ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

ADAPTIVE CONTROL OF A SIX DOF ROBOT MANIPULATOR

M.Sc. THESIS

Muhammet Umut DANIŞ

Department of Mechanical Engineering

System Dynamics and Control Programme

JUNE 2023

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

ADAPTIVE CONTROL OF A SIX DOF ROBOT MANIPULATOR

M.Sc. THESIS

Muhammet Umut DANIŞ
(503201608)

Department of Mechanical Engineering

System Dynamics and Control Programme

Thesis Advisor: Assoc. Prof. Dr. Zeki Yağız BAYRAKTAROĞLU

JUNE 2023

ISTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

ALTI SERBESTLİK DERECELİ BİR ROBOT MANİPÜLATÖRÜN ADAPTİF KONTROLÜ

YÜKSEK LİSANS TEZİ

Muhammet Umut DANIŞ
(503201608)

Makina Mühendisliği Anabilim Dalı

Sistem Dinamiği ve Kontrol Programı

Tez Danışmanı: Doç. Dr. Zeki Yağız BAYRAKTAROĞLU

HAZİRAN 2023

Muhammet Umut DANIŞ, a M.Sc. student of İTU Graduate School student ID 503201608, successfully defended the thesis/dissertation entitled "ADAPTIVE CONTROL OF A SIX DOF ROBOT MANIPULATOR", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Assoc. Prof. Dr. Zeki Yağız BAYRAKTAROĞLU** İstanbul Technical University	.....
Jury Members :	**Prof. Dr. Erdinç Altuğ** Istanbul Technical University	.....
	Assoc. Prof. Dr. Erkin Dinçmen Işık University	.....

Date of Submission : 25.05.2023
Date of Defense : 15.06.2023

To my family,

FOREWORD

I would like to thank Assoc. Prof. Dr. Zeki Yağız BAYRAKTAROĞLU for his guidance throughout the preparation phase of the thesis.

June 2023

Muhammet Umut DANIŞ
(Research Assistant)

TABLE OF CONTENTS

Page

1. INTRODUCTION 1
1.1 Purpose of Thesis 2
1.2 Literature Review 2
1.3 Industrial Robot System 3
2. MATHEMATICAL MODELING 9
2.1 Kinematic Modeling 9
2.1.1 Representation of Position and Orientation 9
2.1.2 Forward Kinematics 12
2.1.3 Inverse Kinematics 14
2.1.4 Velocity Kinematics 14
2.2 Dynamic Modeling 15
2.2.1 Euler – Lagrange Method 15
2.2.2 Newton – Euler Method 16
2.2.3 Friction and Spring Model 17
3. MOTION CONTROL 21
3.1 Trajectory Generation 21
3.2 Computed Torque Control 21
3.3 Passivity-based Control 22
3.4 Adaptive Control 23
4. EXPERIMENTAL RESULTS 27
4.1 Comparison of Controller Performances with Varying Payload 29
4.2 Parameter Convergence of the Adaptive Controller 44
5. CONCLUSIONS 49
REFERENCES 51
APPENDICES 53
APPENDIX A 55
CURRICULUM VITAE 57

ABBREVIATIONS

CTC **:** Computed Torque Control
DOF **:** Degrees of Freedom
DH **:** Denavit – Hartenberg
IDE **:** Integrated Development Environment
ISO **:** International Organiztion for standardization
LLI **:** Low Level Interface
MCP **:** Manuel Control Pendant
MNEA **:** Modified Newton-Euler Algorithm
PB **:** Passivity-Based
RMS **:** Root Mean Square

SYMBOLS

$\boldsymbol{q}$	**:** Joint position vector
$\boldsymbol{\tau}$	**:** Torque vector
R	**:** Rotation matrix
T	**:** Homogeneous transformation matrix
J	**:** Jacobian matrix
t	**:** Time
M	**:** Inertia matrix
C	**:** Centrifugal – Coriolis matrix
g	**:** Gravity vector
Y	**:** Regressor matrix
p	**:** Dynamic parameters vector
$\boldsymbol{\tau_f}$	**:** Friction torque vector
$\boldsymbol{\tau_s}$	**:** Spring torque vector

LIST OF TABLES

Page

Table 1.1 : Amplitude, speed and resolution of joints of RX160..... **5**
Table 2.1: DH Table of RX160..... **13**
Table 2.2: Friction Model Parameters Identified in [13] for Positive Direction..... **18**
Table 2.3: Friction Model Parameters Identified in [13] for Negative Direction **19**
Table 4.1: Waypoints used in the Validation Trajectories **28**
Table 4.2: Adaptive and Passivity-based Controller Gains..... **30**
Table 4.3: Computed Torque Controller Gains..... **33**

