ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

DESIGN , KINEMATICS, DYNAMICS AND CONTROL OF 4-DOF UPPER BODY EXOSKELETON

M.Sc. THESIS

Pegah Nomanfar

Faculty of Mechatronics Engineering

Mechatronics Engineering Programme

December 2019

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

DESIGN , KINEMATICS, DYNAMICS AND CONTROL OF 4-DOF UPPER BODY EXOSKELETON

M.Sc. THESIS

Pegah Nomanfar
(518171031)

Faculty of Electrical and Electronics

Mechatronics Engineering Program

Thesis Advisor: Prof. Dr. Seta Bogosyan

December 2019

ISTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

DESIGN , KINEMATICS, DYNAMICS AND CONTROL OF 4-DOF UPPER BODY EXOSKELETON

YÜKSEK LİSANS TEZİ

Pegah Nomanfar
(51817103)

MekatronkisMühendisliğiAnabilim Dalı

MekatronkisMühendisliğiProgramı

Tez Danışmanı: Prof. Dr Seta Bogosyan

Aralık 2019

Pegah Nomanfar, a M.Sc. student of İTU Graduate School of Science Engineering and Technology student ID 518171031, successfully defended the thesis/dissertation entitled "DESIGN , KINEMATICS, DYNAMICS AND CONTROL OF 4-DOF UPPER BODY EXOSKELETON", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Seta Bogosyan** .....
İstanbul Technical University

Jury Members :

Prof. Dr Metin Gokasan .....
İstanbul Technical University

Assoc. Prof. Türker Türker .....
Yıldız Technical University

Date of Submission :20 January 2020
Date of Defense : 19 December 2019

To:

Honey, Maman, Baba, Mishka and Maryam

FOREWORD

I tribute my thesis to the beloved ones of my life, for all the motivations and their support. Also, I would like to thanks Prof.Dr. Seta Bogasyan and Prof.Dr. MetinGokasan, for their support throughout my scientific journey where it had started inside of me, myself.

January 2020 PEGAH NOMANFAR

TABLE OF CONTENTS

Page

ABBREVIATIONS

DoF	**:** Degree of Freedoms
SsRe	**:** Spherical Joint in Shoulder+ Revolute Joint in Elbow
RCM	**:** Remote Center of Motion
D-H	**:** Denavit–Hartenberg
RoM	**:** Range of Motion
E-E	**:** End-Effector
CoM	**:**Center of Mass
PID	**:** Proportional, Integral and Derivative
CTC	**:**Computed Torque Controller
SISO	**:**Single Input/ Single Output
MIMO	**:**Multi Input/ Multi Output

SYMBOLS

J: Jacobian

$\boldsymbol{D(\theta)}$ **:** Matrix of Inertia

$\boldsymbol{C(\theta, \dot{\theta})}$ **:** Vector ofCoriolis and Centrifugal forces

$\boldsymbol{G(\theta)}$ **:** Vector ofGravity Force

$\boldsymbol{\tau}$ **:** Torque

$\boldsymbol{r_{i-1,ci}}$ **:** Thevectors from the origin of frame $i-1$ to the center of mass (CoM) of the link

$\boldsymbol{r_{i,ci}}$ **:** The vectorfrom the origin of the frame i to the CoM of the link i

$\boldsymbol{r_{i-1,i}}$ **:** The vector from the origin of frame i-1to frame i

$\boldsymbol{f_i}$ **:** The applied force from link i-1 on link i

$\boldsymbol{I_i}$ **:** The inertia tensor of link i about a frame parallel to frame i and the origin of the frame is at the CoM of link i

$\boldsymbol{\omega_i}$ **:** Angular Velocity

$\boldsymbol{R_{i+1}^{i}}$ **:** The rotation matrix from frame i to i+1

$\boldsymbol{\alpha_{c,i}}$ **:** The CoM acceleration

$\boldsymbol{\alpha_{e,i-1}}$ **:** Theacceleration of the end of the link

LIST OF TABLES

Page

LIST OF FIGURES

Page

DESIGN , KINEMATICS, DYNAMICS AND CONTROL OF 4-DOF UPPER BODY EXOSKELETON
SUMMARY

Today the technology can be entered to all aspect of human life such as medical issues and treatments. Robotics with biomedical applications has found its way through our lives and has become a new chapter in medical issues, which is generally called assistive robotics. This new area of science can help treating or improving the quality of life of patients specifically the patients with disabilities. There are numerous great robots and mechanisms with special control deign that help people not only during their therapy sessions but also in their daily life. It has been proven that exoskeletons are one of the best types of robots that can be ergonomic and self-aligned with patient's limb. Exoskeletons are now widely used by therapy centers and individually by people in their daily life and activities.

In this thesis, through studying possible mechanisms regarding to upper arm exoskeleton, the best of them is chosen. The main object of this study is to find the most reliable mechanism and after that, the best control design to fully meet with the requirements of stroke patients that have arm disabilities. Important studies within this concept are kinematics studies such as forward and invers kinematics, Jacobean and singularities and most importantly the workspace of each mechanism. In addition, studying the joint trajectories and finding the proper configurations is important.

Later by bringing control and especially vision-based control by neural networks, this case can be studied in a much better fashion.

In this study by proposing the possible mechanism for upper arm exoskeletons, the best mechanism is found according to kinematics, trajectories, and control.

4-DOF ÜST GÖVDE EXOSKELETONUNUN TASARIMI, KİNEMATİĞİ, DİNAMİĞİ VE KONTROLÜ

ÖZET

Günümüzdeteknoloji, tıbbisorunlarvetedavilergibiinsanyaşamının her alanınagirebilmektedir.
Biyomedikaluygulamalarıolanrobotlarhayatımızboyuncayolunubulduvegenellikleyardımcırobotikolarakadlandırılantıbbikonulardayenibirbölümhalinegeldi. Bu yenibilimalanı,
özellikleengellihastalarınyaşamkalitesinitedavietmeyeveyaiyileştirmeyeyardımcıolabilir. İnsanlarasadeceterapiseanslarındadeğil, günlükyaşamlarında da yardımcıolanözelkontroltenezzülünesahipçoksayıda robot vemekanizmavardır. Dışiskeletlerin, ergonomikvehastanınuzuvlarınagörekendiliğindenhizalanabilen en iyi robot türlerindenbiriolduğukanıtlanmıştır. Dışiskeletlerartıkterapimerkezleritarafındanvegünlükyaşamlarındaveaktivitelerindeinsanlartarafındanayrıayrıkullanılmaktadır.
Bu tezde, üstkoldışiskeletineilişkinolasımekanizmalarincelenerek en iyisiseçilmiştir. Bu çalışmanıntemelamacı en güvenilirmekanizmayıbulmakvebundansonrakolengelliinmehastalarınıngereksinimlerini tam olarakkarşılayacak en iyikontroltasarımıdır. Bu kavramdakiönemliçalışmalar, ileriveterskinematik, Jacobean vetekilliklerve en önemlisi her mekanizmanınçalışmaalanıgibikinematikçalışmalardır. Ekolarak, ortakyörüngeleriincelemekveuygunkonfigürasyonlarıbulmakönemlidir.
Dahasonrakontrolveözelliklesinirağlarıtarafındangörmetabanlıkontrolgetirilerek, bu durum çokdahaiyibirşekildeincelenebilir.
Bu çalışmada, üstkoldışiskeletiçinolasımekanizmaönerilerek, en iyimekanizmakinematik, yörüngelervekontrolegörebulunmuştur.

1. INTRODUCTION – ASSISTING STORK PATIENTS WITH EXOSKELETON ROBOTS

The term exoskeletons means external skeleton. This device has been widely used and researched for many purposes over the decades. The history of very first exoskeleton as a device which is known today, goes back to 1965 when it was called Hardiman which was invented by Ralph Mosher and his team. It was a big wearable device that could assist the human with lifting devices. Although this device was practically useless due to its weights and size, but the exoskeleton found its way from conceptual form to the practical world [1][2]. Today exoskeleton robots are widely researched in the areas related to rehabilitation [3], [4], assistive robotics impairment evaluation [5], resistance exercises [6] and haptic interaction in tele-operated and virtual environments [3], [7]. Medical and biomedical applications are one of the biggest usage areas of exoskeletons. Emerging exoskeletons in curing or assisting patients that suffer from different types of rehabilitations is one of the high usage applications of exoskeletons [7]. In this research, it medical purposes are consideredand more accurately, the application of exoskeletons for assisting upper body with emphasize on shoulder joint assisting is investigated.

1.1Exoskeleton in Assisting Robotics

In literature, "assisting robots" are mainly categorized into two typesin terms of their usages: upper-body exoskeleton and lower-body (or leg exoskeleton), which are worn by human arm and leg consequently. Also, there are two types of highly-used robots in upper limb therapy: one is exoskeleton and the other one is end-effector type of robot [8]. As interaction point of the end-effector robots with human limb is only one point, the posture of arm and each joint required torque cannot be estimated accurately. However, the wearable exoskeletons with more complex dynamics systems and structures can have numerous connection points with human limb.

These characteristics of exoskeletons allow receiving the control feedback from each individual joint [9]. In this research the term “exoskeleton” refers to upper-body and wearable arm exoskeleton.

Assisting robots are one of the highly used devices by therapists in medication [10].Robots that are assisting with rehabilitation cases should be able to lead the movement of the limb with intelligent control and continuous feedback from sensorimotor. This can include repetitive mood and performance monitoring, for physiotherapy centers in order to practice the limb undersupervision of therapists. Therefore, the old school therapy methods have changed by bringing robots that can work with high efficiency and less fatiguefor patients and therapists [11].

Exoskeletons are robots that can be worn by human limb and act in parallel with limb [12] and are one of the best-known robots in limb therapy that can be used as a device in physiotherapy centers with supervision of therapists in fixed-wearable mood for high repetitive limb training or used by patients in complete wearable mood for enhancing limb strength in daily activities.

Shoulder joint mobility is playing an essential role in assisting human in her/his activities of daily living (ADL). It has been estimated that impaired shoulder joints has more than 15% dispersal in population [13] [14] [15]. Moreover, it has been observed that the shoulder dysfunctionality can be grew by aging around 20% to 27%. The main reason is the pain resulted from rotator cuff disease, strength losing and decreasing in shoulder mobility [16][17].

Over the past two decades the active upper limb assisting systems have found their way into medical and rehabilitation purposes [18]. The early studies have been done mostly on end-effector robots and as it was mentioned before, there is only one interface point with human arm. Therefore the human arm movement is not followed completely while the device is being used. Later the rehabilitation robotics researchers switched more towards the exoskeletons-based robots since these devices have the capacity of tracking human motions based on human anatomy and they can align their axis with human limb axis. This characteristic of the exoskeletons guarantee the safe performance when they are attached to human upper limb [19]. Most exoskeletons have been built to enhance the motor functionality for stroke rehabilitations. This means rehabilitation improvements of stroke patients

have been one of the main purposes of designing upper limb exoskeleton [18]. These devices are stationary such as CADEN [20]and ARMin [21] and CLEAVER [22]. The bright side of these devices is that the patient should not overcome the weight of the robot links [23]. One important aspect of this technology is that most of the devices are designed with the fixed based and for clinical purposes. However the lack of portable and wearable devices in the exoskeletons which have been commercialized for the use of every one is seen clearly [24]. There are portable devices for upper body exoskeleton such as MGA [25], RUPERT IV [26] which have been commercialized but they are very expensive and not affordable for every one specially for elderlies and less wealthy people are not able to benefit from the devices with high efficiency and good price to help them with short period or life-period stroke rehabilitations.

1.2 Literature Review on Wearable Arm Exoskeletons in Stroke Patients

Exoskeleton robots which are attached to the limb are able to help with human limb motions directly. Therefore, today there are many active exoskeletons which are designed for arm in order to assist with human limb motions and assisting them [27][4][9][7][3].

Limbs and muscles rehabilitation in stroke paints are curable by using exoskeletons either for practicing in fixed-wearable mood in physiotherapy centersor forincorporating with patients daily activitiesin completely wearable mood outside centers [28].World Health Organization (WHO) statistics shows that by 2050 the over 60 population of the world will be doubled and this elderly population can seriously face the stroke and its side effects [28].

1.3 Human Arm Kinematics

To discover a well-aligned exoskeleton mechanism with the arm, firstly, it is important to discuss about its comfortability and efficient performance [29]. Numerous evidencesshow that the motion in shoulder joint includes translational and rotational movements. The elevation/depression, protraction/retraction are the translational motions, flexion/extension, external/internal rotation and horizontal

abduction/adduction are rotational motions of arm [30][31]. In this case the shoulder joint can be observed as 5 DoF kinematics chain model [31].

By observing tasks in the shoulder joint complex, it is seen that the motion in the shoulder is performed by glenohumeral joint and sous-acromial joint which together form the 3 rotational DoFs [30]. Researchers in the field of exoskeleton robots have mostly designed and proposed upper body exoskeletons based on 3 DoFs. The reason is that the 2 translational DoFs are between joints and they are very small and neglected in the design. Also, in the elbow two types of rotations are flexion/extension (EFE) and internal/external rotation (IE) and researchers consider flexion/extension in the elbow joint during the designing phase. [31] Therefore, the total DoFs in the whole arm for designing the exoskeletons can be varied between minimum 4 DoFs and maximum 8 DoFs . Figure .1 explains the movement types in the shoulder within 5 DoFs.

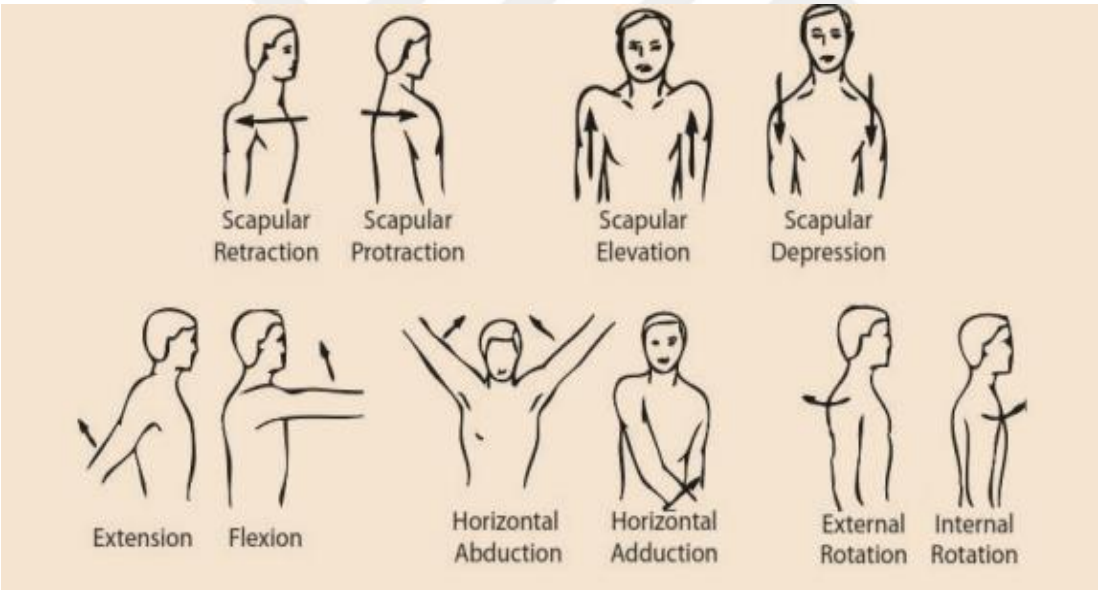

Figure 1.1 Types of movement in shoulder within 5DoF.

1.4 Challenges in Designing Upper-Body Exoskeletons

In order to design upper body exoskeleton, there are number of designing challenge. The reason is mostly relying of number of DoFs in shoulder which leads to require a design with high level of dexterity in motions due to Instantaneous Center of

Rotation (ICR) position that can shift in shoulder and elbow joint by any movement [32]. Also, during the motion the incoordination between human and robot joint axes can happen [33]. There are some problems that can occur due to the explained in coordination between human limb and robot. One of the issue is transmission high level of forces and torques to the connection spot of robot and human limb. This issue not just bringing inconvenience to the user, but also, in some situations can be lead to high level of limb injuries [34]. Another challenge in designing is kinematics design in arm exoskeleton. In designing the lower body exoskeleton there is no such problem because of less level of range of motion. However, this situation is different in arm since arm dexterous motion is high level because of shoulder [35][36].

1.5 Literature review on designed mechanisms for upper arm exoskeletons

There have been many proposed methods throughout the history to face the difficulties in designing arm exoskeletons such as implanting revolute joints serially and applying them in 4-bar mechanism in two parallelograms [36], continuum mechanism in shoulder that can be adjusted with different shoulder and arms size and many models with high capability of holding up the shoulder gridle joint and glenohumeral joint related to wearied exoskeleton device such as CLEAVER ARM [37] and some modern designed devices by considering the Remote Center of Motion (RCM) method have been applied to bringing three rotations around one point by utilizing a two leg-parallel mechanism [38] etc. They have been designed mainly for rehabilitations, enhancing the power and haptic interactions [39].

1.6 Hypothesis

As it has been explained and explored, In many cases of stroke rehabilitation, it is way too improving to use the transferable and wearable exoskeleton in order to assist the patients with their daily life and practice the limb in order not to let the limb get weak. Many studies have been done and many devices have been invented already for which not many of them are economically commercialized for the patients coming from low level wealth segments in the society. Therefore, the lack of wearable devices that can help to improve the upper limb stroke rehabilitation for short/long term is being sensed.

In this research, a new model arm exoskeleton with capabilities of essential shoulder movements will be introduces will be investigatedfrom design, kinematics, dynamics and control aspect of view. In this particular study, the dynamic model of the device will be explored carefully with Newton-Lagrange method, in order to design the controller (specifically torque controller) for the whole device accurately, in order for the injured limb to come up the careful and required amount of interactive torque. This issue will save the user from extra limb injuries and help him/her with curing process in a more efficient way. Also, this device is designed to lift the arm and move it to the desired position and balance the gravity force which is applied to the arm, since stroke patients usually have numbness in their limb and cannot lift their arm and forearm.

2. KINEMATICS DESIGNSOF FULLY WEARBLE ARM EXOSKELETON ROBOTS

In designing a suitable exoskeleton which is user friendly and arm self-aligning, considering the relation between easy forward kinematics and complexity of the designed mechanism is necessary [40] [41].

This means, although, a simple forward kinematics is leading to an easy computations of invers kinematics, Jacobian matrix and maybe an easy control designing method, however, this issue can increase the intricacy of the design by adding extra links and joints, which leads to more weight, higher cost and failure in assembly process.

One optimized and innovative approach for a systematic robot designing is to first calculate the number of required joints and links for a certain number of required DoFs. In this way finding the best-fitted robot mechanisms by determining the possible designed mechanisms and comparing the characteristics of each robot can be revealed.

The mathematical equation for calculating DoFscan be represented as 2.1 equations [42]:

$$M = 3(L - 1) - 2J_1 - J_2 \tag{2.1}$$

By applying equation 2.1, the possible number of links and joints for optimizing the kinematics of exoskeleton robot can be achieved. The possible results for minimum number of links and joints are explained in table 2.1.

These mechanisms can be compared with the mechanism that possesses a spherical joint in the shoulder and revolute joint in the elbow. The reason is that from mechanism designer point of view, human shoulder joint can be modeled as spherical joint. This means spherical joint 3DoFs can be simulated as human shoulder joint by

considering 3 rotational DoFs of shoulder and neglecting 2 translational movements. By bringing the spherical joint into the design with 3 rotational DoFs and the revolute joint in elbow with 1 DoFs, the total robot DoFs becomes as 4. Another reason for this comparison is that since the spherical joint is the best interpretation of the shoulder joint through the mechanism and kinematics viewpoint, however, there are doubts about its efficiency compared to other mechanisms. Although, considering a spherical joint in shoulder can lead to an easy designing procedure compared to the proposed mechanisms in the table 2.1, the control design can be more complicated on the other hand since in spherical joint there are three variables that should be cared for. Other mechanisms may seem more complex but they have formed from 1DoF joints. By studying these mechanisms and comparing, them with the spherical joints in the shoulder plus the revolute joint in the elbow (SsRe), the pros and cons of each design will be revealed.

Table 2.1 : Possible mechanisms for arm exoskeleton robot

Mechanisms	Links (L)	1 DoF Joint(J_1)	2 DoF Joint (J_2)	Total DoFs
Series Links in Shoulder + Elbow Revolute Joint	5	4	0	3+1
3RRR Mechanism in Shoulder + Elbow Revolute Joint	9	10	0	3+1
4 Bar Chain Mechanism in Shoulder + Elbow Revolute Joint	6	5	0	4+1
5 Bar Chain in Shoulder + Elbow Revolute Joint	7	6	0	5+1

By applying equation 2.1 and finding out the feasible mechanisms based on the minimum and maximum requirements of DoFs which are 4 and 6 DoFs successively and are shown in table 2.1, wasting time on failure in design mechanism can be prevented. In table 2.1, the "Elbow Revolute Joint" is considered to be 1 DoF and this one DoF is shown as "+1" on the "Total Joint" column section. This means that the shoulder joint itself is the main challenge in designing. Also, this table is not unique, since numbers of different designed mechanism can be found in different joints and links for the shoulder, but it conveys enough information regarding to optimized way of designing.

As the number of possible mechanisms for designing an arm exoskeleton was discussed above, it will be much beneficiary if the methods in designing each mechanism is considered. One of the best-known design methods which is highly practical in medical robotics and bio mechatronics is Remote Center of Motion (RCM) types of mechanism which lets the revolute axis of joints or the rotation center of a mechanism to have an intersection point on a space where no mechanical component exists [41].

This is an effective characteristics since it can be practical in medical applications and let the rotation point to be fixed at patient's limb. In addition, different DoFs of the shoulder have intersection points with each other. This fact leads to conclude that the best mechanism for wearable upper arm exoskeleton is an RCM mechanism.

2.1 Kinematics Design of Possible mechanisms in Upper-Body Exoskeletons

In this section the kinematics characteristics of each mechanism such as forward and invers kinematics, Jacobian and singularities will be discussed. The approach in proving each calculation is done by using MATLAB, MATLAB robotic toolbox, Simulink and Simscape Multibody. For this section there are 2 designs that are done by SOLIDWORKS previously which include SsRe and serial links and are brought to Simscape Multibody environment for simulations. Other mechanisms are also studied by using Simscape Multibody.

2.2 Forward and invers kinematics

Forward kinematics is the way of calculating the end-effector orientation and position, based on the joint variables. One of the notable methods in this calculation is Denavit–Hartenberg (D-H) method.

2.2.1 Spherical joint in the shoulder and revolute joint in the elbow (SsRe)

Figure 2.1 consists of the schematics of this design related to figure 1.1 which is the designed mechanism in SOLIDWORKS. also table 2.2 is showing the Denavit–Hartenberg (D-H) parameters equivalent to joint transformation matrices which can be found in table 2.2 and equation 2.2. The results appear in 2.3 equation.

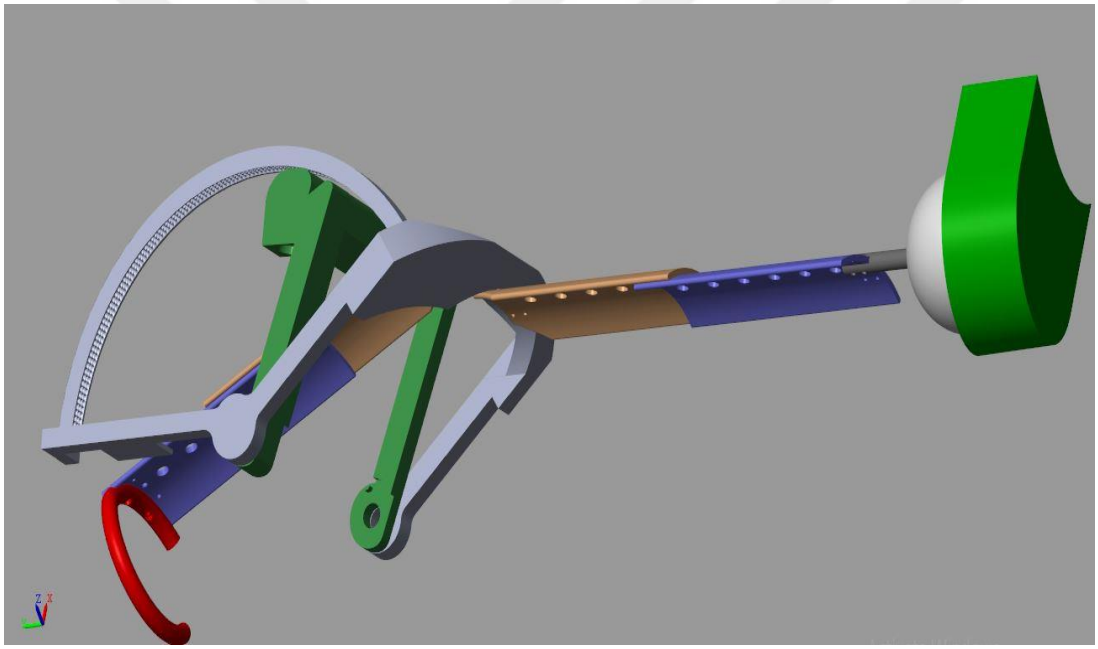

Figure 2.1 : Spherical joint in shoulder and revolute joint in elbow model SsRe.

Table 2.2 : D-H parameters of spherical joint in shoulder and revolute joint in elbow

Links (i)	α_{i-1} ***(rad)***	a_{i-1}	d_{i-1}	θ_{i-1} ***(rad)***
1	$\pi/2$	0	0	$\pi/2+\theta_1$
2	$\pi/2$	0	0	$\pi/2+\theta_2$
3	$\pi/2$	0	L_1	$\pi/2+\theta_3$
4	0	L_2	0	θ_4

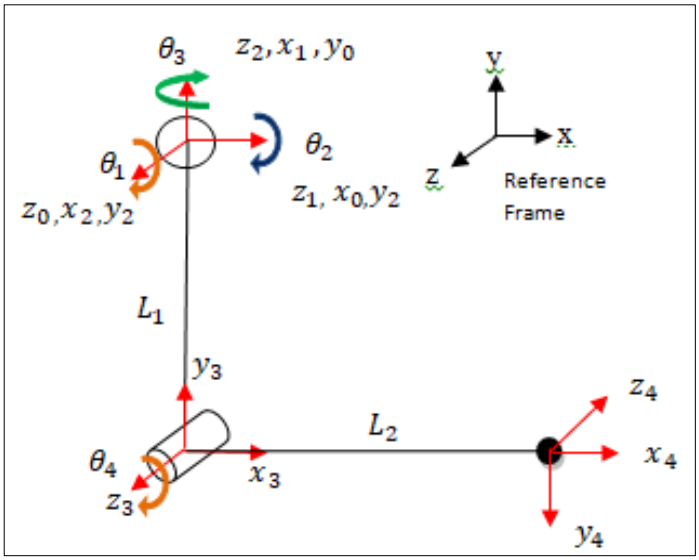

Figure 2.2 : Kinematics configurations of SsRe.

$$T_1^0 = \begin{bmatrix} -s_1 & 0c_1 & 0 \\ c_1 & 0s_1 & 0 \\ 0 & 10 & 0 \\ 0 & 00 & 1 \end{bmatrix}, \quad T_2^1 = \begin{bmatrix} -s_2 & 0c_2 & 0 \\ c_2 & 0s_2 & 0 \\ 0 & 10 & 0 \\ 0 & 00 & 1 \end{bmatrix} \tag{2.2}$$

$$T_3^2 = \begin{bmatrix} -s_3 & 0c_3 & 0 \\ c_3 & 0s_3 & z0 \\ 0 & 10 & L_1 \\ 0 & 00 & 1 \end{bmatrix}, \quad T_4^3 = \begin{bmatrix} -s_4 & 0c_4 & L_2c_4 \\ c_4 & 0s_4 & L_2s_4 \\ 0 & 10 & 0 \\ 0 & 00 & 1 \end{bmatrix}$$

$$T = T_4^0 = T_1^0 T_2^1 T_3^2 T_4^3 = \begin{bmatrix} \mathrm{T}_{11} & \mathrm{T}_{12}\mathrm{T}_{13} & \mathrm{X} \\ \mathrm{T}_{21} & \mathrm{T}_{22}\mathrm{T}_{23} & Y \\ \mathrm{T}_{31} & \mathrm{T}_{32}\mathrm{T}_{33} & \mathrm{Z} \\ 0 & 0\ 0 & 1 \end{bmatrix} \tag{2.3}$$

$$T_{11} = c_4(c_1c_3 - s_1s_2s_3) - c_2s_1s_4$$
$$T_{12} = -s_4(c_1c_3 - s_1s_2s_3) - c_2c_4s_1$$
$$T_{13} = c_1s_3 - s_1s_2c_3$$
$$T_{21} = c_4(s_1c_3 + c_1s_2s_3) + c_1c_2s_4$$
$$T_{22} = -s_4(s_1c_3 - c_1s_2s_3) + c_2c_4c_1$$
$$T_{23} = s_1s_3 - c_1c_3s_2$$
$$T_{31} = s_2s_4 - c_2c_4s_3$$
$$T_{32} = s_2c_4 + c_2s_3s_4$$
$$T_{33} = c_2c_3$$
$$X = L_2c_4(c_1c_3 - s_1s_2s_3) - L_1c_2s_1 - L_2c_2s_1s_4$$
$$Y = L_2c_4(s_1c_3 + c_1s_2s_3) + L_1c_1c_2 + L_2c_2c_1s_4$$
$$Z = L_1s_2 + L_2s_2s_4 - L_2c_2s_4s_3$$

Also, by using MATLAB robotics toolbox, the validity of obtained results for D-H method are tested. Figure 2.3, demonstrates the forward Kinematics simulation of inputs based on D-H parameters for every link regarding table 2.2 and equation 2.3. Figure 2.3 is showing the conceptual design worn by human arm. After achieving the results from the forward kinematics for commuting the end effector position and orientation, invers kinematics calculation is also one of the necessary calculations in further steps in control design.

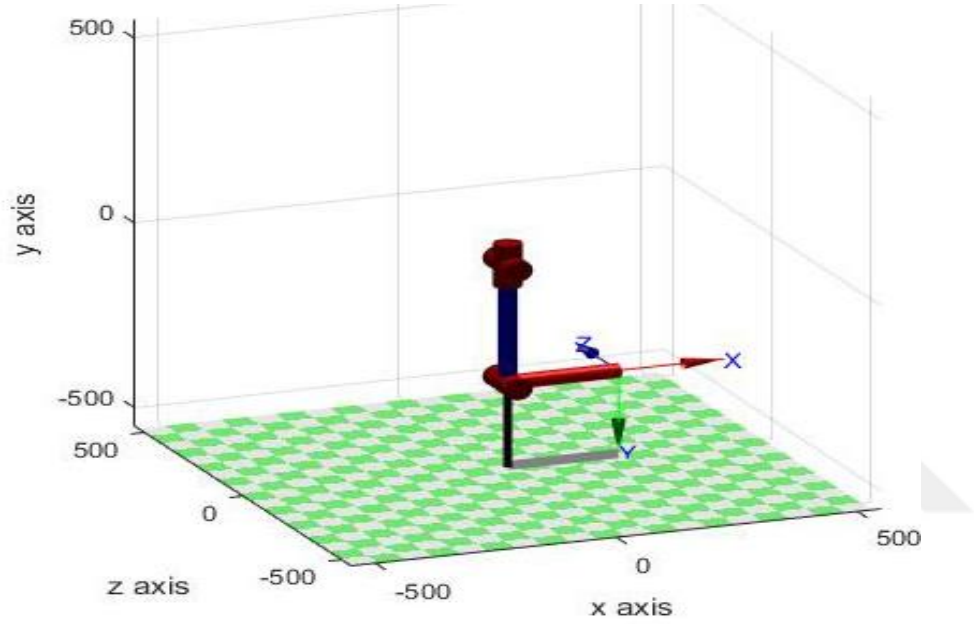

Figure 2.3 : Forward kinematics simulation of SsRe by using MATLAB robotics toolbox.

Invers kinematics job is to find joint variables according to robot end-effector position and orientation. Figure 2.4 explains the invers kinematics parameters such as "W"and "E" which imply wrist and elbow successively. Equations 2.7 to 2.10 describe the θ1, θ2 and θ3, as the angle between L1 and z , x and y axis respectively.

From figure 2.4, φ_3 is the angle between *y*-axis and R also, φ is the angle between *L1* and *R*. The details can be found in equations 2.4 to 2.7.

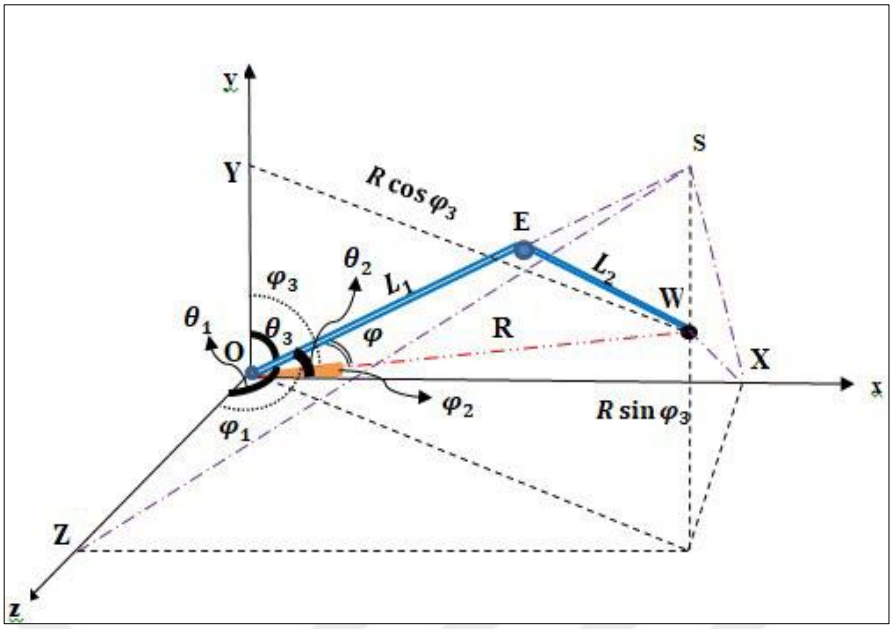

Figure 2.4 : : Invers kinematics variables of SsRe..

$$R = \sqrt{{L_1}^2 + {L_2}^2 + 2L_1L_2\cos\theta_4}$$

$$\theta_4 = \cos^{-1}\left(\frac{R^2 - {L_1}^2 - {L_2}^2}{2L_1L_2}\right) \quad (2.4)$$

$$\varphi_3 = \sin^{-1}\left(\frac{Y}{R}\right)$$

$$\varphi = \cos^{-1}\left(\frac{{L_2}^2 - {L_1}^2 - R^2}{-2L_1R}\right)$$

$$\theta_3 = \varphi_3 - \varphi \quad (2.5)$$

$$\theta_{zx} = \tan^{-1}\left(\frac{Z}{X}\right)$$

$$SX = \sqrt{(R\sin\varphi_3 + L_2\sin\theta_4)^2 + (R\cos\varphi_3\sin\theta_{zx})^2}$$

$$\theta_2 = \cos^{-1}\left(\frac{SX^2 - (L_1 + L_2\cos\theta_4)^2 - X^2}{-2(L_1 + L_2\cos\theta_4)X}\right) (2.6)$$

$$SZ = \sqrt{(R\sin\varphi_3 + L_2\sin\theta_4)^2 + (R\cos\varphi_3\cos\theta_{zx})^2}$$

$$\theta_1 = \cos^{-1}\left(\frac{SZ^2 - (L_1 + L_2\cos\theta_4)^2 - Z^2}{-2(L_1 + L_2\cos\theta_4)Z}\right) (2.7)$$

2.2.2Serial links in shoulder and revolute joint in elbow

As discussed above, the serial link is the best economical approach in designing since there is less complexity in assembling parts and less number of links and joints are used for the design. Figure 2.5 shows the designed mechanism in SOLIDWORKS environment and figure 2.6 explains the schematics form of the designed mechanism. Also, table 2.3 reveals the D-H parameters and equations 2.8 and 2.9 refer to transform matrix and forward kinematics equations respectively.

Table 2.3 : D-H parameters of serial links joint in shoulder and revolute joint in elbow

Links (i)	$\boldsymbol{\alpha}_{\mathbf{i-1}}$ ***(rad)***	$\mathbf{a_{i-1}}$	$\boldsymbol{d_{i-1}}$	$\boldsymbol{\theta_{i-1}}$ ***(rad)***
1	$-\frac{\pi}{2}$	$\mathrm{Rsin}\frac{\pi}{4}\sin\frac{67.5\pi}{180}$	$\mathrm{Rsin}\frac{\pi}{4}\sin\frac{22.5\pi}{180}$	θ_1
2	$\frac{\pi}{2}$	$\mathrm{Rsin}\frac{\pi}{4}$	0	$\theta_2+\frac{\pi}{2}$
3	0	L_1	0	$\theta_3-\frac{\pi}{2}$
4	0	L_2	0	θ_4

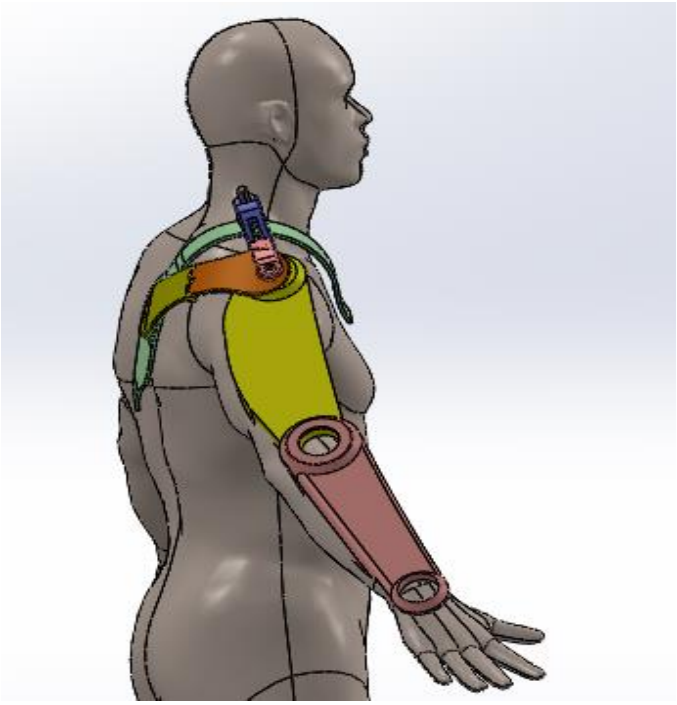

Figure 2.5 : : Serial links mechanism in SOLIDWORKS environment.

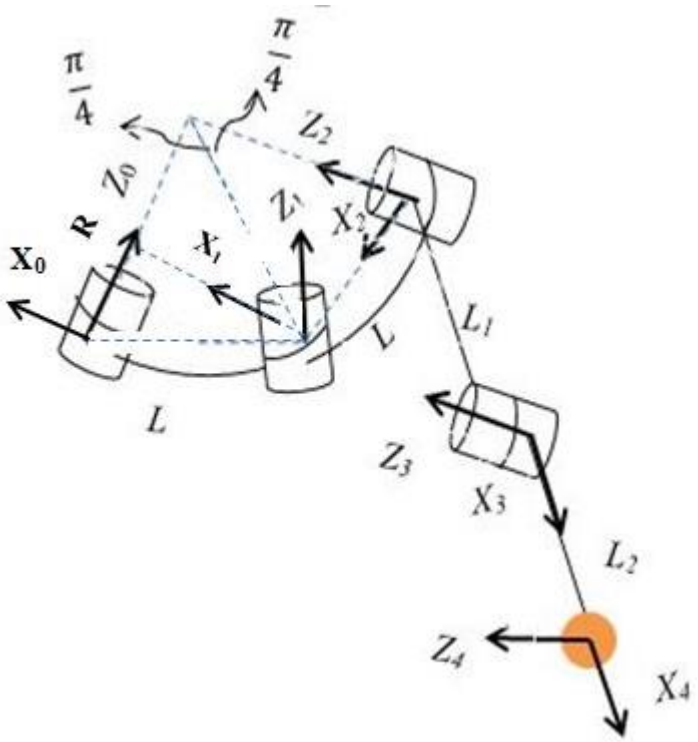

Figure 2.6 : : Schematics of serial links mechanism.

$$T_1^0 = \begin{bmatrix} c_1 & 0 & -s_1 & 0.65\,R\,c_1 \\ s_1 & 0 & c_1 & 0.65\,R\,s_1 \\ 0 & -1 & 0 & 0.27\,R \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^1 = \begin{bmatrix} -s_2 & 0 & c_2 & -0.71\,R\,s_2 \\ c_2 & 0 & s_2 & 0.71\,R\,c_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

$$\boldsymbol{T_3^2} = \begin{bmatrix} \boldsymbol{s_3} & \boldsymbol{c_3\,0} & \boldsymbol{L_1 s_3} \\ \boldsymbol{-c_3} & \boldsymbol{s_3\,0} & \boldsymbol{-L_1 c_3} \\ \boldsymbol{0} & \boldsymbol{0\,1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0\,0} & \boldsymbol{1} \end{bmatrix}, \quad \boldsymbol{T_4^3} = \begin{bmatrix} \boldsymbol{c_4} & \boldsymbol{-s_4} & \boldsymbol{0} & \boldsymbol{L_2 c_4} \\ \boldsymbol{s_4} & \boldsymbol{c_4} & \boldsymbol{0} & \boldsymbol{L_2 s_4} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} \end{bmatrix}$$

$$T = T_4^0 = T_1^0 T_2^1 T_3^2 T_4^3 = \begin{bmatrix} \mathrm{T}_{11} & \mathrm{T}_{12}\mathrm{T}_{13} & \mathrm{X} \\ \mathrm{T}_{21} & \mathrm{T}_{22}\mathrm{T}_{23} & \mathrm{Y} \\ \mathrm{T}_{31} & \mathrm{T}_{32} - c_2 & Z \\ 0 & 0\ 0 & 1 \end{bmatrix} (2.9)$$

Also, the proved forward kinematics calculations by using MATLA robotic toolbox can be found by figure 2.7. The angle configuration in figure 2.7 is $[0^o 0^o 0^o 0^o]$.

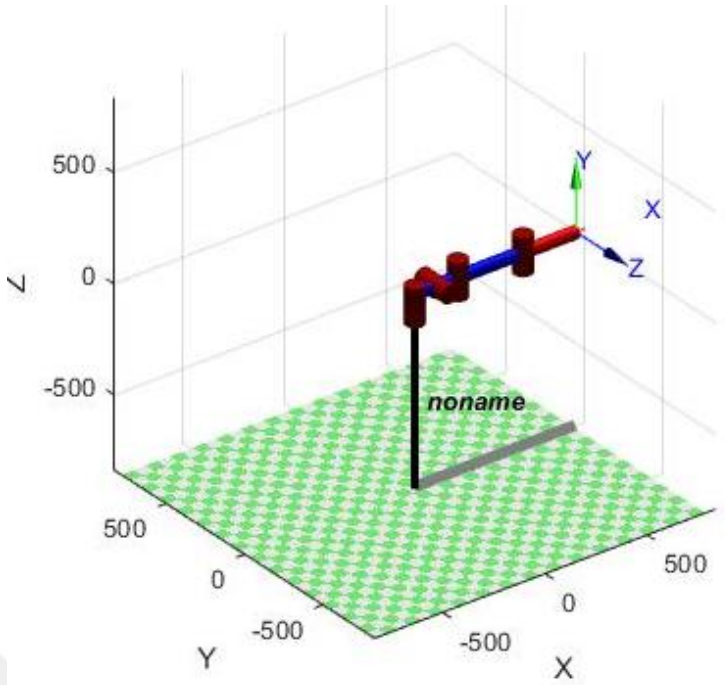

Figure 2.7 : : Forward kinematics simulation of Serial link by using MATLAB robotics toolbox.

The angle configuration in figure 2.8 is $[\mathbf{0}^{o}\ \frac{\boldsymbol{\pi}}{\mathbf{2}}\ \frac{\boldsymbol{\pi}}{\mathbf{2}}\ \mathbf{0}^{o}]$

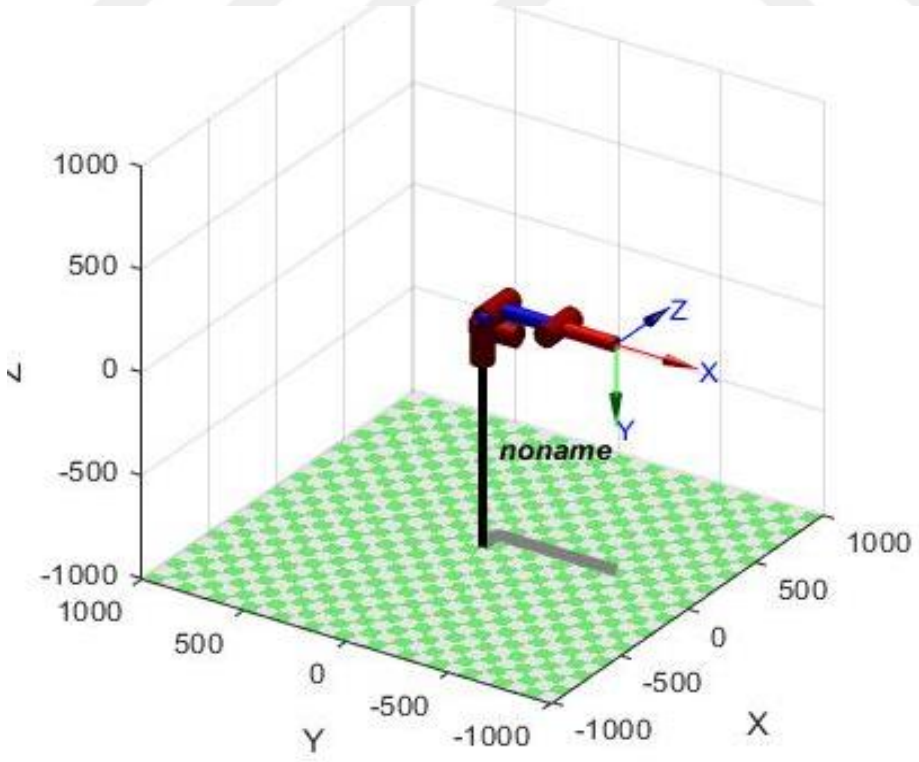

Figure 2.8 : : Forward kinematics simulationofSeriallinkbyusing MATLAB robotics toolbox.

2.3 Jacobian and singularities

Jacobian matrices are one of the important calculations in robotics to realize the level of motion for forward and inverse kinematics. Jacobian calculations are highly used in control of robot. Another concept in robot control is singularity. Singularities give information about the special situations that robot joints are moving but the end-effector losses its ability to move in some direction.

