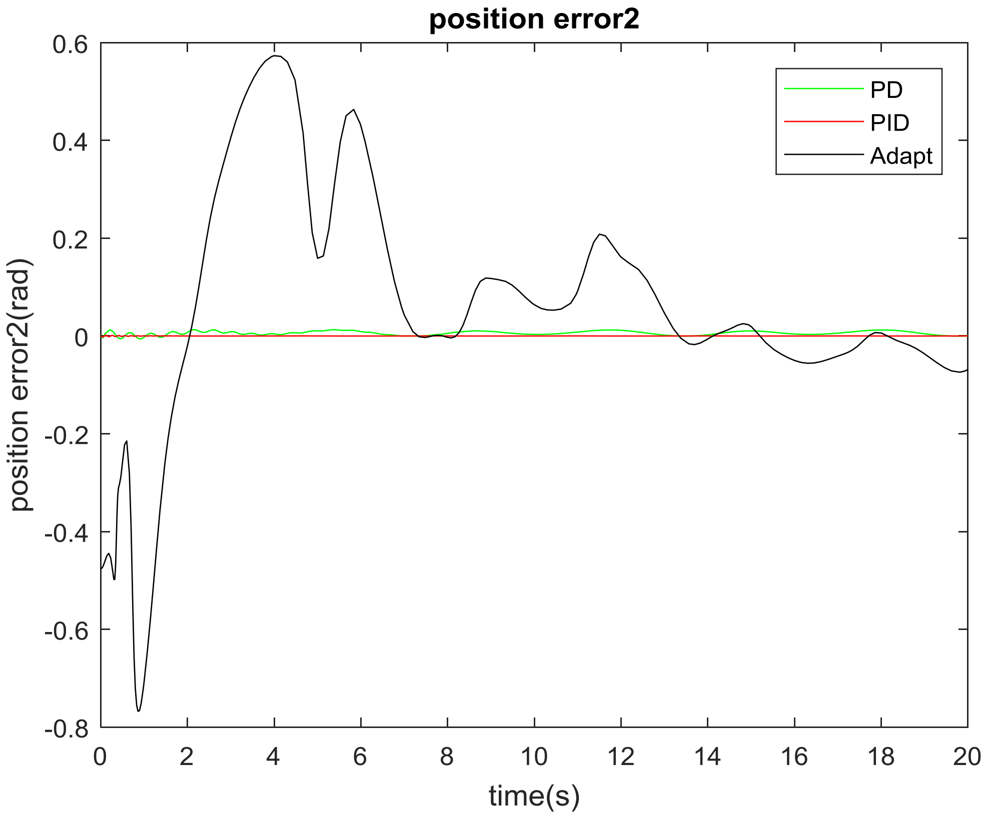


Figure 23: PD, PID and adaptive position errors for link 2

The PID controller gives us the best tracking error which remains almost equal to zero even on the smallest scale.
Adaptive error from -0.55rad to about 0.57 rad in the first 8 minutes then approaches zero thereafter.
The PD controller shows an error 0.02rad in the first 2 seconds and does not show much deviation from of eror throughout the time of observation.