2.3.1 Spherical joint in shoulder and revolute joint in elbow (SsRe)

The Jacobian matrix of SsRe can be found as equation 2.10.

$$\boldsymbol{J}(\boldsymbol{q}) = \begin{bmatrix} J_{11} & J_{12} & J_{13}J_{14} \\ J_{21} & J_{22} & J_{23}J_{24} \\ 0 & J_{32} & J_{33}J_{34} \\ 0 & c_1 & -c_2 s_1 \; J_{44} \\ 0 & s_1 & c_1 c_2 \; J_{54} \\ 1 & 0 & s_2 \quad c_2 c_3 \end{bmatrix} \tag{2.10}$$

$$J_{11} = -L_2 c_4 (c_3 s_1 + c_1 s_2 s_3) - L_1 c_1 c_2 - L_2 c_2 c_1 s_4$$
$$J_{12} = s_1 (L_1 s_2 + L_2 s_2 s_4 - L_2 c_2 s_4 s_3)$$
$$J_{13} = c_1 c_2 (L_1 s_2 + L_2 s_2 s_4 - L_2 c_2 s_4 s_3) - s_2 (L_2 c_4 (s_1 c_3 + c_1 s_2 s_3) + L_1 c_1 c_2 + L_2 c_2 c_1 s_4)$$
$$J_{14} = (L_2 s_2 s_4 - L_2 c_2 c_4 s_3)(s_1 s_3 - c_1 c_3 s_2) - c_2 c_3 (L_2 c_4 (s_1 c_3 + c_1 s_2 s_3) + L_2 c_2 c_1 s_4)$$
$$J_{21} = L_2 c_4 (c_1 c_3 - s_1 s_2 s_3) - L_2 c_2 s_1 - L_2 c_2 s_1 s_4$$
$$J_{22} = -c_1 (L_1 s_2 + + L_2 s_2 s_4 - L_2 c_2 s_4 s_3)$$
$$J_{23} = c_2 s_1 (L_1 s_2 + L_2 s_2 s_4 - L_2 c_2 c_4 s_3) - s_2 (L_1 c_2 s_1 - L_2 c_4 (c_1 c_3 - s_1 s_2 s_3) + L_2 c_2 s_1 s_4$$
$$J_{24} = c_2 c_3 (L_2 c_4 (c_1 c_3 - s_1 s_2 s_3) - L_1 c_2 s_1 s_4 - L_2 s_2 s_4 - L_2 c_4 c_2 s_3) c_1 s_3 c_3 s_1 s_2)$$
$$J_{32} = c_1 (L_2 c_4 (c_3 s_1 + c_1 s_2 s_3) + L_1 c_2 c_1 + L_2 c_2 c_1 s_4 + s_1 (L_1 c_2 s_1 - L_2 c_4 (c_1 c_3 - s_1 s_2 s_3) + L_2 c_2 s_1 s_4$$
$$J_{33} = c_1 c_2 (L_1 c_2 s_1 - L_2 c_4 (c_1 c_3 - s_1 s_2 s_3) + L_2 c_2 s_1 s_4) - c_2 s_1 (L_2 c_4 (s_1 c_3 + c_1 s_2 s_3) + L_1 c_2 c_1 + L_2 c_2 c_1 s_4)$$
$$J_{34} = (c_2 s_1 + c_3 s_1 s_2)(L_2 c_4 (s_1 c_3 + c_1 s_2 s_3) + L_2 c_2 c_1 s_4 - s_1 s_3 - c_1 c_3 s_2)\,(L_2 c_4 (s_1 c_3 - s_1 s_2 s_3) - L_2 c_2 s_1 s_4$$

$$\boldsymbol{J_{44} = c_1 s_3 + c_3 s_1 s_2}$$

Another essential design consideration is singularity and specially with spherical joint which has 3 DoFs. In order to study the singularity in spherical joints, firstly by translating spherical joint to 3 consequence rotational joint, regarding to $\boldsymbol{q_i}$, the velocity equation can be achieved as equation 2.11.

$$\dot{q} = J_s(q)^{-1} w_s \tag{2.11}$$

As the assumption of finding the velocity equation is considering the spherical joint as a three serial revolute joint, following the analysis about the singularity can also be used for serial link mechanism. Equation 2.12 explains the Jacobian matrix of spherical joint with the mentioned assumption.

$$J_{ss} = \begin{bmatrix} 0 & c_1 & -c_2 s_1 \\ 0 & s_1 & c_1 c_2 \\ 1 & 0 & s_1 \end{bmatrix} \tag{2.12}$$

The singularity calculation regarding J_s matrix is shown in equation 2.13.

$$det\,(J_{ss}) = c_2 {c_1}^2 + c_2 {s_1}^2 = 0 \tag{2.13}$$

The result is:

$$\theta_2 = k\pi + \frac{\pi}{2} \qquad K = -1,0$$

This result implies that the singularity for this mechanism happens when Link L1 which is attached to spherical joint is in ±90° from the sagittal view. Also, it is important to consider that this issue will never happen completely since the workspace of this robot mechanisms is ±89.5°. In addition, since the first concepts of designing this mechanism were to let the arm to meet its fully range of motions, it has been designed for healthy human, and thus the human arm comfort ability was the first concern of the design and it was decided to add control considerations later

2.3.2 Serial links in the shoulder and revolute joint in the elbow

$$J_4 = \begin{bmatrix} J4_{11} & J4_{12} & J4_{13} & J4_{41} \\ J4_{21} & J4_{22} & J4_{23} & J4_{24} \\ 0 & J4_{32} & J4_{33} & -L_2 s_2 c_{34} \\ 0 & s_1 & c_1 s_2 & c_1 s_2 \\ 0 & -c_1 & s_1 s_2 & s_1 s_2 \\ 1 & 0 & -c_2 & -c_2 \end{bmatrix} \quad (2.14)$$

$$J4_{11} = L_1(c_1 c_3 + s_1 s_3 c_2) - 0.707R(s_1 + c_2 s_1) + L_2(c_1 c_3 c_4 - c_1 s_3 s_4 + c_2 c_3 s_1 s_4 + c_2 c_4 s_1 s_3)$$

$$J4_{12} = c_1 s_2(-0.707R + L_1 s_3 + L_2 s_{3,4})$$

$$J4_{13} = -L_1(s_1 s_3 + c_1 c_2 c_3) + L_2(c_1 c_2 s_3 s_4 - s_4 c_3 s_1 - c_4 s_3 s_1 - c_1 c_2 c_3 c_4)$$

$$J4_{14} = L_2(c_1 c_2 s_3 s_4 - c_4 s_3 s_1 - c_1 c_2 c_3 c_4 - s_4 c_3 s_1)$$

$$J4_{21} = 0.707R(c_1 + c_1 c_2) + L_1(s_1 c_3 - s_1 s_3 c_2) + L_2(s_1 c_3 c_4 - s_1 s_3 s_4 - c_1 c_2 c_3 s_4 - c_1 c_2 c_4 s_3)$$

$$J4_{22} = s_1 s_2(-0.707R + L_1 s_3 + L_2 s_{3,4})$$

$$J4_{23} = L_1(c_1 s_3 - s_1 c_2 c_3) + L_2(s_1 c_2 s_3 s_4 + s_4 c_3 c_1 + c_4 s_3 c_1 - s_1 c_2 c_3 c_4)$$

$$J4_{24} = L_2(s_1 c_2 s_3 s_4 + s_4 c_3 c_1 - s_1 c_2 c_3 c_4 + c_4 s_3 c_1)$$

$$J4_{32} = -c_2(-0.707R + L_1 s_3 + L_2 s_{3,4})$$

$$J4_{33} = -s_2(L_1 c_3 + L_2 c_{3,4})$$

As singularity was explained above, it is a necessary calculation in control designing and since intelligent control of the chosen device will be analyzed in the following chapters, therefore these calculations will be applied. More importantly, realizing and considering the points in 3D space that end-effector cannot be moved to will make the control design more efficient by avoiding those joint positions. In this design, the serial link has 3 successive rotational joints on the shoulder which may put the robot in a singularity situation. Equation 2.15 refers to the singularity calculations.

$$J_{sL} = \begin{bmatrix} 0 & s_1 & c_1 s_2 \\ 0 & -c_1 & s_1 s_2 \\ 1 & 0 & -c_2 \end{bmatrix} \tag{2.15}$$

$$det\,(J_{sL}) = s_2 {c_1}^2 + s_2 {s_1}^2 = 0 \tag{2.16}$$

The result for θ_2 is:

$$\theta_2 = k\pi + \frac{\pi}{2} \qquad K = -1,0$$

Although J_{sL} and J_{ss} are two different matrices due to the different setup of joints in both mechanisms, however, as it was expected the result for θ_2 is the same for both mechanisms. In serial links mechanism, this result implies that positions of robot which put the serial joints of shoulder in a line are defined as singularities space for this robot.

2.4Workspace of exoskeletons

Workspace of the robot is one of the major issues in design considerations. The workspace of a robot defines the points in space that end-effector can reach. In this particular issue, it is necessary to consider human arm range of motion (RoM) for each joint to examine the efficiency of the device for human. The design assumption for both design was to let the devices cover human RoM as much as possible and later, by applying control methods the limitations of joint and link motions will be decided. Table 2.4 compares the human range of motion with RoM of both conceptual designs. As the results shown for the spherical joint mechanism in the shoulder can achieve to more range of motions, it also should be considered that a joint with 3DoFs at the same time and 3 variables may face to fusilier more probably than the serial mechanism with 3 separated joints. Also, the internal and external rotations of a shoulder and adduction and abduction of elbow joint can be fully covered by both mechanisms. The reason is that these robots are assumed to be

attached by human limb and these movements are around the limb axis. Therefore, they can be covered fully by both robots.

Table 2.4 : RoM for human arm, SsRe and serial links and joint robot mechanisms

Movements	Human Arm	SsRe	Serial Links and Joint
Shoulder (Frontal) Adduction/Abduction	45◦/180◦	89.5◦/89.5◦	90◦/90◦
Shoulder Extension / Flexion	50◦/180◦	203.06◦/143.06◦	180◦/90
Shoulder (Horizental) Adduction/Abduction	45◦/135◦	0◦/179◦	90/135◦
Shoulder Internal/External Rotation	90◦/90◦	90◦/90◦	90◦/90◦
Elbow Extension / Flexion	0◦/145◦	0◦/135◦	0◦/135◦
Elbow Adduction/Abduction	0◦/200◦	0◦/200◦	0◦/200◦

As the results show, the spherical joint mechanism in shoulder can achieve to more ranges of motion, but it also should be considered that, a joint with 3 DoFs at the same time and 3 variables may face to fuailier most probably than the serial mechanism with 3 separated joints. In addition, the internal and external rotations of a shoulder and adduction and abduction of the elbow joint can be fully covered by both mechanisms. The reason is these robots are assumed to be attached by human limb and these movements are around the limb axis. Therefore, both robots can cover them fully.

2.5Conclusions

In this chapter by considering two different types of exoskeleton robot, the kinematic study was done. The results show the forward kinematics, invers kinematics and Jacobian for the serial links robot can be solved in easier way the than the spherical joint mechanism. Also, spherical joint shows more dexterity in RoM. However, as mentioned above from the aspects of frictions and failure in the movement, the spherical mechanism most probably can face defeat since the motions cannot be divided between joints, all three motions should be performed with one joint, and therefore three variables should be considered during the motions.

3. DYNAMICS MODEL OF 4 DoF WEARBLE ARM EXOSKELETON ROBOTS

In this chapter, the dynamics model of the robot will be discussed. Dynamic model of the robot is essential in order to add the control to the robot. As much as the robot dynamics and parameters are defined and calculated precisely, better and reliable results in controlling the robot can be achieved and better result for control as well.

3.1 Dynamics Calculations Approach

In order to calculate the dynamics model of the exoskeleton, firstly, it is important to consider the range of motion (RoM) of the device and its level of dexterity. Since the device is 4 DoFs which means it has 3-D workspace it would be complicated to solve the problem by following the classical Newton-Euler method. This renowned equations is known better in solving the motion equations of the usual planner robot[43]. Since the high RoM in shoulder can bring complexity in finding the motion equations, to find a practical and precise way for calculating the exoskeleton's dynamic model in 3D space, it is much reliable to follow the Newton-Euler approach. The dynamic model of the robot is found in equation 3.1[44].

$$D(\theta)\ddot{\theta} + C\left(\theta, \dot{\theta}\right)\dot{\theta} + g(\theta) = \tau \tag{3.1}$$

In equation 3.1, $\boldsymbol{\tau}$ is the vector of actuator torque, g is vector regarding to gravity force, $\boldsymbol{C}$ is a vector and D is inertia matrix. [43] [44] .The $\boldsymbol{C}\left(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}\right)$can be found as 3.2 equation [44].

$$C\left(\theta, \dot{\theta}\right) = \frac{1}{2}\left(\frac{\partial D_{ij}}{\partial \theta_k} + \frac{\partial D_{ik}}{\partial \theta_j} - \frac{\partial D_{kj}}{\partial \theta_i}\right)\dot{\theta}_k \tag{3.2}$$

The inertia matrix equation can be expended as below equation for 4 links serial robot [44]:

$$D(\theta) = \sum_{i=1}^{n} (m_i J_{vi}(q_i) J_{vi}(q_i)^T + J_{wi}(q_i) I_i J_{wi}(q_i)^T) \quad (3.3)$$

As explained before, the final dynamics model of the robot will be as 3.1. It means that the dynamics model contains the inertia matrix of the whole robot and here is 4× 4, since the robot is 4 DoFs. The $C(\theta, \dot{\theta})$ vector is 4× 1 which contains $\dot{q}_i{}^2$ terms including centrifugal properties of the robot dynamics and $\dot{q}_i q_j$ $(i \neq j)$ is called Coriolis term which is known better as Coriolis vector. In addition, there are terms with q_i where there is no derivative expressing the potential energy related to dynamics model of the robot. The inertia matrix and Coriolis terms will be discussed in the next chapter.

First assumption in this approach is to consider every link to be attached to a frame and that there are vectors from the origin of frame $i-1$ to the center of mass (CoM) of the link i which are $r_{i-1,ci}$, and from the origin of the frame i to the CoM of the link$i$, $r_{i,ci}$.In addition, there is vector from the origin of frame $i-1$to frame$i$, which is called$r_{i-1,i}$.Momentum equation of every link can be found as 3.4.

$$\tau_i = \sum_{i=1}^{n} \omega \times (I\omega) + I\dot{\omega} \quad (3.4)$$

3.4 can be expanded as 3.5, which is more practical in numerical calculations.

$$\tau_i = R_{i+1}^{i} \tau_{i+1} - f_i \times r_{i-1,ci} + (R_{i+1}^{i} f_i) \times r_{i,ci} + \omega_i \times (I_i \omega_i) + I_i \alpha_i \quad (3.5)$$

In 3.5, f_i is the applied force from link $i-1$ on link i (equation 3.9) and I_i is the inertia tensor of link i about a frame parallel to frame i and the origin of the frame is at the CoM of link$i$.Also, the angular velocity of frame i related to frame zero isω_i and is defined as equation in 3.6:

$$\omega_i = R_{i+1}^{i}\omega_{i-1} + b_i\,\dot{q}_i \tag{3.6}$$

The b_i is the rotation of joint i in frame$i$, which can be expressed as 3.7:

$$b_i = (R_i^0)^T R_{i-1}^0 z_0 \tag{3.7}$$

Also, α_i is the angular acceleration of frame i respecting to frame *0* and is explained in 3.8:

$$\alpha_i = (R_i^{i-1})^T \alpha_{i-1} + b_i\,\ddot{q_i} + \omega_i \times b_i\,\dot{q}_i \tag{3.8}$$

$z_0$is the base frame actuation axis. Moreover, from equation 3.5, the R_{i+1}^{i} is the rotation matrix from frame i to $i+1$ frame and $f_i$can be expressed as 3.9.

$$f_i = R_{i+1}^{i}\,f_i + m_i\alpha_{c,i} - m_i g_i \tag{3.9}$$

In 3.9, the $\alpha_{c,i}$ is the CoM acceleration and its equation comes as 3.10.

$$\alpha_{ci} = (R_i^{i-1})^T \alpha_{e\,,i-1} + \dot{\omega}_i \times\, r_{i-1\,,ci} + \omega_i \times \left(\omega_i \times\, r_{i-1\,,ci}\right) \tag{3.10}$$

$\alpha_{e\,,i-1}$is the acceleration of the end of the link in 3.11:

$$\alpha_{e,i} = R_i^{i-1}\alpha_{e\,,i-1} + \dot{\omega}_i \times\, r_{i-1\,,i} + \omega_i \times \left(\omega_i \times\, r_{i-1\,,i}\right) \tag{3.11}$$

The Euler-Newton method contains forward dynamics and backward dynamics in order to solving the n-links manipulator robot with following conditions:

Forward dynamics: It started with initial condition, which is coming as follow:

$$\omega_0 = \alpha_0 = \alpha_{c,0} = \alpha_{e,0} = 0$$

Backward dynamics: It started with final condition, which is coming as follow:

$$\tau_{i+1} = f_{i+1} = 0$$

3.2 Dynamics Model of 4 DoFs Exoskeleton in 3-D

In order to find out the dynamics model of the robot, first it is important to calculate the rotation of every joint related to its frame. The frame here is referred to as the defined frame in D-H method for calculating the forward kinematics. Therefore, $\boldsymbol{b_i}$become as 3.12:

$$\boldsymbol{b_1} = \left(\boldsymbol{R_1^0}\right)^{\boldsymbol{T}} \boldsymbol{z_0} = [\mathbf{0}\ \ \mathbf{1}\ \ \mathbf{0}]^{\boldsymbol{T}}$$

$$b_2 = (R_2^0)^T R_1^0\, z_0 = [0\ \ 1\ \ 0]^T \tag{3.12}$$

$$\boldsymbol{b_3} = \left(\boldsymbol{R_3^0}\right)^{\boldsymbol{T}} \boldsymbol{R_2^0}\, \boldsymbol{z_0} = [\mathbf{0}\ \ \mathbf{0}\ \ \mathbf{1}]^{\boldsymbol{T}}$$

$$\boldsymbol{b_4} = \left(\boldsymbol{R_4^0}\right)^{\boldsymbol{T}} \boldsymbol{R_3^0}\, \boldsymbol{z_0} = [\mathbf{0}\ \ \mathbf{0}\ \ \mathbf{1}]^{\boldsymbol{T}}$$

In 3.12 $\boldsymbol{z_0} = [\mathbf{0}\ \ \mathbf{0}\ \ \mathbf{1}]^{\boldsymbol{T}}$. Figure 3.1 is referred to as the assumed frames for D-H calculations. It can be concluded that every joint rotation referred to as its frame in initial condition occurs in XZ and XY plane.

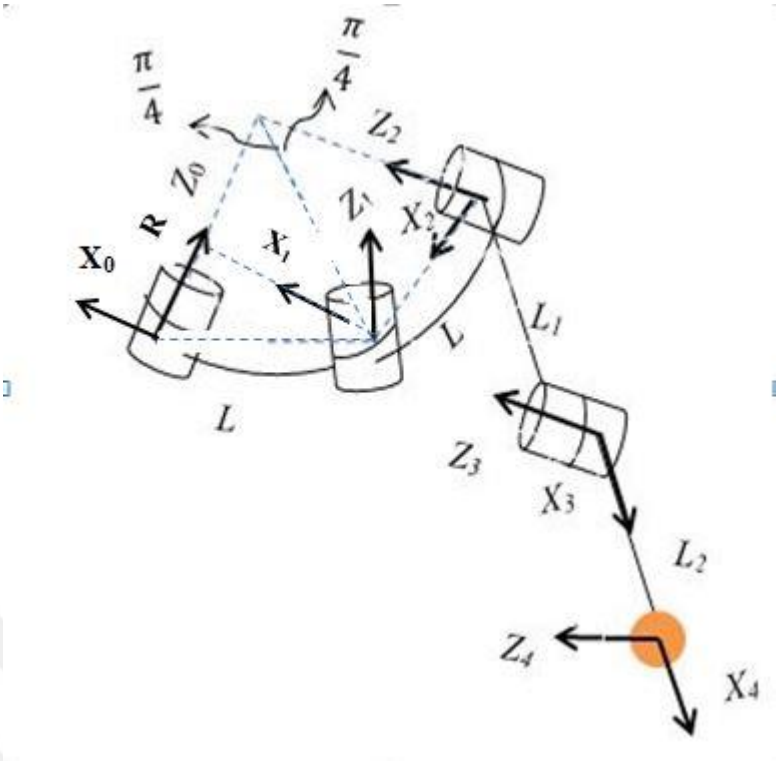

Figure 3.1: Schematics of serial links mechanism and frames related to D-H method

The next step is to find$r_{i-1,i}$ and $r_{c,i}$ which are the base frame and link frames and are playing an important role. Since this method is based on defined frame for every link, it is important to assign frames correctly. Specially for base frame, the axis should be the same as what is defined for D-H and forward kinematics calculations. The links measurements are done based on the conceptual design in SolidWorks 2017 software. Figure 3.2 explains the based frame and initial condition of the robot which is consistent with the assumed frame in D-H and forward kinematics calculations. Moreover, other vectors related to the origin of frame *i* to origin of the frame $i+1$ should be well determined. In this way, the frames should be introduced first. It is important to consider same frames as defined frame in D-H methods and forward kinematics calculations. Also, the vectors are specified related to origin of frame $+1$. Figure 3.3 depicts the frames at the origin of every link. The frames are related to $r_{i-1,i}$ vectors in SolidWorks environments and are explained in 3.13. The vectors unit is mm. Note that the $r_{i-1,ci}$, $r_{i,ci}$vectors are assigned from frame $i-1$ to the CoM of link i. Equation 3.14 is about $r_{i-1,ci}$, $r_{i,ci}$vectors.

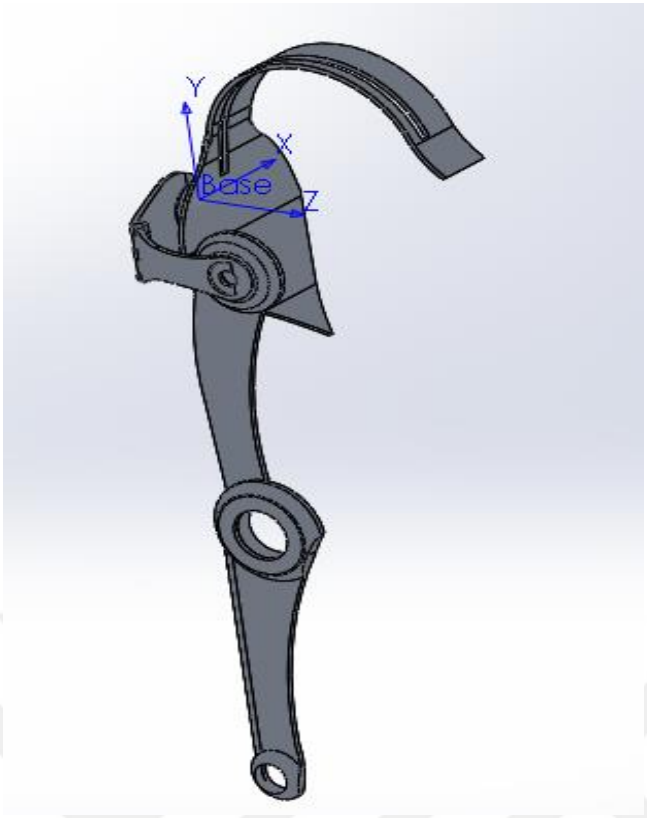

Figure 3.2: Defined base frame and robot initial condition in SolidWorks environment

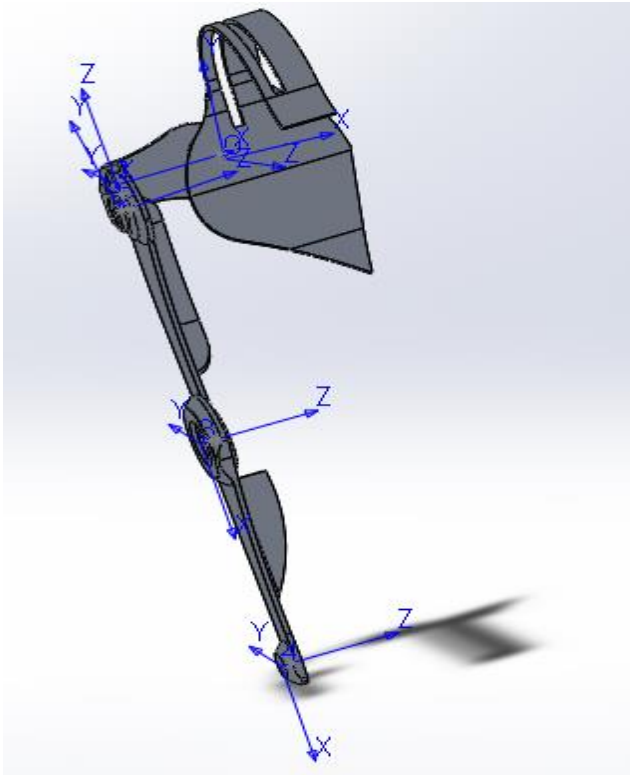

Figure 3.3: Defined frame related to origin of every link in SolidWorks environment

$$r_{0,1} = [-107.56 \quad 0 \quad 0]^T \tag{3.13}$$

$$r_{1,2} = [27.89 \;\; 98.18 \;\; 5.02]^T$$

$$r_{2,3} = [13.29 \;\; 258.44 \;\; 37.32]^T$$

$$r_{3,4} = [253.0 \;\; 0 \;\; 0]^T$$

$$r_{0,c1} = [-49.4 \;\; 1.24 \;\; -7.78]^T$$

$$r_{1,c1} = r_{0,c1} - r_{0,1} \tag{3.14}$$

$$r_{1,c2} = [-19.69 \;\; -46.42 \;\; -5.05]^T$$

$$r_{2,c2} = r_{1,c2} - r_{1,2}$$

$$r_{2,c3} = [14.5 \;\; -99.45 \;\; -3.21]^T$$

$$r_{3,c3} = r_{2,c3} - r_{2,3}$$

$$r_{3,c4} = [79.36 \quad 8.97 \quad 13.17]^T$$

$$r_{4,c4} = r_{3,c4} - r_{3,4}$$

Next step is to calculate moment of inertia for every link. Inertia matrix plays an essential role in kinetics energy of the system. Therefore, finding the elements of this matrix precisely is necessary. The first assumption in order to start the calculation is to consider the frames parallel to the frame i whose origin is at the CoM of link i. The inertia matrix related to every link can be found by mass properties section in SolidWorks environment if the initial condition of the joint, link and frames are consisting with the defined frames and gesture in D-H and the forward kinematics calculation.

By calculating the essential parameters, the process of finding the dynamic plant of the robot is started. The first important assumption is to put zero for all the variables at the initial conditions, $\boldsymbol{\omega_0 = \alpha_0 = \alpha_{c,0} = \alpha_{e,0} = 0}$.

3.2.1 Froward recursion

In forward recursion, the angular velocity and acceleration of every link are calculated. The axis of rotation for each joint , b_i in equation 3.12 should be used in order to start angular velocity and acceleration computations 3.12 . As it was mentioned, the initial condition is to put zero for all $\boldsymbol{\omega_0, \alpha_0, \alpha_{c,0}, \alpha_{e,0}}$.Angular acceleration and velocity equations for link 1 to link 4 are shown in 3.16.

Link 1:

$$\omega_1 = b_1 \dot{q}_1$$

$$\alpha_1 = b_1 \ddot{q_1} + \omega_1 \times b_1 \dot{q}_1$$

Link 2:

$$\omega_2 = R_2^1 \omega_1 + b_2 \dot{q}_2$$

$$\alpha_2 = (R_2^1)^T \alpha_1 + b_2 \ddot{q_2} + \omega_2 \times b_2 \dot{q}_2$$

Link 3:

$$\omega_3 = R_3^2 \omega_2 + b_3 \dot{q}_3 \qquad (3.16)$$

$$\alpha_3 = (R_3^2)^T \alpha_2 + b_3 \ddot{q_3} + \omega_3 \times b_3 \dot{q}_3$$

Link 4:

$$\omega_4 = R_4^3 \omega_3 + b_4 \dot{q}_4$$

$$\alpha_4 = (R_4^3)^T \alpha_4 + b_4 \ddot{q_4} + \omega_4 \times b_4 \dot{q}_4$$

Acceleration for end of link and acceleration for center of mass can be found as 3.17 equation:

Link 1:

$$\boldsymbol{\alpha_{c1} = \alpha_1 \times r_{0,c1} + \omega_1 \times (\omega_1 \times r_{0,c1})}$$

$$\alpha_{e,1} = \alpha_1 \times r_{0,1} + \omega_1 \times (\omega_1 \times r_{0,1}) \qquad (\mathbf{3.17})$$

Link 2:

$$\alpha_{c2} = (R_2^1)^T \alpha_{e,1} + \alpha_2 \times r_{1,c2} + \omega_2 \times (\omega_2 \times r_{1,c2})$$

$$\alpha_{e,2} = R_2^1 \alpha_{e,1} + \alpha_2 \times r_{1,2} + \omega_2 \times (\omega_2 \times r_{1,2})$$

Link 3:

$$\boldsymbol{\alpha_{c3} = (R_3^2)^T \alpha_{e,2} + \alpha_3 \times r_{2,c3} + \omega_3 \times (\omega_3 \times r_{2,c3})}$$

$$\alpha_{e,3} = R_3^2 \alpha_{e,2} + \alpha_3 \times r_{2,3} + \omega_3 \times (\omega_3 \times r_{2,3})$$

Link 4:

$$\boldsymbol{\alpha_{c4} = (R_4^3)^T \alpha_{e,3} + \alpha_4 \times r_{3,c4} + \omega_4 \times (\omega_4 \times r_{3,c4})}$$

Note that no need for finding the $\alpha_{e,4}$ since there is no link number 5.

3.2.2Backward recursion

By finding the angular velocity and acceleration and specially CoM accelerations in the forward recursion, the forces and torques can be found by backward recursion method. It means the calculation should start from the last link to the first link. In this

method, firstly, it is important to calculate the gravity vector for every link. Since in rotations the orientation of links are changing repeatedly. Equation 3.18 refers to gravity torque and force vector for every link.

Link 4:

$$g_4 = (R_4^0)^T g_0$$

$$f_4 = m_4 \alpha_{c,4} - m_4 g_4$$

$$\tau_4 = -f_4 \times r_{3,c4} + \omega_4 \times (I_4 \omega_4) + I_4 \alpha_4$$

Link 3:

$$g_3 = (R_3^0)^T g_0$$

$$f_3 = R_4^3 f_4 + m_3 \alpha_{c,3} - m_3 g_3 \quad (3.18)$$

$$\tau_3 = R_4^3 \tau_4 - f_3 \times r_{2,c3} + (R_4^3 f_4) \times r_{3,c3} + \omega_3 \times (I_3 \omega_3) + I_3 \alpha_3$$

Link 2:

$$g_2 = (R_2^0)^T g_0$$

$$f_2 = R_3^2 f_3 + m_2 \alpha_{c,2} - m_2 g_2$$

$$\tau_2 = R_3^2 \tau_3 - f_2 \times r_{1,c2} + (R_3^2 f_3) \times r_{2,c2} + \omega_2 \times (I_2 \omega_2) + I_2 \alpha_2$$

Link 1:

$$g_1 = (R_1^0)^T g_0$$

$$f_1 = R_2^1 f_2 + m_1 \alpha_{c,1} - m_1 g_1$$

$$\tau_1 = R_2^1 \tau_2 - f_1 \times r_{0,c1} + (R_2^1 f_2) \times r_{1,c1} + \omega_1 \times (I_1 \omega_1) + I_1 \alpha_1$$

There are many benefits due to using Newton-Euler method for calculating required torques and forces. In this method the whole problem can be divided in to smaller

pieces. In fact the problem is divided into the number of the links. Therefore, every single link can be studied as 1 DoF link with the torque and force which are coming from the link itself and are bringing the effect of force and torque of next link in to the calculation by using the rotation matrix in backward recursion. Using Newton-Euler method is very helpful with high degree of freedom robot and the robot with special shape links. The reason is that in this kind of robots the inertia matrices can be acquired with difficulty. However, this method is much precise in calculation that there would be a small chance of mistakes in calculation and specially in g derivations.

Also, it is important to know that since in backward recursion the process of calculation starts from the last link and continues to the first link, the torques and forces of the next links are added in to the current link force and torques calculation. Therefore, the vector of torque and force regarding the first link result in a huge vector. In appendix the details about the torque and force vectors can be found.

Moreover, Newton-Euler is the useful and optimized approach for deriving required forces and torques. However, in torques control it is essential to achieve the inertia matrix , Coriolis vector and potential energy vector for the whole robot. In next chapter the torque control for the whole device will be discussed. Next step is to find the inertia matrix, C-vector and gravitational vector.

3.3 Invers Dynamics of Serial link Exoskeleton

As discussed before, the Newton-Euler method is helpful with finding the torques. However, exploring other terms such as inertia matrix, Coriolis and centrifugal term and gravitational properties of the robot dynamics require a different approach. Moreover, the achieved torque is in the form of huge mathematical expression, which made the attempt for finding the mentioned terms harder. Fortunately MATLAB *subs()*function can ease the work. In order to achieve the inertia matrix by considering each explored torque and putting zero for variables such as $\dot{q}_\iota$ **,** $\ddot{q}_\iota$, q_i , g except $\ddot{q}_\iota$ with the same index as *j*from inertia matrix in 3.19

$$\boldsymbol{M_{ij}(q)} = \begin{bmatrix} \boldsymbol{M11} & \boldsymbol{M12M13} & \boldsymbol{M14} \\ \boldsymbol{M21} & \boldsymbol{M22M23} & \boldsymbol{M24} \\ \boldsymbol{M31} & \boldsymbol{M32M33} & \boldsymbol{M34} \\ \boldsymbol{M41} & \boldsymbol{M42M43} & \boldsymbol{M44} \end{bmatrix} \quad (3.19)$$

Also for discovering the gravitational properties of the robot in MTLAB, the *sub()* can be used by putting zero to all $\dot{q}_i$, $\ddot{q}_i$ in for each link achievedtorque. As it was discussed before, the gravitational vector which is discussed in 3.20 b is 4× 1.

$$G(q) = \begin{bmatrix} G1 \\ G2 \\ G3 \\ G4 \end{bmatrix} \quad (3.20)$$

Finally by subtracting inertia matrix and gravitational vector from torque vector, the centrifugal and Coriolis vector C is detectable. Due to the length of this vector, as will be specified in the appendix, 3.21 explains the C vector equation. As well as G, C vector is 4×1.

$$\boldsymbol{C(q,\dot{q})\dot{q} = \tau - M(q)\ddot{q} - G(q)} \quad (3.21)$$

$$\boldsymbol{C(q,\dot{q})\dot{q}} = \begin{bmatrix} \boldsymbol{C1} \\ \boldsymbol{C2} \\ \boldsymbol{C3} \\ \boldsymbol{C4} \end{bmatrix}$$

One positive aspect of using Newton-Euler method is easing the calculations related to the C vector since in Euler-Lagrange the process of calculations of centrifugal and Coriolis matrix is much more difficult.

After specifying each element, the dynamics block diagram of the robot can be built in MATLAB / Simulink environment. In the next chapter, the dynamics properties of

the robot and the approach for achieving the proper torque control will be discussed. But first, it is important to do analysis for the achieved dynamic model of the robot.

The $M(q)$, $C(q,\dot{q}), and\ G(q)$ can be seen in appendix.

3.4 Simulation of Robot Dynamical Model

By considering the robot model and using MATLAB Simscape Multibody, the designed robot can be imported from SolidWorks to MATLAB Simulink environment easily. Simscape Multibody has the good capacity for analyzing the model from many aspects such as force/torque, motion analyzing, control design etc. Figure 3.4 shows the SolidWorks model that has been plugged in to MATLAB Simulink environment. As it is shown, the model has 4 revolute joints which make it 4 DoFs and the properties such as mass, inertia and center of mass can be well organized with the block called "Solid". Figure 3.5 is depicting the arm first link "solid" block. It is essential to set the parameters correctly since the model and calculated motions with Simulink are responding well. The first analysis is to find out how each joint and link can respond to the assigned torque.

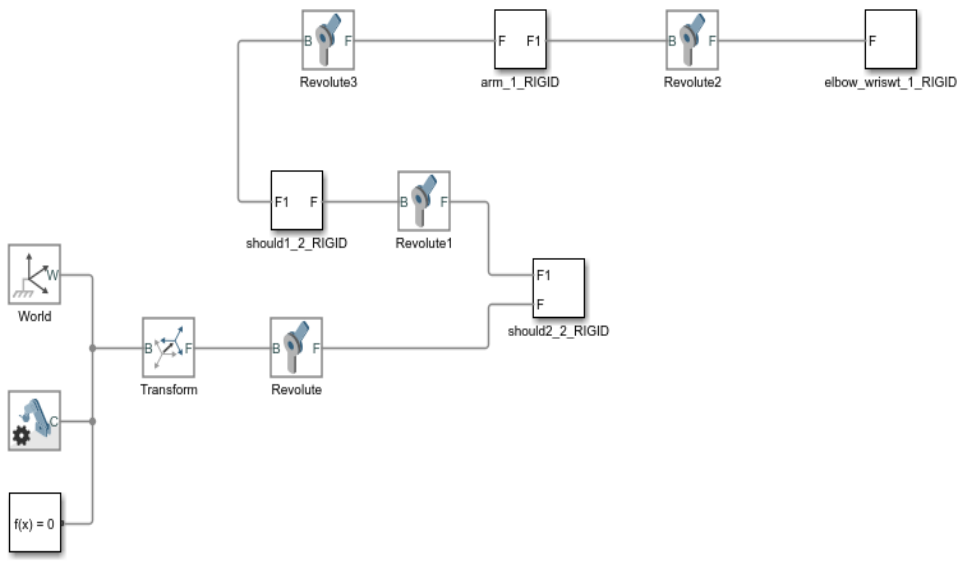

Figure 3.4: Imported robot model from Solid Works to MATLAB Simulink environment

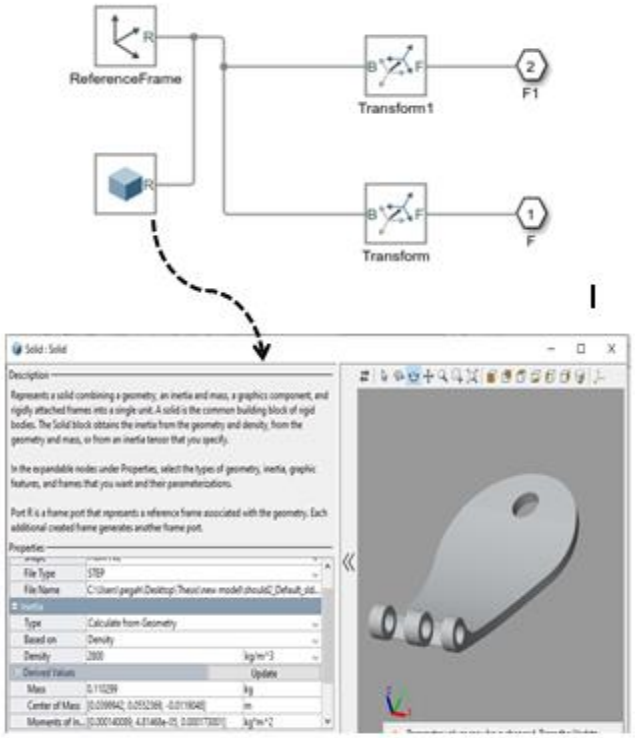

Figure 3.5: The "solid" block diagram and its characteristics.

The first reason is to find out the best fitted motion and frequency for each joint; since each joint will be connected to the DC motor, therefore there will be four DC motors with different torque frequencies which can make the system oscillate. This study illustrates the accuracy of the achieved dynamics model and recognition on whether the model can follow the resulting torque from the joint motion or not.

Another study is to figure out the range of motion (RoM) for four essential arm movements that every renowned movement (such as abduction/adduction, flection/extension .etc.) can be translated to one joint movement and more importantly to find each joint workspace. In this study the joint under study is active and receives the motion signal but all other joints should be removed from the system in order to let other links stand steal and act as the rigid body. Also, in order to achieve the best fitted result, controlling and choosing the right mass and inertia parameters are very important. Table 3.1 shows the mass and inertia parameters. For better functionality, the first two links are assumed to be made fromAluminum Alloy and the last two links are made of PVC 0.0007. Figure 3.6 is depicting the links position in the robot.

Table 3.1 : Mass and material properties for every link in dynamic simulation

Link number	Density (Kg/m^3)	Mass (Kg)	Elastic Module (N/m^2)	Shear Module (N/m^2)
Link 1	2800	0.110	7.3×10^{10}	2.8×10^{10}
Link 2	2800	0.064	7.3×10^{10}	2.8×10^{10}
Link 3	1290	0.164	6×10^{6}	2×10^{6}
Link 4	1290	0.174	6×10^{6}	2×10^{6}

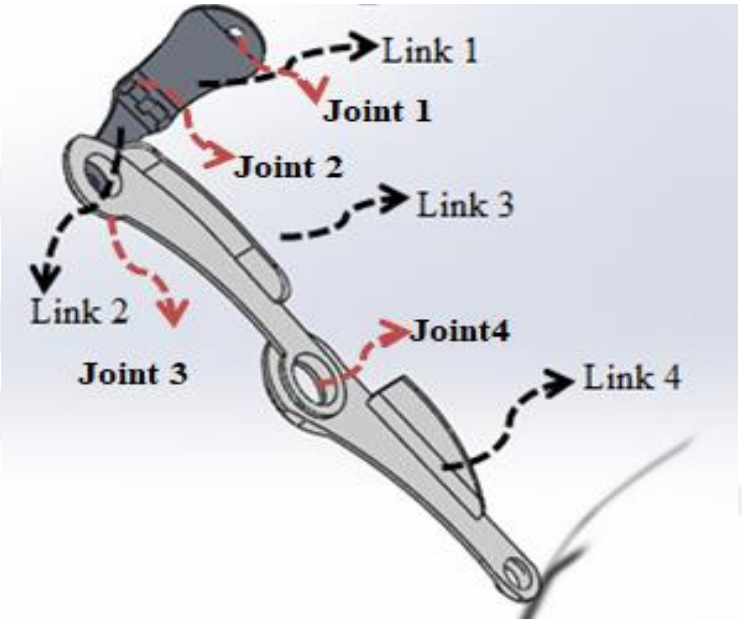

Figure 3.6: The links position in the robot

Figure 3.7, is depicting the whole system in Simscape Multibody environment.

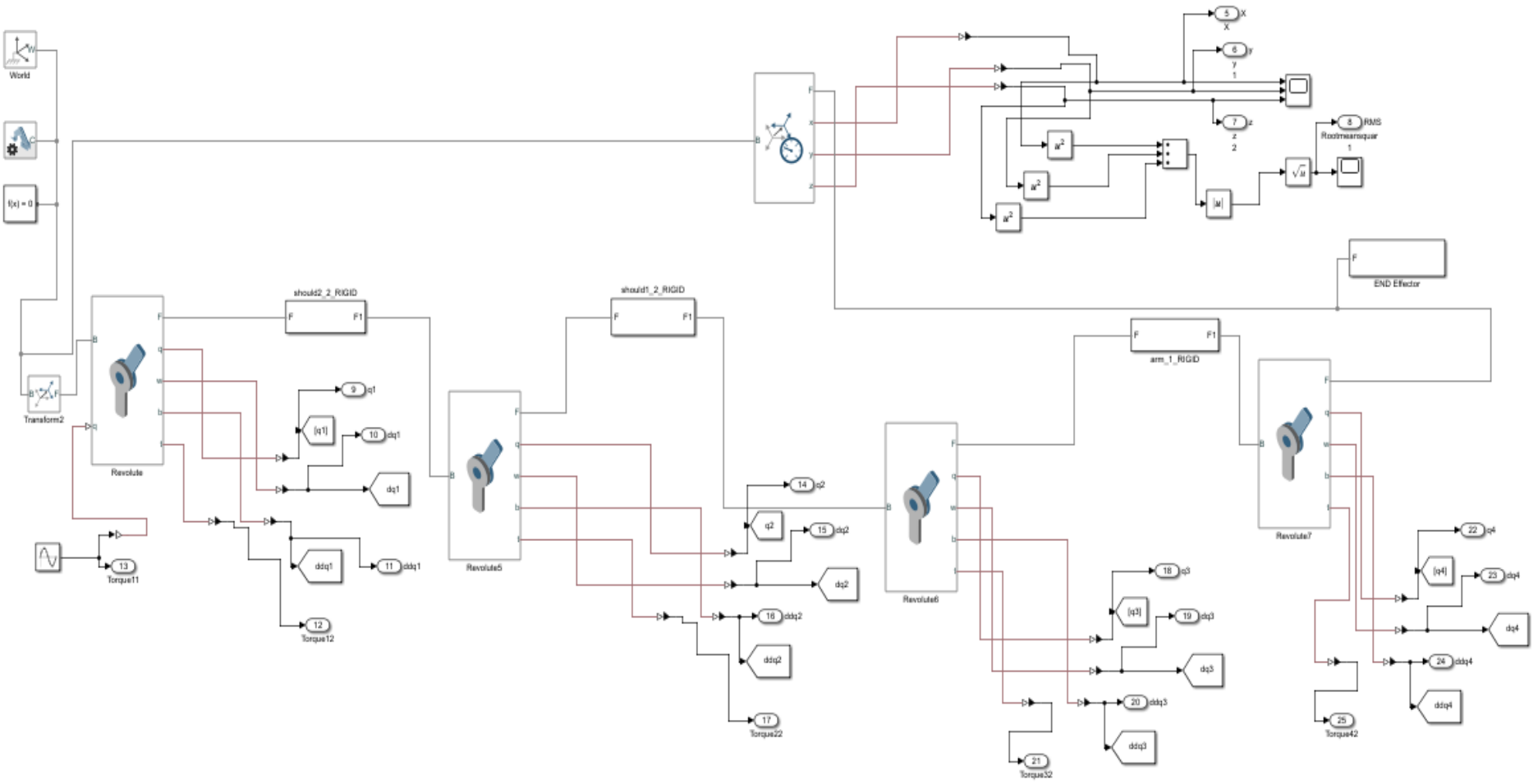

Figure 3.7: The Whole robot in SimscapeMultibody

3.4.1 First study: torque and oscillation frequency simulation

In this simulation, by applying torque to every joint and studying on every joint reaction, the best frequency that makes the dynamic model give the best response and non-oscillatory frequency for the rest of the system is found. Therefore, this study begins by starting from the first joint.

3.4.1.1 1st joint

Figure 3.8 shows the first joint in the Simulink in more details. As it can be seen from figure 3.8 the input to the system is angular displacement and the outputs from the first joint which are the motion variables such as calculated torques, angular displacement, angular velocity and angular acceleration become the inputs to the dynamicsmodel of the system and lead to torque calculations. Figures 3.9 to 3.12 show the computed torque by dynamic model and torque out of the applied angular motion in joint one with different frequencies and amplitude within 10 seconds.

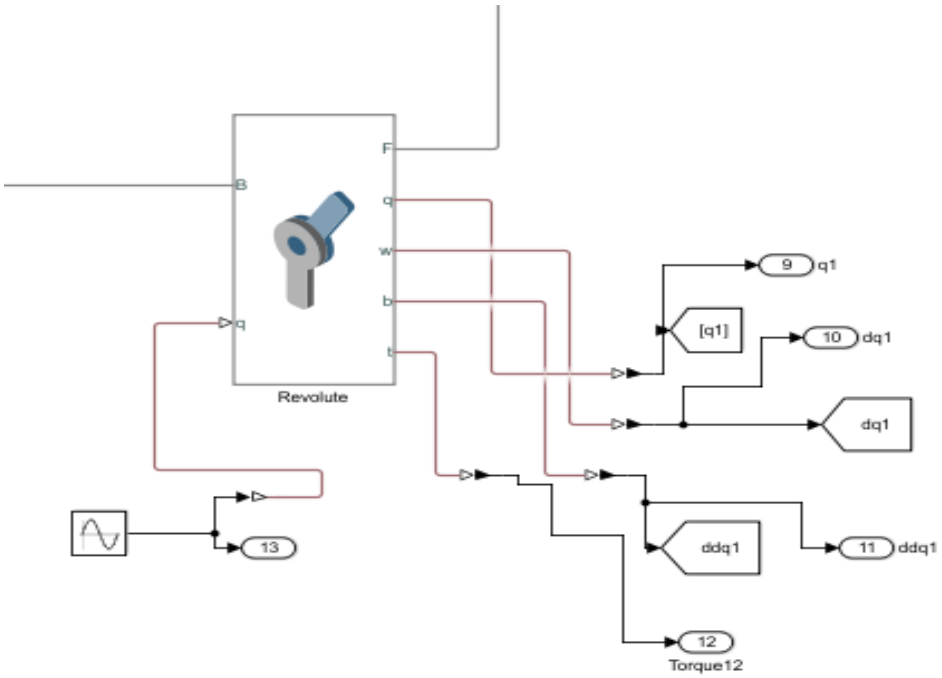

Figure 3.8: Details of first joint.

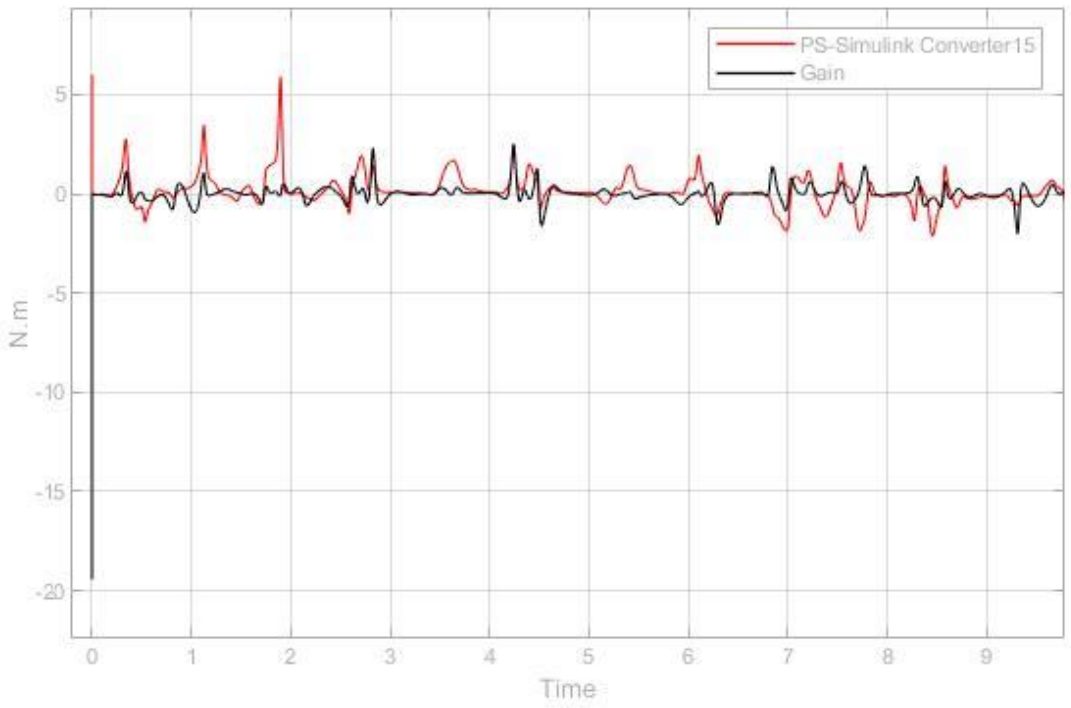

Figure 3.9: The computed torque by dynamic model and torque out of applied angular motion in joint one in 0.5 (rad/sec) frequency and amplitude 1.

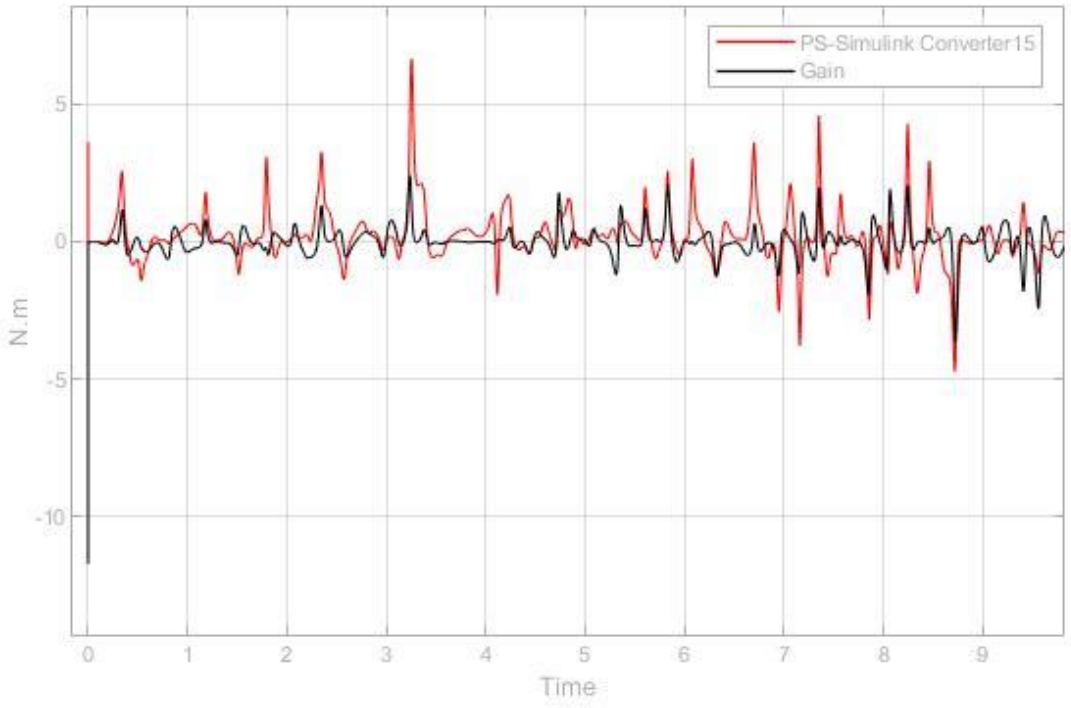

Figure 3.10: The computed torque by dynamic model and torque out of applied angular motion in joint one in 0.4 (rad/sec) frequency and amplitude 0.75.

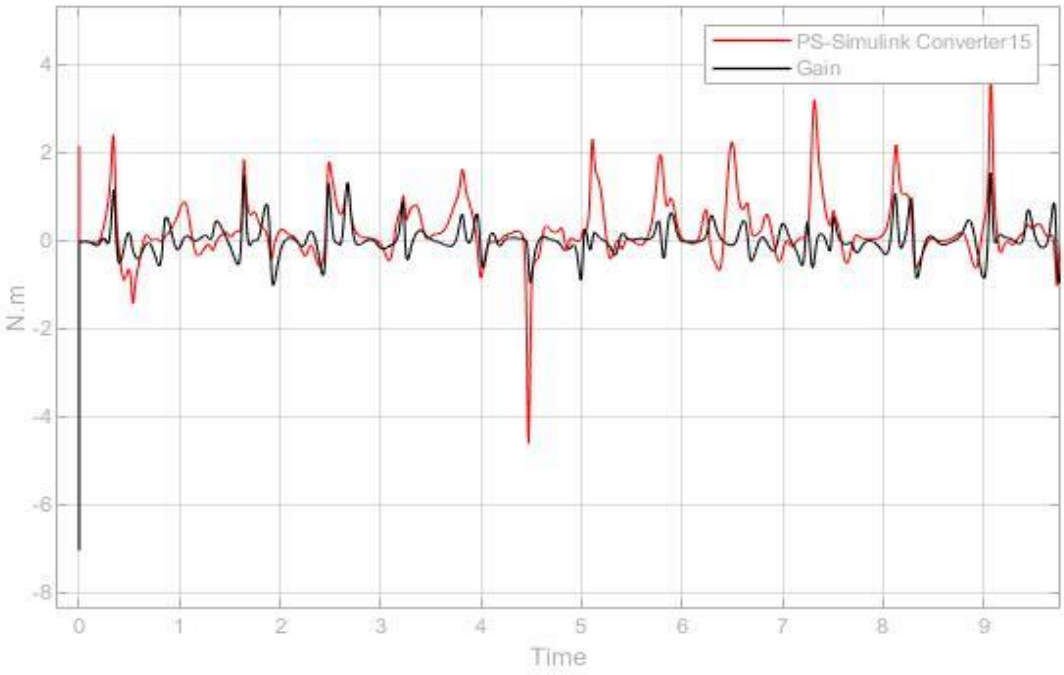

Figure 3.11: The computed torque by dynamic model and torque out of applied angular motion in joint one in 0.3 (rad/sec) frequency and amplitude 0.6.

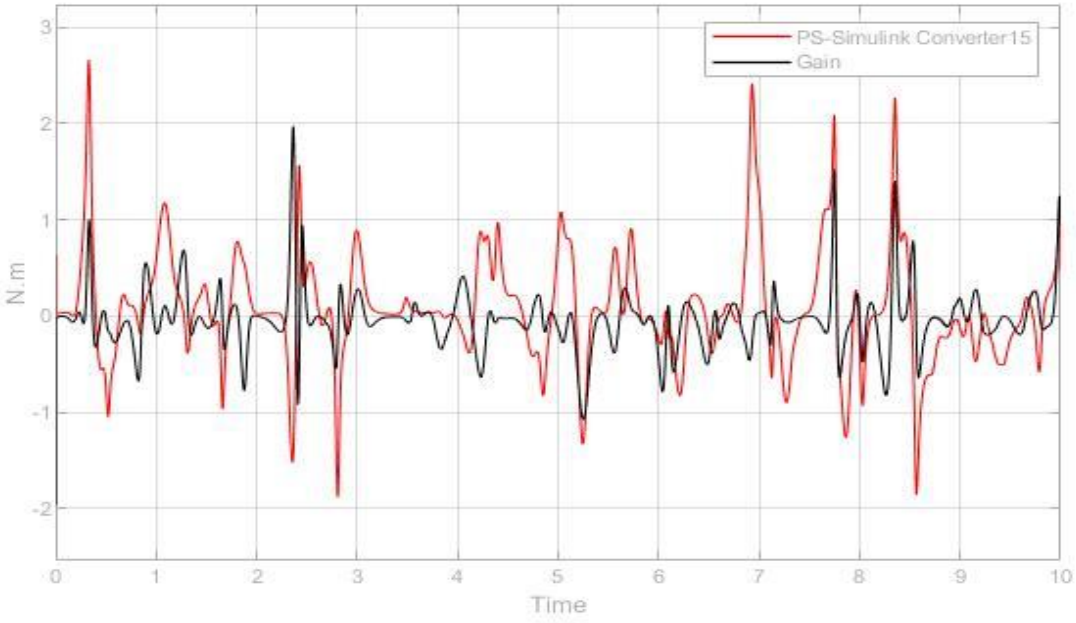

Figure 3.12: The computed torque by dynamic model and torque out of applied angular motion in joint one in 0.25 (rad/sec) frequency, amplitude 0.4 and pi/3 phase.

As it appear, the best motor signal for angular motion of the first joint can be seen in figure 3.12 which$q_{m1} = 0.4\,sin(0.25t + \frac{\pi}{3})$. Other signals make the system have a great range of overshoot, which cannot be followed by dynamic model in 0 sec,and torque from the joint cannot follow the torque from the dynamic model very well. Also, there are small ranges of disturbance which is due to other joints that are not connected to a signal and are acting accidentally in their range of motion boundaries and this was discussed in chapter 2. As was mentioned before, this study is just based on finding the best signal to prevent the system from oscillating and in fact the first joint is playing an important role since in the future for applied control it must to compensate the disturbances from other links gravity, Coriolis and centrifugal force which are coming from their motion. In the last section of this chapter more details about angular motion that have been applied to all joints is discussed.

3.4.1.2 2nd joint

As the first study for joint one has been discussed in details, for other joints the, result will be studied. This joint also plays an important role since it lets the device to move around *y-axis*. Due to the dimensional properties of the joint, it is better not to apply the high range of angular motion since it can be resulted in high stress and break the robot. There results can be found within figures 3.13 to 3.16.

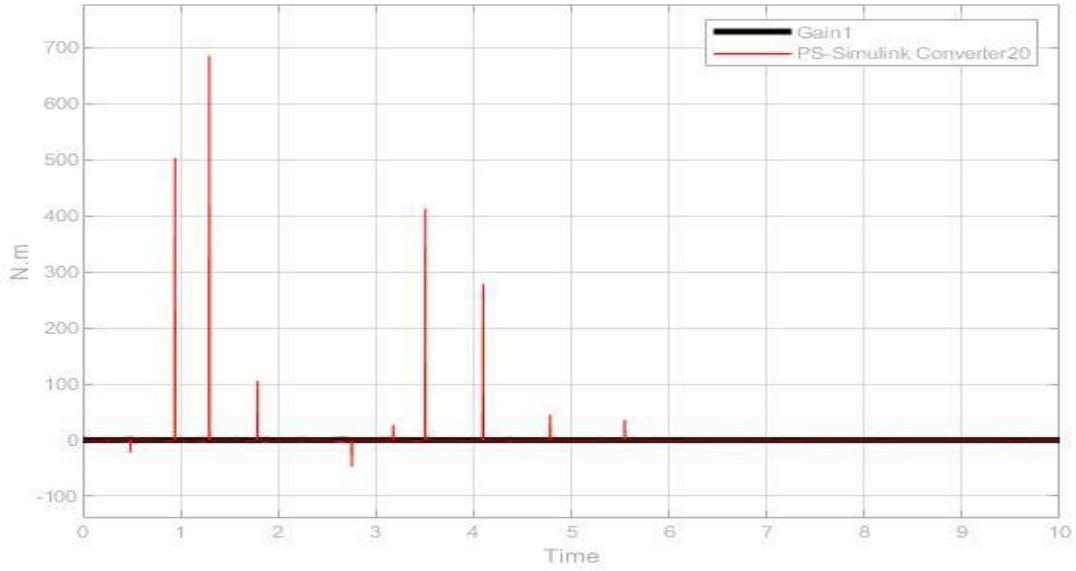

Figure 3.13: The computed torque by dynamic model and torque out of applied angular motion in joint two in 0.2 (rad/sec) frequency, amplitude0.5 .

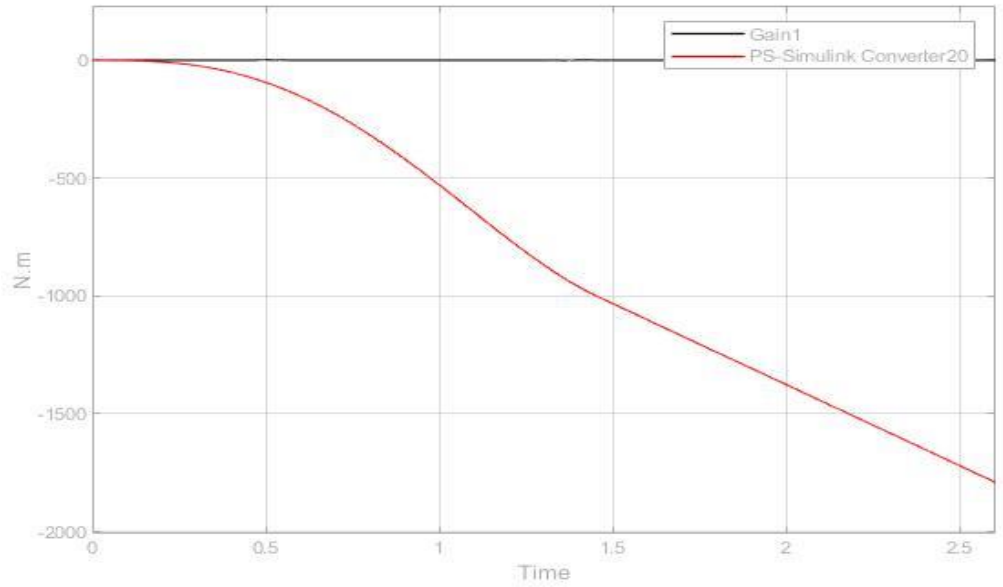

Figure 3.14: The computed torque by dynamic model and torque out of applied angular motion in joint two in 0.012 (rad/sec) frequency, amplitude 0.04 .

As it can be seen in both simulations, joint 2 behave oscillatory specially in figure 3.10 which is completely oscillating.

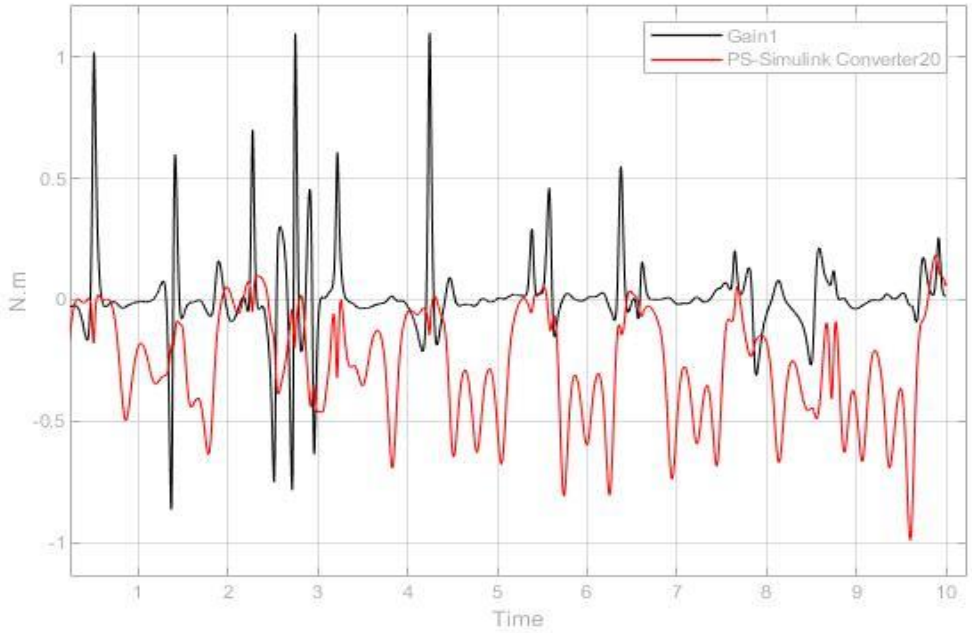

Figure 3.15: The computed torque by dynamic model and torque out of applied angular motion in joint two in 0.012 (rad/sec) frequency, amplitude 0.1 .

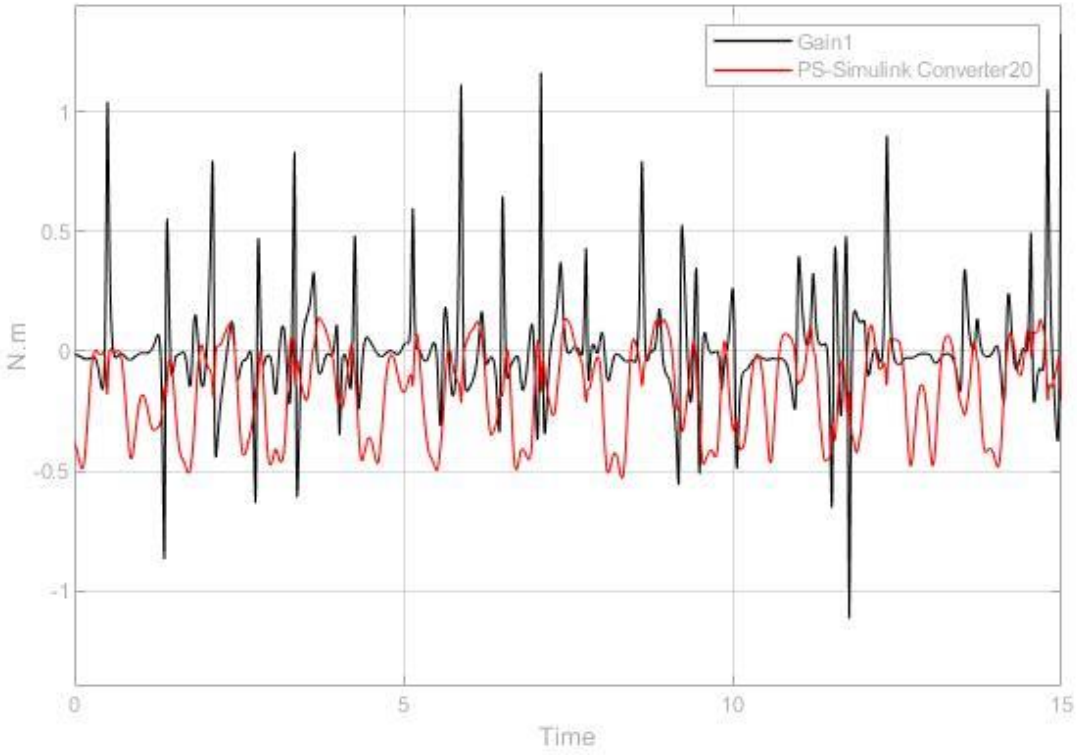

Figure 3.16: The computed torque by dynamic model and torque out of applied angular motion in joint two in 0.05 (rad/sec) frequency, amplitude 0.05.

The dynamic model can partially follow the torque from the motion signal $q_{m2} = 0.05\ sin(0.05t)$ in figure 3.16 which means the dynamic model is working correctly however the amount of overshoots are higher than expected. One reason is that the joints boundaries can add extra disturbances. That is because when the joint reaches to its boundary for moving, in order to start the return motion, some extra disturbances are added to the system, especially joint 2 has a smaller range of motion than other joints. It should also be considered that the joint two has the special situation in the robot and special range of motion as well and can be affected by motion of the other joints from its left and right. Therefore, in the end of this chapter after implementing all joints with motion signals the decision about the applied motion to this joint will be taken.

3.4.1.3 3rd joint

Joint three also has an special circumstance, it is the joint that should be acting well in, as the arm and shoulder are connected within this joint. Therefore, its robustness is very important. Figures 3.17, 3.18, 3.19 and 3.20 are the results of joint three dynamic models.

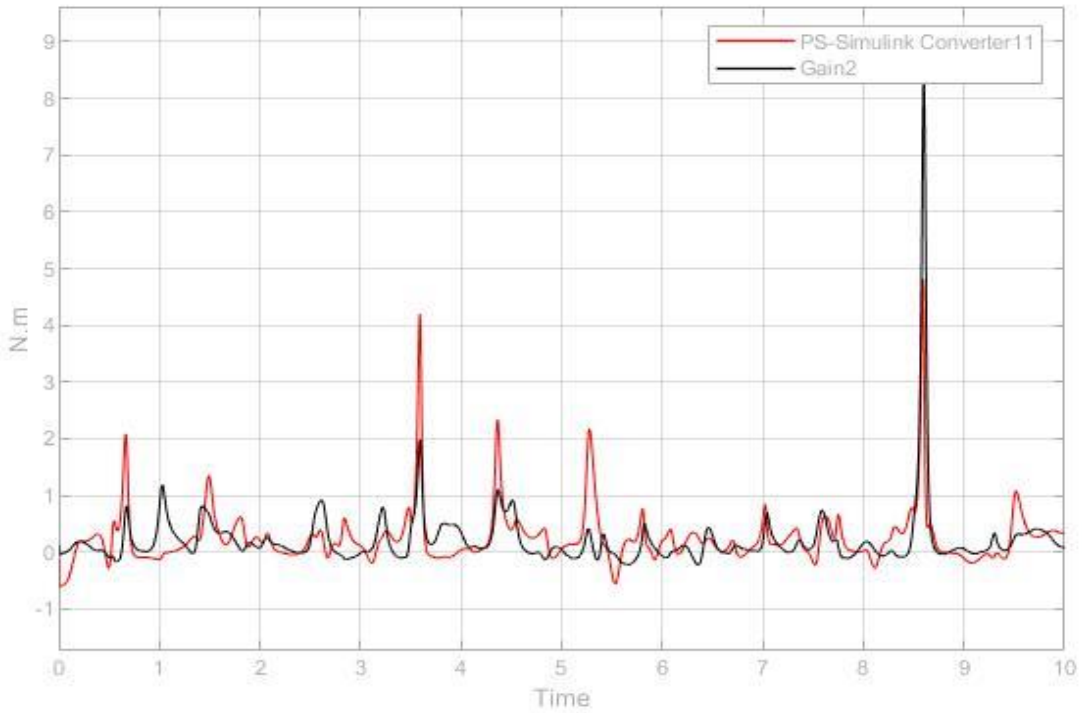

Figure 3.17: The computed torque by dynamic model and torque out of applied angular motion in joint three in 0.09 (rad/sec) frequency, amplitude 0.1.

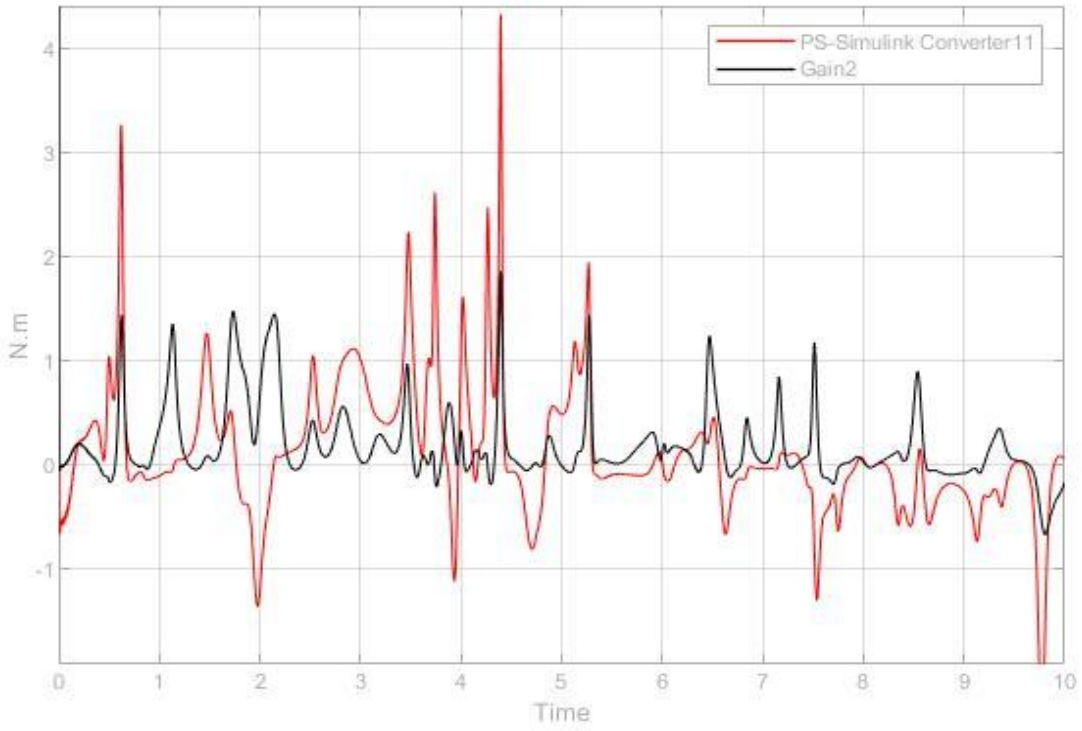

Figure 3.18: .The computed torque by dynamic model and torque out of applied angular motion in joint three in 0.5 (rad/sec) frequency, amplitude 0.8.

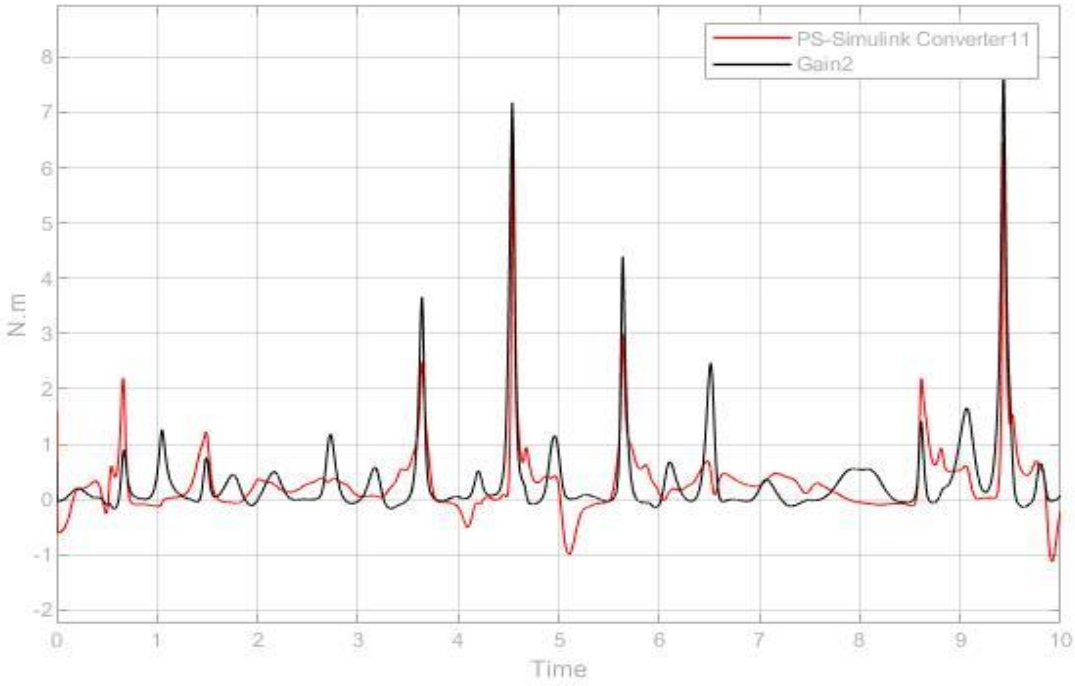

Figure 3.19: The computed torque by dynamic model and torque out of applied angular motion in joint three in 0.1 (rad/sec) frequency, amplitude 0.4.

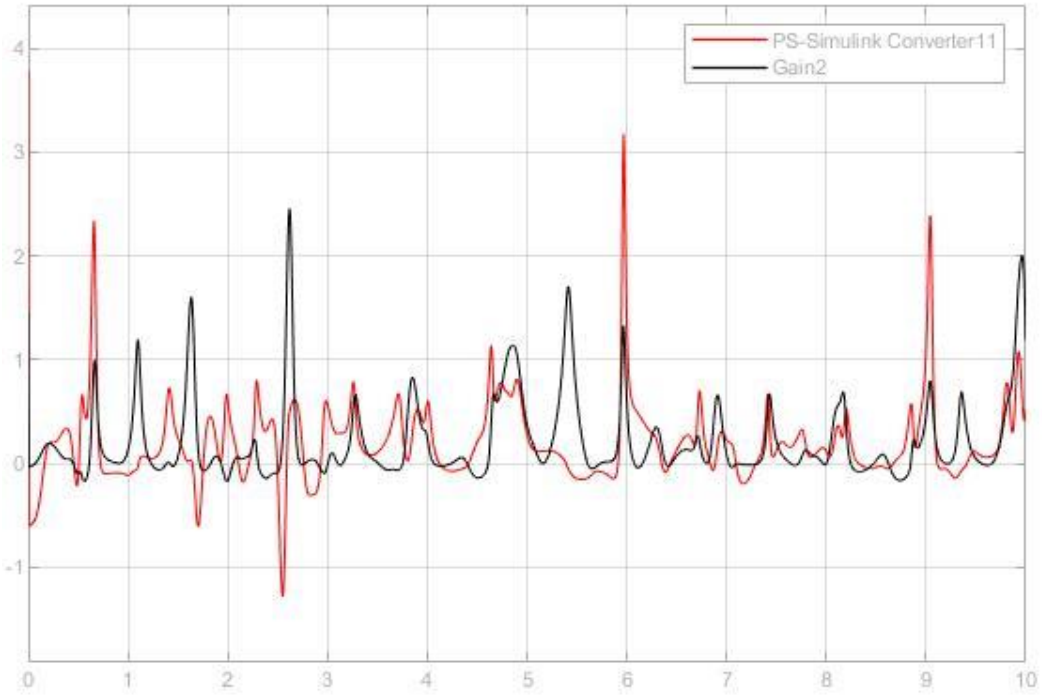

Figure 3.20: The computed torque by dynamic model and torque out of applied angular motion in joint three in 0.4 (rad/sec) frequency, amplitude 0.2.

Figure 3.20 shows the joint with $q_{m3} = 0.2\ sin(0.4t)$ signal and shows the best response and that the model can follow the torque properly.

3.4.1.4 4th joint

The last joint belongs to the elbow movement. This movement is vital not only because of the elbow but also because if the motor could not be implemented for the second joint, this joint with the help of first joint will help the user reach to the desired position. As explained before, the second joint is assisting the user to cover the frontal arm movement. Figures 3.21 to 3.22 show the results of dynamic model alignment with joint torque.

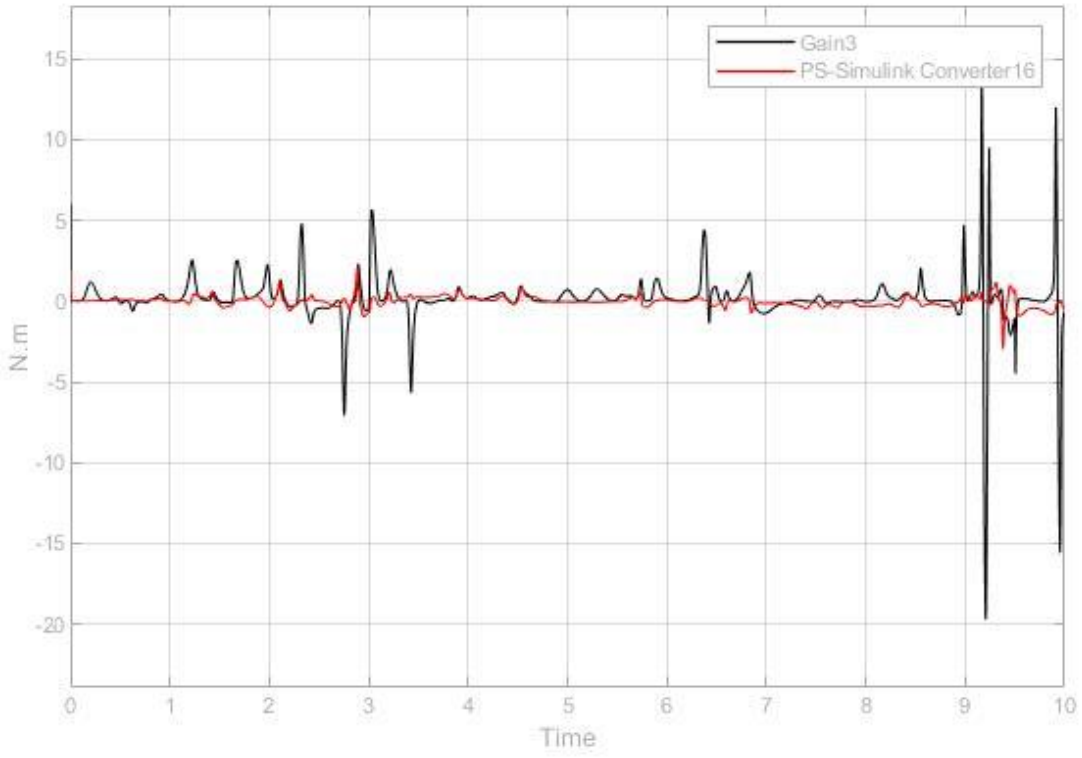

Figure 3.21: The computed torque by dynamic model and torque out of applied angular motion in joint four in 0.5 (rad/sec) frequency, amplitude 1.

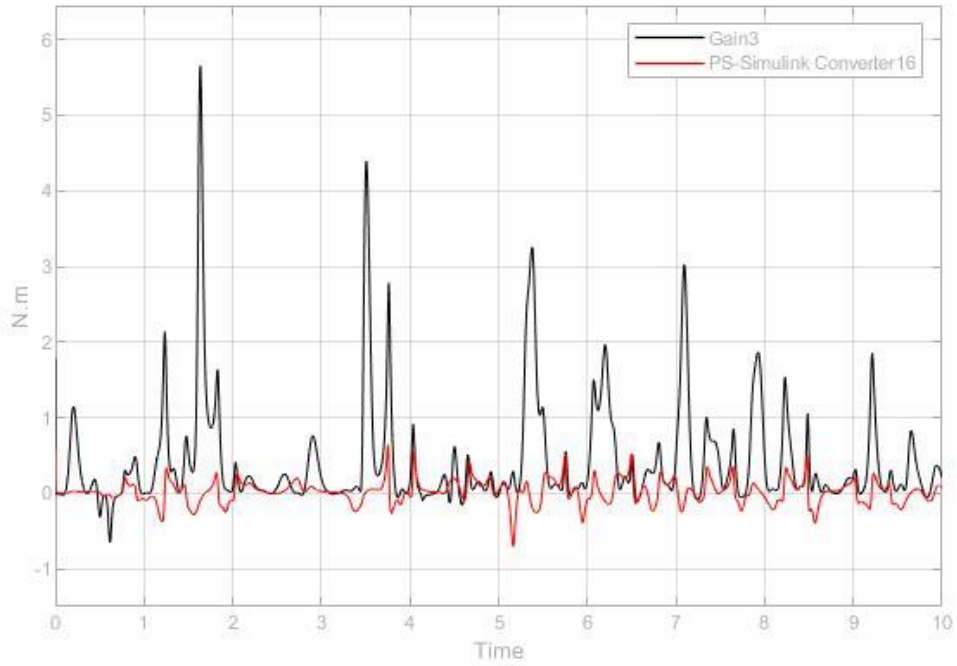

Figure 3.22: The computed torque by dynamic model and torque out of applied angular motion in joint four in 0.3 (rad/sec) frequency, amplitude 0.5.

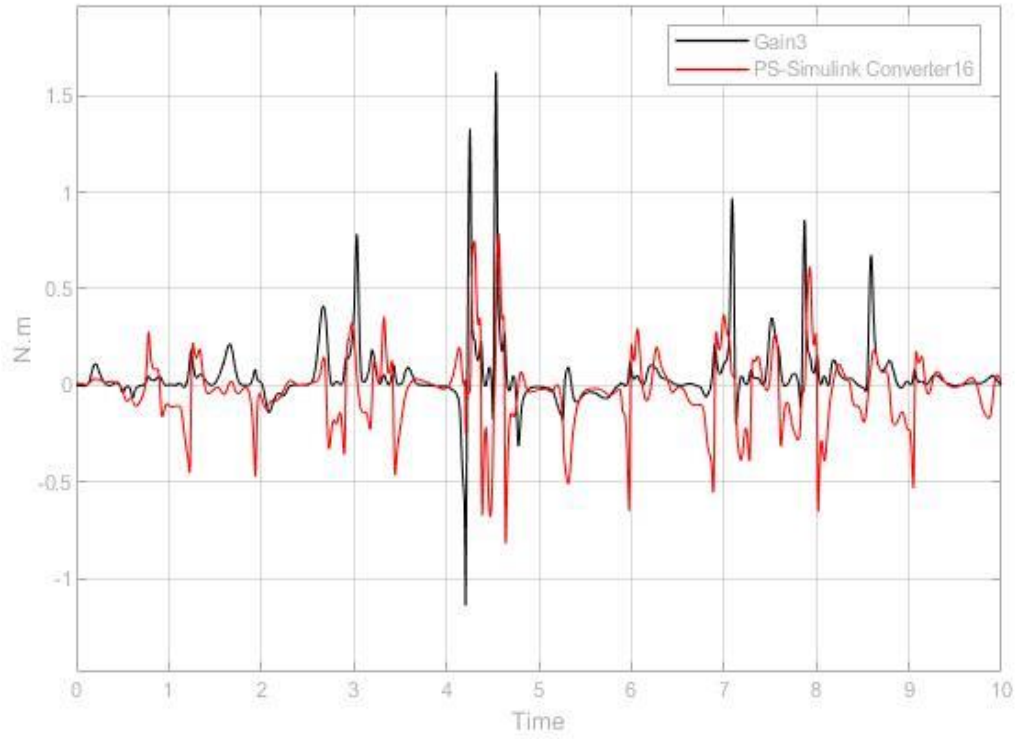

Figure 3.23: The computed torque by dynamic model and torque out of applied angular motion in joint four in 0.02 (rad/sec) frequency, amplitude 0.05.

Figure 3.22 with $q_{m4} = 0.02\ sin(0.05t)$ is the best choice for joint 4th. After this study by implementing each joint with little-modified chosen signals, their dynamical behaviors are studied. It is important that the system behaverobustly and without oscillation. Figures 3.24 to 3.27are about all joints with angular motions.

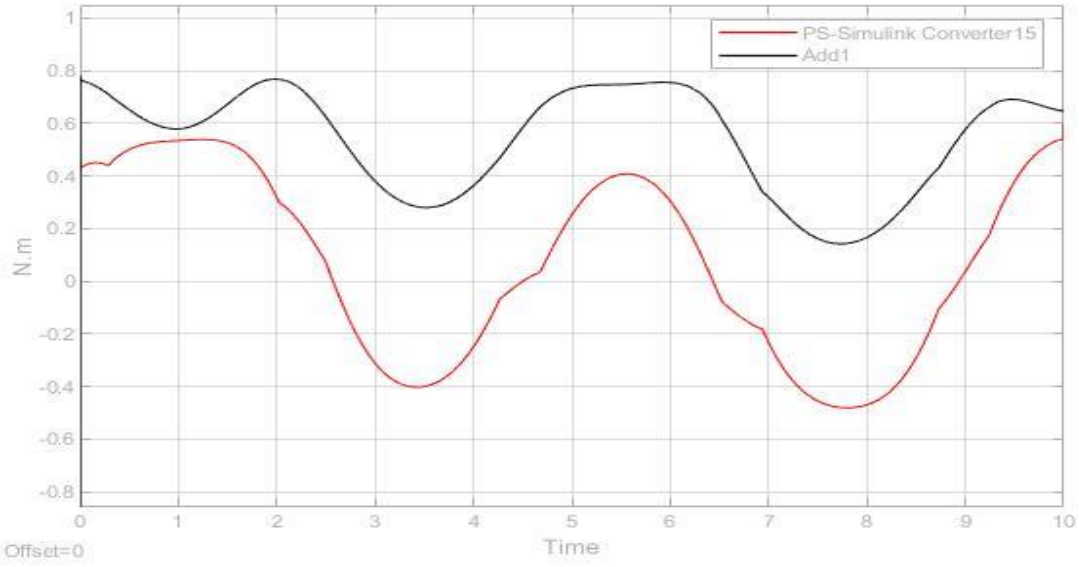

Figure 3.24: The computed torque by dynamic model and torque out of applied angular motion in joint one in 0.4 (rad/sec) frequency, amplitude 0.6 and pi/3 phase

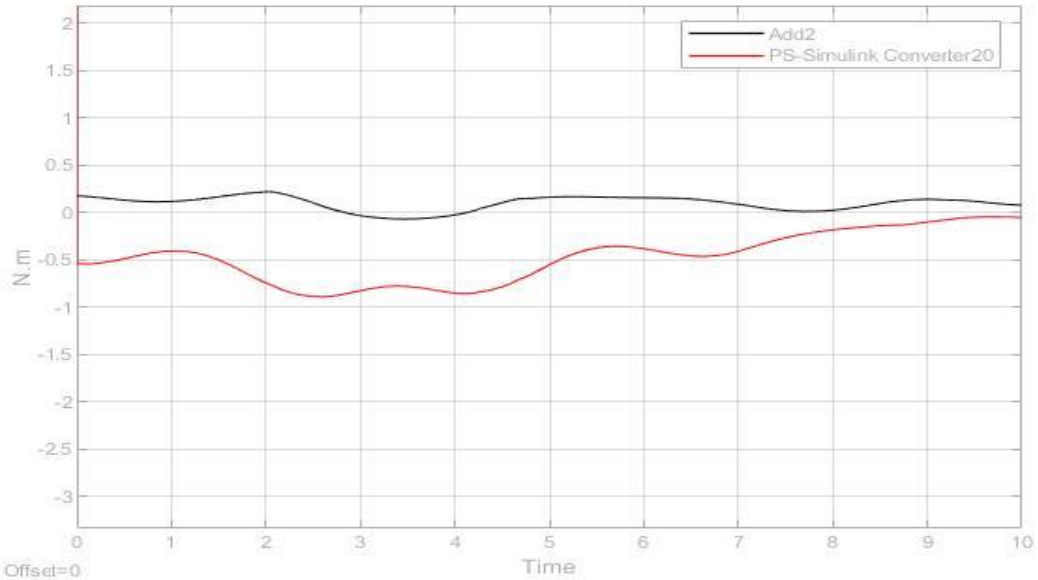

Figure 3.25: The computed torque by dynamic model and torque out of applied angular motion in joint two in 0.5 (rad/sec) frequency, amplitude 0.5.

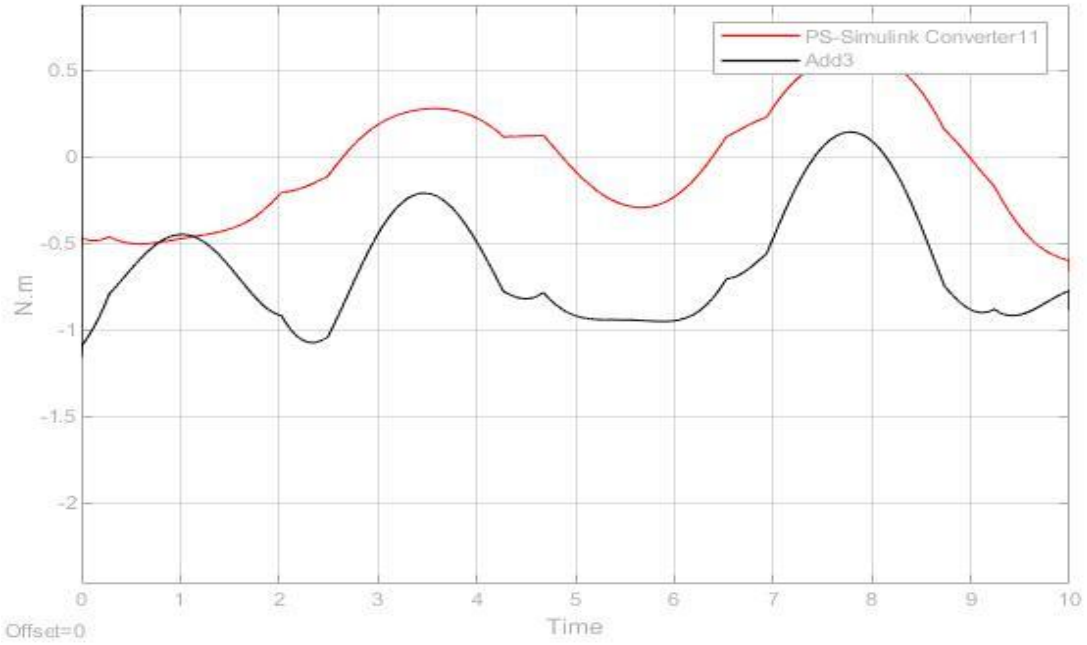

Figure 3.26: The computed torque by dynamic model and torque out of applied angular motion in joint three in 1.4 (rad/sec) frequency, amplitude 1.

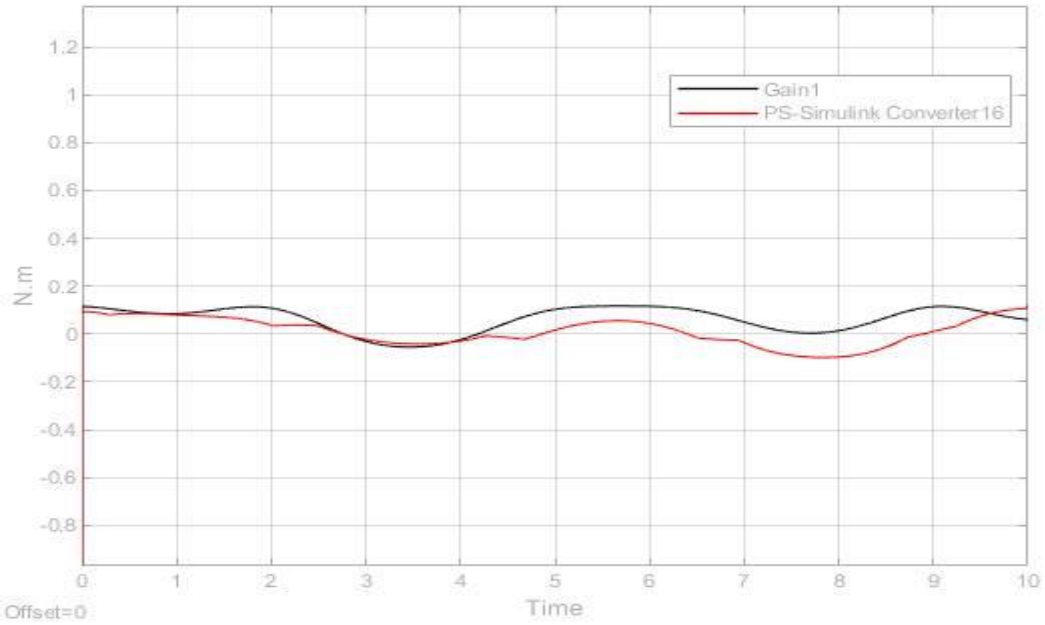

Figure 3.27: The computed torque by dynamic model and torque out of applied angular motion in joint four in 0.08 (rad/sec) frequency, amplitude 0.05.