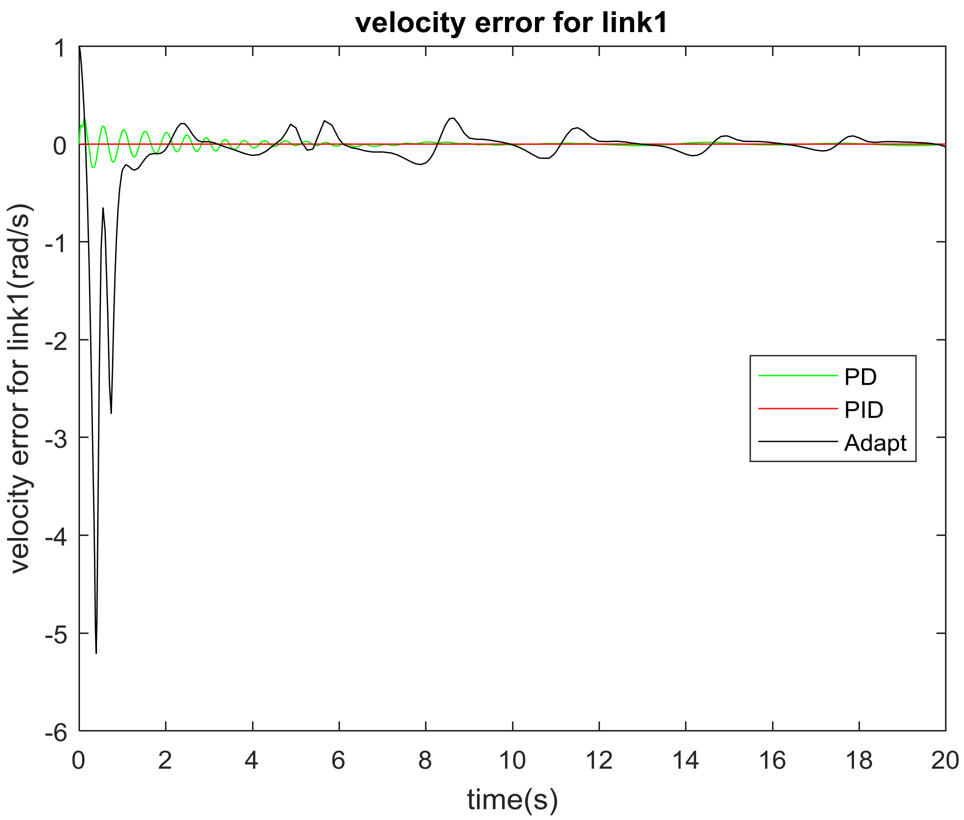


Figure 4.24: PD, PID and adaptive velocity errors for link 1

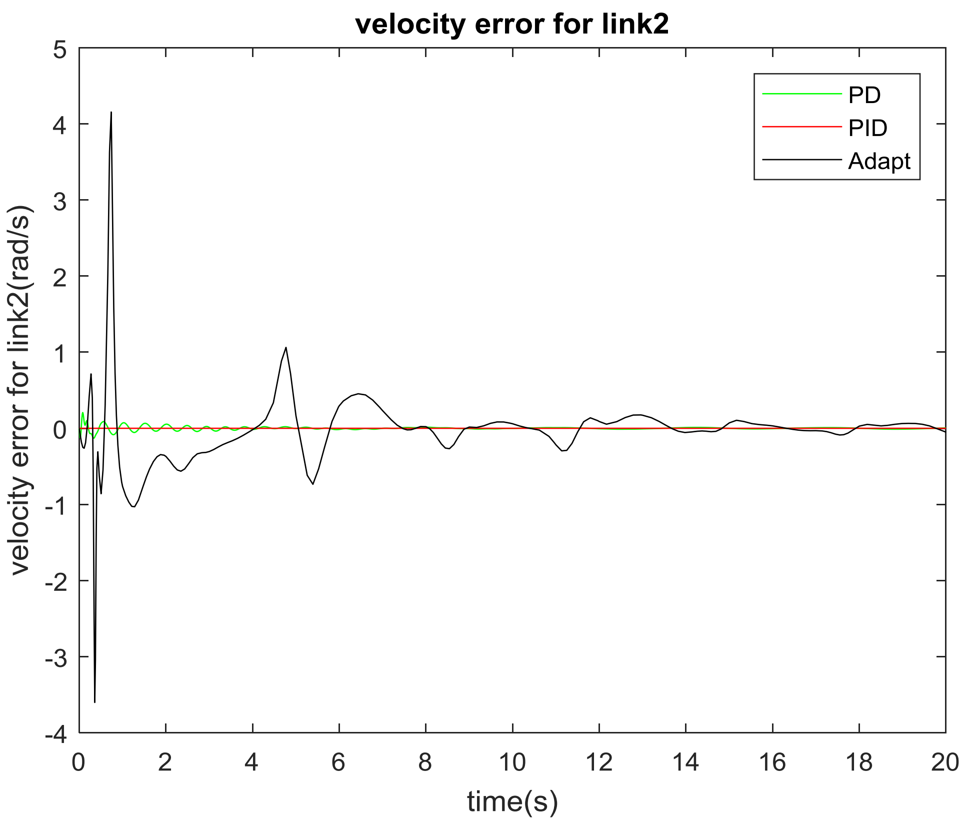


Figure 4. 25: PD, PID and adaptive velocity errors for link 2

As was in all the other figures PID control is the most stable followed by the PD and finally the adaptive controller which shows a n overshoot in the beginning and then oscillates around a constant value with time.
In general all our control systems have shown a very tracking error as the errors were more or less equal to zero

4.6 Comments

4.6.1 PD Controller:

The aim of using P-D controller is to increase the stability of the system by improving control since it has an ability to predict the future error of the system response. In order to avoid effects of the sudden change in the value of the error signal, the derivative is taken from the output response of the system variable instead of the error signal. Therefore, D mode is designed to be proportional to the change of the output variable to prevent the sudden changes occurring in the control output resulting from sudden changes in the error signal. In addition D directly amplifies process noise therefore D-only control is not used.