As it appears from the figures, the calculated torques from dynamic model can follow the torque out of each joint with a very high amount of accuracy, also in joint number two angular motions is possible. It is important to know that when every change in frequency is going to be made, running simulation for all the joints is necessary. Since every change to the angular motion of a single joint is effecting the torques of other joints. Therefore, the combinations of the signals for the system are important. In the next chapter, the implemented control for the system will help the system to compensate the calculated torques.

3.4.2 Second study: workspace and equivalency to arm essential movements

In this study, the workspace of every joint and the whole system is going to be investigated. In order to start the study for every joint, the rest of the body must stay rigidly. Therefore, in the Simulink environment, all the joints except the one, which is under the study, should be removed. Starting with the first joint, putting the desired motion for the input of the first joint, and calculating the dynamic variables, the position of end effector will be acquired with respect to the base frame. Figure 3.28 is the workspace for the first joint.

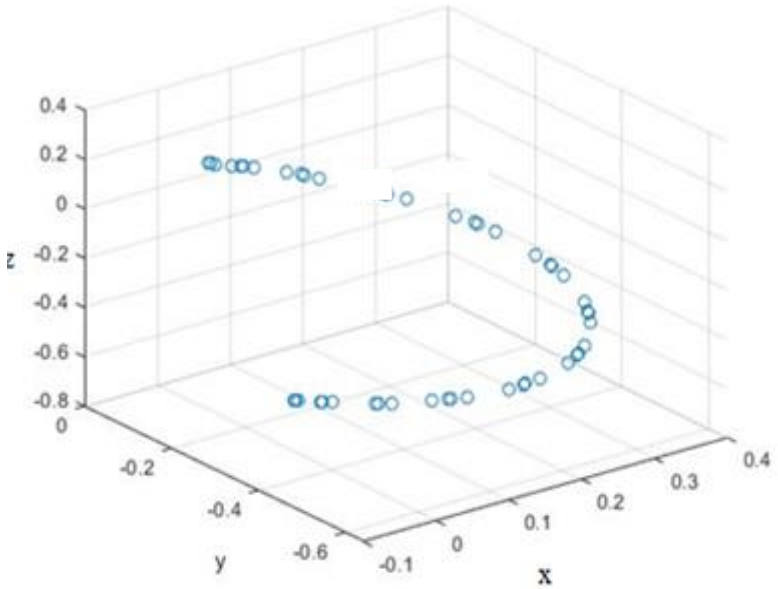

Figure 3.28: Workspace for the first joint.

As it has been discussed before, the movement type of the joint one in the shoulder can be translated as abduction/adduction. Figure 3.29 is depicting joint one moving the robot and the equivalency of the motion in the arm. As it is clear, the motion profile of the joint one is around the z-axis of the base frame.

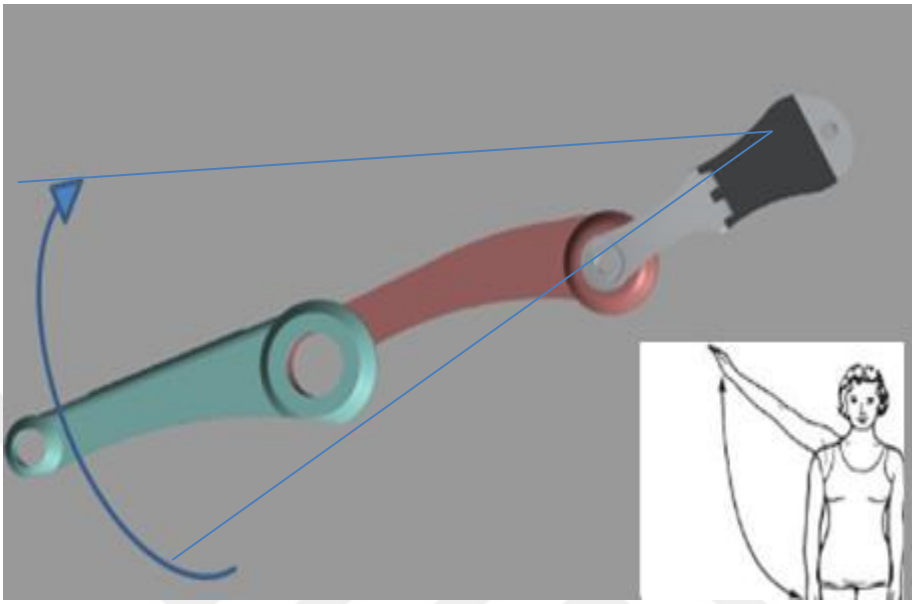

Figure 3.29: First joint motion and its equivalency in shoulder

For the second join, by removing all the joints except joint two, the same simulation is done. The workspace of the second joint can be found in figure 3.30.

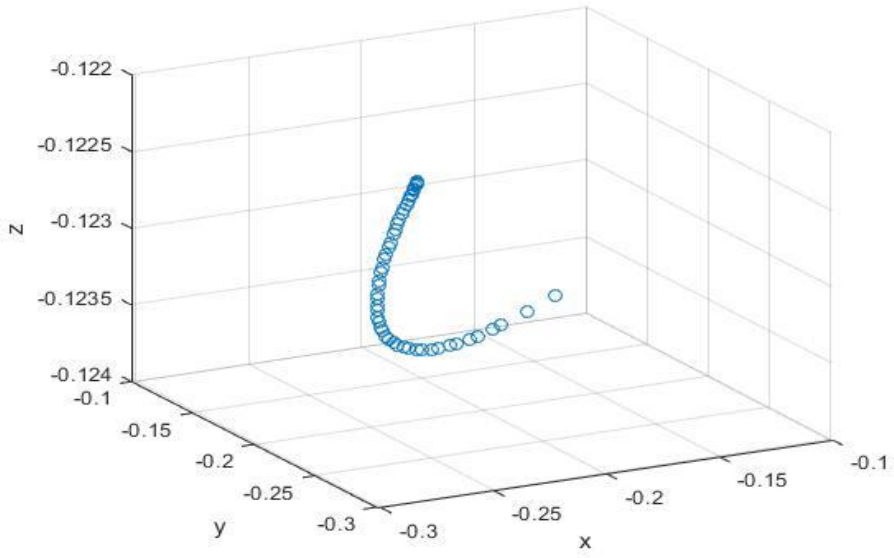

Figure 3.30: Workspace of the second joint.

The motion in the robot can be translated as horizontal abduction/adduction. Figure 3.31 shows the joint two moving the robot rigidly and the equivalency of the human motion in the arm. Not that in Figure 3.32 the motion profile is around y-axis of base frame in joint one.

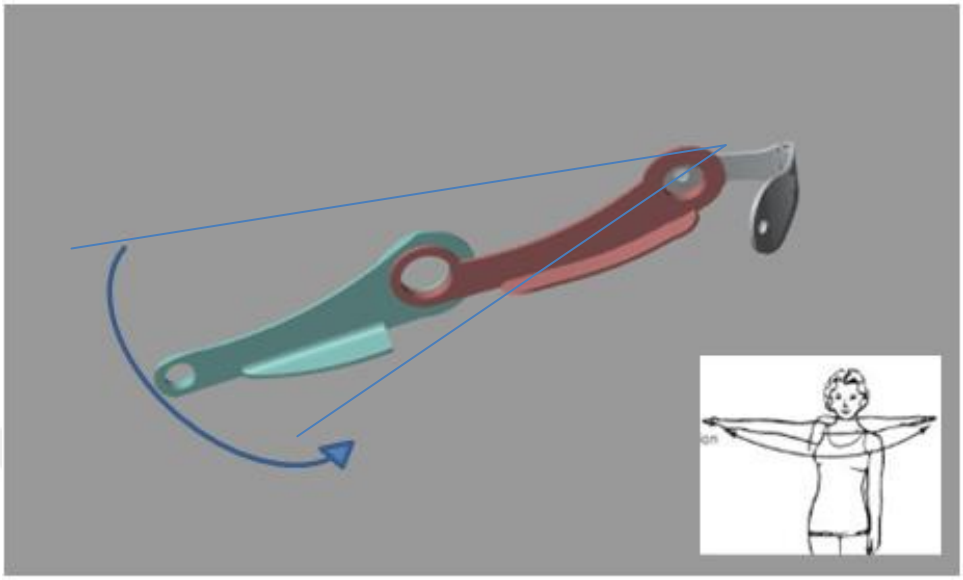

Figure 3.31: Secon joint motion and its equivalency in shoulder

For joint three the result of its workspace can be seen in figure 3.32.

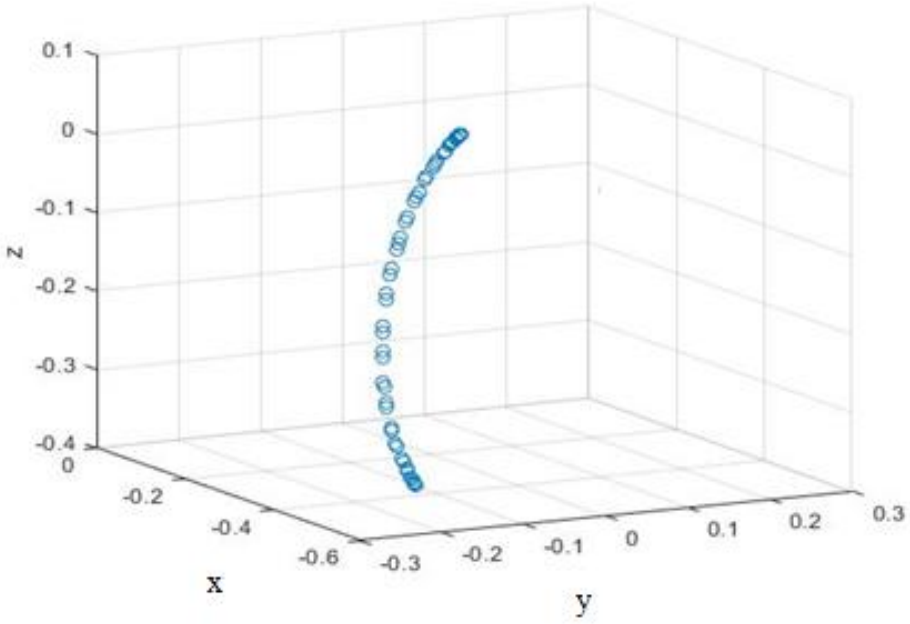

Figure 3.32: Third joint motion and its equivalency in shoulder

The types of movement in the joint three can be found as flection/extension of the arm. Figure 3.33 is about the joint three motions and its translation to type of arm essential movement.

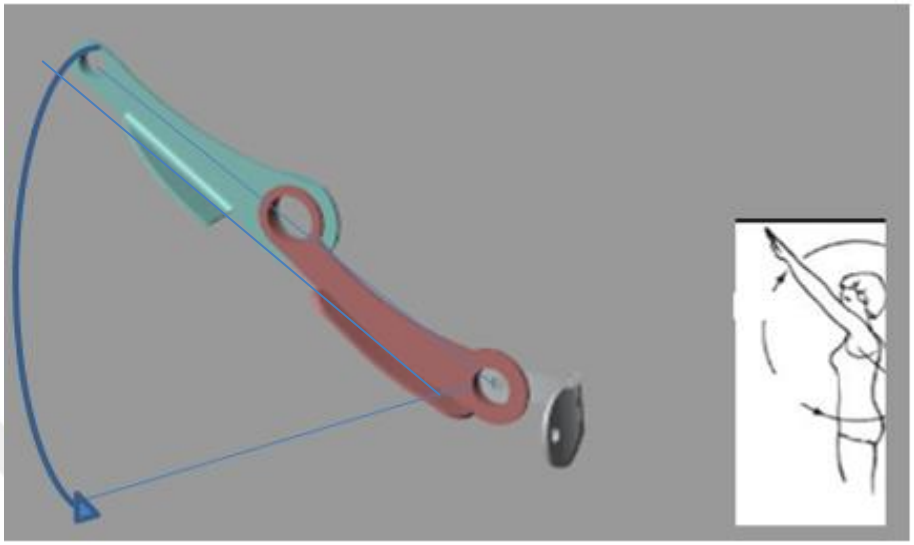

Figure 3.33: Secon joint motion and its equivalency in shoulder

All these 3 joint work sapces let the arm to move freely in the sphere shape of surface. Figure 3.34 shows the whole workspace for the upper body exoskskleton.

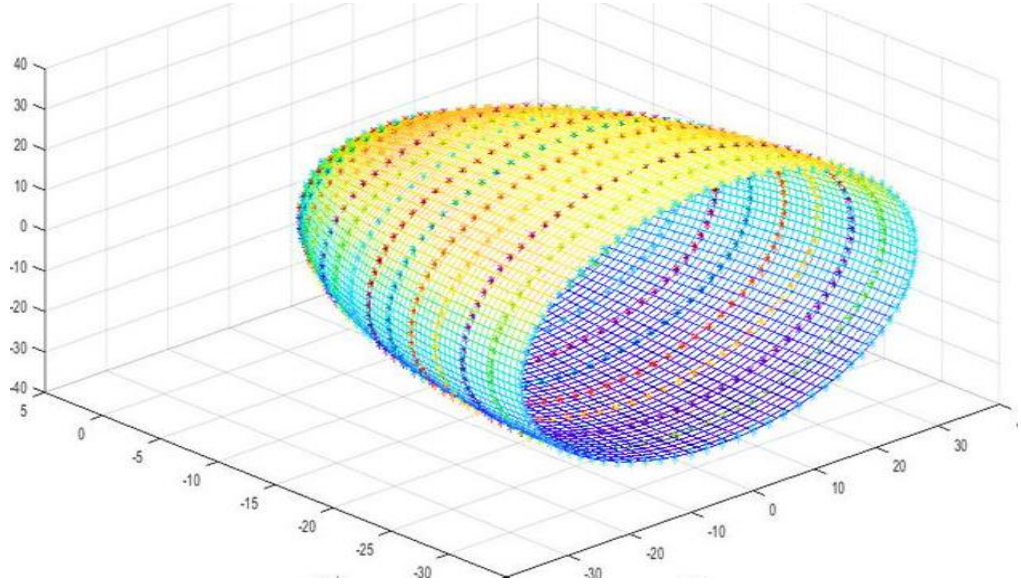

Figure 3.34: The whole workspace for upper body exoskskleton.

3.5Concolusions

In this chapter, the scientific approach for achieving the dynamic model of the robot has been discussed and the accuracy of the mathematical model has been proven with Simscape and MATLAB/Simulink. The workspace of each joint and their equivalency to arm motion has been studied as well. The motion study was donebecause one of the works in future would assign a motor to each joint. Therefore, it is important to find good frequencies for the system.

Dynamic simulations have been resulted to aboveexplanations, that $\boldsymbol{q_i \dot{q}_i \ddot{q}_i}$ as variables for each link have been discussed in chapter 3. Therefore, by applying the detected result, first, the robot plant can be created by using state-space method and then the required torque and force for some special situation will be investigated. In inverse dynamics sub chapter, more detail on how to calculate and how to design the control system will be discovered. Chapter 3 was all about forward and backward dynamics. In next chapter, the approach for designing torque controller, simulations and results will be discussed.

4.CONTROLLER DESIGN METHODS AND SIMULATIONS

In this chapter the methods and approach for a well-fitted controller will be discussed. According to the previous chapter the first assumption of designing this upper body exoskeleton was for the use of stroke patients. Since the stroke patients are suffering from numbness of the limb, the device should assist with arm lifting and enhancing the arm motion. Therefore, it is necessary to equip the device with motors and controllers for this purpose. The chosen controller has huge effect on exoskeleton performance. Moreover, the mechanical design of the device is another significant factor to enhance the performance of the robot. But, note that the complexity in mechanical design of the robot usually leads in utilizing more sophisticated control methods [45].

Previously in chapter three, the dynamical model was achieved with Newton-Euler Method and simulations related to torques and angular motions resulted in the system dynamical model accuracy. In this chapter, by applying PID controller and Computed Torque Controller (CTC) to the achieved dynamic model of the system, the best controller for the system will be introduced. Note that, the dynamic model of the system is transformed into the *MATLAB s-function level 2.*

4.1 The PID controller

The classical joint controller is a renowned approach for torque calculations, whichis defined as 4.1 equations.

$$e = \theta_d - \theta \qquad \tau = K_p e + K_i \int e + K_d \dot{e} \tag{4.1}$$

K_p, $K_i$and $K_d$are called the proportional, integral and derivative gain matrices, respectively from desired angle. Since the dynamic model of the robot has been discussed in previous chapter, the applied controller through the dynamical model can be written as below equation 4.2.

$$D(\theta)\ddot{\theta} + C\left(\theta, \dot{\theta}\right)\dot{\theta} + g(\theta) = K_p e + K_i \delta + K_d \dot{e} \ (4.2)$$

In 4.1, the θ_d is the desired angular position and θ is the actual position and $\dot{\delta} = e$

PID controller is best known for controlling the most industrial robots. However, the methodology for tuning the PID controller which is choosing the fittedK_p, K_i, K_d for the controller response is crucial. The PID controller can gives the best response with step inputs [46].

Figure 4.1 explains about this controller.

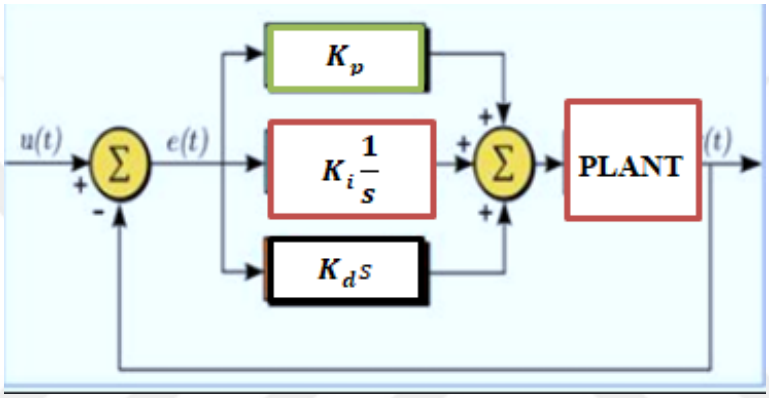

Figure 4.1: The PID controller for joint

As it can be seen from the figure, the PID controller had been applied to every joint. It means forever joint there should be one PID controller, which leads to four PID controllers in total. By choosing the units of the signal, it can be imported and exported correctly. Figure 4.2 and 4.3 are the results for every joint actual and desired angular motion. The input to the system was the step signal and frequencies and amplitudes are 1. As it has been mentioned previously, there is a methodology for the desired response of the controller. The controller can gives 3 different speed response based on the damping ration behavior. These responses are better known as “under damp”, “critical damped” and “over damped” which they will be explained in more detail. However, before get into the PID controller,first it is essential to introduce the assumed system for 1-Dof and the PD controller which is applies to the system that can be find in figure 4.2. In order to make the calculations easier, the calculations will be followed with laplas form. This type of controller is called independent joint control and each

axis of manipulator or robot is controlled as single input/single output (SISO) system. In this case, all coupling effect from other links motions are considered as disturbance[45].

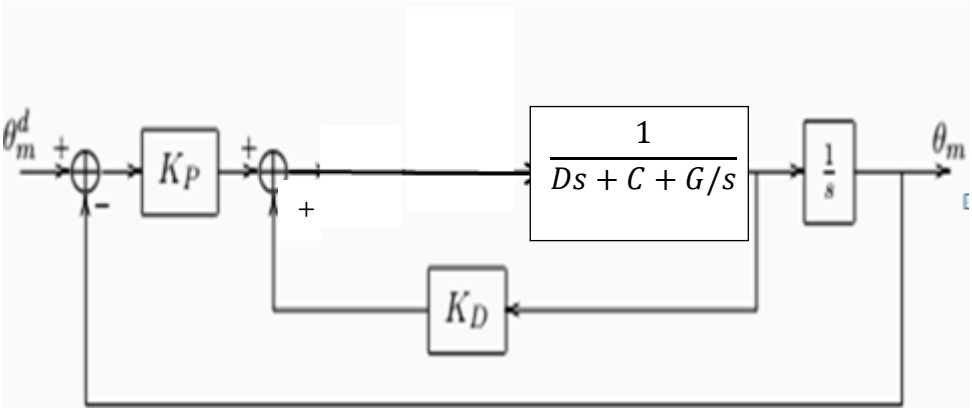

Figure 4.2. The block diagram of the 1-DoF Plant and the PD controller

As it can be seen there is no external disturbance impacting the system. Next step is the methodology on how to achieve K_d and K_p controller for assumed system. As the system is close loop, first the total equation is necessary to be calculated. Equation 4.3 is the close loop system.

$$G = \frac{K_p + K_d s}{Ds^2 + (C + K_d)s + (K_p + G)} \tag{4.3}$$

Equation 4.4 explains about the response characteristics of the system.

$$s^2 + \frac{(C + K_d)}{D}s + \frac{(K_p + G)}{D} = s^2 + 2\zeta\omega + \omega^2 \tag{4.4}$$

Hence, K_d and $K_p$can be found as below:

$$K_p = D\,\omega^2 - G\& \quad K_d = 2\zeta\omega\, D - C$$

As it has been mentioned previously, the speed response of the system is dependson damping ratio. If the damping ratio is between 0 and 1, the response of the system is called “under damped”. If the damping ratio is equal to 1, the response of the system is called “critically damped”. If the damping ration is more than 1, the response of the system is called “over damped”. In many robotics applications, it is relevant to work with critically damped response.

Next step is to find the system that impact with PID controller. Therefore the term. $K_i$is adding to the system. Note that, `PID controller is the best for compensating the system when there is an uncertainty in the system. Figure 4.3 shows the assumed block diagram for 1-DoF system.

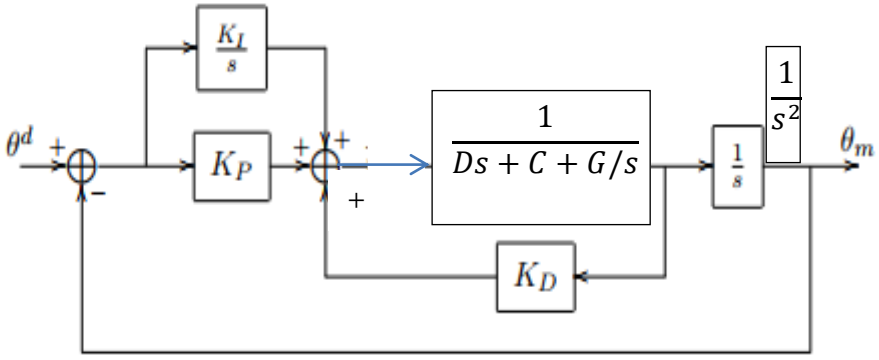

Figure 4.3: Block diagram of 1-DoF system andPID controler..

For PID conroller the transfer function of the whole system can be define as equation 4.5:

$$G = \frac{\frac{K_i}{s} + K_p + K_d s}{Ds^3 + (C + K_d)s^2 + (K_p + G)s + K_i} \tag{4.5}$$

As it can be seen from 4.4, the order of the whole system has been increased. In order to find K_i, the stability method should be apply to the system the result for rinding $K_i$can be found in 4.6.

$$0 < K_i < \frac{(G + K_p)(C + K_d)}{D} \tag{4.6}$$

One essential aspect of choosing the parameters for PID controller is the restriction on the chosen number. Altough increasing in parameters can lead to reaching to the steady-state situation sooner, but also can lead to huge oscilations in the system specially in K_p. Therefore, stifness and osscilations of the system are two important factors in limiting K_p[45].

Next step is to define the system precisely. Since, the mechanical design has been transported to the Simulinkenvironment by the means of Simscape in the previous chapter; it can be used for following the desired trajectories and end-effector PID independent joint control. Note that, this defined system in Simscape is SISO, in which for every joint there is one input and one output from the system, leads to 4 independent controllers. The Simulink is a great tool for building the system accurately based on the mass properties that has been calculated by the means of SolidWorks mass and material section. These details have been well explained in previous chapters.After arranging the system and discussion on how to find the system response, the three types of the response for every joint should be studied. Figure 4.4 to 4.7 shows the response of the system to the step inputs in “under damped” mood.

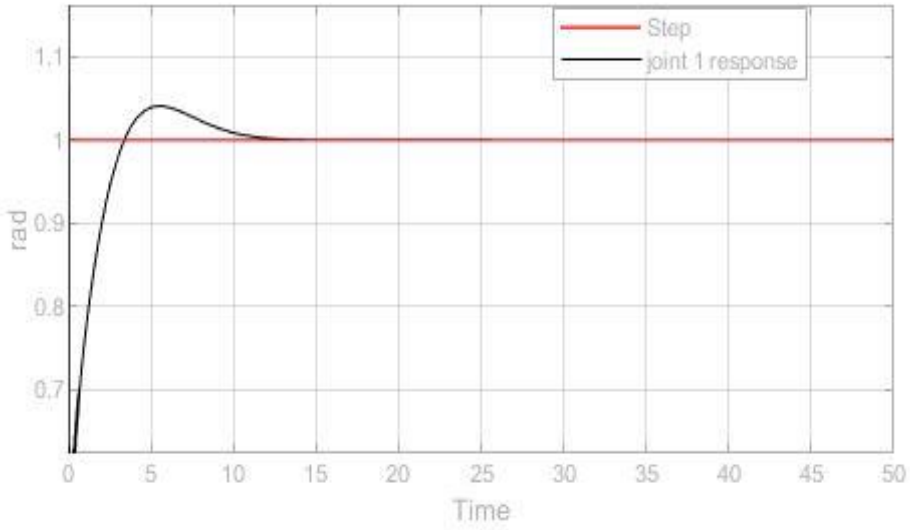

Figure 4.4: Joint 1 “under damped” response.

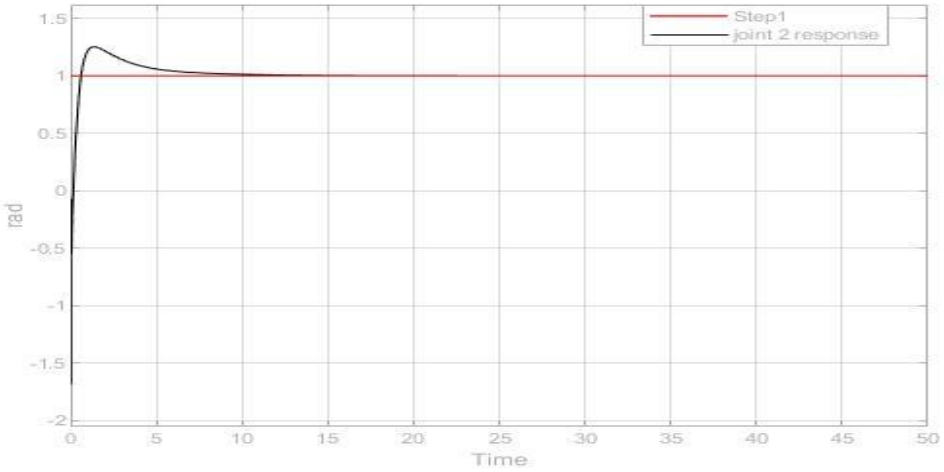

Figure 4.5: Joint 2 “under damped” response.

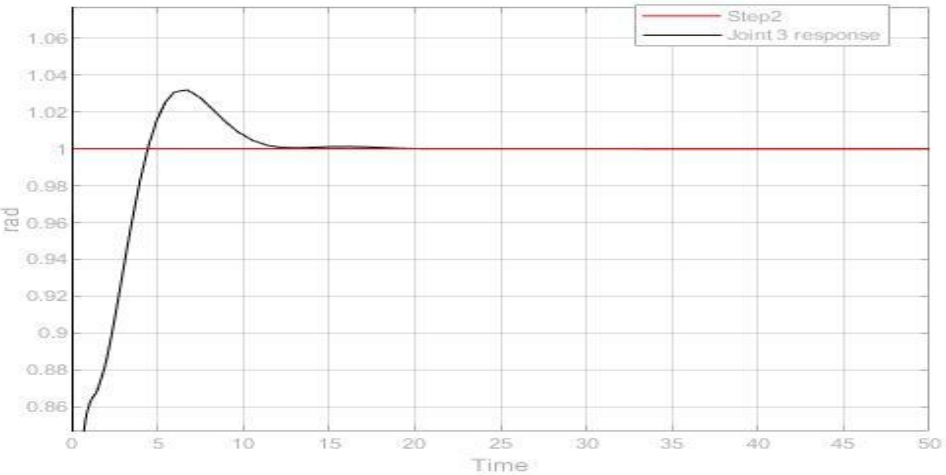

Figure 4.6: Joint 3 “under damped” response.

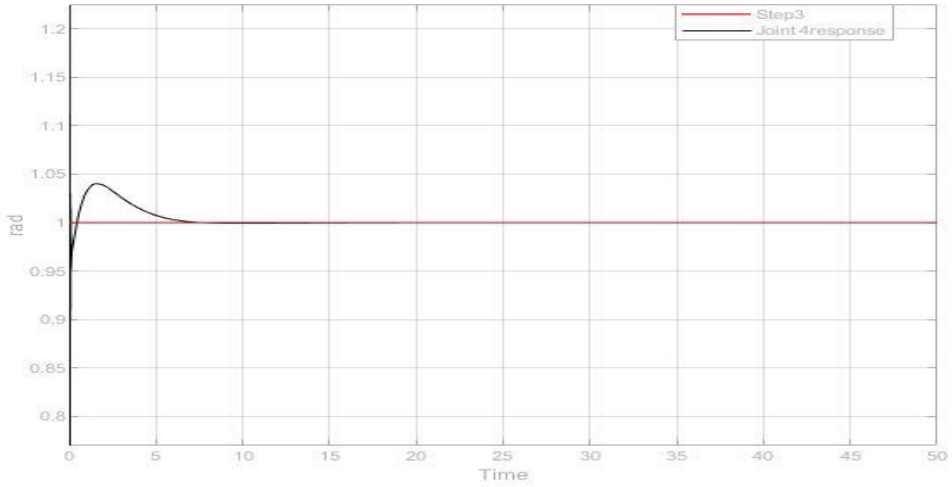

Figure 4.7: Joint 4 “under damped” response.

Figures 4.8 to 4.11 are the tracking error for all the joints.

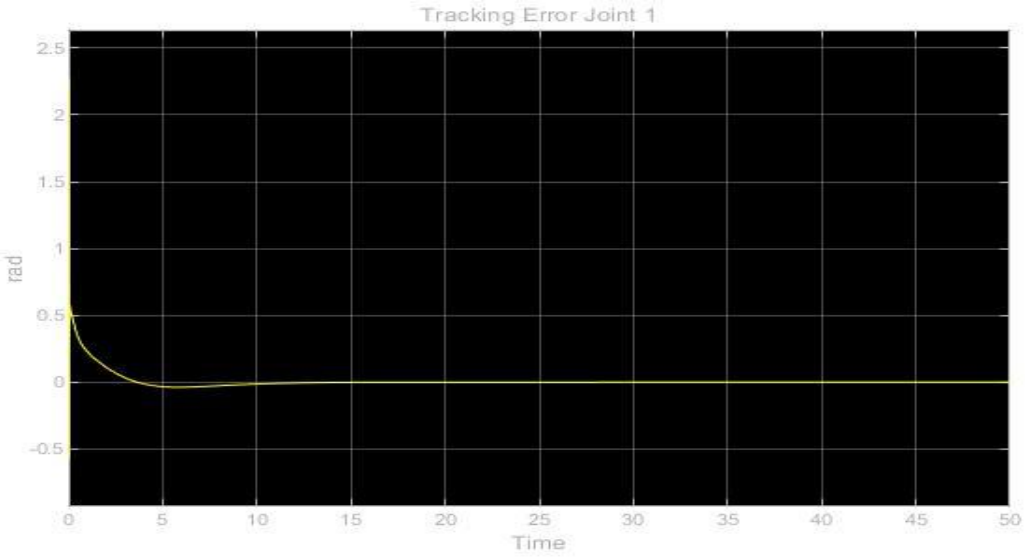

Figure 4.8: Joint 1 tracking error in “under damped” response.

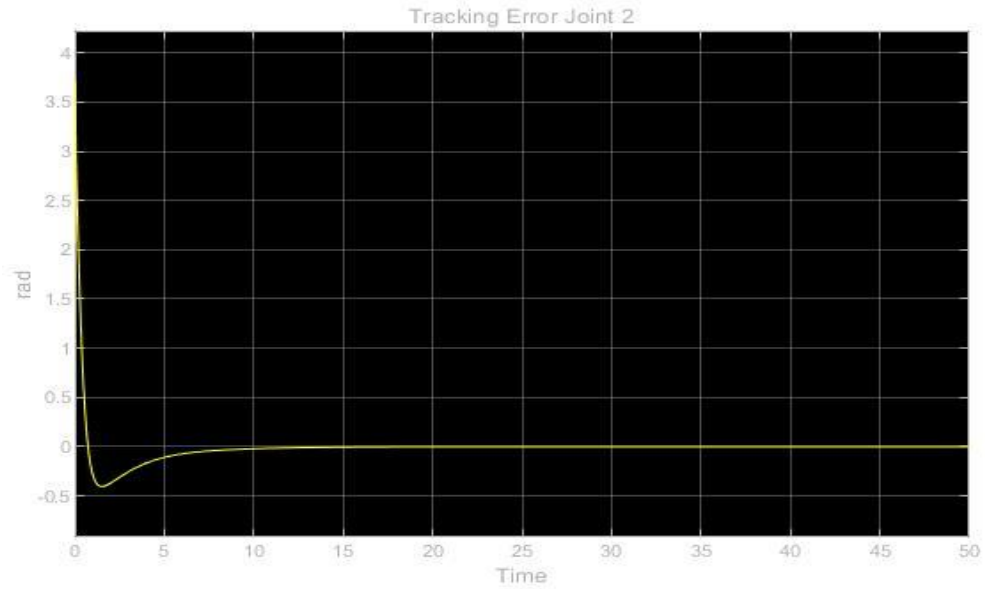

Figure 4.9: Joint 2 tracking error in "under damped" response.

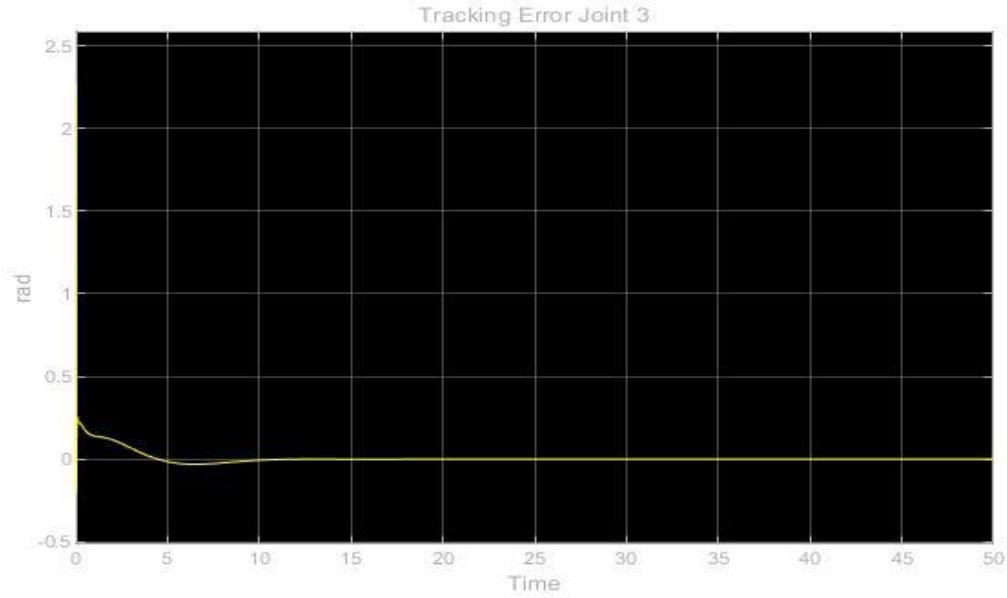

Figure 4.10: Joint 3 tracking error in "under damped" response.

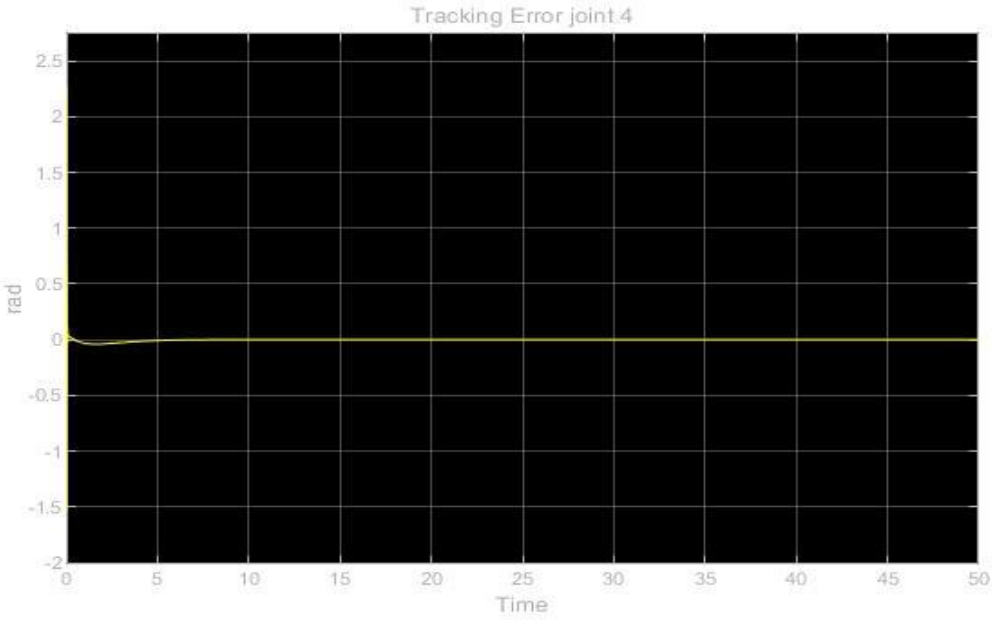

Figure 4.11: Joint 4 tracking error in "under damped" response

As it clearly observable from figures ,before the 15th second, the actual position is reaching to the desired position in "under damped" response. It is interesting to know the joints 2 and 4, response of the system are quite the same. The reason should be searched in dynamic behavior of the robot. In joint 2 the gravity does not have the huge influence due to the behavior of the joint. Also, in joint 4, the situation is quite the same with joint 2 but the reason is the position of the joint in the robot, since it is located as the last joint before end-effector. According to what has been explained previously, choosing the high range of control parameters specially for these two joint can bring the system to the state of oscillations, since the amount of gravity that is considered as the disturbance is lower than other two joints. This behavior in the system will be persistent in any types of response. Not only just in Simscape model-based controller, but also later when the controller is applied to the achieved dynamics model this behavior, this can be observed which implies that the dynamical model of the system is explored and defined precisely. Figures 4.12 to 4.15 are about response of the system in over damped state.

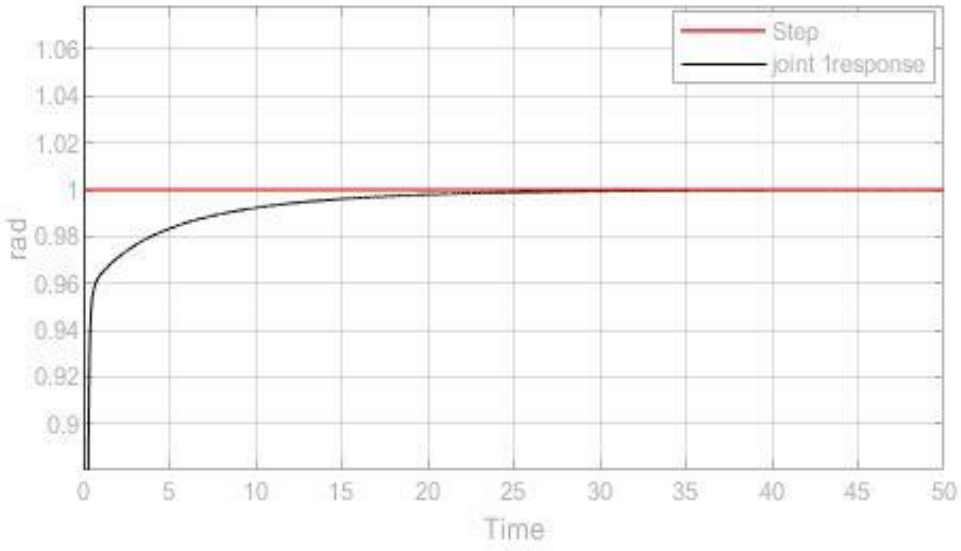

Figure 4.12: Joint 1 :over damped" response.

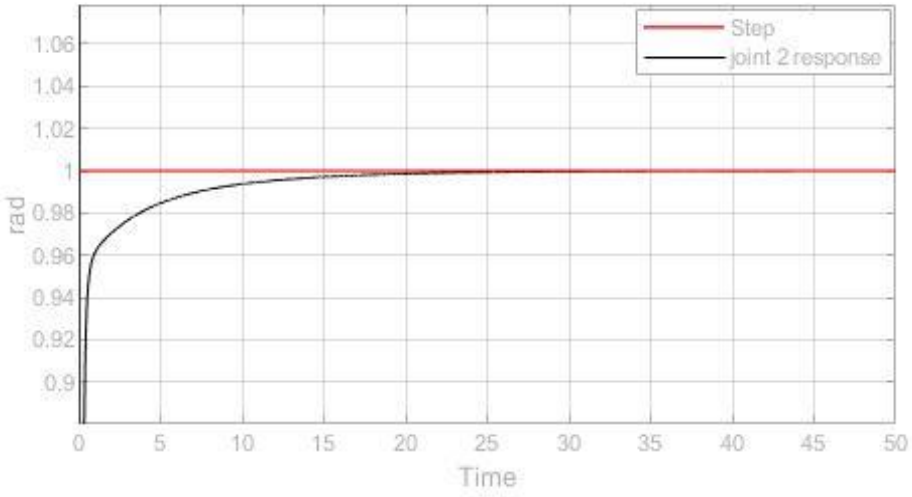

Figure 4.13: Joint 2 over damped" response.

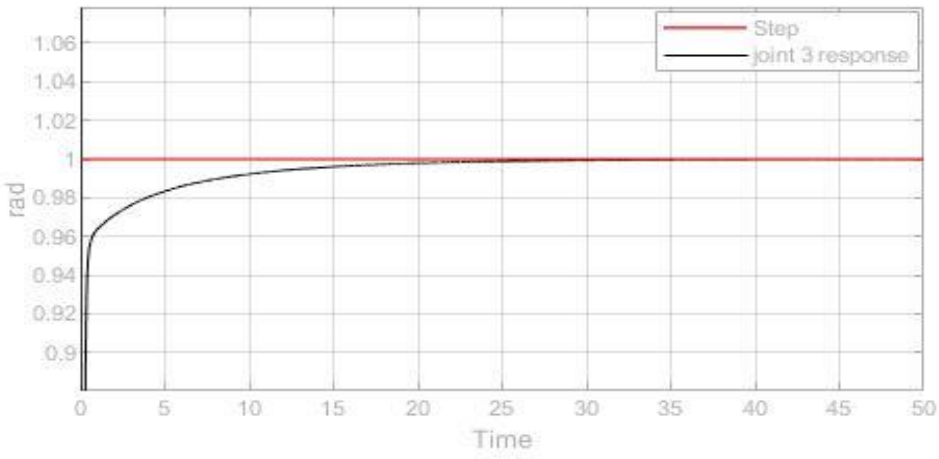

Figure 4.14: Joint 3 over damped" response.

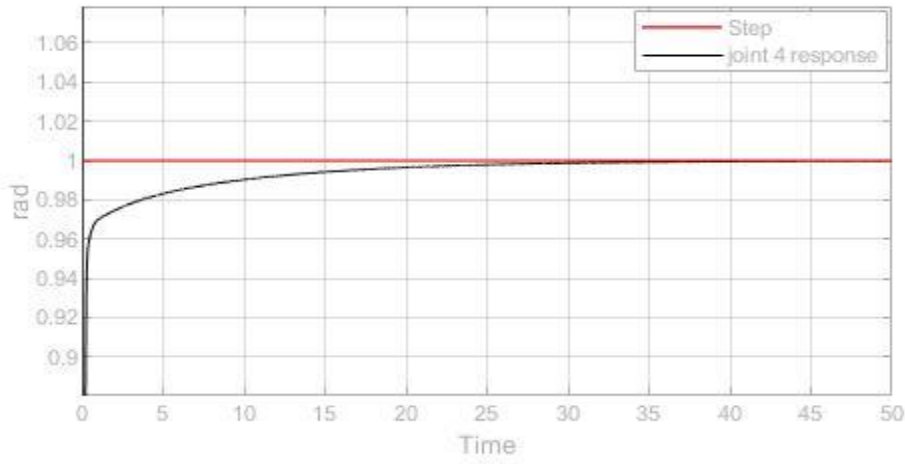

Figure 4.15: Joint 4 over damped" response.

The tracking error for this response in all joints can be find in figures 4.16 to 4.19.

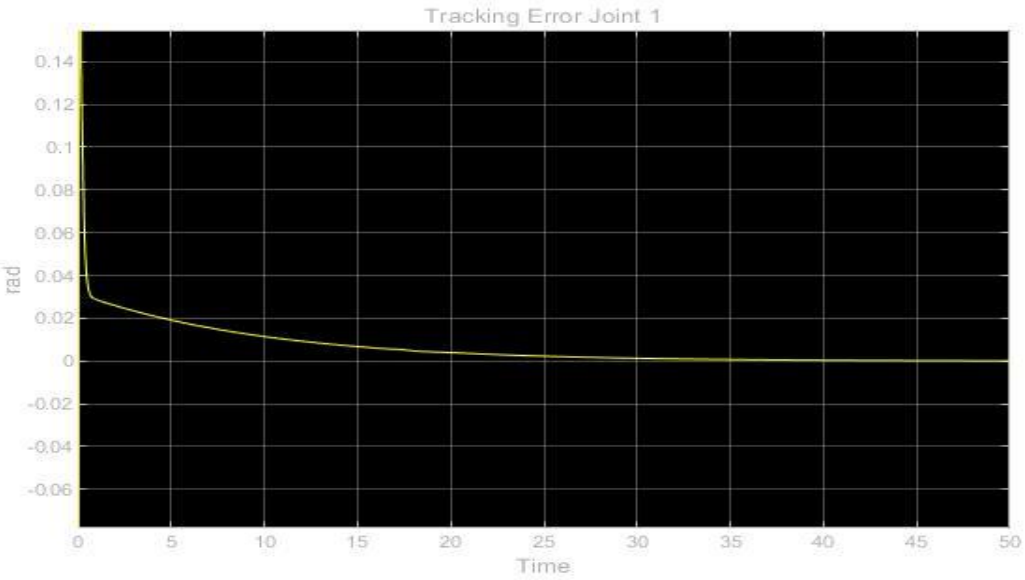

Figure 4.16: Joint 1 tracking error in "over damped" response.

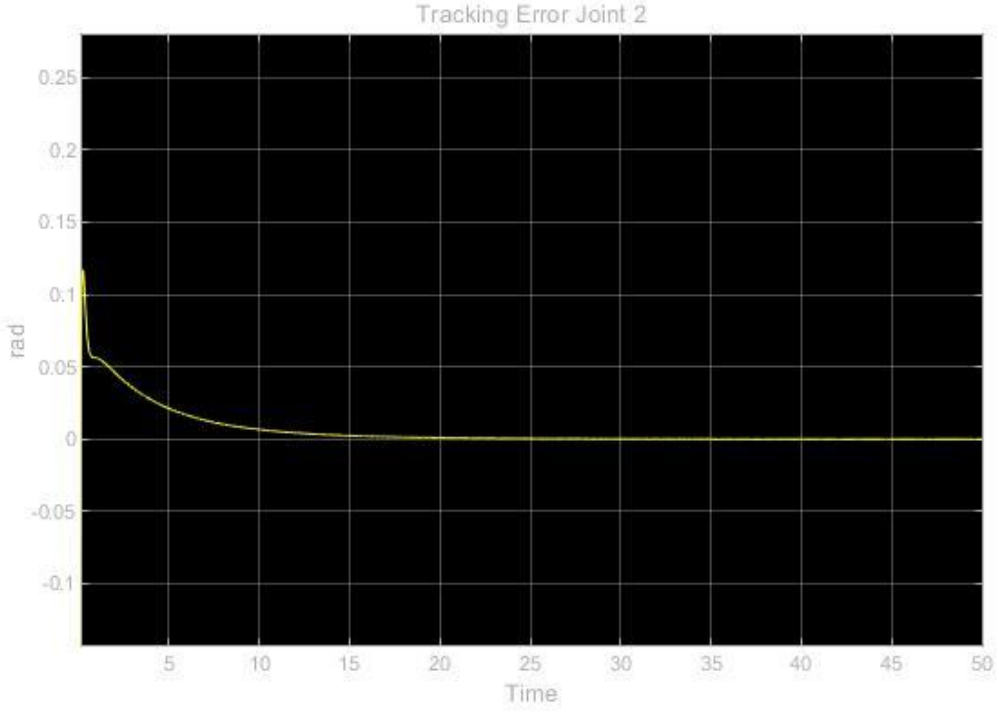

Figure 4.17: Joint 2 tracking error in "over damped" response.

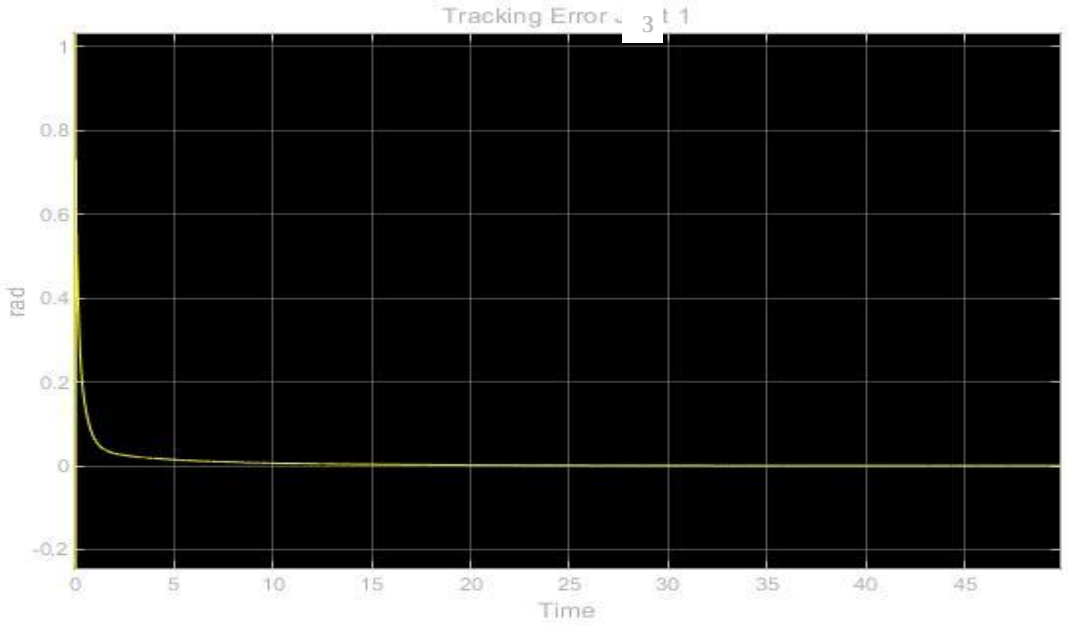

Figure 4.18: Joint 3 tracking error in "over damped" response.

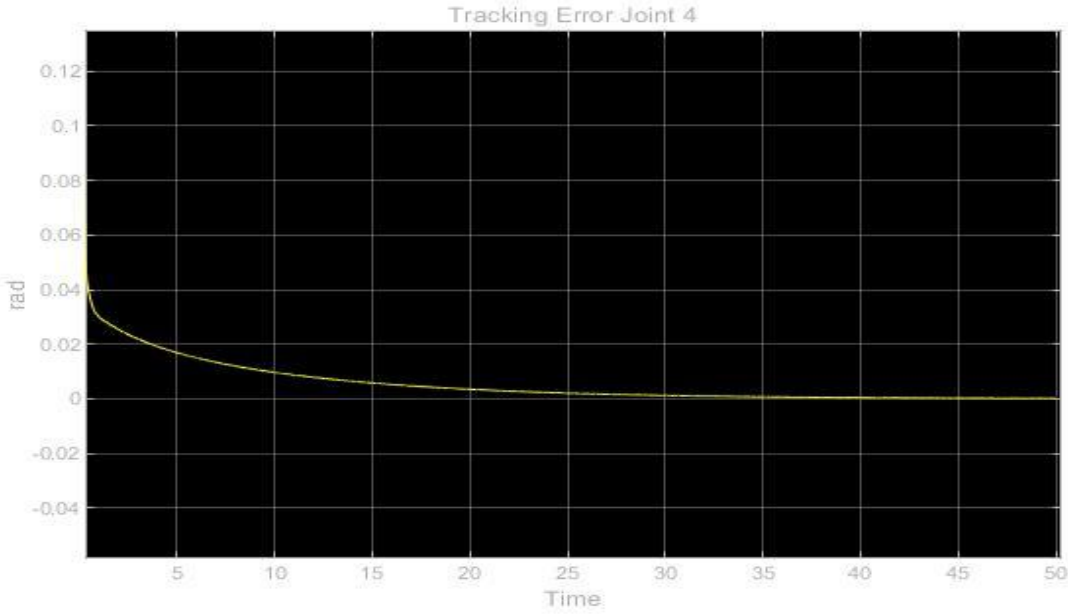

Figure 4.19: Joint 4 tracking error in "over damped" response.

From figures it can be recognized that in 30th second , the actual position of the joints are matched with the desired position. Figures 4.20 to 4.23 are illustrating the critical damped response of the system.

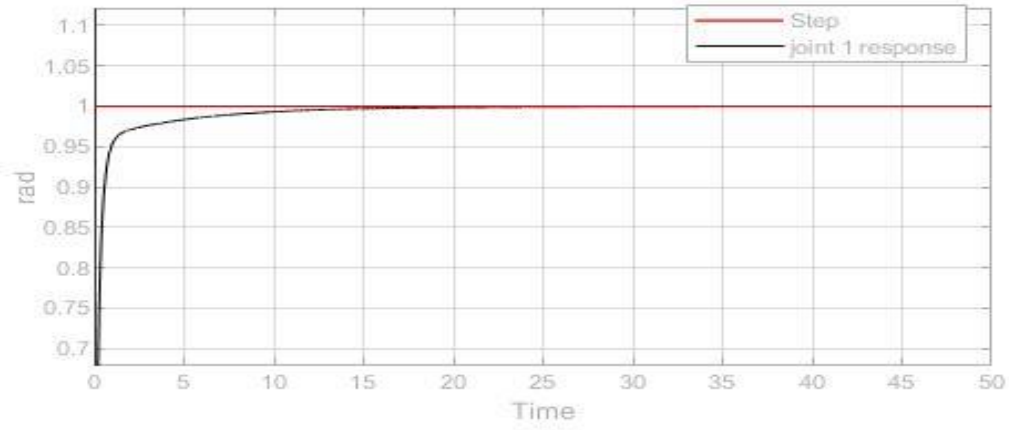

Figure 4.20: Joint 1 “critical damped” response.

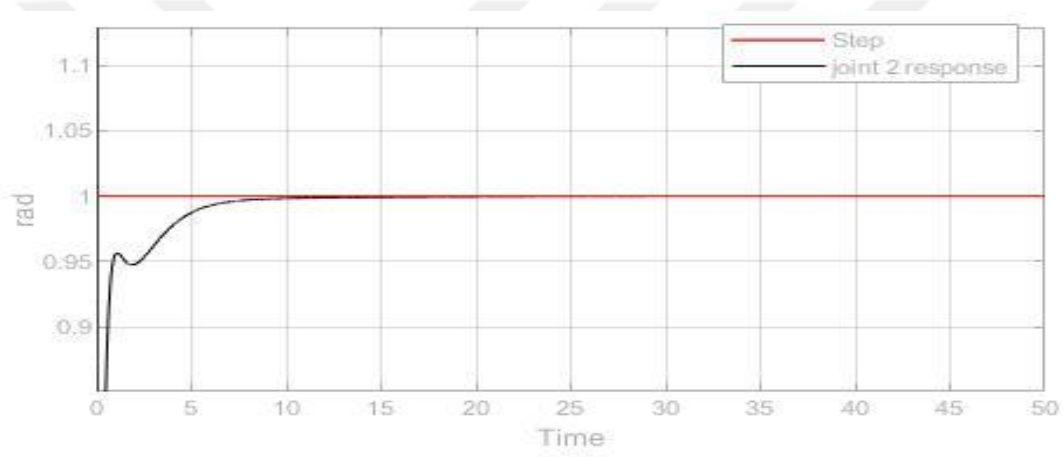

Figure 4.21: Joint 2 “critical damped” response.

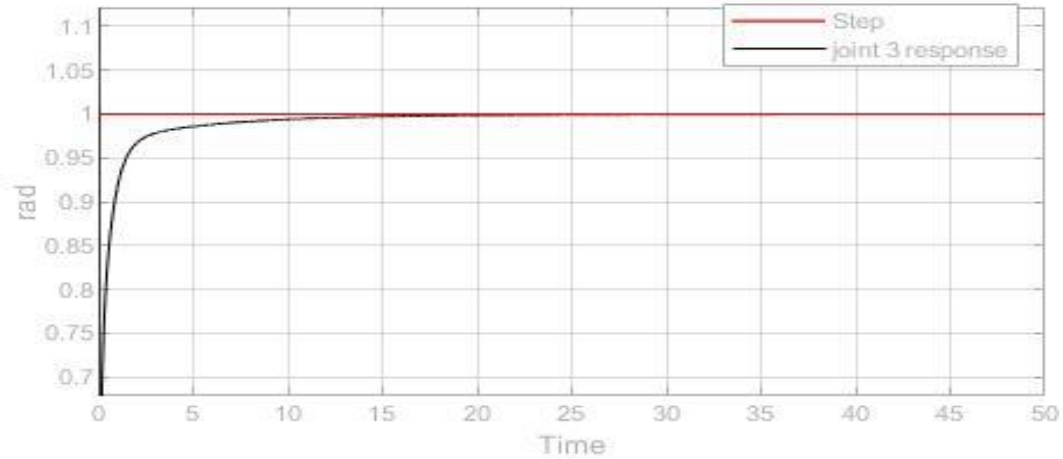

Figure 4.22: Joint 3 “critical damped” response.

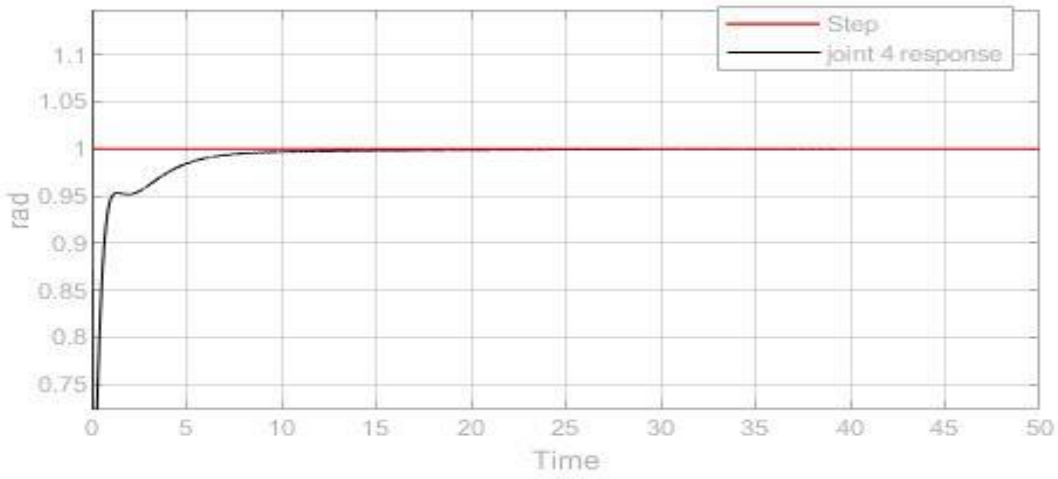

Figure 4.23: Joint 4 "critical damped" response.

Figures 4.24 to 4.27 are showing the tracking error for the critical damped response.

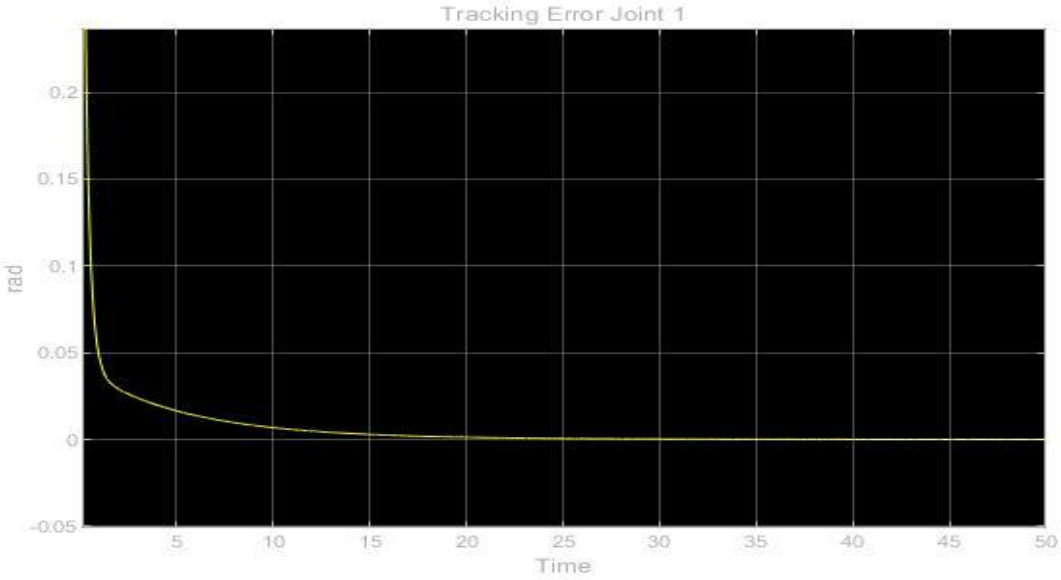

Figure 4.24: Joint 1 tracking error in "critically damped" response.

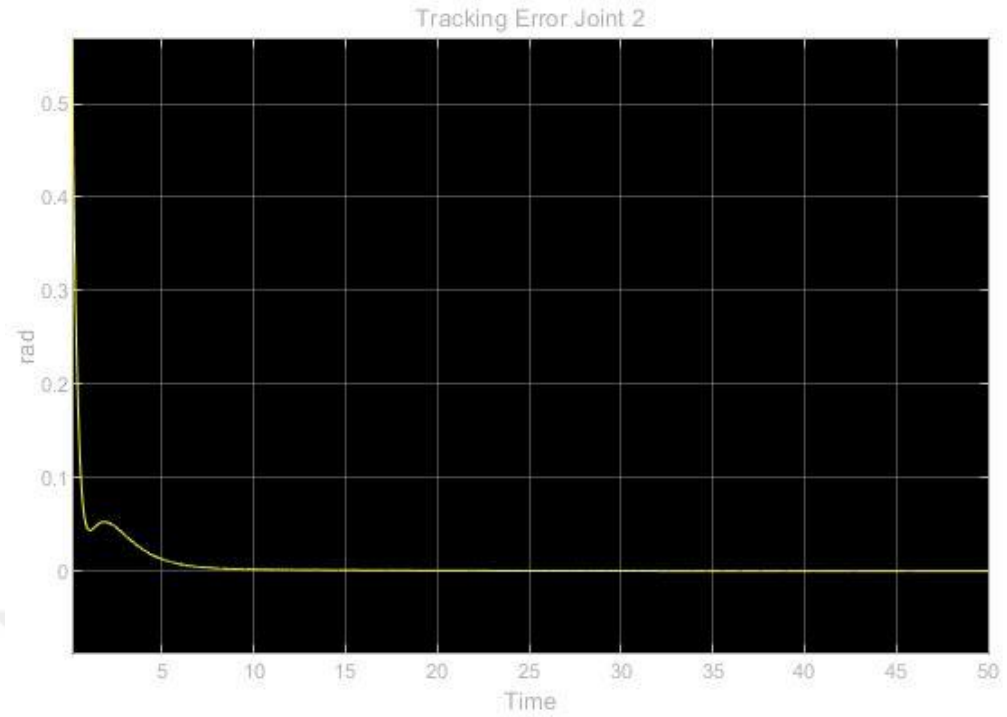

Figure 4.25: Joint 2 tracking error in "critically damped" response.

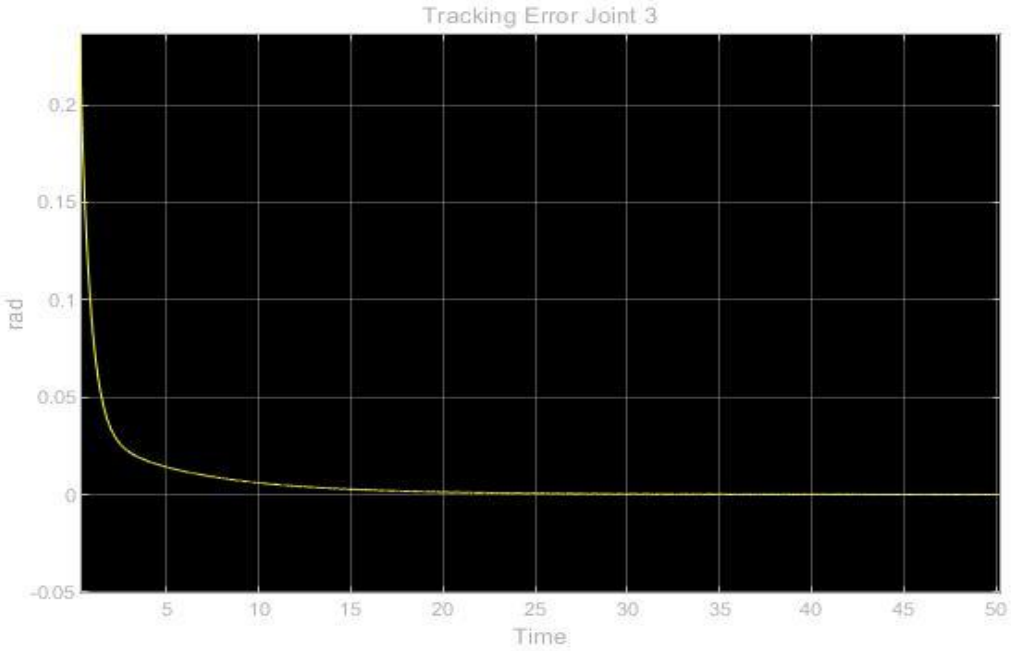

Figure 4.26: Joint 3 tracking error in "critically damped" response.

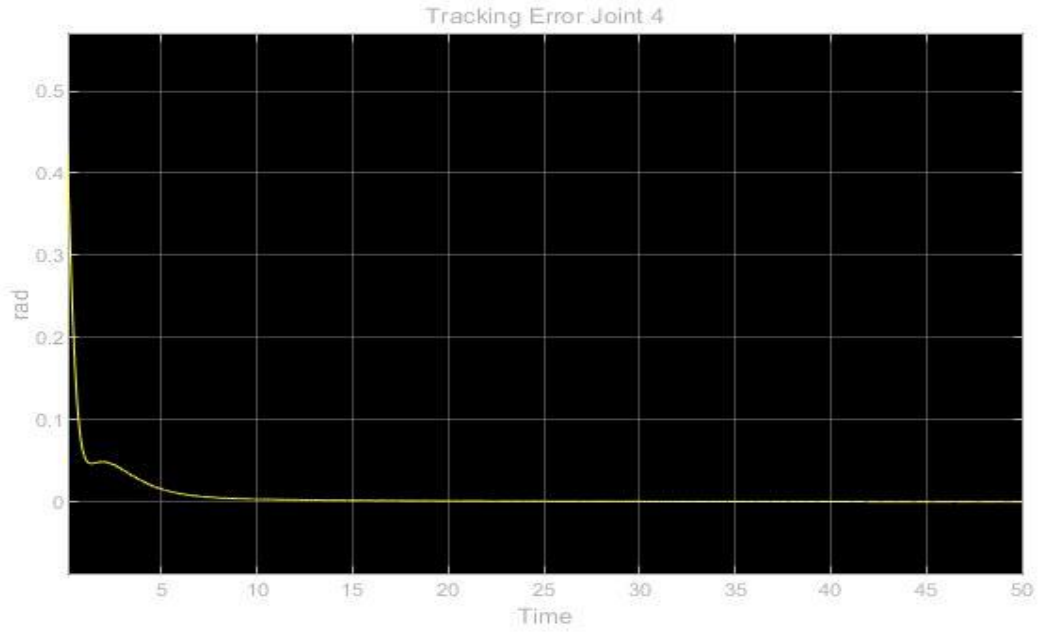

Figure 4.27: Joint 4 tracking error in "critically damped" response.

As it was been explained above, joints 2 and 4 show the same behavior which is reflected in their response to the controller. From critically damped response figure, it can be achieved that after 15^{th} second, the actual joint position meets the joint desired position. Critically damped is an important response in robotics. Many robots are utilizing this response for their applications. It is also used widely within rehabilitation robotics as well. Critically damped response lets the system to reach to the desired position fast and without over shoot which is important for the human user to not face the sudden impact from the system, especially in rehabilitation cases which may cause serious limb damage to the user [47].Although the over damped response may be seen much friendlier for rehabilitation case, but the energy consumption and fatigue of the device should be considered as well. Therefore, in many exoskeletons, critically damped response is chosen [47][48].

What have been explained, was about the response of the system to step input. However, in reality there is sinusoidal types of motion for this robot which will be applied by each assumed pancake motor in each joint. Therefore, the system response to the sinusoidal input should be studied. Figures 4.28 to 4.31 are the response of the system to the sinusoidal input to each individual joint. The PID parameters are the same as what have been achieved for critically damped response.

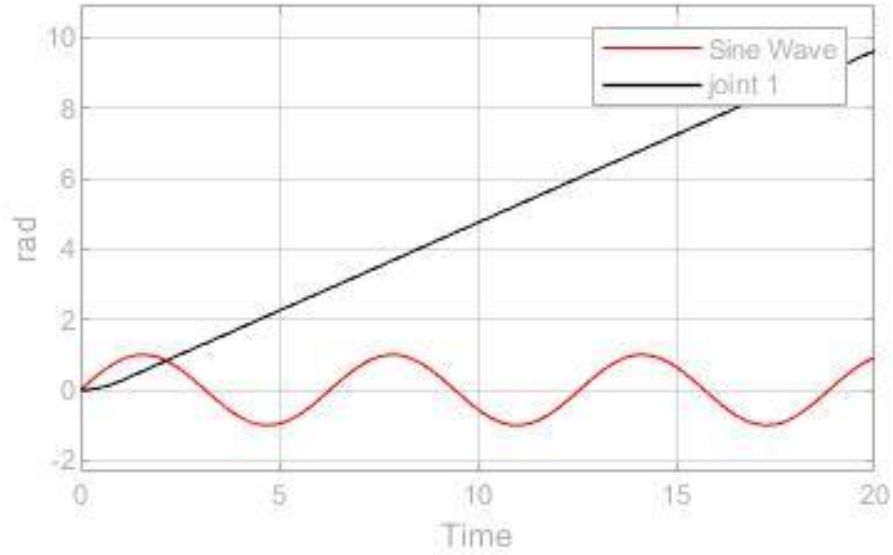

Figure 4.28: Joint 1 "critical damped" response to sinusoidal input.

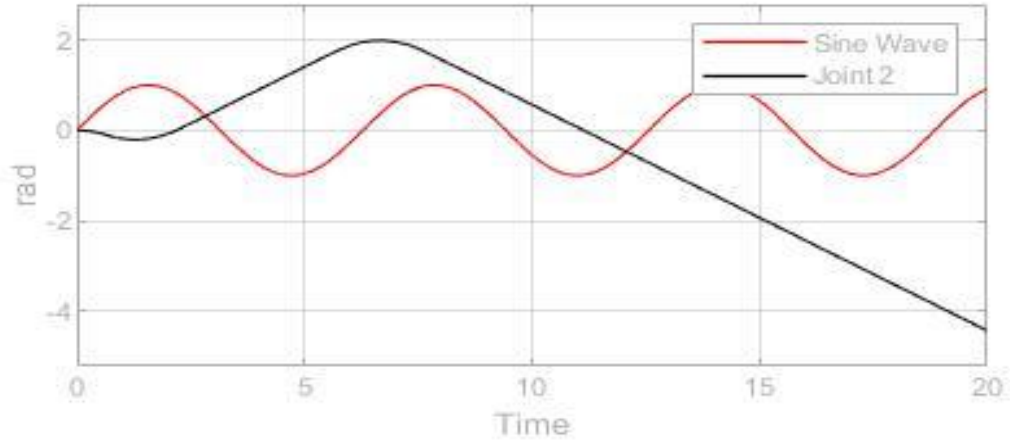

Figure 4.29: Joint 2 "critical damped" response to sinusoidal input.

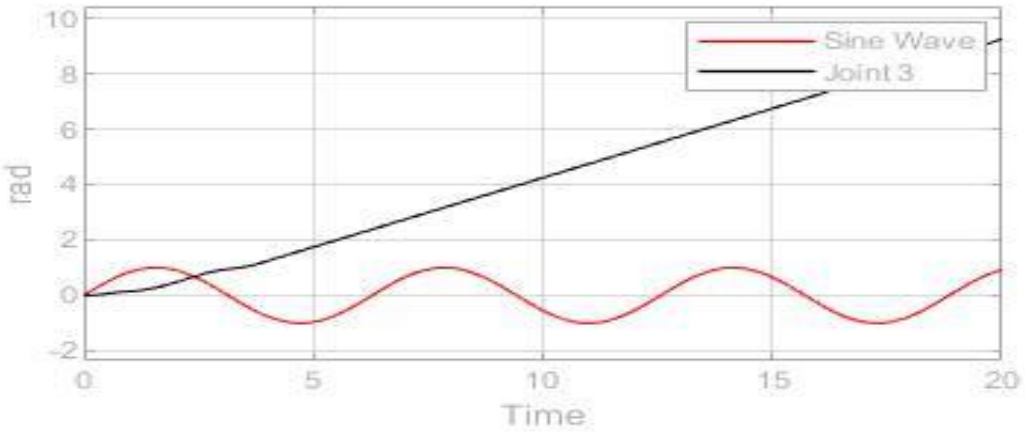

Figure 4.30: Joint 3 "critical damped" response to sinusoidal input.

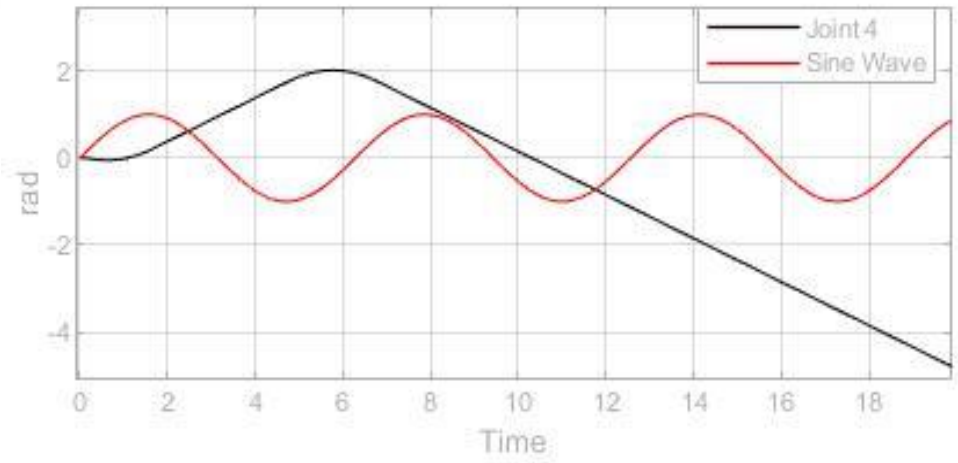

Figure 4.31: Joint 4 "critical damped" response to sinusoidal input.

As the results are showing, a huge oscillation can be observed in the system as the response to the sinusoidal inputs. The reason is that the independent joint controller has not that capacity to work with sophisticated inputs such as sinusoidal. PID controller is working great when the input is linear and simple such as step input. In this case, it is important to find the best-fitted alternative solution in controller. In this scenario, one of the reliable solutions is computed torque controller (CTC) which is suitable for nonlinearities in the system. In order to use this controller, it is important to bring the dynamic model in to the transfer function in the MATLAB/Simulink environment and the transfer function of the system becomes as multi

Input/multi Output (MIMO) .Moreover, in reality the dynamic model of the system is in the form of nonlinear and multivariable system and therefore, multivariable controller,

4.2 Computed Torque Controller (CTC)

Applying linear control for controlling the robot motion purpose leads in omission of nonlinear forces which couple with robot motions. This situation reduces the accuracy of the controller, since there are forces such as Coriolis and centrifugal in robot links that inherited the square of speed in itself [47].

Computed torque controller (CTC) is a nonlinear and strong controller, which is commonly used for the robot manipulator. The mechanism of this controller is based on calculating the arm torque by using for linearizing the nonlinear feedback [49]. This controller performance is highly efficient in the environment when there are no uncertainties such as dynamic parameters. Otherwise, the performance of the controller would not be acceptable. Note that, practically most of physical systems have unknown parameters or pose the time variant parameters. In these cases, the torque controller plays its role on compensating the dynamics equation of a manipulator [49][50]. Many researches have been done on computing robot and it is still a growing research topic [51][49].

As discussed, this controller is built on feedback linearization. Therefore, the first step is to linearize the feedback. The first assumption is to consider $\boldsymbol{q_d^{(t)}}$ as desired motion trajectory and desired displacement for manipulator end effector and $\boldsymbol{q_a^{(t)}}$ is the actual displacement of manipulator. Thus, the tracking error becomes as 4.7.

$$e(t) = q_d^{(t)} - q_a^{(t)} \tag{4.7}$$

In 4.1, $e(t)$ is the error of the dynamic plant? Next step is defining the state-space equation in the linear form of$\dot{x} = Ax + BU$. This equation can be expressed as 4.8.

$$\dot{x} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ I \end{bmatrix} U \tag{4.8}$$

By employing the BrunouskyCononical Form, the $\boldsymbol{U}$ can be find as 4.9

$$U = -D(q)^{-1}[C(q,\dot{q})\dot{q} + g(q)] + D(q)^{-1} \quad (4.9)$$

Equation 4.9 can be explained in the form of state-space in 4.10, since the term $x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix}$:

$$\frac{d}{dt}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} U \quad (4.10)$$

Since $\ddot{q}_i$ can be found by the inertia reverse matrix, $\ddot{q}_i$ becomes as 4.11.

$$\ddot{q}_i = D(q)^{-1}(\tau - [C(q,\dot{q})\dot{q} + g(q)]) \quad (4.11)$$

It is assumed that the inertia matrix is invertible. The inertia matrix plays an important role since it is the main factor of kinetics energy. This presumption is acceptable since the kinetics energy is always positive. Thus, equation 4.5 is valid [44][52]. Therefore, 4.3 can be specified as 4.12.

$$U = \ddot{q}_d + D(q)^{-1}([C(q,\dot{q})\dot{q} + g(q)] - \tau) \quad (4.12)$$

Equation 4.6 shows that the required torque can be computed as 4.13.

$$\tau = D(q)(\ddot{q}_d - U) + [C(q,\dot{q})\dot{q} + g(q)] \quad (4.13)$$

As it is depicted in 4.7, the nonlinear feedback control law tracks the desired trajectory. This equation can be transformed in to the PD computed torque controller (proportional +derivative feedback) by assigning PD feedback for U. Equation 4.14 illustrates the final equation.

$$\tau = D(q)\left(\ddot{q_d} + K_v\dot{e} + K_p e\right) + [C(q,\dot{q})\dot{q} + g(q)] \qquad (4.14)$$

Moreover, the result in linear dynamic error can be found as 4.15.

$$\left(\ddot{q_d} + K_v\dot{e} + K_p e\right) = 0 \quad (4.15)$$

Based on linear system theory, the isotropic behavior of tracking error to zero is ensured [49][53][54] if the control gains are K_p and K_v. Figure 4.29 is the schematic block diagram of PD- computed torque controller. As shown in the figure, there are two feedback loops, which the inner loop is a compensating loop and the outer loop is called tracking error loop.

As properly shown in figure 4.29, the PD-CTC equation has two parts. According to figure 4.32, the nonlinearity part of the controller can be written as 4.16.

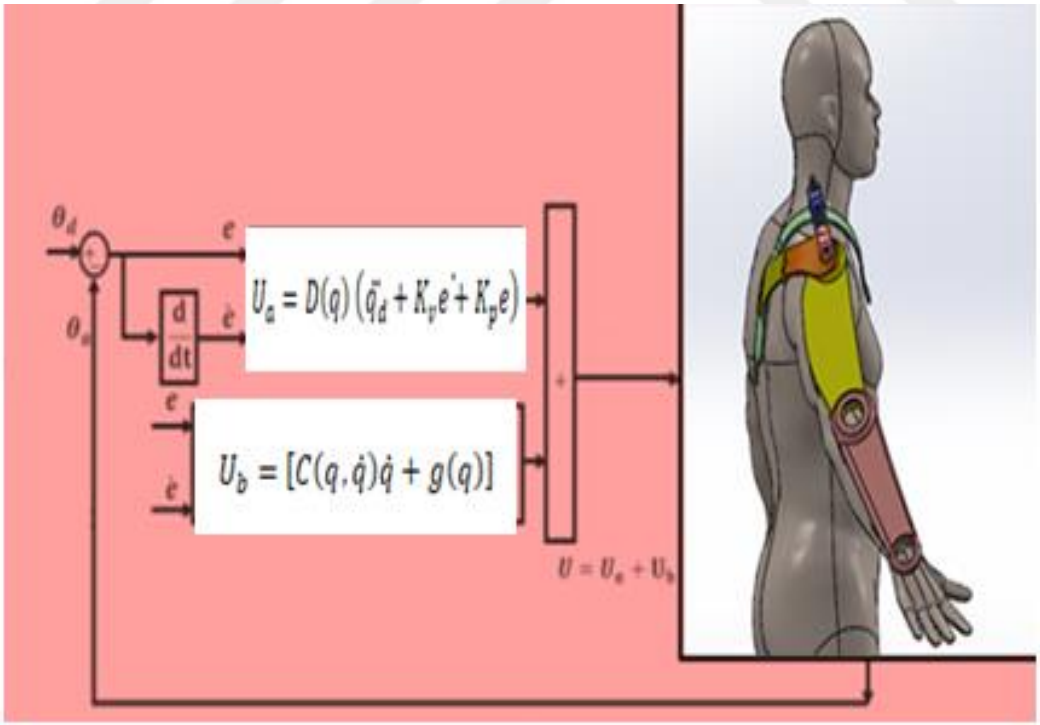

Figure 4.32:PD-CTC Block diagram

$$\begin{bmatrix}\tau 1\\ \tau 2\\ \tau 3\\ \tau 4\end{bmatrix}=\begin{bmatrix}M11 & M12M13 & M14\\ M21 & M22M23 & M24\\ M31 & M32M33 & M34\\ M41 & M42M43 & M44\end{bmatrix}\begin{bmatrix}\ddot{q_{1d}}+K_{v1}\dot{e_1}+K_{p1}e_1\\ \ddot{q_{2d}}+K_{v2}\dot{e_2}+K_{p2}e_2\\ \ddot{q_{3d}}+K_{v3}\dot{e_3}+K_{p3}e_3\\ \ddot{q_{4d}}+K_{v4}\dot{e_2}+K_{p2}e_2\end{bmatrix}+\begin{bmatrix}C1\\ C2\\ C3\\ C4\end{bmatrix}+\begin{bmatrix}G1\\ G2\\ G3\\ G4\end{bmatrix} \quad (4.16)$$

Next step is to design the computed torque controller and find the tracking error. According to the previous controller method, there are 3 types of the response based on the damping ratio. CTC has strong tools for canceling the gravitational, Coriolis and centrifugal forces by the help of feed forward types of controller. If the controller works well, after canceling all the disturbance what is left is the desired motion based on 4.15 equation.

4.3PD Computed Torque Controller Designing

As Claimed by equation 4.15, the PD equation is as$\left(\ddot{q_d}+K_v\dot{e}+K_pe\right)$. The linearized part for one joint can be built as figure 4.33. $K_v$andK_p are the proportional and derivative gains, $\ddot{q_d}$is the double derivative of desired joint variable and $\dot{e}$ is the error rate of the change. The details of implemented PD controller for all of the joints in the robot are available in figure4.34.

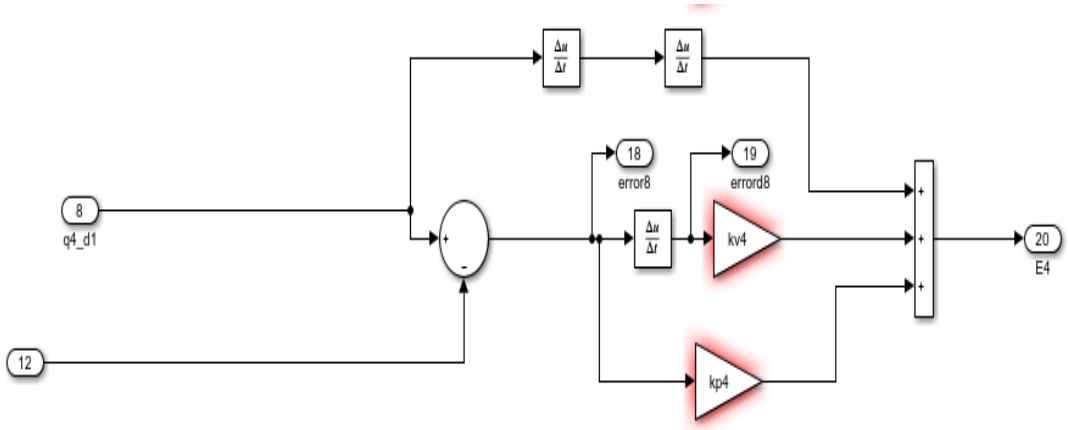

Figure 4.33:PD Computed torque controller for one joint

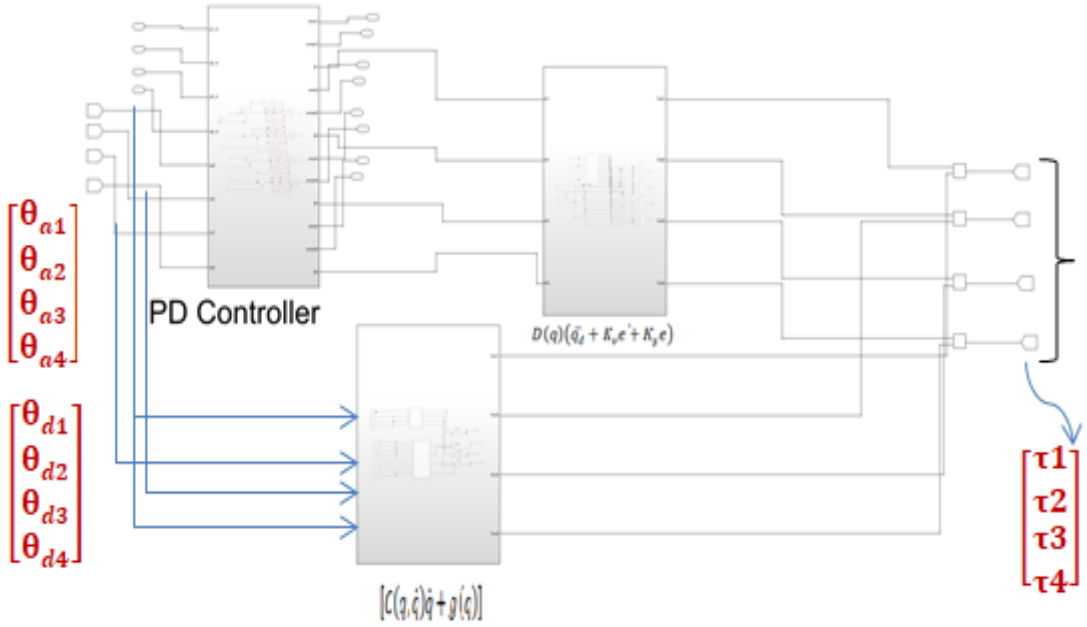

Figure 4.34: The controller for all serial exoskeleton joints

In figure 4.34 the output of the controller is the compensated torque which is going to the dynamics block in figure 4.35. In dynamics section by assist of equation 4.5 the new actual joint properties such as angle and angular velocity are calculated. Note that, finding the inverse of inertia of the matrix is the tough process, which occupies all the RAM and Disk space of the used server or computer. In order to optimize the process, once the inverse of inertia matrix is achieved, the elements of matrix had been put into the *MATLAB s-function level 2* for dynamic model.

Next step is to find out the position of new actual joint variables regarding to desired joint variables. In this, by calling the kinematics of the robot and transformation matrices, calculating the RMS error of detected result will be discussed. Figure 4.36 contains dynamics, controller and kinematics blocks .Note that, this controller is calculating the position not the orientation of the robot end-effector.

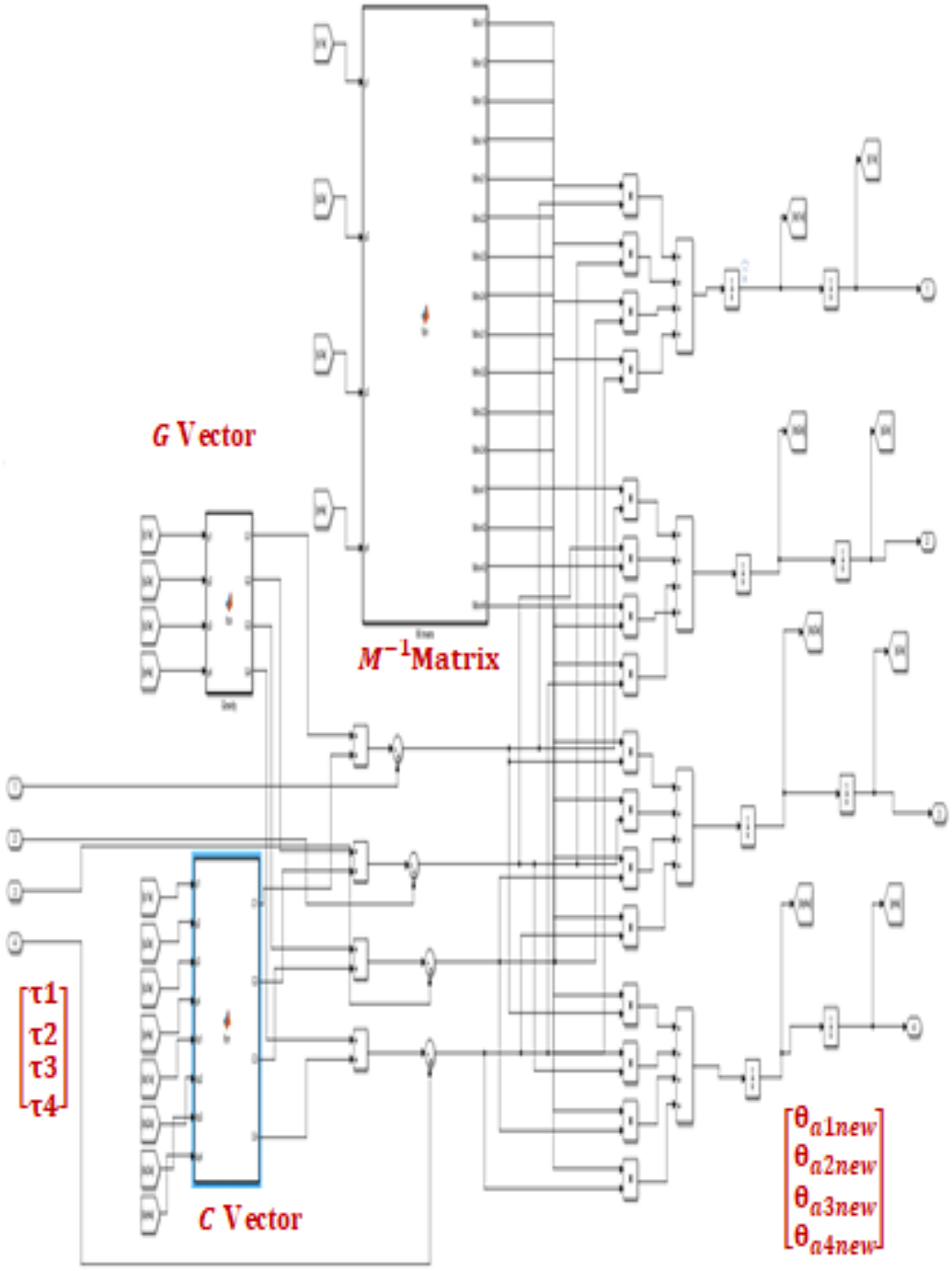

Figure 4.35: The block diagram of dynamic plant

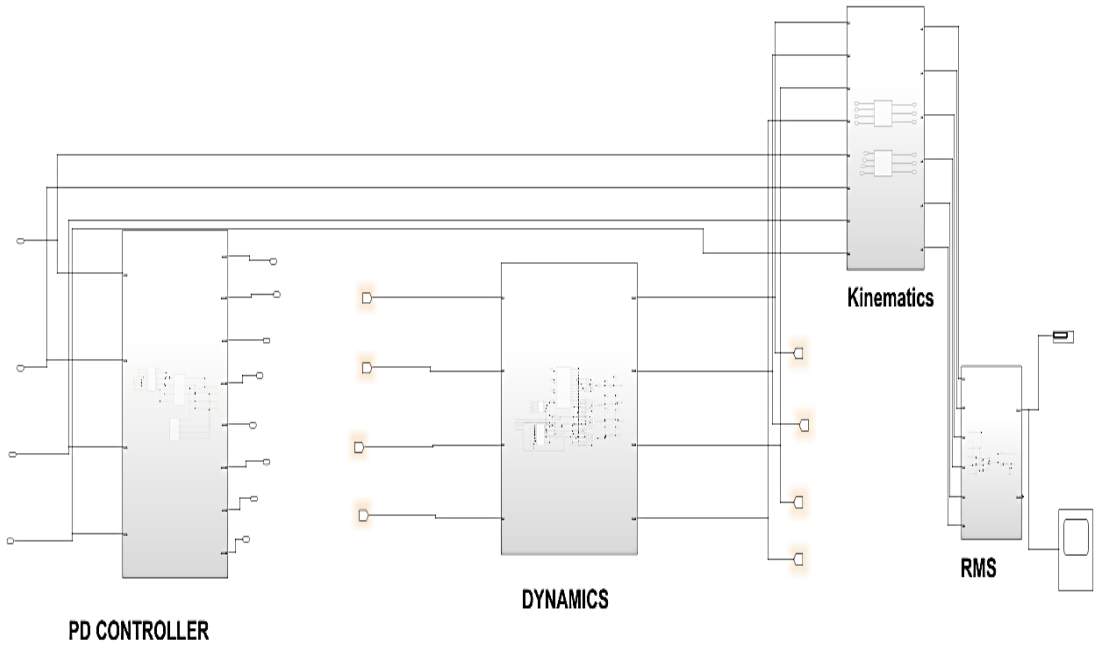

Figure 4.36: Dynamics, controller and kinematics blocks

After implementing the block diagram, the next step is running simulations and finding the best results for coefficients. Following figures are the step response of the system. Figures 4.37 to 4. 38 are about the under damped response of PD CTC conroller to the step inputs.