4.6.2 P-I-D Controller:

The performance of two-link robotic manipulator is investigated with PD, PID control. The PID controller resulted in the best performance and very effective and accurate trajectory tracking capability as compared to PD controller. Also the response with PID controller was having reduced oscillations about the desired trajectory as compared to PD controller
The P-I-D controller has the optimum control dynamics including zero steady state error, fast response (short rise time), no oscillations and higher stability. The necessity of using a derivative gain component in addition to the PI controller is to eliminate the overshoot and the oscillations occurring in the output response of the system. One of the main advantages of the P-I-D controller is that it can be used with higher order processes including more than single energy storage. This is coherent to what was observed by .[O. J. Ogntoyinbo]

4.6.3 Adaptive Controller

It is known that no system is constant and some parameters are likely to vary with time or to the working condition. In classical control, the controller is designed just for the current system model and thus may lose performance or even be unstable due to the system change and uncertainties. In such aspect, adaptive control may be more applicable. In the adaptive approach, the controller attempts to study the uncertain parameters of the system and, if properly designed, will eventually be the best controller for the system in question. The adaptive approach is applicable to a lot of uncertainties. Since the control parameters in the adaptive controller are changing every time, it is difficult to prove its stability. Since the real control problems are not always ideal, it is meaningful to take uncertainties and disturbance into the system model.

4.7 Choosing the right controller

However, no matter how powerful the control method is, there are rarely situations where we do not need to make trade-offs. For example, we are often not only interested in minimizing the tracking error but actually minimizing a cost function such as minimizing the time to reach a set point or to minimize the energy used to reach the set point. We always need to make decisions to increase the income and reduce the expense. We should keep all these factors in mind when choosing a controller. In practice, a compromise controller that reduces our costs while giving a relatively good tracking is what is best.

4.8 Conclusion

In this research we have the history of robotics and much much it has evolved up to this day. We have seen that it increases productivity in the industry and is so much handy as it has taken over repetitive, monotonous and heavy work that used to be dirty and heavy for humans. We have seen different types of classifying robots and we dwelt more into classification by function where we saw robot manipulators and saw why they are of interest for research. We have seen the mechanical structure of a robot and how to come up with a dynamical mdel of a robot with two degrees of freedom when we are given the parameters at each join. We derived the dynamical equations of motion of Lagrange from first principles. We have gone on to look at the different types of computed torque control for designing robot manipulators. These three different types are PD, PID and adaptive. We have seen the result of simulation in Matlab using the three different types of computed torque controllers.

References

[1]Carpin Steffano, **USARSim: a robot simulator for research and education, USA:** 2015

[2]I. C Hofstede Victor, **The importance and purpose of simulation in robotics**, University of Amsterdam, Netherlands 2015

[3].Lewis Frank. L, **Robot Manipulatore Control Theory and Practice:** Dekker Mercel, 2004

[4]. Angels Jorge, Fundamentals of Robotic Mechanical Systems, Ling Frederick F. , Texas 2007

[5] Duysinx. P and GERADIN. M**, An introduction to robotics: mechanical systems**; University of Liège: Novembre 2004

[6]. C. Pepper, C .Scrapper and S. Balarkirsky, Robot Simulation Physics Validation, Gaithersburg USA 2016
[7] E. Tom, T. Yuval and T. Emmanuel **Simulation Tools for Model-Based Robotics**
[8] J. Steven **3D Simulated Robot Manipulation Using Deep Reinforcement Learning,** London June 2016
[9] V. CHANDANA, **Simulation of a humanoid robot** Ohio USA, November 2007
[10] Modélisation et commande des robots manipulateurs par les outils de l'intelligence artificielle
[11] I. Serena, N Francisco and P. Vincent **, Tools for dynamics simulation of robots: a survey based on user feedback,** Feb 2014
[12] Smith . B **Classical vs Intelligent Control:** Memorial University, 2002
[13] Campa. R, Valenzuela. J. M **A Class of OFT Controllers for Torque-Saturated Robot Manipulators:** Spain 2008
[14] Gill. S. S **Two link manipulator Control:** Canada 2012
[15] Jenkins, Control and Dynamics, Brown University
[16] P. H Farzin, M . Samsodini, M. H. Yarmahmoudi, A. Hosainpour, E. Mazlomian **UMA-560 Robot Manipulator Position Computed Torque Control Methods Using MATLAB/SIMULINK and Their**
[17] Integration into Graduate Nonlinear Control and MATLAB Courses: International Journal of Robotics and Automation, (IJRA), Volume (3) : 2012
[18] [The Limitations of Classical PID Controller and Its Advanced Derivations]
[19]R.C. Dorf & R.H. Bishop, Modern Control Systems (Twelfth Edition), Pearson, USA.
[20] O. J. Ogntoyinbo PID control of brushless DC motor and robot trajectory planning and simulation with MATLAB, Vasaan : 2009