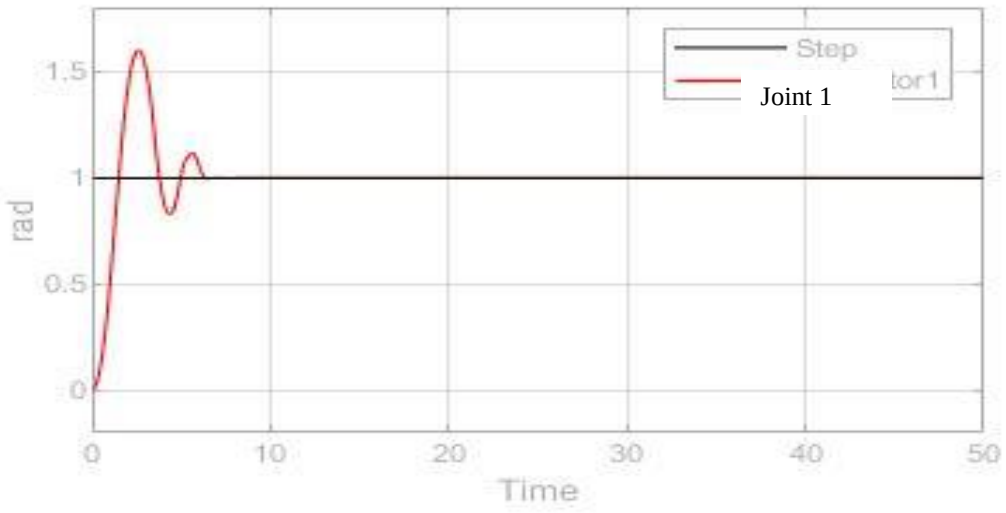

Figure 4.37: Joint 1 “under damped” response of PD-CTC to step input.

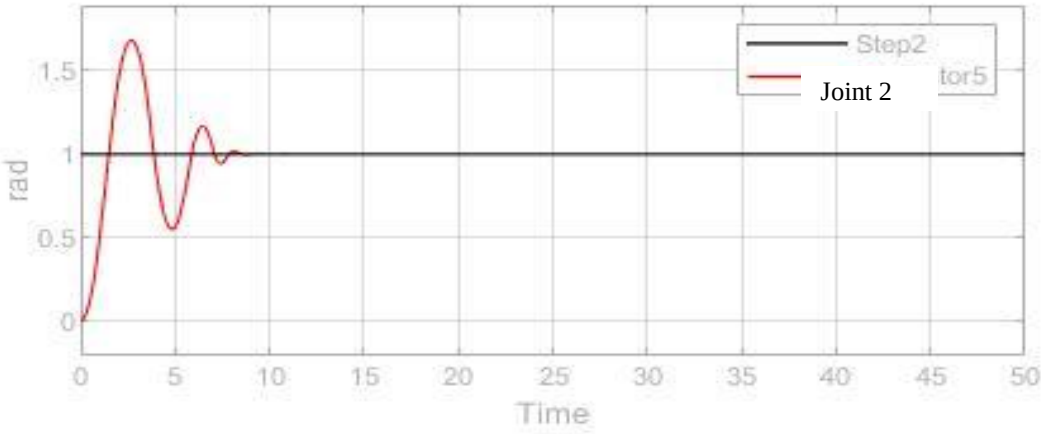

Figure 4.38: Joint 2 "under damped" response of PD-CTC to step input.

Figure 4.39: Joint 3 "under damped" response of PD-CTC to step input.

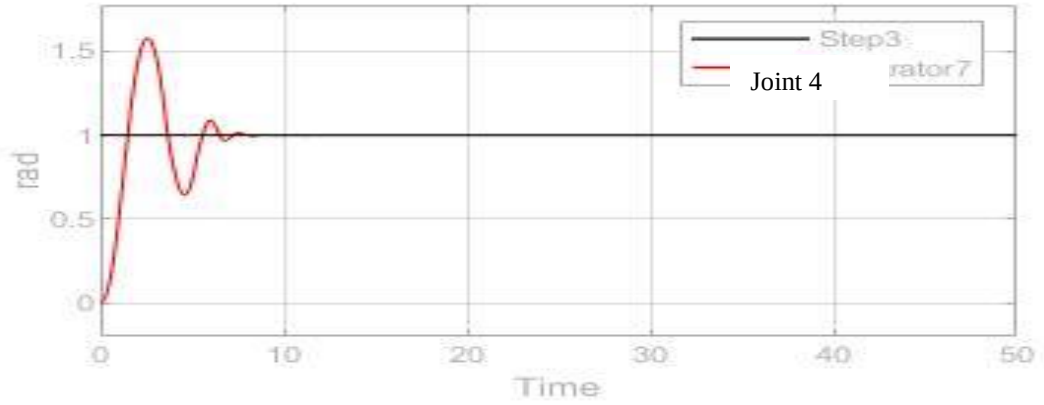

Figure 4.40: Joint 4 "under damped" response of PD-CTC to step input.

The tracking error can be find within figure 4.41 to 4.44.

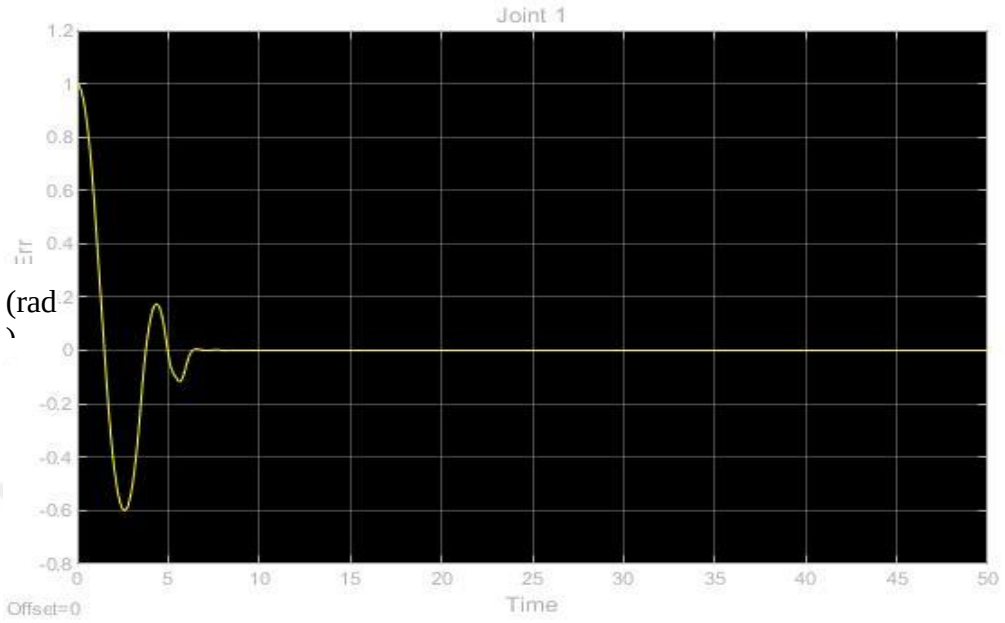

Figure 4.41: Joint 1 tracking error in “under damped” response of PD-CTC to step input

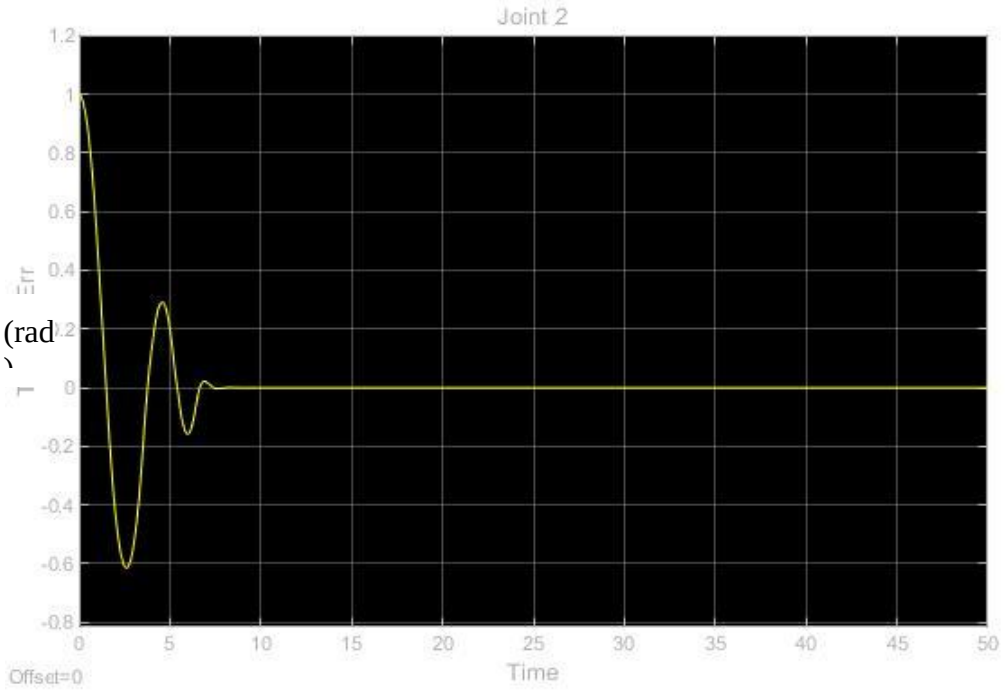

Figure 4.42:Joint2 tracking error “under damped” response of PD-CTC to step input

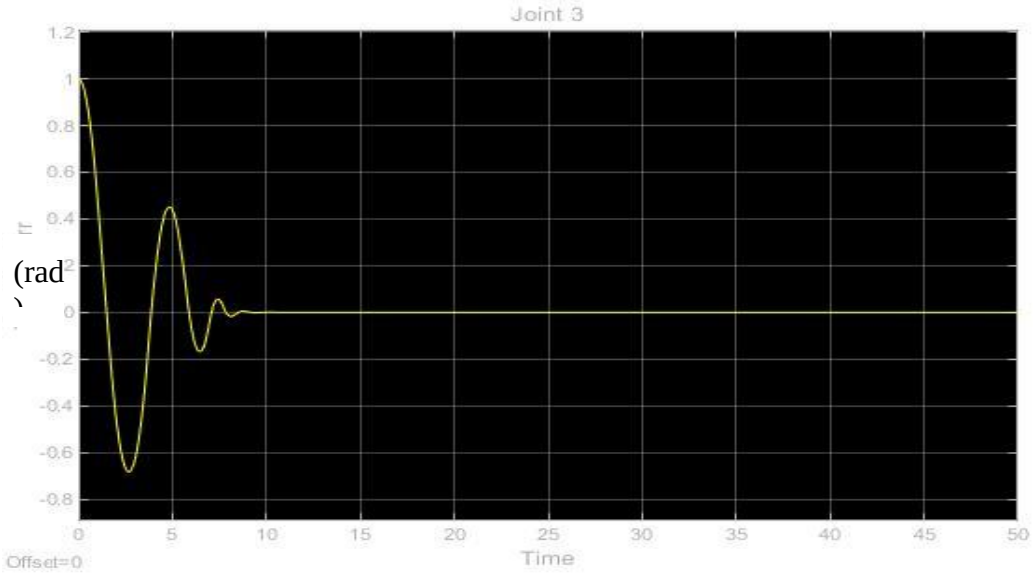

Figure 4.43:Joint 3 tracking error "under damped" response of PD-CTC to step input

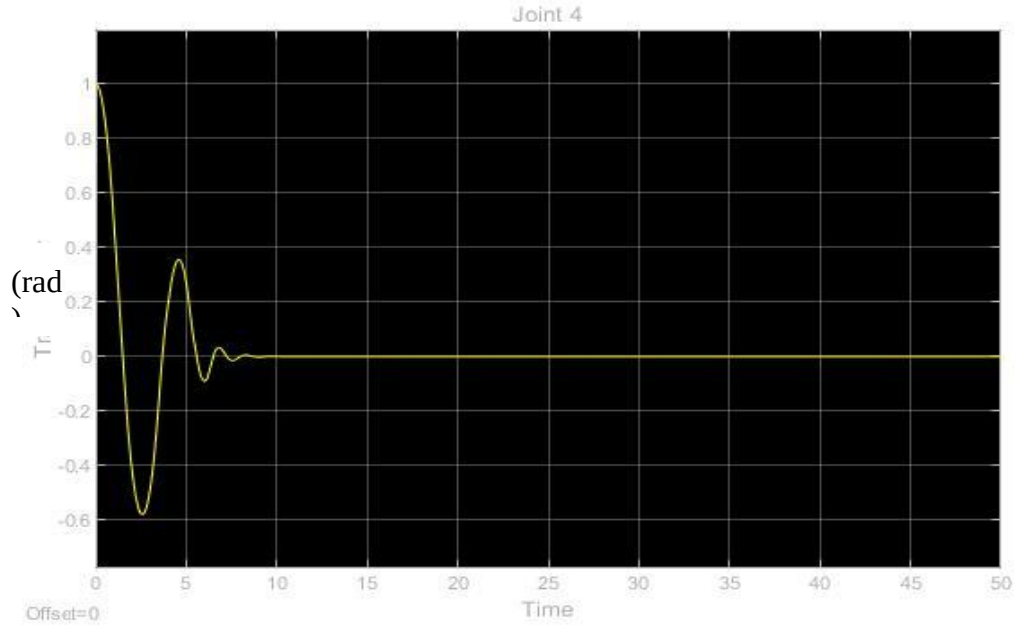

Figure 4.44: Joint 4 tracking error "under damped" response of PD-CTC to step input

In comparison to PID independent joint controller, PD-CTC gives much better responses in shorter period. As the figures are demonstrating, before the 10^{th} second, the actual motion

meets with desired motion. However, of course the range of over shoot is much higher than PID controller is in the SISO system under assumption.

According to what has been stated previously, the most important response for the system is "critically damped" response with wide applications in robotics and notable number of exoskeleton robots. Figures 4.45 to 4.48 are displaying the "critically damped response "of MIMO system with PD-CTC controller.

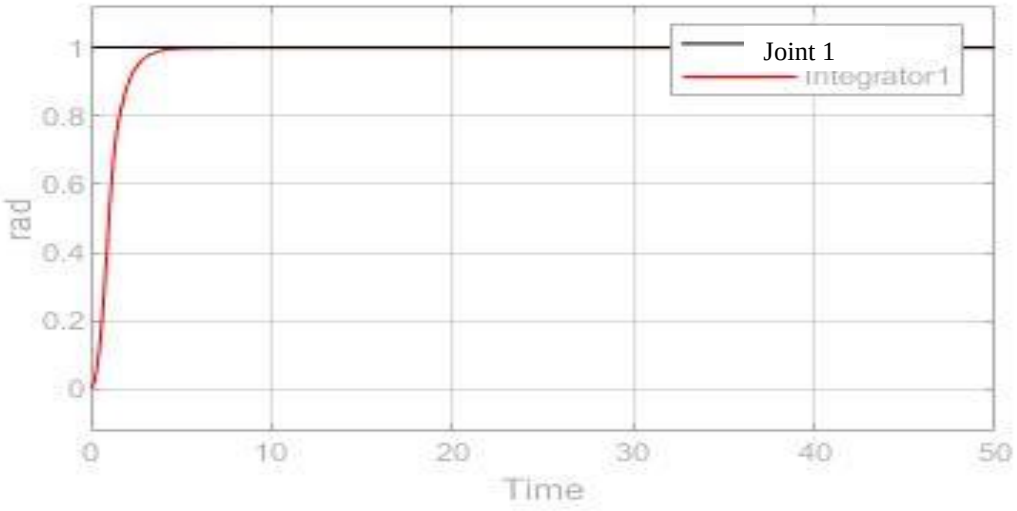

Figure 4.45: Joint 1 "critically damped" response of PD-CTC to step input.

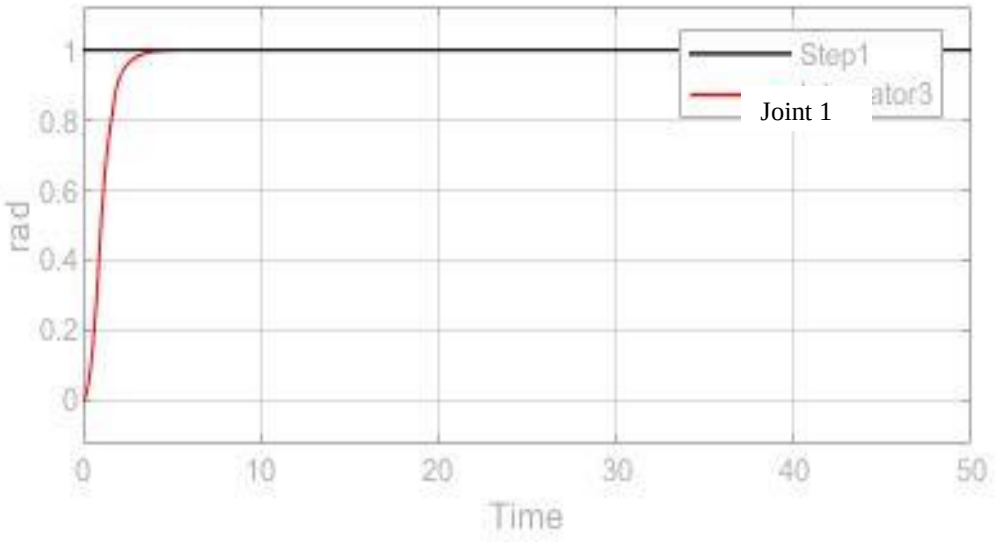

Figure 4.46: Joint 2 "critically damped" response of PD-CTC to step input.

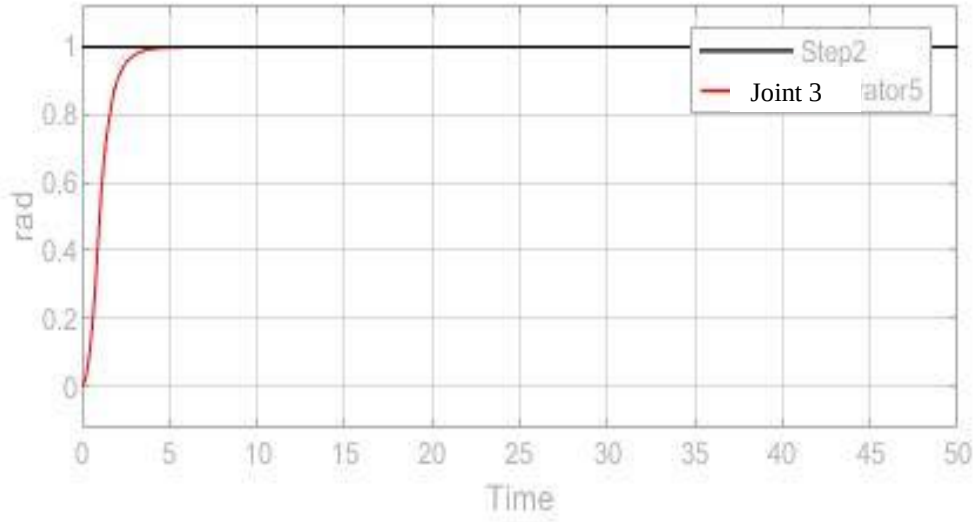

Figure 4.47: Joint 3 "critically damped" response of PD-CTC to step input.

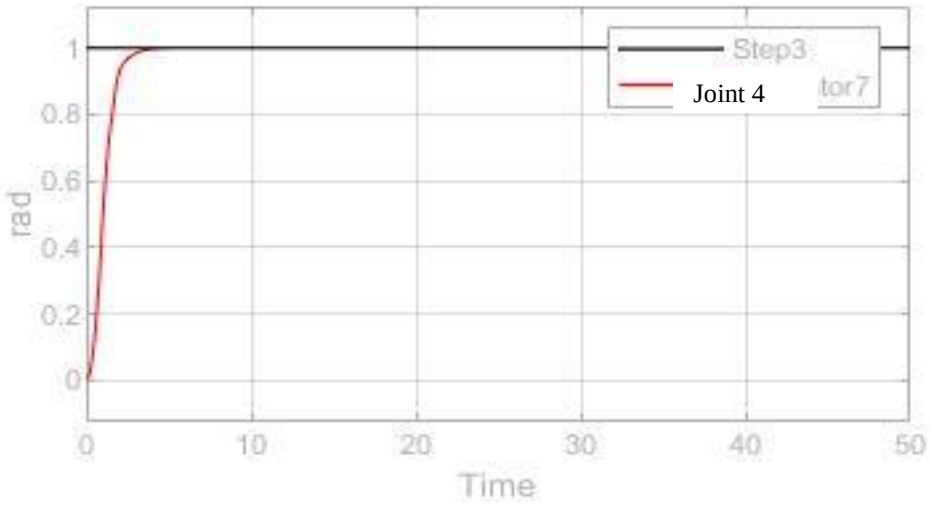

Figure 4.48: Joint 4 "critically damped" response of PD-CTC to step input.

As the figures are representing before 5th second, the actual position meets the desired position which is faster in response than PID independent joint controller. Moreover, one of the important aspects of using CTC is saving time. For the same response in these two

controllers, the time that the system spends for simulating first 50 seconds was around 7 minutes in CTC, while in PID independent joint controller it was around 28 minutes

Figure 4.49 to 4.52 are about the tracking error of PD-CTC.

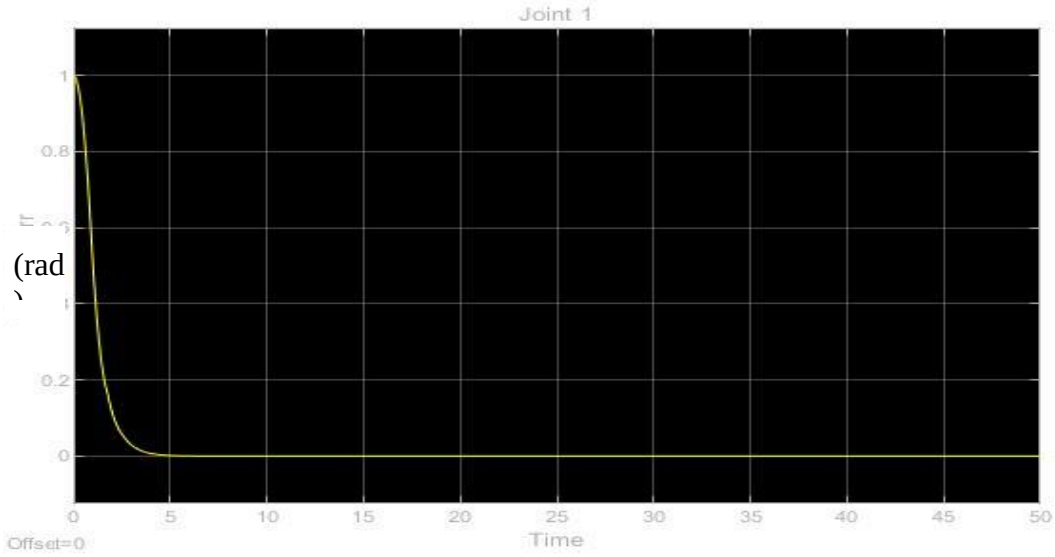

Figure 4.49: Joint 1 tracking error "critically damped" response of PD-CTC to step input

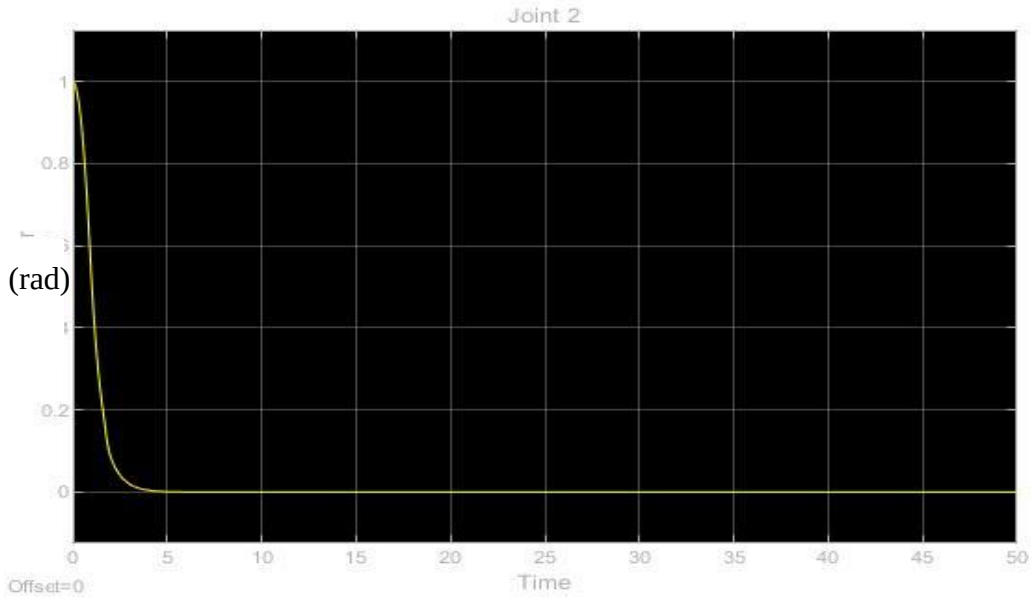

Figure 4.50: Joint 2 tracking error "critically damped" response of PD-CTC to step input

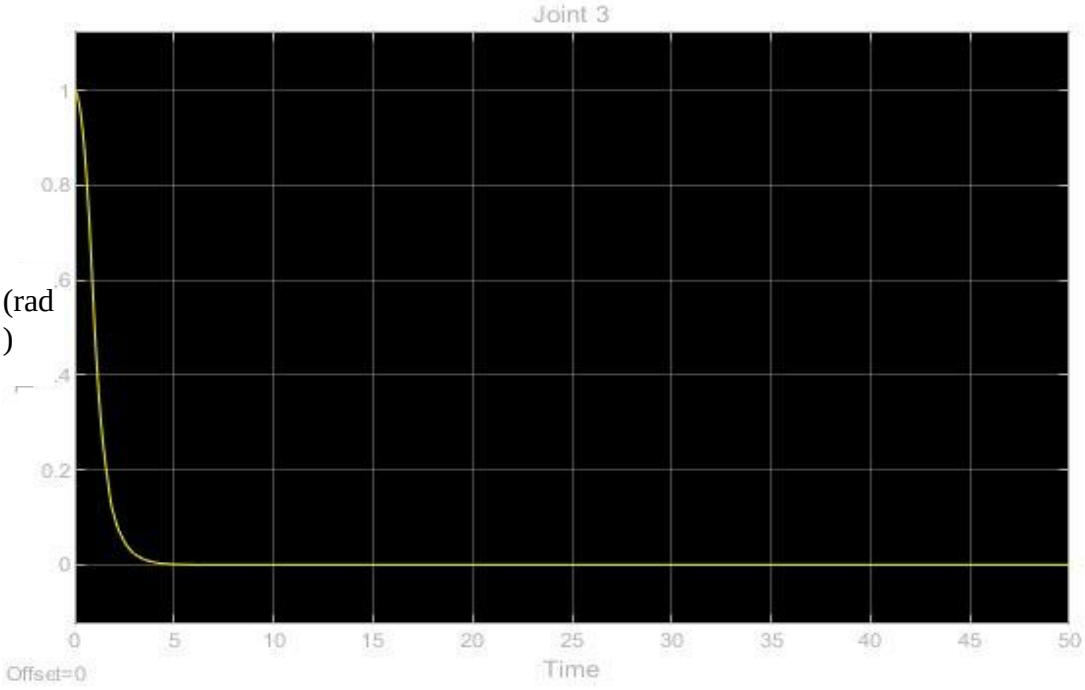

Figure 4.51: Joint 3 tracking error "critically damped" response of PD-CTC to step input

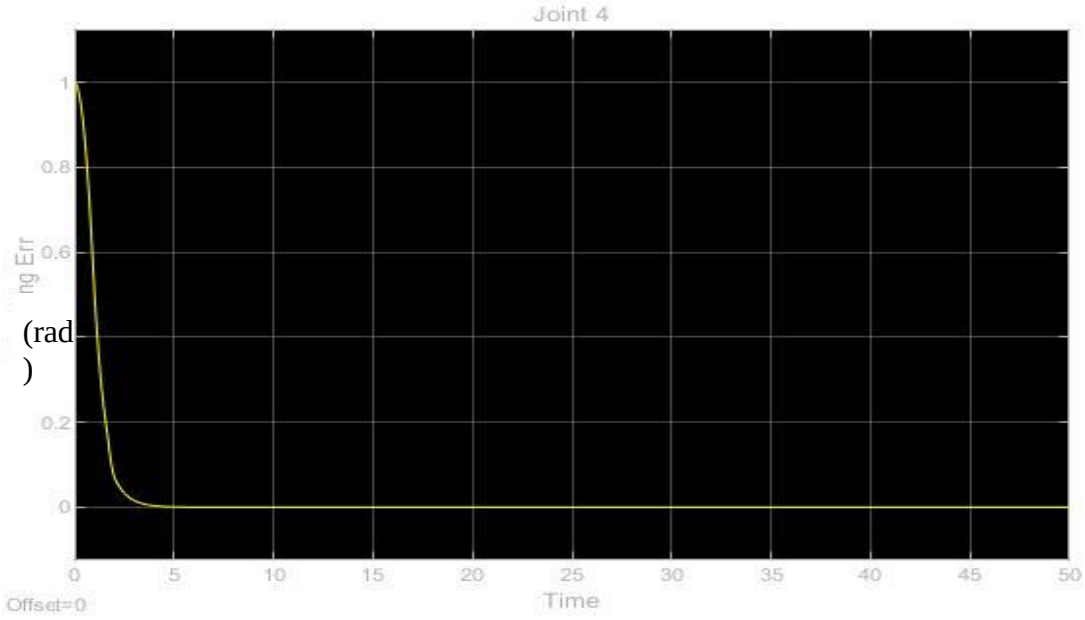

Figure 4.52: Joint 4 tracking error "critically damped" response of PD-CTC to step input

The over damped response of the system with step input can be found in figures4.53 to 4.56:

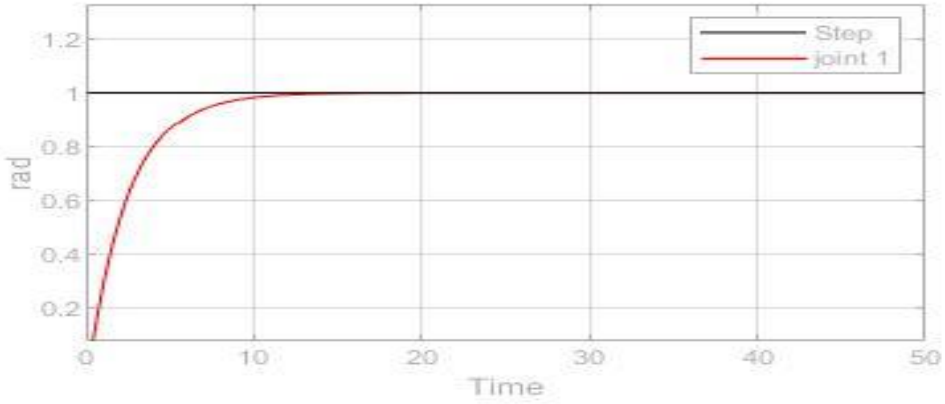

Figure 4.53: Joint 1 "over damped" response of PD-CTC to step input.

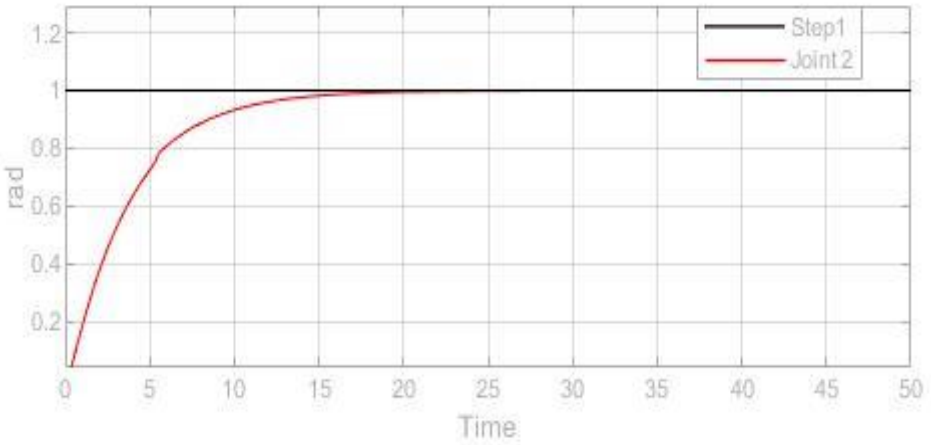

Figure 4.54: Joint 2 "over damped" response of PD-CTC to step input.

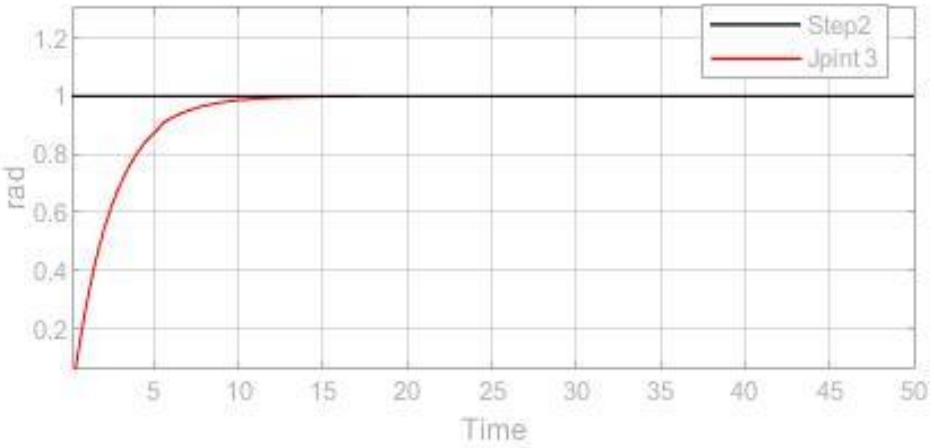

Figure 4.55: Joint 3 "over damped" response of PD-CTC to step input.

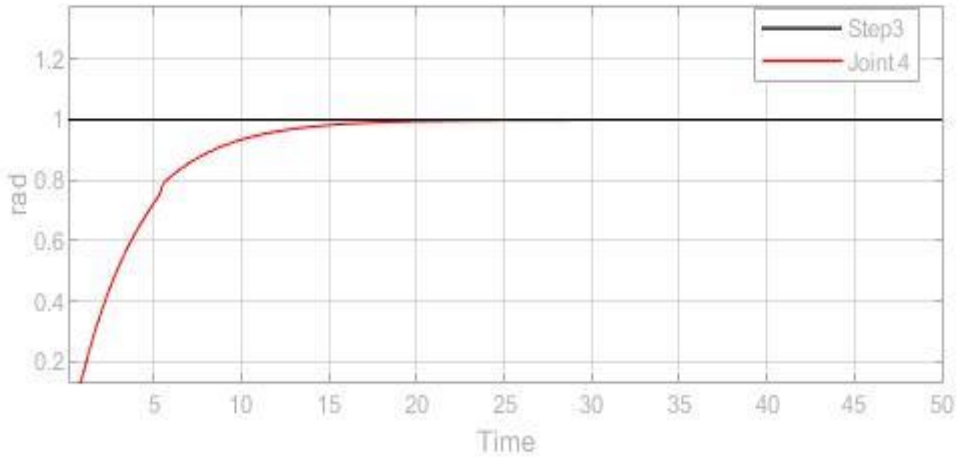

Figure 4.56: Joint 4 "over damped" response of PD-CTC to step input.

As it is clear from the figures, before 20th second, the actual angular position meets the desired angular position. The tracking error of each joint regarding to over damped response can be seen in figures 4.57 to 4.60.

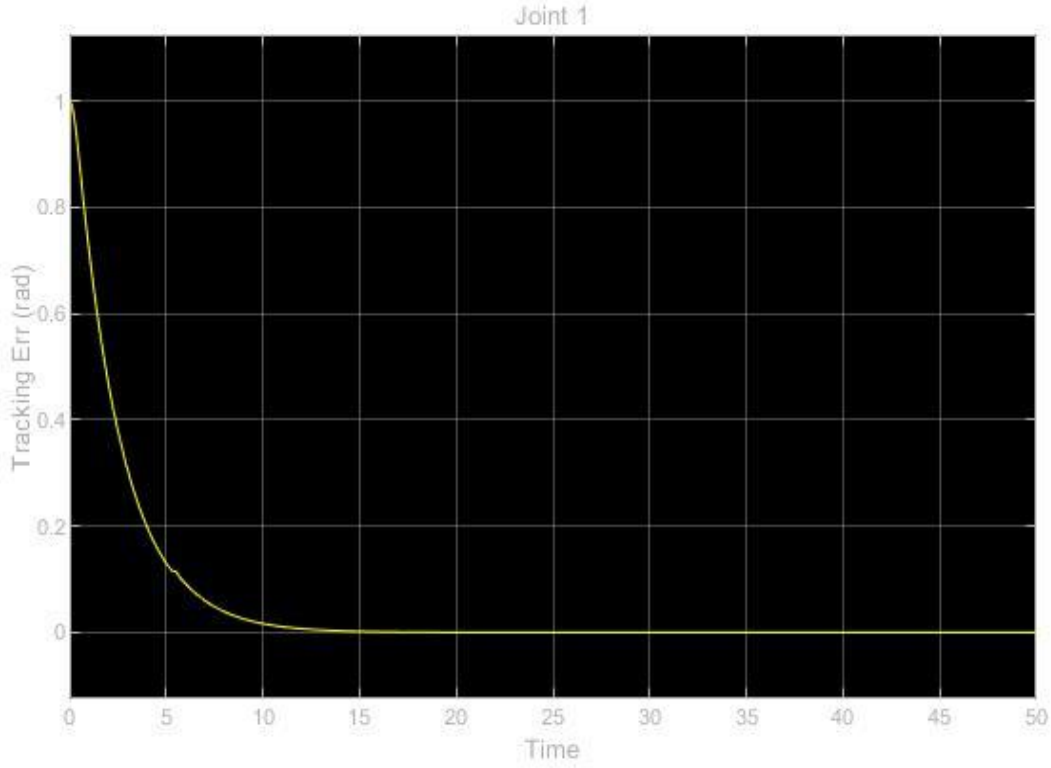

Figure 4.57: Joint 1 tracking error "over damped" response of PD-CTC to step input

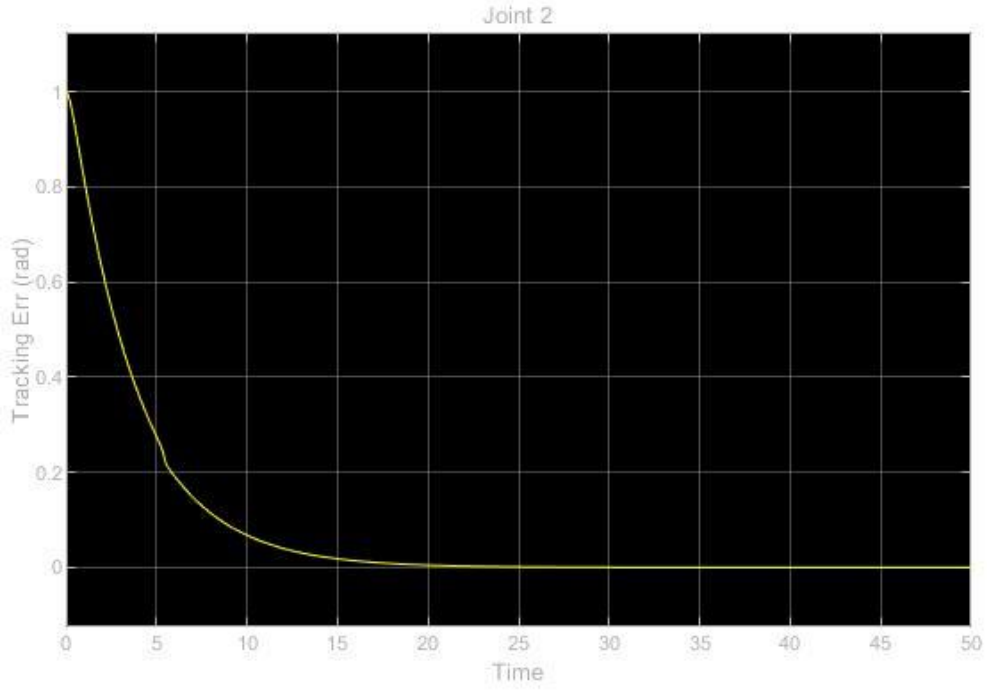

Figure 4.58: Joint 2 tracking error "over damped" response of PD-CTC to step input

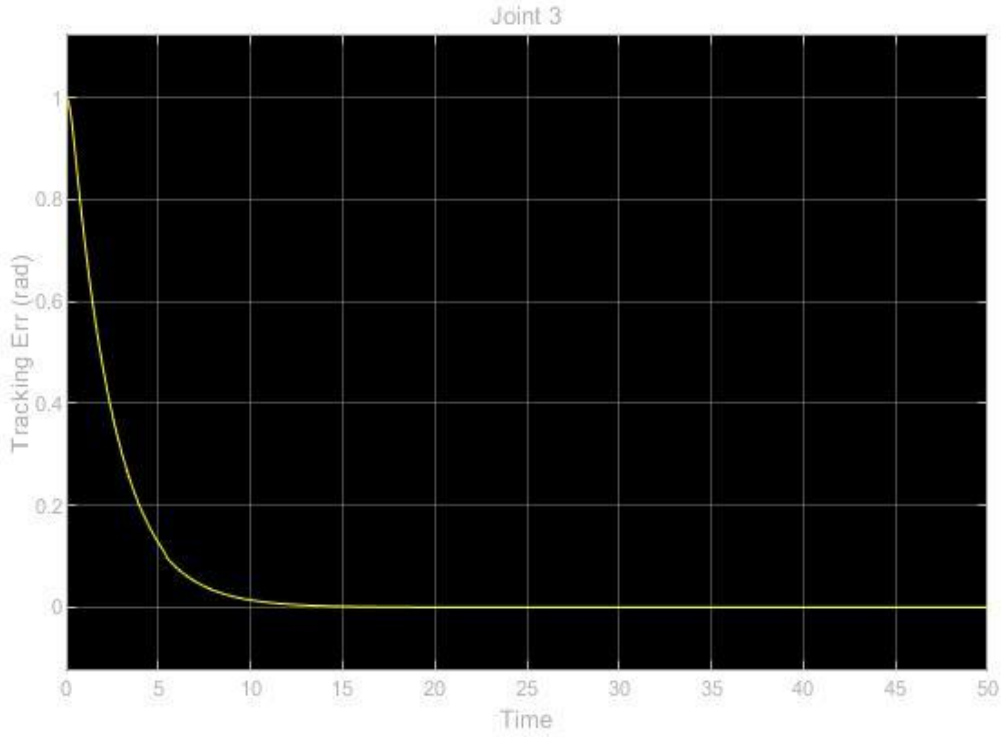

Figure 4.59: Joint 3 tracking error "over damped" response of PD-CTC to step input

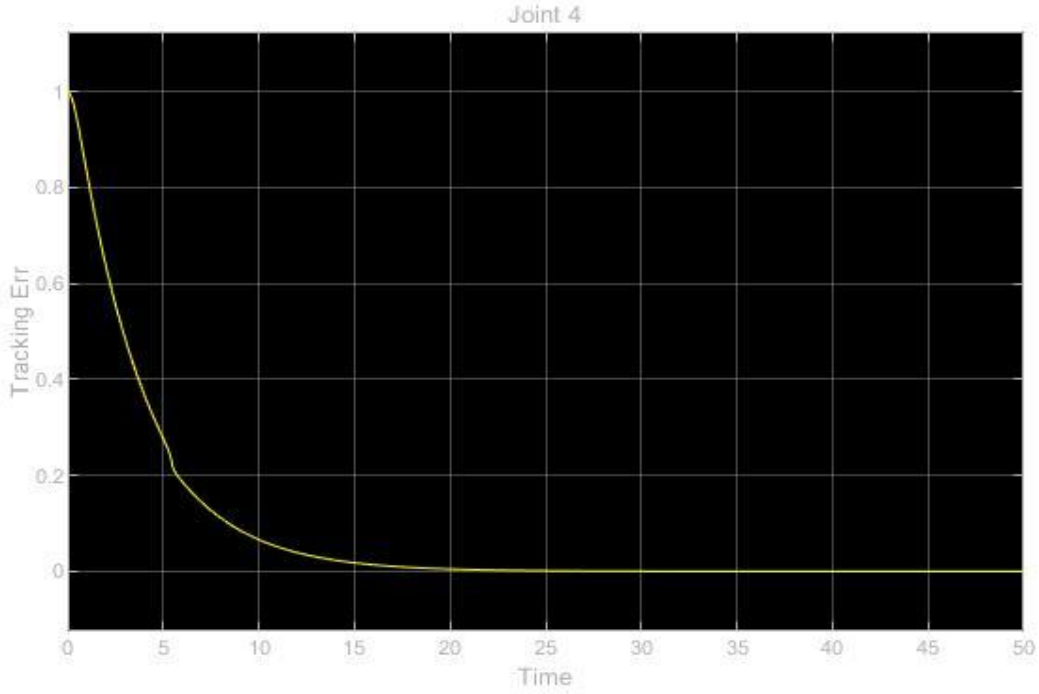

Figure 4.60: Joint 4 tracking error "over damped" response of PD-CTC to step input

As discussed before, the main purpose of using this controller was due to sinusoidal inputs to the system. Figures 4.61 to 4.64 are displaying the response of the system to the sinusoidal inputs in critical damped response in the first 50 seconds. Not that, the frequency and amplitude of sinusoidal inputs are based on what has been achieved in chapter 3.

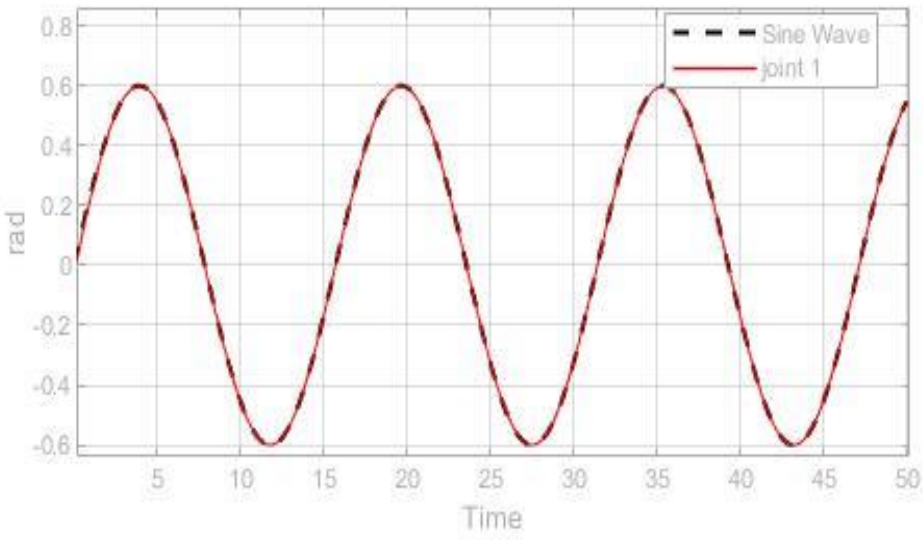

Figure 4.61: Joint 1"critically damped" response of PD-CTC to sinusoidal input.

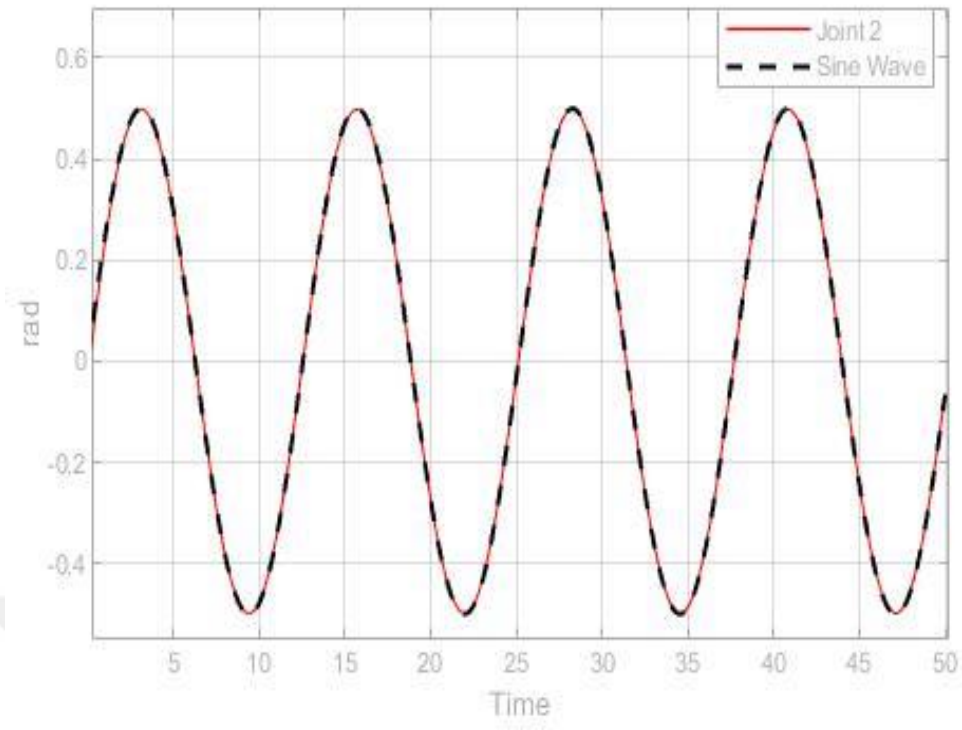

Figure 4.62: Joint 2 “critically damped” response of PD-CTC to sinusoidal input.

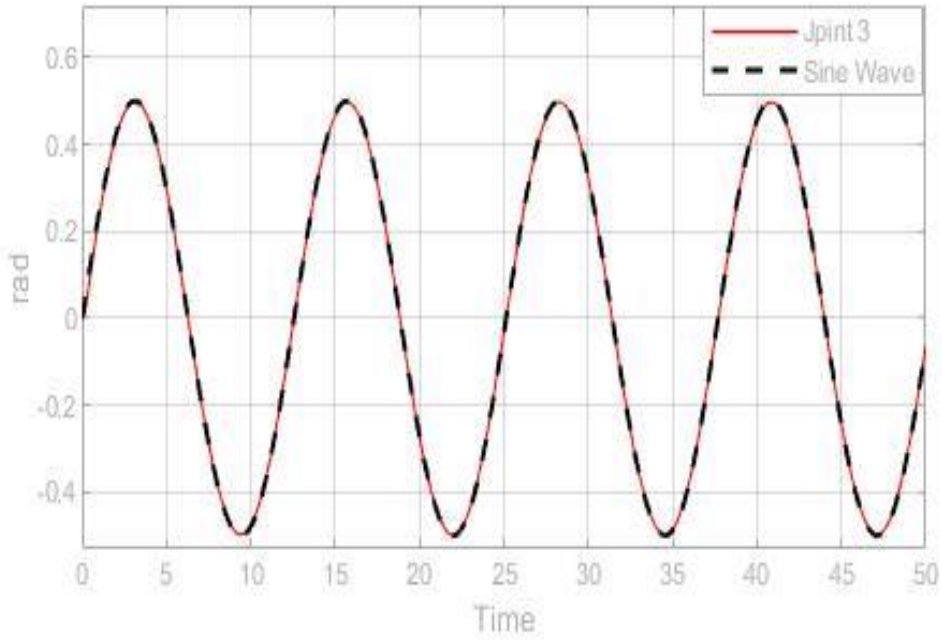

Figure 4.63: Joint 3 “critically damped” response of PD-CTC to sinusoidal input.

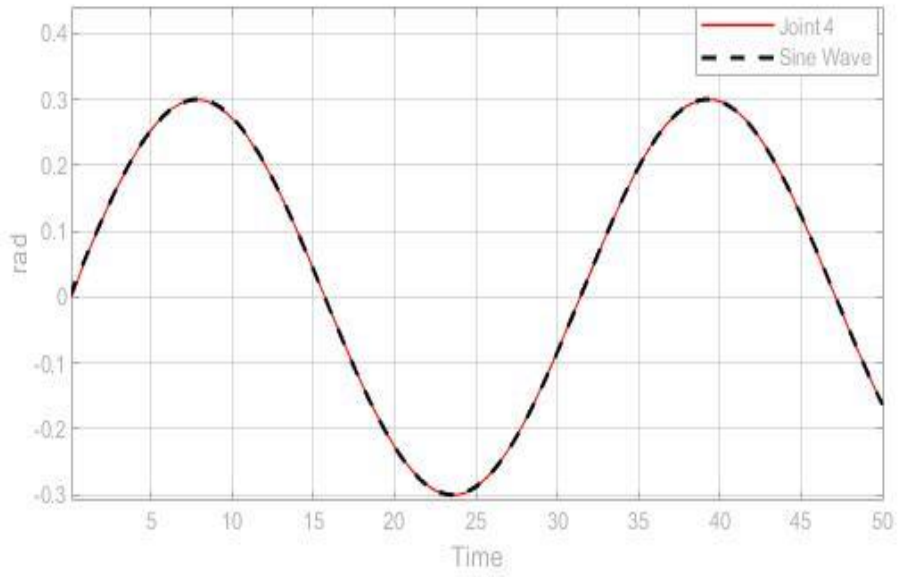

Figure 4.64: Joint 4 "critically damped" response of PD-CTC to sinusoidal input.

As it can be observed the CTC-PD controller works great in MIMO system and with sinusoidal inputs. Figures 4.65 to 4.68 are about the tracking error.

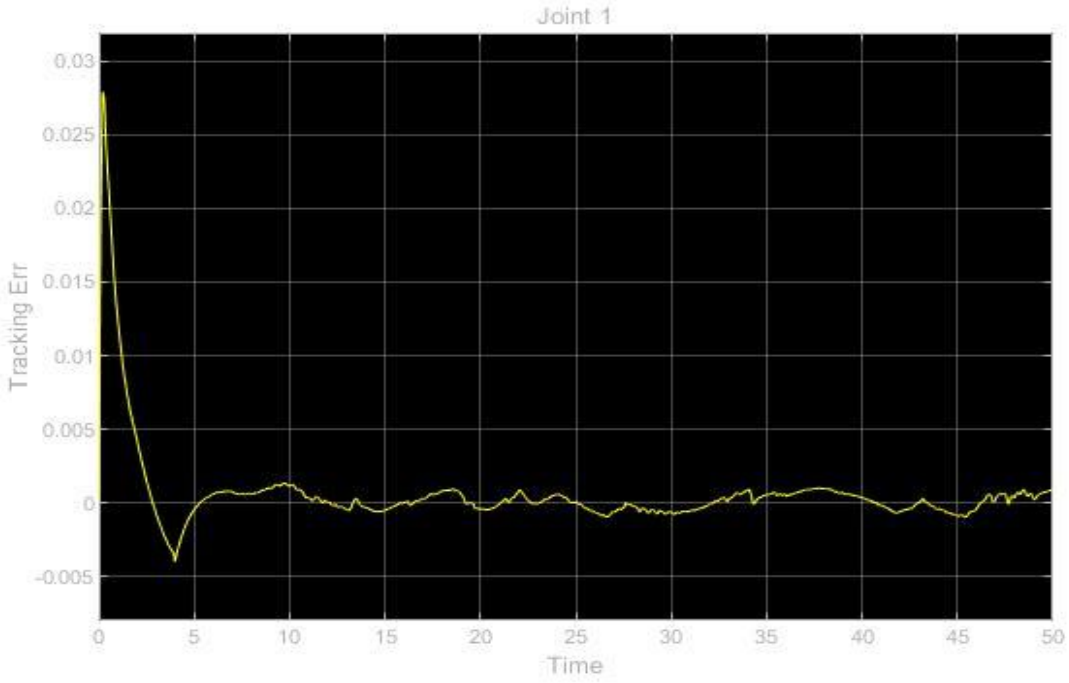

Figure 4.65: Joint 1 tracking error "critically damped" response of PD-CTC to sinusoidal input

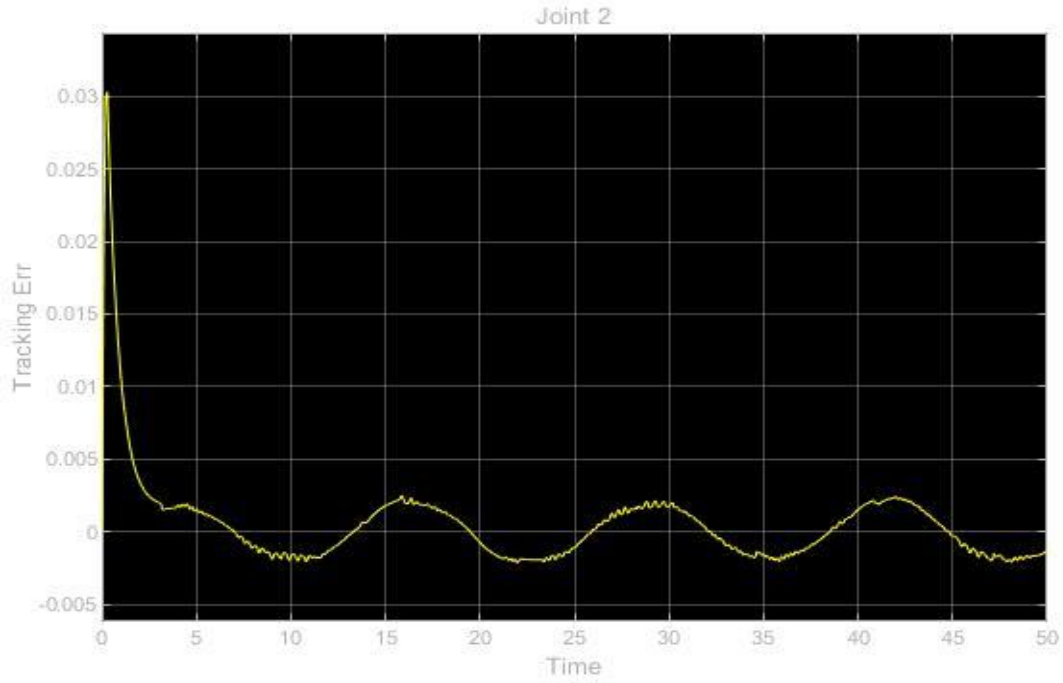

Figure 4.66: Joint 2 tracking error "critically damped" response of PD-CTC to sinusoidal input

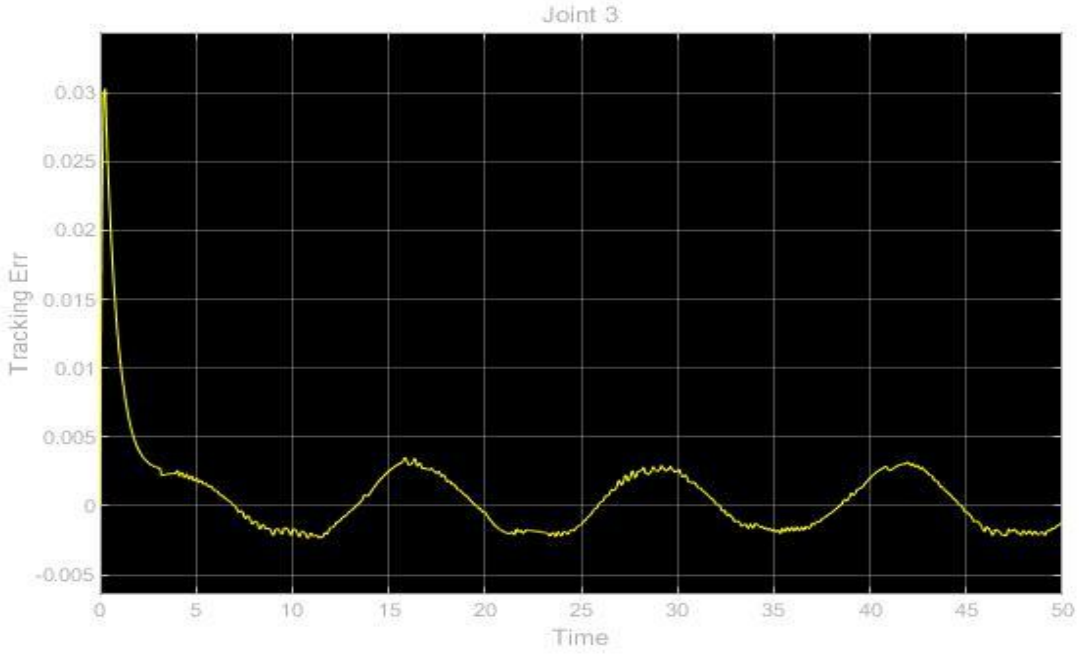

Figure 4.67: Joint 3 tracking error "critically damped" response of PD-CTC to sinusoidal input

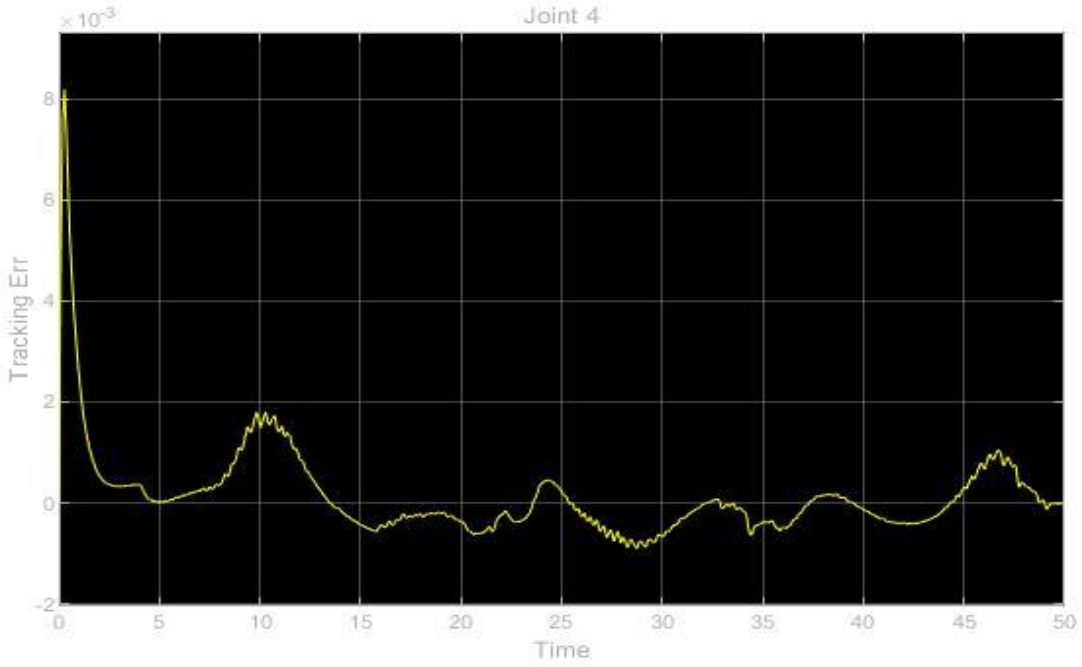

Figure 4.68: Joint 4 tracking error "critically damped" response of PD-CTC to sinusoidal input

The actual and desired position of end-effector in space can be find in figure 4.69

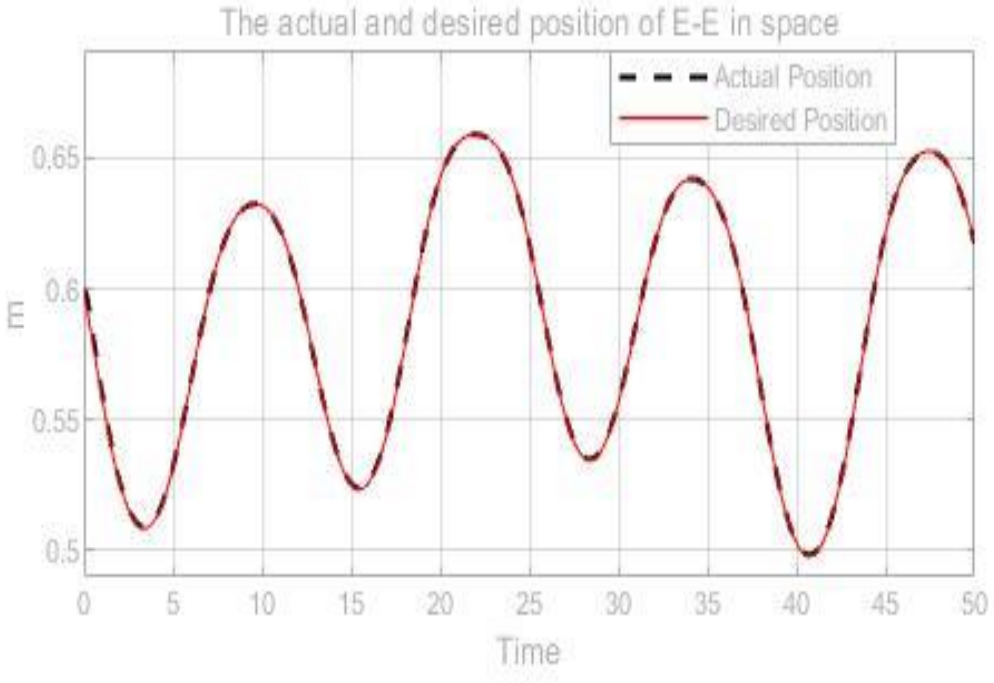

Figure 4.69: The actual position and desired position of end-effector in the space.

The RMS error between desired position of end effector and the actual position of end effector can be observed in figure 4.70.

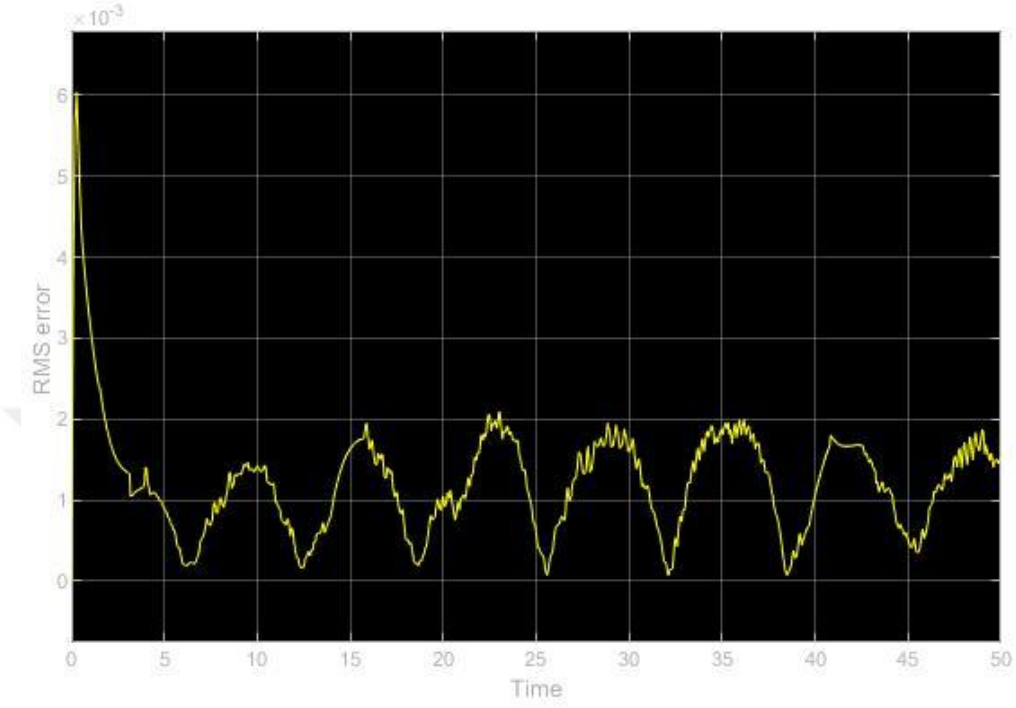

Figure 4.70: The RMS error between actual and desired position of end-effector in the space.

As it is clear from the tracking error results, after reaching to the steady state, the result include small amounts of error that can be seen in the figures which is less than 0.003 radians. Also the RMS error at the maximum value which is the first condition is around 0.006 and after the steady state, the maximum RMS is less than 0.0002.

According to the results, PD-CTC is the controller that can work in best shape with the nonlinearities in the system and sinusoidal inputs in the optimized time and best with very small tracking error. However, it is very important to consider that, the controller is working very well due to the recognitions of all parameters and variables. This means that in reality and under the circumstances with uncertainties, which are added to the system, the performance surely decreases. For this situation, it is reasonable to utilize PID-CTC to compensate the uncertainties in the system. In addition, in real world, the uncertainties with exoskeleton can be over and lower than normal arm weight. Before this, all the studies were

done without considering human arm weights and just robot links weights. Therefore, it is essential to predict the behavior of the system in real environment with uncertainties.

4.3PID controller and PID computed torque controller with human arm weight

As it was mentioned previously, PID-CTC is reliable for rejecting the disturbances and compensating the system under the situation with uncertainties. In this particular part of study it is important to understand the effect of overweight arm and lower weight arm on the system. It has been assumed that the exoskeleton is flexible with any arm length. Therefore, in order to start the study, this study is begun by applying some different forearm and upper arm weights through the device.

In many studies it have been indicated that normally the distribution of women weight in arms is 4.9% of total weight and in men is 5.7% of total human weight[55][56].

By considering this information and by assuming the impact of human arm weight on PID-CTC, the performance will is investigated. Firstly, by applying the step input in to the system and later the sinusoidal input, the rate of system compensation will be explored.

One of the important aspects of this issue is to find the best response to the disturbance. This response is usually considered as under damped types of response, first because the system is facing the disturbance and second, over damped and under damped response cannot reach the system to the desired steady state and system can be seen as unstable. These are means that there is essential requirement of choosing parameters for controller for all joints. The reason is that in MIMO system, moving or oscillating attached link to the joint can highly affect the other joints and links movements. Another issue is to consider the fact that all the simulations are done without considering the motors into the system. The type of motor that has been assumed for this exoskeleton is pancake motors. These small motors with high peak torque with an efficient size and dimensions are good choices for the system. Table 4.1 is showing the assumed motor characteristics.

Table 4.1 : Assumed pancake motor characteristics for all joints

Moment of Inertia(Kg.cm2)	Viscous Damping Constant (N..m/KRPM)	Torque Constant (N.m/Amp)	Back EMF Constant (V/kRPM)	Armature Resistance (Ra)	Armature Inductance (μH)
0.39	0.005	0.0477	5	0.719	<0.03

4.3.1 The PID controller with exoskeleton

PID controller is known for its great response to the step input. Adding disturbance to the system will let the SISO system become as following in figure 4.71 whichis about the system with PID controller and disturbance.

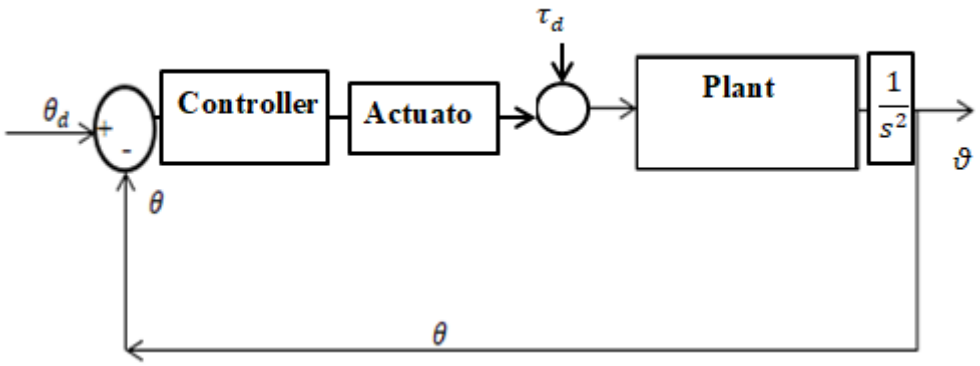

Figure 4.71: The system with PID.

As it was mentioned the disturbance, τ_d is in the form of torque that is coming from the gravity. For this system, it has been assumed that the woman who wears the device is 58 Kg and the man is 73 Kg. Figures 4.72 to 4.75 are about the assumed woman with 58 Kg who wears the device and the device is compensating the human weight and lifts it up to the desired position.

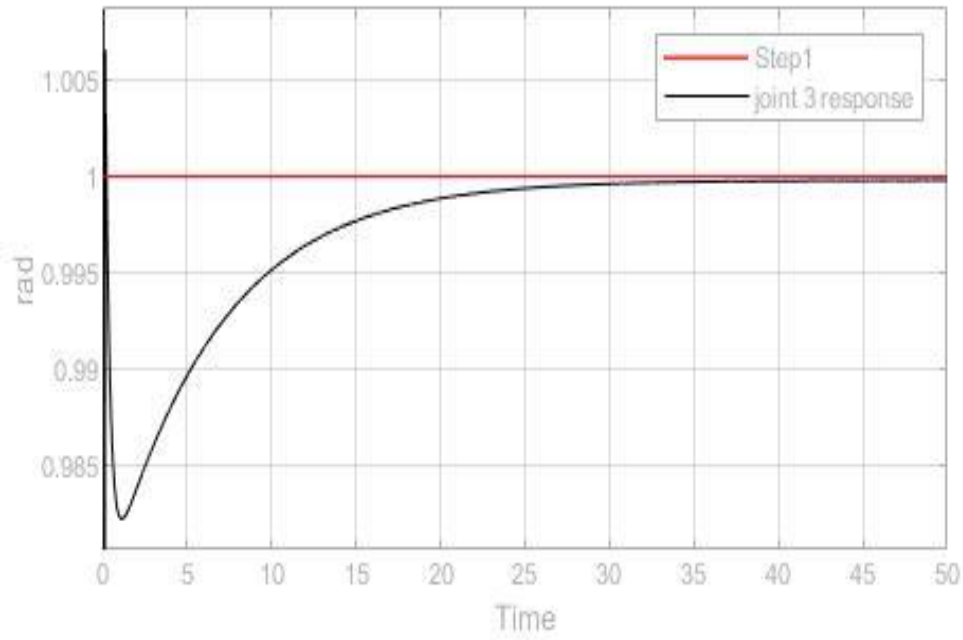

Figure 4.72: The effect of PID controller on the 1st joint with that wearied by a woman with 58 Kg.

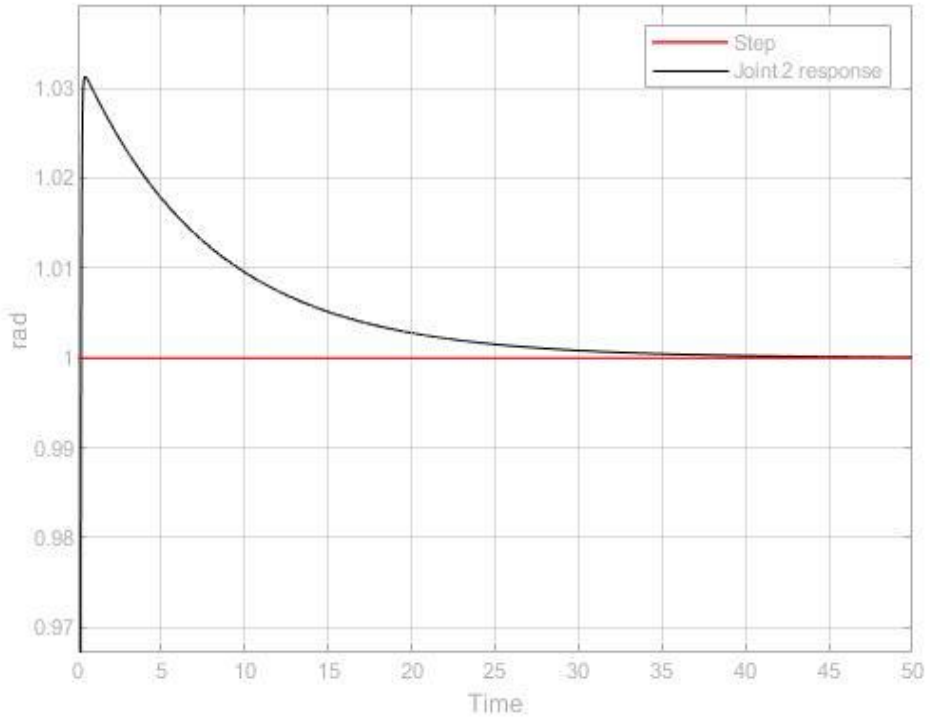

Figure 4.73: The effect of PID controller on the 2nd joint with that wearied by a woman with 58 Kg.

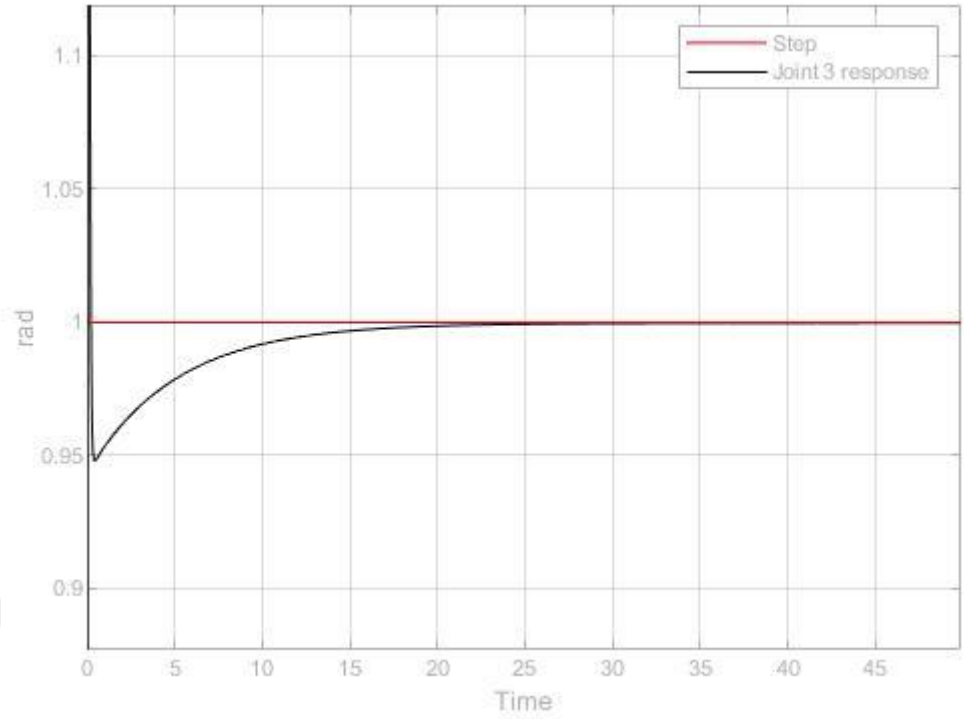

Figure 4.74: The effect of PID controller on the 3rd joint with that wearied by a woman with 58 Kg.

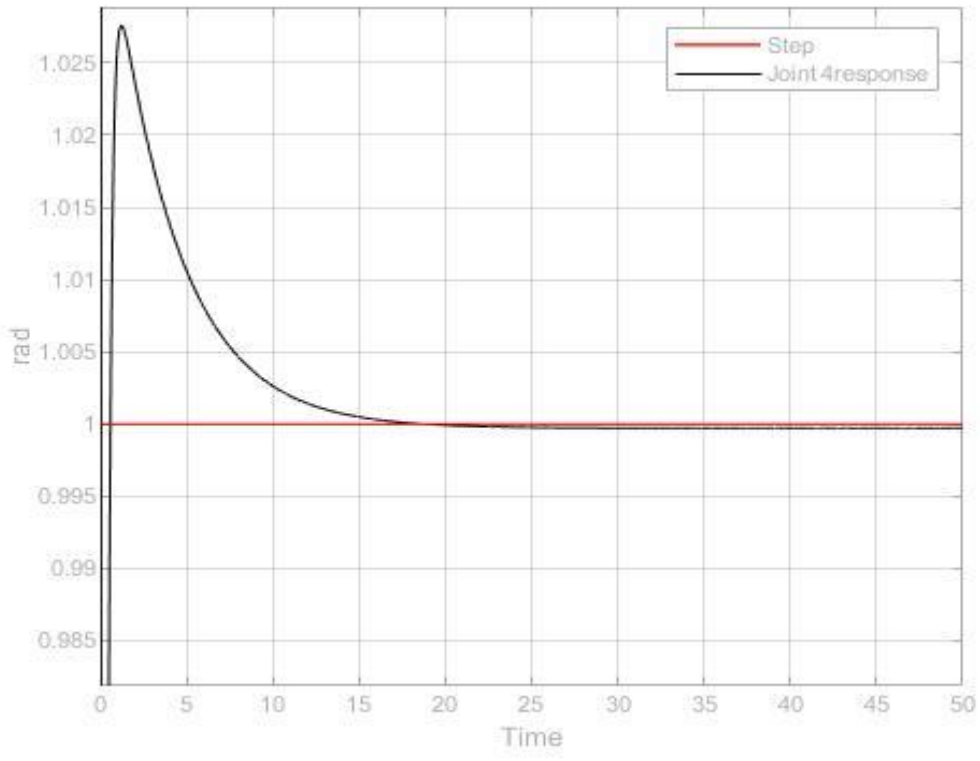

Figure 4.75: The effect of PID controller on the 4rd joint with that wearied by a woman with 58 Kg.

As it is clear, the joint 1 and 2 are mostly affected by this disturbance. This issue can be observed from the figures as well, that in the very first seconds of the simulation, the joints are trying to compensate these disturbances. In joint 2 and joint 4 which were explained before, the joints are less affected by gravity. Figures 4.76 to 4.79 are about the tracking error for all joints.

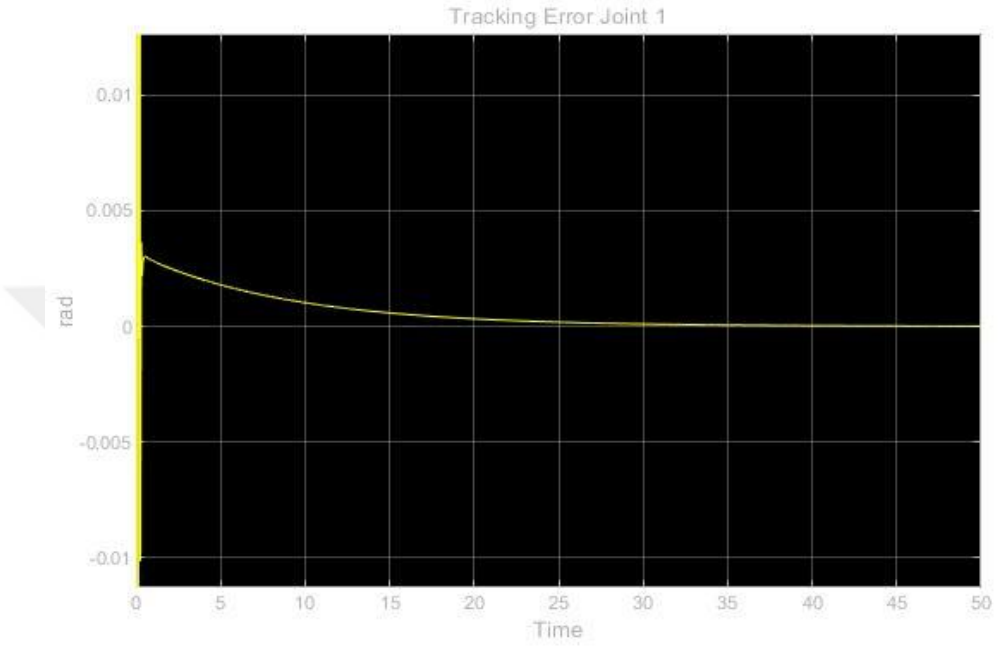

Figure 4.76: The 1st joint tracking error with for a woman with 58 Kg.

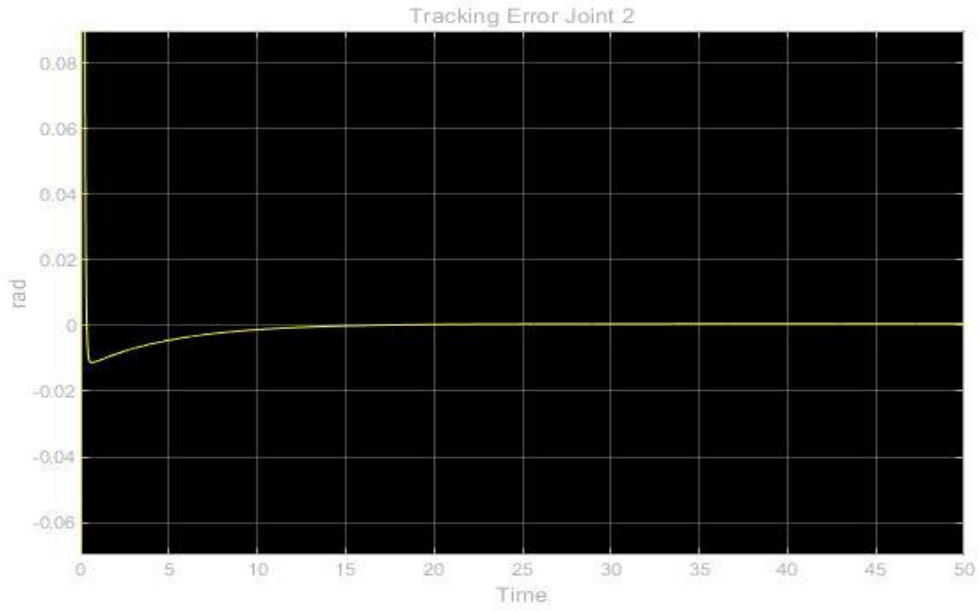

Figure 4.77: The 2nd joint tracking error with for a woman with 58 Kg.

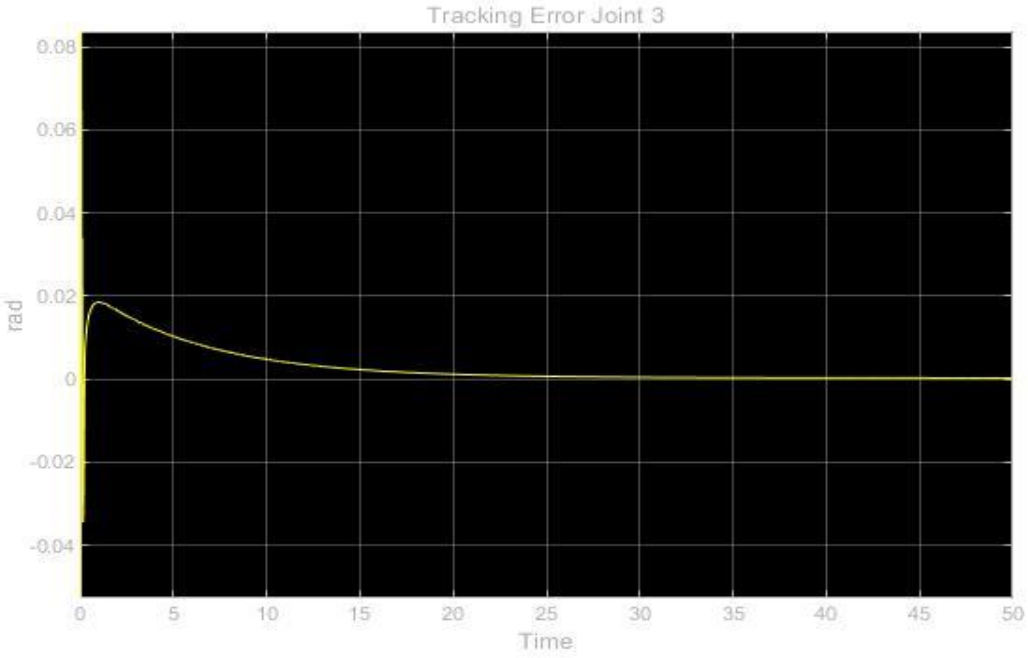

Figure 4.78: The 3rd joint tracking error with for a woman with 58 Kg.

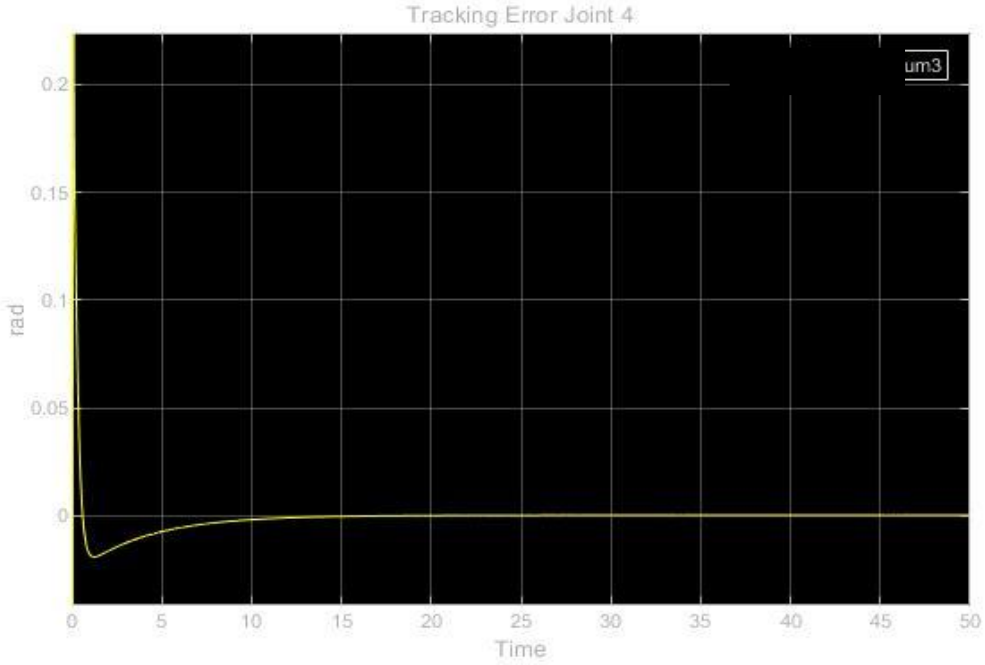

Figure 4.79: The 4th joint tracking error with for a woman with 58 Kg.

As the results are showing, the system and controller can compensate the gravity torque and lift the arm through the desired position in a good manner. There would be very small amount of tracking error for joint 3, but it is very small and negligible. Joint 3 has critical positions in the system from the aspect of applied torque from gravity and the major of arm weight to be

tolerated. Therefore, this joint should be studied carefully. It is also important to find the behavior of the system for the assumed man which is 73 Kg. Figures 4.80 to 4.83 are showing the response of the system to the disturbance which is coming from the assumed man arm weight, wearing the exoskeleton.

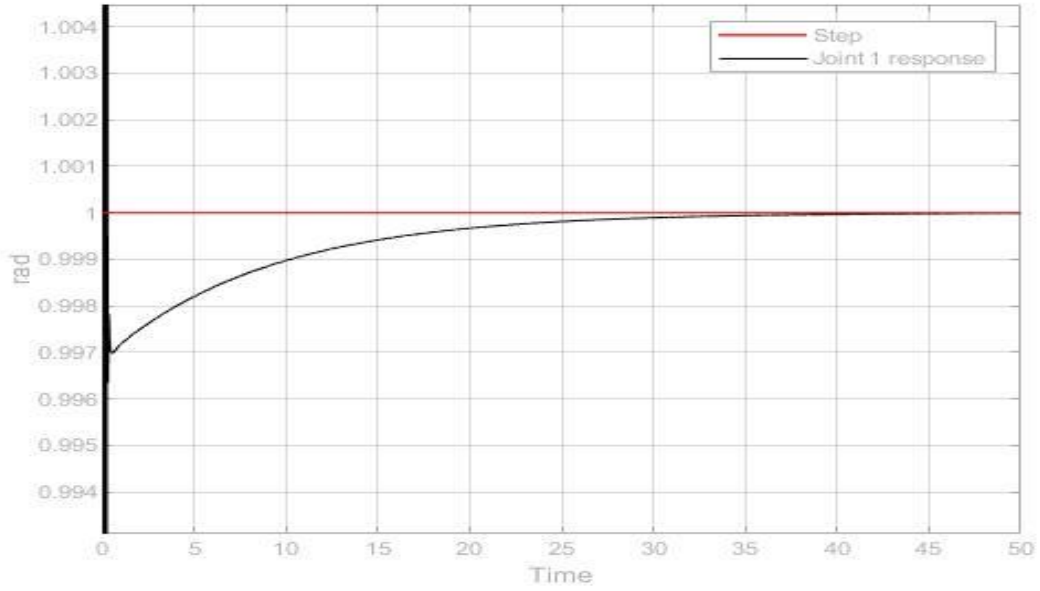

Figure 4.80: The effect of PID controller on the 1st joint with that wearied by a man with 73 Kg.

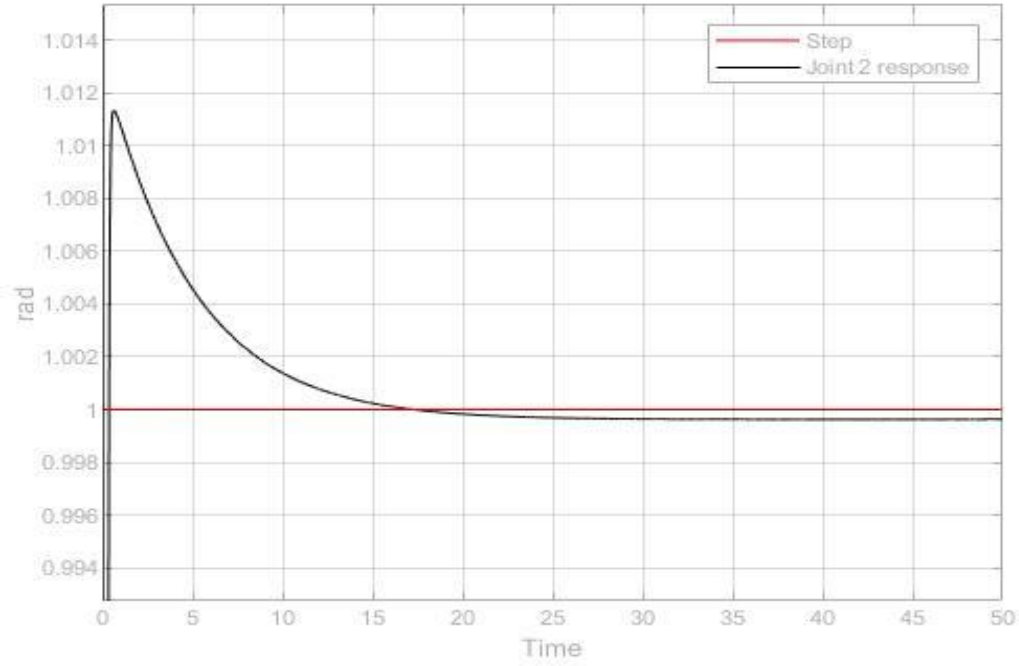

Figure 4.81: The effect of PID controller on the 2nd joint with that wearied by a man with 73 Kg.

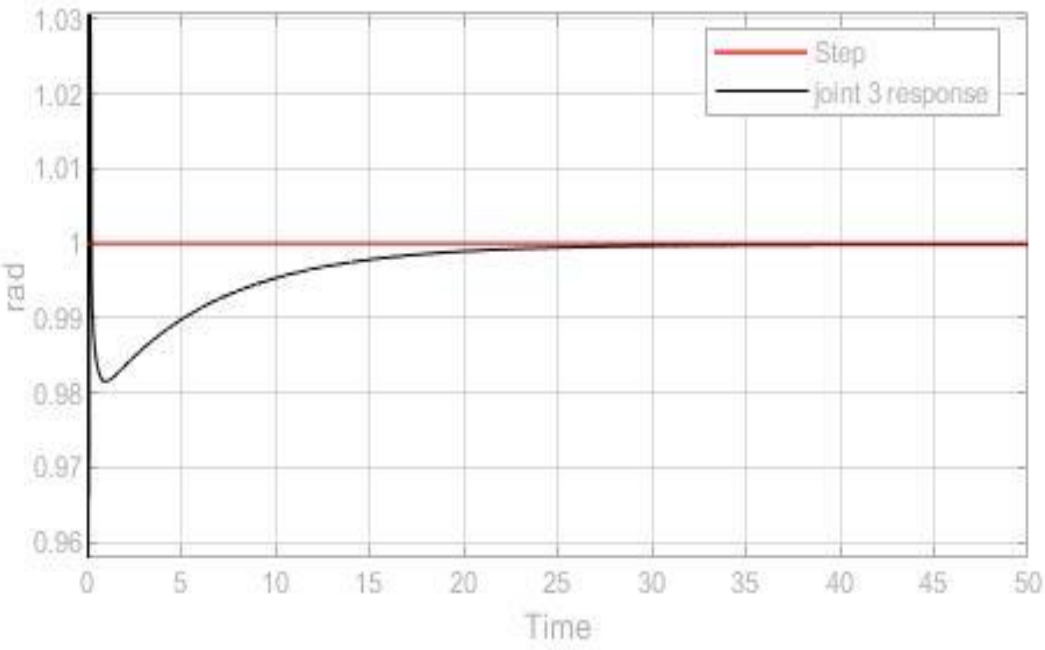

Figure 4.82: The effect of PID controller on the 3rd joint with that wearied by a man with 73 Kg.

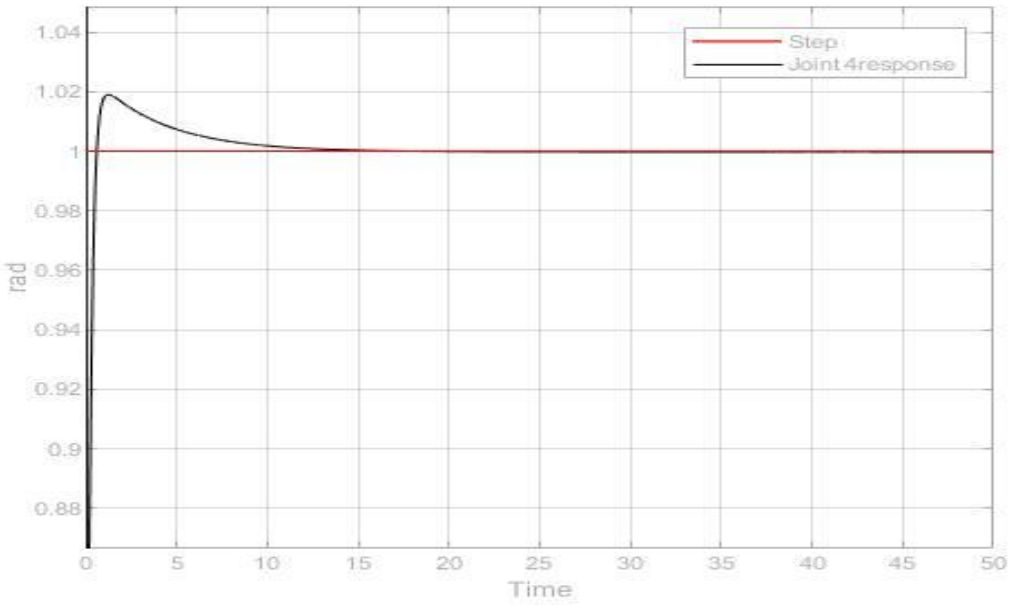

Figure 4.83: The effect of PID controller on the 4th joint with that wearied by a man with 73 Kg.

Figures 4.84 to 4.87 are about the tracking error in the system.

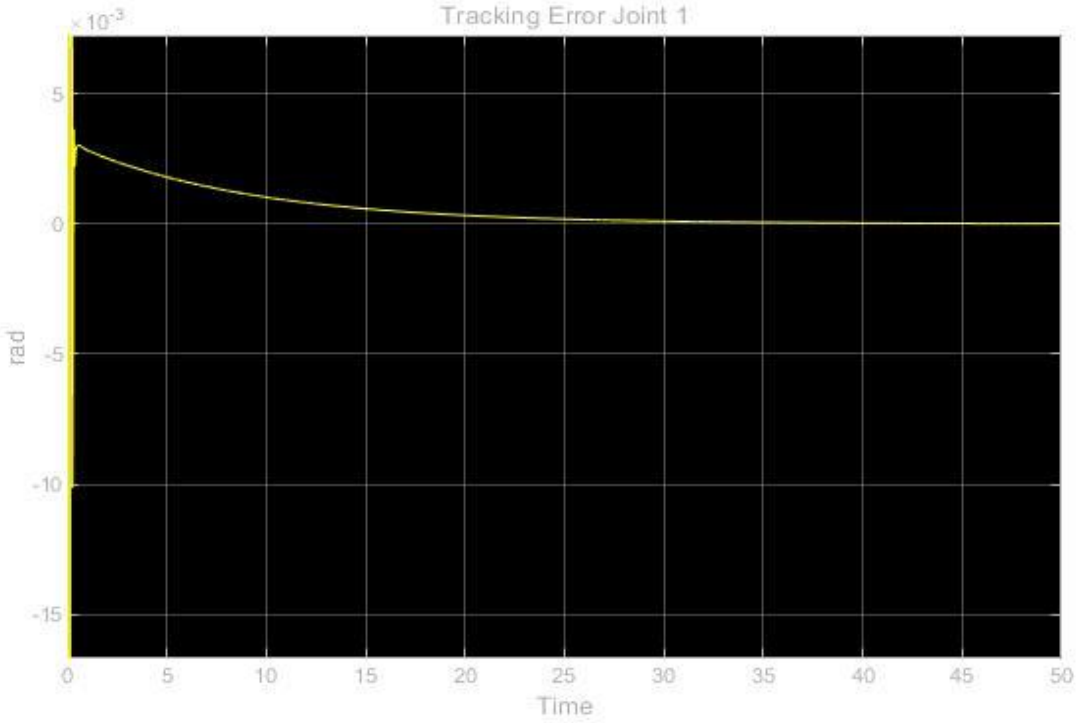

Figure 4.84: The 1st joint tracking error with for a man with 73 Kg.

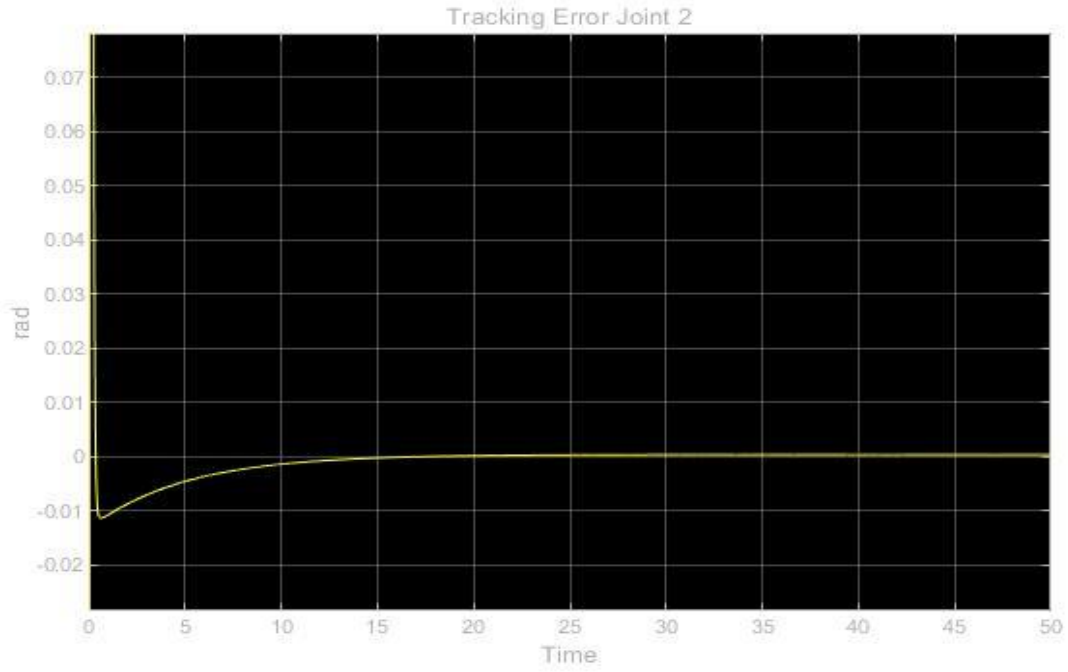

Figure 4.85: The 2nd joint tracking error with for a man with 73 Kg.

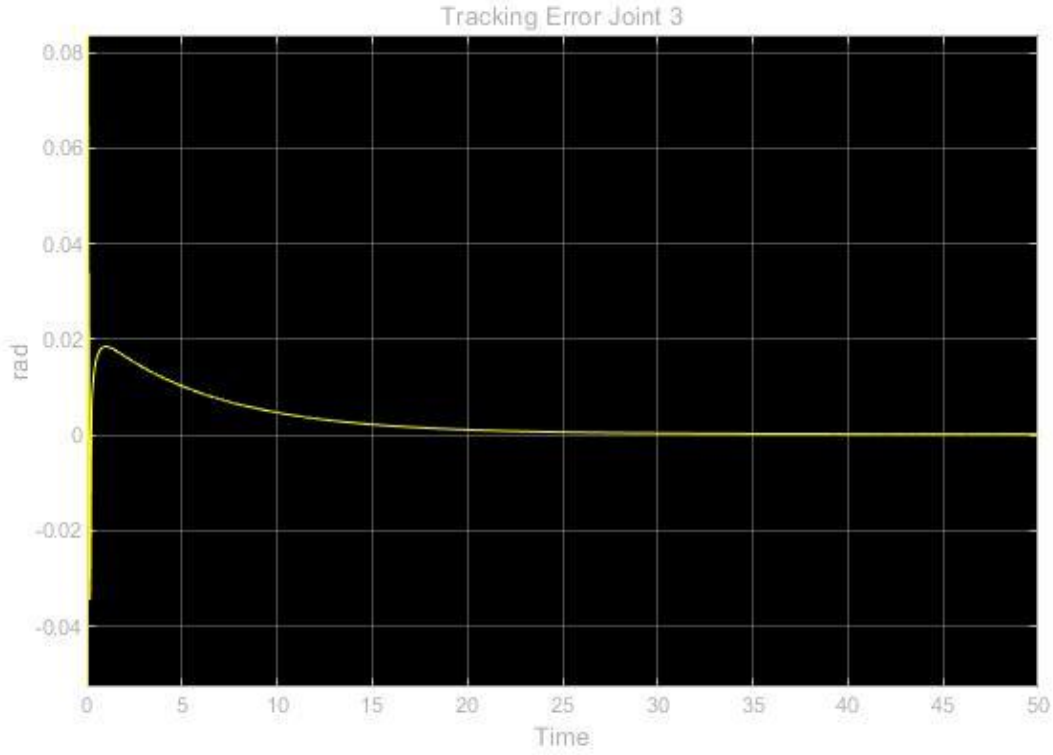

Figure 4.86: The 3rd joint tracking error with for a man with 73 Kg.

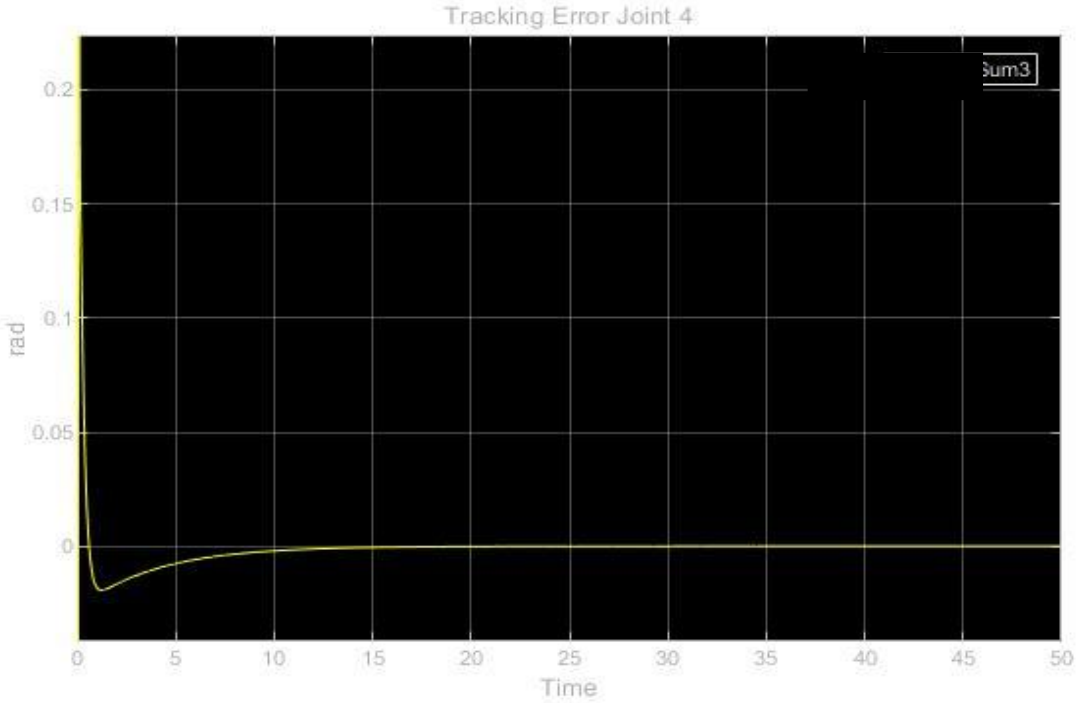

Figure 4.87: The 4th joint tracking error with for a man with 73 Kg.

The system can come up with compensating 73 Kg from a man arm weight as well. As it can be obviously seen from both weights, the disturbance-effecting joint 1, 3 and a little bit of joint 4. However, the amount of overshoots in joint 3 and 1 in the beginning of the simulation are very much the same. Joint 1 is the joint that is influenced by the centrifugal, Carioles and gravity effects of other joints. Therefore, it is clear to find that this joint deals with disturbance. Joint 3 also has the same situation. Since the major weight of the arm is belonging to the upper arm and joint 3 should compensate full arm weight, forearm weight can also bring disturbance to the joint 3. Note that the PID parameter has not changed at all and all the results in this chapter for PID controllers have the same PID parameters. Also, for the sinusoidal input in the CTC-PD, the results are satisfying.

In the future, there will be more studies for sinusoidal input to the system with CTC-PID controller as well.

4.4 Results and discussion

In this dissertation, design, kinematics, dynamics and control of 4-DoF wearable transported upper body exoskeleton was explored and discussed. As discussed in the introduction, the main aim of this research was to design an economic and commercialized type of exoskeleton in order to help and assist in short/long periods of life of stroke patients. The design of this device has been provenuser-friendly for patients in terms of kinematics studies and follows the natural arm motions dynamically. The two types of controller also are applied for this exoskeleton, both of which show good responses and therefore, both PID and PID-CTC are applicable for this device and PID is good for compensating the human arm weight as a disturbance.

In future studies the effect of sinusoidal input to the system for PID-CTC controller will be studied and also there will be more effort for improving the controller, for the situations that the system faces uncertainties.

REFERENCES

[1] Habib Ali1. (2014). Bionic Exoskeleton: History, Development and the Future..*International Conference on Advances in Engineering & Technology*, ,Singapore, 29-30March58-62

[2] Neil J Mizen. (1966). *U.S. Patent No. 3449769A.* Powered exoskeletal apparatus for amplifying human strength in response to normal body movements, Buffalo, N.Y., assignor to Cornell Aero- 5 nautical Laboratory, Inc.,

[3] A. H. A. Stienen, J. G. McPherson, A. C. Schouten, and J. P. A. Dewald. (2011).The ACT-4d: A Novel Rehabilitation Robot for the Quantification of Upper Limb Motor Impairments Following Brain Injury, *IEEE International Conference on Rehabilitation Robotics*, Jun 2011.

[4] D. Ragonesi, S. Agrawal, W. Sample, T. Rahman. (2011) . Series Elastic Actuator Control of a Powered Exoskeleton, *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Aug 2011.

[5] T.-M. Wu, S.-Y. Wang, and D.-Z. Chen. (2011). Design of an Exoskeleton for Strengthening the Upper Limb Muscle for Overextension Injury Prevention, *Mechanism and Machine Theory.*, 46,1825–1839. December 2011.

[6]A. Frisoli, F. Rocchi, S. Marcheschi, A. Dettori, F. Salsedo, M. Bergamasco, (2005). A New Force-Feedback arm Exoskeleton for Haptic Interaction in Virtual Environments, *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. IEEE, April 2005.

[7] B, S, Rupal., S, Rafique., A, Singla., E, Singla., M, Isaksson., G,S,Virk., (2017). Lower-limb exoskeletons, *International Journal of Advanced Robotic Systems,* 14,6. November 2017.

[8]C. Chu, and R. M. Patterson. (2018).Soft Robotic Devices for Hand Rehabilitation and Assistance: A Narrative Review, *IEEE Journal of NeuroEngineering and Rehabilitation,* 15,1.February 2018.

[9] J, Lu., K, Haninger., W, Chen., S ,Gowd., M, Tomizuka., and Jose M. Carmena .(2016). Design of a Passive Upper Limb Exoskeleton for

Macaque Monkeys, *Journal of Dynamic Systems, Measurement, and Control*, 138,11, July 2016.

[10] M, Iosa., G, Morone., A, Cherubini., S, Paolucci. (2016). The Three Laws of Neurorobotics: A Review on What Neurorehabilitation Robots Should Do for Patients and Clinicians, *Springer Nature.*, 36, 1-11. February 2016.

[11] S, Masiero., P, Poli., G, Rosati., D, Zanotto., M, Iosa., S,Paolucci., G, Morone(2014). The Value of Robotic Systems in Stroke Rehabilitation, *Expert Review of Medical Devices.*,2, 187-198, March 2014.

[12]P, Agarwal., D, Deshpande .(2019) *Human Performance Optimization, Exoskeletons.*, 234-259

[**13] Parson S, Breen A, Foster NE, Letley L, Pincus T, Vogel S,**(2007) et al. Prevalence and comparative troublesomeness by age of musculoskeletal pain in different body locations. Fam Pract. 2007;24:308–16.

[**14] Urwin M, Symmons D, Allison T, Brammah T, Busby H, Roxby M, et al.**(1998). Estimating the burden of musculoskeletal disorders in the community: the comparative prevalence of symptoms at different anatomical sites, and the relation to social deprivation. Ann Rheum Dis.1998;57:649–55.

[**15] Picavet H, Schouten JS.**(2003) Musculoskeletal pain in the Netherlands: prevalences, consequences and risk groups, the DMC(3) study. Pain.;102:167–78.

[16]Van Schaardenburg D, Van den Brande KJ, Ligthart GJ, Breedveld FC, Hazes JM.(1994) Musculoskeletal disorders and disability in persons aged 85 and over: a community survey.Ann Rheum Dis. 53:807–11.

[17] Lin JC, Weintraub N, Aragaki DR.(2009) Nonsurgical treatment for rotator cuff injury in the elderly. J Am Med Dir Assoc. 2008;9:626–32.

[18] Gopura RARC, Kiguchi K. (2009).Mechanical design of active upperlimb exoskeleton robots. ICORR Conf Proc. 2009;178–87.

[19] Lin HS, Xie SQ.(2012) Exoskeleton robots for upper-limb rehabilitation: state of the art and future prospects. Med Eng Phys. 2012;34:261–268.

[**20]Perry JC, Rosen J.** (2006) Design of a 7 degree-of-freedom upper-limb powered exoskeleton. BioRob Conf Proc. 2006;805–10.

[21] Nef T, Riener R. Shoulder actuation mechanisms for arm rehabilitation exoskeletons. BioRob Conf Proc. 2008;862–68.

[22] Tobias Nef and Robert Riener , (2008) ,Shoulder actuation mechanisms for arm rehabilitation exoskeletons, *2nd IEEE International Conference on Biomedical Robotics and Biomechatronics.* October 2008

[23] Calabrò RS, Russo M, Naro A, Milardi D, Bulletta T, Leo A, et al. (2019) .Who May Benefit From Armeo Power Treatment? A Neurophysiological Approach to Predict Neurorehabilitation.*American Academy of Physical Medicine and Rehabilitation. Published by Elsevier* .Inc. 2016 Feb 20.

[24] P. Nomanfar , A. Tekin , S Bogosyan and P.Sendur. (2019), Design and Kinematics of 4- {DoF} Multi-Purpose Wearable Mechanical Arm ({MUWA}) Support for Enhanced Operation Stability, *IEEE International Conference on Mechatronics*. March 2019.

[**25] Carignan C, Liszka M**.(2005) Design of an arm exoskeleton with scapula motion for shoulder rehabilitation. ICAR Conf Proc. 2005;524–531

[26] Balasubramanian S, Wei R, Perez M, Shepard B, Koeneman E, Koeneman J, et al.(2008). RUPERT: an exoskeleton robot for assisting rehabilitation of arm functions. Virtual Rehabil Conf Proc.2008;163–7

[27] T, Nef., M, Guidali ., R, Riener., (2009). Arm Therapy Exoskeleton with an Ergonomic Shoulder Actuation, *Applied Bionics and Biomechanics,* 6, 2, 127-142. July 2009.

[28] S, Masiero., P, Poli., G, Rosati., D, Zanotto., M, Iosa., S, Paolucci., G, Morone. (2014), The value of robotic systems in stroke rehabilitation, *Expert Review of Medical Devices,* 11,187-198. June 2014.

[29] H. Kawasaki, S. Ito, Y. Ishigure, Y. Nishimoto, T. Aoki, T. Mouri, H. Sakaeda, and M. Abe. (2007), Development of a HandM assist Robot for Rehabilitation Therapy by Patient Self-Motion Control,. *IEEE 10th International Conference on Rehabilitation Robotics,* June, 2007.

[30] P. W. McClure, L. A. Michener, B. J. Sennett, and A. R. Karduna, Direct 3-Dimensional Measurement of Scapular Kinematics During Dynamic Movements In Vivo, *Journal of Shoulder and Elbow Surgery,*. 10, 3, 269–277, May 2001.

[31] M. A. Ergin and V. Patoglu, (2012) .ASSISTON-SE: A Self-Aligning Shoulder Elbow Exoskeleton, *2012 IEEE International Conference on Robotics and Automation*. IEEE, May 2012.

[**32] K. Kiguchi and T. Fukuda**, (2004). A 3 DOF Exoskeleton for Upper Limb Motion Assist: Consideration of the Effect of Bi-Articular Muscles, *IEEE International Conference on Robotics and Automation,* April 2004.

[33] M. Esmaeili, K. Gamage, E. Tan, and D. Campolo, (2011). Ergonomic Considerations for Anthropomorphic Wrist Exoskeletons: A Simulation Study on the Effects of Joint Misalignment, *International Conference on Intelligent Robots and Systems*. IEEE, September 2011.

[34] J. Li., Z. Zhang., C. Tao., and R. Ji., (2017). A number Synthesis Method of the Self-Adapting Upper-Limb Rehabilitation Exoskeletons,*International Journal of Advanced Robotic Systems.*,14, 172988141771079, May 2017.

[35] X. Jiang, C. Xiong, R. Sun, and Y. Xiong, (2010). Characteristics of the Robotic arm of a 9-DoF Upper Limb Rehabilitation Robot Powered by Pneumatic Muscles, *Intelligent Robotics and Applications*. Springer, Berlin Heidelberg,. 463–474.

[36] S. Bai, S. Christensen, and M. R. U. Islam., (2017) .An upper-body Exoskeleton with a Novel Shoulder Mechanism for Assistive Applications," in *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM).*, Jul 2017.

[37]G. Turchetti, N. Vitiello, L. Trieste, S. Romiti, E. Geisler and S. Micera. (2013). Why Effectiveness of Robot-Mediated Neuro rehabilitation Does Not Necessarily Influence Its Adoption, *IEEE Reviews in Biomedical Engineering.,* 143-153.

[**38] M. Bottlang., S. M. Madey, C. M. Steyers, J. L. Marsh, and T. D. Brown.** (2000) .Assessment of Elbow Joint Kinematics in Passive Motion by Electromagnetic Motion Tracking, *Journal of Orthopedics Research*, 18, 195–202.

[39] D. Fekrache., M. Guiatni. (2015). Kinematics and Design a 5 DoF Exoskeleton for the Rehabilitation of the Upper Limb, *4th International Conference on Electrical Engineering (ICEE)*. IEEE, December 2015.

[40] H. Lee, W. Kim, J. Han, and C. Han., (2012). The Technical Trend of the eExoskeleton Robot System for Human Power Assistance, *International Journal of Precision Engineering and Manufacturing*13, 8, 1491–1497, July 2012.

[41] T, Essomba ., L, Nguyen ,Vu., (2018), Kinematic analysis of a new five-bar spherical decoupled mechanism with two-degrees of freedom remote center of motion, *Mechanism and Machine* Theory,119, 184-197.

[42] J.E. Shigley and J.J. Uicker, *Theory of Machines and Mechanisms*, 2nd Edition, McGraw-Hill, 1995.

[43] H, Høifødt., J. T, Gravdahl., (2011), Dynamic Modeling and Simulation of Robot Manipulators, Norwegian University of Science & Technology

Department of Engineering Cybernetics, *Master Thesis*,

[44] M. W. Spong, S. Hutchinson, M. Vidyasagar., (2004), Chapter 9: Dynamics*Robot Dynamics and Control*, second edition.

[45] M. W. Spong, S. Hutchinson, M. Vidyasagar., (2004), Chapter 10: *Contro of Single Joint*, second edition.

[46] H. M. Al-Qahtani ,A.A. Mohammed and M. Suna(2016) ,Dynamics and Control of a Robotic Arm Having Four Links, *Arabian Journal for Science and Engineering*, October 2016, 42,5, pp 1841-1852

[47] J. A. French , C.G. Rose and Marcia K. O'Malley(2014)System Characterization of Exo: A Robotic Exoskeleton for Upper Extremity Rehabilitation,*Sensing and Actuation for Control and mathsemicolon$ Biomedical Systems* **,**Volume 3, Ocober.

[48]Eduardo Rocon , Jos. L. Pons (2011) Exoskeletons in Rehabilitation Robotics Springer Berlin Heidelberg.

[49]F. Piltan, M. HosseinYarmahmoudi, M. Shamsodini, E. Mazlomian, A. Hosainpour.,(2012) PUMA-560 Robot Manipulator Position Computed Torque Control Methods Using MATLAB/SIMULINK and Their Integration into Graduate Nonlinear Control and MATLAB Courses, *International Journal of Robotics and Automation*, (IJRA), Volume (3) : Issue (3) : 2012.

[50] F.Piltan , N. Sulaiman, Zahra Tajpaykar, PaymanFerdosali, M. Rashidi, (2011) "Design Artificial Nonlinear Robust Controller Based on CTLC and FSMC with Tunable Gain, *"International Journal of Robotic and Automation*, 2 (3): 205-220, 2011.

[**51**] **B. Siciliano and O. Khatib**, (2008) Springer handbook of robotics: Springer-Verlag New York Inc, 2008

[52] F.Bendary, K.Elserafi, M.Soliman, R. Hassen.,(2015), Trajectory Tracking Control for Robot Manipulator Using Fractional-Order $\boldsymbol{PI\lambda D\mu}$ Control, *17th International Middle East Power Systems Conference*, Mansoura University, Egypt, December 15-17, 2015

[53]F. Piltan, S. Emamzadeh, Z. Hivand, F. Shahriyari, M. Mirzaei(2012) UMA-560 Robot Manipulator Position Sliding Mode Control Methods Using MATLAB/SIMULINK and Their Integration into Graduate /Undergraduate Nonlinear Control, Robotics and MATLAB Course., *International Journal of Robotic and Automation*, (IJRA), Volume (6) : Issue (3) : 2012

[54] F.Piltan , N. Sulaiman, Zahra Tajpaykar, PaymanFerdosali, M. Rashidi, (2011) "Design Artificial Nonlinear Robust Controller

Based on CTLC and FSMC with Tunable Gain, *"International Journal of Robotic and Automation*, 2 (3): 205-220, 2011.

[55] Paolo de Leval,(1996), Adjustments to Zatsiorsky-Seluyanovsegment inertia parameters,*Journal of Biomechanics* 29, 9, pp1223--1230

[56] S. Plagenhoef ,F. G. Evans and T. Abdelnour ,(1983), Anatomical Data for Analyzing Human Motion, *Research Quarterly for Exercise and Sport*,Informa {UK} Limited, 54,2, pp 169-178. June 1983

,

APPENDIX A

Chapter 3

Matrix of Inertia:

```
M=[ 5.2e-3*cos(q4) - 3.3e-3*sin(q2) + 1.8e-3*cos(q3)*sin(q4) + 1.8e-
3*cos(q4)*sin(q3) - 6.1e-3*sin(q2)*sin(q3) - 3.9e-3*cos(q2)^2 -
5.2e-3*cos(q2)^2*cos(q4) - 5.1e-4*cos(q2)^2*cos(q3)*sin(q4) - 5.1e-
4*cos(q2)^2*cos(q4)*sin(q3) - 3.4e-3*cos(q3)*sin(q2)*sin(q4) - 3.4e-
3*cos(q4)*sin(q2)*sin(q3) + 7.6e-3,
1.3e-3*cos(q3 + q4)*cos(q2),
6.1e-3*sin(q3) - 3.9e-3*sin(q2) - 5.2e-3*cos(q4)*sin(q2) + 1.7e-
3*cos(q3)*sin(q4) + 1.7e-3*cos(q4)*sin(q3),
1.7e-3*sin(q3 + q4) - 3.9e-3*sin(q2);

1.3e-3*cos(q3 + q4)*cos(q2),
3.7e-3*sin(q3 + q4) - 6.8e-3*cos(2.0*q3) - 5.2e-3*cos(2.0*q3 + q4) +
5.2e-3*cos(q4) - 6.3e-4*cos(2.0*q3 + 2.0*q4) + 0.013*sin(q3) +
0.014,
0,                                   0;

6.1e-3*sin(q3) - 3.9e-3*sin(q2) - 5.2e-3*cos(q4)*sin(q2) + 1.7e-
3*cos(q3)*sin(q4) + 1.7e-3*cos(q4)*sin(q3),
0,
0.01*cos(q4) + 0.02,
5.2e-3*cos(q4) + 3.9e-3;

1.7e-3*sin(q3 + q4) - 3.9e-3*sin(q2),
0,
5.2e-3*cos(q4) + 3.9e-3,
  3.9e-3;];
```

Gravity :

```
G=[ 0.04*g*cos(q1) - 0.017*g*cos(q1)*cos(q3)*sin(q2)*sin(q4) -
0.017*g*cos(q1)*cos(q4)*sin(q2)*sin(q3) +
0.017*g*cos(q3)*cos(q4)*sin(q1)*sin(q2)^2 -
0.017*g*sin(q1)*sin(q2)^2*sin(q3)*sin(q4);

-1.0e-34*g*cos(q2)*sin(q1)*(1.7e+32*sin(q3 + q4) + 6.2e+32*sin(q3) +
3.4e+32);
 0.062*g*cos(q1)*sin(q3) + 0.017*g*cos(q1)*cos(q3)*sin(q4) +
0.017*g*cos(q1)*cos(q4)*sin(q3) - 0.062*g*cos(q3)*sin(q1)*sin(q2) -
0.017*g*cos(q3)*cos(q4)*sin(q1)*sin(q2) +
0.017*g*sin(q1)*sin(q2)*sin(q3)*sin(q4);

0.017*g*cos(q1)*cos(q3)*sin(q4) + 0.017*g*cos(q1)*cos(q4)*sin(q3) -
0.017*g*cos(q3)*cos(q4)*sin(q1)*sin(q2) +
0.017*g*sin(q1)*sin(q2)*sin(q3)*sin(q4);];
```

Coriolis and CentrigiugalForces:

```
C    =[6.1e-3*dq1^2*cos(q3)    +    0.01*dq1*dq3*sin(q4)    -    6.3e-
4*dq2^2*sin(2.0*q3)*sin(q2)   -   6.3e-4*dq2^2*sin(2.0*q4)*sin(q2)   +
1.3e-3*dq1*dq2*sin(2.0*q2)  +  1.7e-3*dq3^2*cos(q3)*cos(q4)  +  1.7e-
3*dq4^2*cos(q3)*cos(q4)   -   5.2e-3*dq1^2*sin(q2)*sin(q4)   -   5.2e-
3*dq2^2*sin(q2)*sin(q4)   -   5.2e-3*dq3^2*sin(q2)*sin(q4)   -   1.7e-
3*dq3^2*sin(q3)*sin(q4)   -   1.7e-3*dq4^2*sin(q3)*sin(q4)   -   6.7e-
3*dq1*dq2*cos(q2)  +  1.7e-3*dq1^2*cos(q2)^2*cos(q3)*cos(q4)  +  5.2e-
3*dq1^2*cos(q2)^2*sin(q2)*sin(q4)               -               1.7e-
3*dq1^2*cos(q2)^2*sin(q3)*sin(q4)               +               5.2e-
3*dq2^2*cos(q3)^2*sin(q2)*sin(q4) - 0.01*dq1*dq3*cos(q2)^2*sin(q4) +
1.8e-3*dq1^2*cos(q3)*cos(q4)*sin(q2)             +             1.8e-
3*dq2^2*cos(q3)*cos(q4)*sin(q2)                 -               1.8e-
3*dq1^2*sin(q2)*sin(q3)*sin(q4)                 -               1.8e-
3*dq2^2*sin(q2)*sin(q3)*sin(q4)  +  3.4e-3*dq3*dq4*cos(q3)*cos(q4)  -
0.012*dq1*dq2*cos(q2)*sin(q3)   -   0.012*dq1*dq3*cos(q3)*sin(q2)   -
3.4e-3*dq3*dq4*sin(q3)*sin(q4)                  +               3.1e-
3*dq1^2*cos(q2)^2*sin(q2)*sin(q3)*sin(q4)           +           2.6e-
3*dq1*dq3*cos(q2)^2*cos(q3)*cos(q4)               +             2.6e-
3*dq1*dq4*cos(q2)^2*cos(q3)*cos(q4)               +             2.5e-
3*dq1*dq2*cos(q2)*cos(q3)^2*sin(q2)               +             2.5e-
3*dq1*dq2*cos(q2)*cos(q4)^2*sin(q2)               -             2.6e-
3*dq1*dq3*cos(q2)^2*sin(q3)*sin(q4)               -             2.6e-
3*dq1*dq4*cos(q2)^2*sin(q3)*sin(q4)               -             5.2e-
3*dq1^2*cos(q2)^2*cos(q3)^2*sin(q2)*sin(q4)         +           5.2e-
3*dq2^2*cos(q3)*cos(q4)*sin(q2)*sin(q3)                         +
0.01*dq1*dq2*cos(q2)*cos(q4)*sin(q2)              -             3.4e-
3*dq1*dq2*cos(q2)*cos(q3)*sin(q4)                 -             3.4e-
3*dq1*dq2*cos(q2)*cos(q4)*sin(q3)                 -             3.4e-
3*dq1*dq3*cos(q3)*cos(q4)*sin(q2)                 -             3.4e-
3*dq1*dq4*cos(q3)*cos(q4)*sin(q2)                 -             2.6e-
3*dq2*dq3*cos(q2)*cos(q3)*sin(q4)                 -             2.6e-
3*dq2*dq3*cos(q2)*cos(q4)*sin(q3)                 -             2.6e-
3*dq2*dq4*cos(q2)*cos(q3)*sin(q4)                 -             2.6e-
3*dq2*dq4*cos(q2)*cos(q4)*sin(q3)                 +             3.4e-
3*dq1*dq3*sin(q2)*sin(q3)*sin(q4)                 +             3.4e-
3*dq1*dq4*sin(q2)*sin(q3)*sin(q4)                 -             3.1e-
3*dq1^2*cos(q2)^2*cos(q3)*cos(q4)*sin(q2)           +           1.3e-
3*dq1^2*cos(q2)^2*cos(q3)*sin(q2)*sin(q3)           +           1.3e-
3*dq1^2*cos(q2)^2*cos(q4)*sin(q2)*sin(q4)           +           2.5e-
3*dq2^2*cos(q3)*cos(q4)^2*sin(q2)*sin(q3)           +           2.5e-
3*dq2^2*cos(q3)^2*cos(q4)*sin(q2)*sin(q4)           -           2.5e-
3*dq1^2*cos(q2)^2*cos(q3)*cos(q4)^2*sin(q2)*sin(q3)      -      2.5e-
3*dq1^2*cos(q2)^2*cos(q3)^2*cos(q4)*sin(q2)*sin(q4)      -      5.0e-
3*dq1*dq2*cos(q2)*cos(q3)^2*cos(q4)^2*sin(q2)         +         3.6e-
3*dq1*dq2*cos(q2)*cos(q3)*sin(q2)*sin(q4)           +           3.6e-
3*dq1*dq2*cos(q2)*cos(q4)*sin(q2)*sin(q3)           -           5.2e-
3*dq1^2*cos(q2)^2*cos(q3)*cos(q4)*sin(q2)*sin(q3)                   -
0.01*dq1*dq2*cos(q2)*cos(q3)^2*cos(q4)*sin(q2)                      +
0.01*dq1*dq2*cos(q2)*cos(q3)*sin(q2)*sin(q3)*sin(q4)      +     5.0e-
3*dq1*dq2*cos(q2)*cos(q3)*cos(q4)*sin(q2)*sin(q3)*sin(q4);
-     9.8e-3*dq1^2*sin(2.0*q2)     +     3.3e-3*dq1^2*cos(q2)      -
0.01*dq2*dq3*sin(q4)          -          0.01*dq2*dq4*sin(q4)          +
0.012*dq2*dq3*sin(2.0*q3)   -   1.3e-3*dq2*dq3*sin(2.0*q4)   -   1.3e-
3*dq2*dq4*sin(2.0*q3)     -     1.3e-3*dq2*dq4*sin(2.0*q4)+     6.1e-
3*dq1^2*cos(q2)*sin(q3)     +     0.033*dq1*dq3*cos(q2)     +     2.6e-
3*dq1*dq4*cos(q2)           +           0.013*dq2*dq3*cos(q3)           +
0.012*dq1^2*cos(q2)*cos(q3)^2*sin(q2)             -             1.3e-
3*dq1^2*cos(q2)*cos(q4)^2*sin(q2)  -  0.024*dq1*dq3*cos(q2)*cos(q3)^2
+          2.5e-3*dq1*dq3*cos(q2)*cos(q4)^2          +          2.5e-
```

```
3*dq1*dq4*cos(q2)*cos(q3)^2   +   2.5e-3*dq1*dq4*cos(q2)*cos(q4)^2   +
0.021*dq2*dq3*cos(q3)^2*sin(q4)   +   0.01*dq2*dq4*cos(q3)^2*sin(q4)-
0.01*dq1^2*cos(q2)*cos(q4)*sin(q2)              +              1.7e-
3*dq1^2*cos(q2)*cos(q3)*sin(q4)                +              1.7e-
3*dq1^2*cos(q2)*cos(q4)*sin(q3)                                   -
0.013*dq1^2*cos(q2)*sin(q2)*sin(q3)  +  0.021*dq1*dq3*cos(q2)*cos(q4)
+  0.01*dq1*dq4*cos(q2)*cos(q4)  +  3.7e-3*dq2*dq3*cos(q3)*cos(q4)  +
3.7e-3*dq2*dq4*cos(q3)*cos(q4)  +  0.013*dq1*dq3*cos(q2)*sin(q3)    -
3.7e-3*dq2*dq3*sin(q3)*sin(q4)  -  3.7e-3*dq2*dq4*sin(q3)*sin(q4)  -
0.021*dq1*dq3*cos(q2)*cos(q3)^2*cos(q4)                           -
0.01*dq1*dq4*cos(q2)*cos(q3)^2*cos(q4)            +            5.0e-
3*dq2*dq3*cos(q3)*cos(q4)^2*sin(q3)             +              5.0e-
3*dq2*dq3*cos(q3)^2*cos(q4)*sin(q4)             +              5.0e-
3*dq2*dq4*cos(q3)*cos(q4)^2*sin(q3)             +              5.0e-
3*dq2*dq4*cos(q3)^2*cos(q4)*sin(q4)                 +          2.5e-
3*dq1^2*cos(q2)*cos(q3)^2*cos(q4)^2*sin(q2)         -          5.0e-
3*dq1*dq3*cos(q2)*cos(q3)^2*cos(q4)^2            -             5.0e-
3*dq1*dq4*cos(q2)*cos(q3)^2*cos(q4)^2            -             3.7e-
3*dq1^2*cos(q2)*cos(q3)*sin(q2)*sin(q4)           -            3.7e-
3*dq1^2*cos(q2)*cos(q4)*sin(q2)*sin(q3)           +            3.7e-
3*dq1*dq3*cos(q2)*cos(q3)*sin(q4)               +              3.7e-
3*dq1*dq3*cos(q2)*cos(q4)*sin(q3)               +              3.7e-
3*dq1*dq4*cos(q2)*cos(q3)*sin(q4)               +              3.7e-
3*dq1*dq4*cos(q2)*cos(q4)*sin(q3)                                 +
0.021*dq2*dq3*cos(q3)*cos(q4)*sin(q3)                             +
0.01*dq2*dq4*cos(q3)*cos(q4)*sin(q3)                              +
0.01*dq1^2*cos(q2)*cos(q3)^2*cos(q4)*sin(q2)                      -
0.01*dq1^2*cos(q2)*cos(q3)*sin(q2)*sin(q3)*sin(q4)                +
0.021*dq1*dq3*cos(q2)*cos(q3)*sin(q3)*sin(q4)                     +
0.01*dq1*dq4*cos(q2)*cos(q3)*sin(q3)*sin(q4)          -           2.5e-
3*dq1^2*cos(q2)*cos(q3)*cos(q4)*sin(q2)*sin(q3)*sin(q4)    +    5.0e-
3*dq1*dq3*cos(q2)*cos(q3)*cos(q4)*sin(q3)*sin(q4)       +       5.0e-
3*dq1*dq4*cos(q2)*cos(q3)*cos(q4)*sin(q3)*sin(q4);
5.2e-3*dq2^2*sin(q4)      -      5.2e-3*dq4^2*sin(q4)      -      6.1e-
3*dq2^2*sin(2.0*q3)     +     6.3e-4*dq2^2*sin(2.0*q4)     -     6.6e-
3*dq1^2*cos(q3)   -   6.6e-3*dq2^2*cos(q3)   -   0.01*dq3*dq4*sin(q4)   +
6.6e-3*dq1^2*cos(q2)^2*cos(q3)   -   5.2e-3*dq1^2*cos(q2)^2*sin(q4)   -
0.01*dq2^2*cos(q3)^2*sin(q4)  -  1.8e-3*dq1^2*cos(q3)*cos(q4)  -  1.8e-
3*dq2^2*cos(q3)*cos(q4)   +   6.1e-3*dq1^2*cos(q3)*sin(q2)   +   1.8e-
3*dq1^2*sin(q3)*sin(q4)      +      1.8e-3*dq2^2*sin(q3)*sin(q4)      -
0.033*dq1*dq2*cos(q2)   +   1.8e-3*dq1^2*cos(q2)^2*cos(q3)*cos(q4)   +
0.012*dq1^2*cos(q2)^2*cos(q3)*sin(q3)             -             1.3e-
3*dq1^2*cos(q2)^2*cos(q4)*sin(q4)               -               2.5e-
3*dq2^2*cos(q3)*cos(q4)^2*sin(q3)               -               2.5e-
3*dq2^2*cos(q3)^2*cos(q4)*sin(q4)               -               1.8e-
3*dq1^2*cos(q2)^2*sin(q3)*sin(q4)  +  0.024*dq1*dq2*cos(q2)*cos(q3)^2
-                2.5e-3*dq1*dq2*cos(q2)*cos(q4)^2                 +
0.01*dq1^2*cos(q2)^2*cos(q3)^2*sin(q4)            +             1.7e-
3*dq1^2*cos(q3)*cos(q4)*sin(q2)  -  0.01*dq2^2*cos(q3)*cos(q4)*sin(q3)
-              1.7e-3*dq1^2*sin(q2)*sin(q3)*sin(q4)               -
0.021*dq1*dq2*cos(q2)*cos(q4)   -   0.013*dq1*dq2*cos(q2)*sin(q3)   +
0.01*dq1*dq4*sin(q2)*sin(q4)                                      +
0.021*dq1*dq2*cos(q2)*cos(q3)^2*cos(q4)           +             2.5e-
3*dq1^2*cos(q2)^2*cos(q3)*cos(q4)^2*sin(q3)         +         2.5e-
3*dq1^2*cos(q2)^2*cos(q3)^2*cos(q4)*sin(q4)         +         5.0e-
3*dq1*dq2*cos(q2)*cos(q3)^2*cos(q4)^2            -            3.7e-
3*dq1*dq2*cos(q2)*cos(q3)*sin(q4)               -             3.7e-
3*dq1*dq2*cos(q2)*cos(q4)*sin(q3)                                 +
0.01*dq1^2*cos(q2)^2*cos(q3)*cos(q4)*sin(q3)                      -
0.021*dq1*dq2*cos(q2)*cos(q3)*sin(q3)*sin(q4)         -         5.0e-
3*dq1*dq2*cos(q2)*cos(q3)*cos(q4)*sin(q3)*sin(q4);
```

```
5.2e-3*dq1^2*sin(q4) + 5.2e-3*dq2^2*sin(q4) + 5.2e-3*dq3^2*sin(q4) +
6.3e-4*dq2^2*sin(2.0*q3)   +   6.3e-4*dq2^2*sin(2.0*q4)   -   5.2e-
3*dq1^2*cos(q2)^2*sin(q4)  -  5.2e-3*dq2^2*cos(q3)^2*sin(q4)  -  1.8e-
3*dq1^2*cos(q3)*cos(q4)   -   1.8e-3*dq2^2*cos(q3)*cos(q4)   +   1.8e-
3*dq1^2*sin(q3)*sin(q4)   +   1.8e-3*dq2^2*sin(q3)*sin(q4)   -   2.6e-
3*dq1*dq2*cos(q2)  +  1.8e-3*dq1^2*cos(q2)^2*cos(q3)*cos(q4)  -  1.3e-
3*dq1^2*cos(q2)^2*cos(q3)*sin(q3)              -              1.3e-
3*dq1^2*cos(q2)^2*cos(q4)*sin(q4)              -              2.5e-
3*dq2^2*cos(q3)*cos(q4)^2*sin(q3)              -              2.5e-
3*dq2^2*cos(q3)^2*cos(q4)*sin(q4)              -              1.8e-
3*dq1^2*cos(q2)^2*sin(q3)*sin(q4)  -  2.5e-3*dq1*dq2*cos(q2)*cos(q3)^2
-          2.5e-3*dq1*dq2*cos(q2)*cos(q4)^2          +          5.2e-
3*dq1^2*cos(q2)^2*cos(q3)^2*sin(q4)              +              1.7e-
3*dq1^2*cos(q3)*cos(q4)*sin(q2)                -               5.2e-
3*dq2^2*cos(q3)*cos(q4)*sin(q3)                -               1.7e-
3*dq1^2*sin(q2)*sin(q3)*sin(q4)   -   0.01*dq1*dq2*cos(q2)*cos(q4)   -
0.01*dq1*dq3*sin(q2)*sin(q4)                                          +
0.01*dq1*dq2*cos(q2)*cos(q3)^2*cos(q4)            +            2.5e-
3*dq1^2*cos(q2)^2*cos(q3)*cos(q4)^2*sin(q3)          +          2.5e-
3*dq1^2*cos(q2)^2*cos(q3)^2*cos(q4)*sin(q4)          +          5.0e-
3*dq1*dq2*cos(q2)*cos(q3)^2*cos(q4)^2             -             3.7e-
3*dq1*dq2*cos(q2)*cos(q3)*sin(q4)              -              3.7e-
3*dq1*dq2*cos(q2)*cos(q4)*sin(q3)              +              5.2e-
3*dq1^2*cos(q2)^2*cos(q3)*cos(q4)*sin(q3)                             -
0.01*dq1*dq2*cos(q2)*cos(q3)*sin(q3)*sin(q4)         -         5.0e-
3*dq1*dq2*cos(q2)*cos(q3)*cos(q4)*sin(q3)*sin(q4);];
```

CURRICULUM VITAE

Name Surname **: PegahNomanfar**

Place and Date of Birth **: Iran/Tehran**

E-Mail **: nomanfar17@itu.edu.tr**

EDUCATION **:**

- **B.Sc.** **:** 2014, University of Tehran, Mechanics of Biosystems,

Papers:

"Design and Kinematics of 4-DoF Multi-Purpose Wearable Mechanical Arm (MUWA) Support for Enhanced Operation Stability". IEEE ICM 2019 Conferances, with Prof. Dr. S.Boygosan(professor at Istanbul Technical University),Dr, P. Sendur and Dr. A. Tekin (Assistant Professors at Ozyegin University) and M. A. Osama (Graduate Master Student from Ozyegin University)

AND PATENTS:

Patent in "A Multi-Purpose Wearable Mechanical Arm Support for Enhanced Operation Stability" Tracking number 2018/00340 in www.turkpatent.gov.tr ,Ozyegin University (2018)