LIST OF FIGURES

Page

Figure 1.1 : Annual installations of industrial robot units [2] **1**
Figure 1.2 : Stäubli RX160 industrial robot manipulator [9] **3**
Figure 1.3 : Dimensions of Stäubli RX160 industrial robot manipulator [9] **4**
Figure 1.4 : Work envelope of Stäubli RX160 industrial robot manipulator [9] **4**
Figure 1.5 : CS8C Controller [9] **5**
Figure 1.6 : General architecture of CS8 controller [10] **6**
Figure 1.7 : Manual Control Pendant [9]. **6**
Figure 1.8 : Structure of the joint control system[10] **7**
Figure 2.1 : Representation of Position in Cartesian Coordinates. **10**
Figure 2.2 : Coordinates of a vector w.r.t. Two Coordinate Frames **10**
Figure 2.3 : Euler Angles **11**
Figure 2.4 : Coordinate Frame Assignment to RX160 Robot **13**
Figure 2.5 : Joint Structures of RX160 Robot Manipulator **18**
Figure 2.6 : Gravity Counter-Balance Spring System **19**
Figure 3.1 : Block Diagram of the Computed Torque Control **22**
Figure 3.2 : Block Diagram of the Passivity-based Control **23**
Figure 3.3 : Block Diagram of the Adaptive Control **25**
Figure 4.1 : Payload of 3 kg. **27**
Figure 4.2 : Validation Trajectories **28**
Figure 4.3 : Path Followed by the Robot End Effector **29**
Figure 4.4 : End Effector Velocities **29**
Figure 4.5 : Position Errors for No Payload(Adaptive-PB) **30**
Figure 4.6 : Position Errors for Payload of 3 kg(Adaptive-PB) **31**
Figure 4.7 : Position Errors for Payload of 5 kg(Adaptive-PB) **31**
Figure 4.8 : Position Errors for Payload of 7 kg(Adaptive-PB) **32**
Figure 4.9 : Position Errors for Payload of 9 kg(Adaptive-PB) **32**
Figure 4.10 : Position Errors for No Payload(Adaptive-CTC) **33**
Figure 4.11 : Position Errors for Payload of 3 kg(Adaptive-CTC) **34**
Figure 4.12 : Position Errors for Payload of 5 kg(Adaptive-CTC) **34**
Figure 4.13 : Position Errors for Payload of 7 kg(Adaptive-CTC) **35**
Figure 4.14 : Position Errors for Payload of 9 kg(Adaptive-CTC) **35**
Figure 4.15 : Velocity Errors for No Payload(Adaptive-PB) **36**
Figure 4.16 : Velocity Errors for Payload of 3 kg(Adaptive-PB) **37**
Figure 4.17 : Velocity Errors for Payload of 5 kg(Adaptive-PB) **37**
Figure 4.18 : Velocity Errors for Payload of 7 kg(Adaptive-PB) **38**
Figure 4.19 : Velocity Errors for Payload of 9 kg(Adaptive-PB) **38**
Figure 4.20 : Velocity Errors for No Payload(Adaptive-CTC) **39**
Figure 4.21 : Velocity Errors for Payload of 3 kg(Adaptive-CTC) **39**
Figure 4.22 : Velocity Errors for Payload of 5 kg(Adaptive-CTC) **40**
Figure 4.23 : Velocity Errors for Payload of 7 kg(Adaptive-CTC) **40**
Figure 4.24 : Velocity Errors for Payload of 9 kg(Adaptive-CTC) **41**
Figure 4.25 : Torque Values for No Payload(Adaptive-PB) **42**

Figure 4.26 : Torque Values for Payload of 3 kg..... **43**
Figure 4.27 : Torque Values for Payload of 5 kg..... **43**
Figure 4.28 : Torque Values for Payload of 7 kg..... **44**
Figure 4.29 : Torque Values for Payload of 9 kg..... **44**
Figure 4.30 : RMS Values of Position Errors..... **45**
Figure 4.31 : Percent Change in RMS Values of Position Errors..... **45**
Figure 4.32 : Dynamic Parameter Values for No Payload..... **46**
Figure 4.33 : Dynamic Parameter Values for Payload of 3 kg..... **46**
Figure 4.34 : Dynamic Parameter Values for Payload of 5 kg..... **47**
Figure 4.35 : Dynamic Parameter Values for Payload of 7 kg..... **47**
Figure 4.36 : Dynamic Parameter Values for Payload of 9 kg..... **48**
Figure A.1 : MATLAB Code for MNEA..... **56**

ADAPTIVE CONTROL OF A SIX DOF ROBOT MANIPULATOR

SUMMARY

Robot manipulators are used in the industry for industrial applications such as pick and place, painting, packaging, assembly, welding and machining. Performance improvement of the robots is important for increasing efficiency and accuracy in the tasks where the robots are used. Control algorithm is a vital part of the robot that effects the performance of the robot. The performance of the control algorithms based on the dynamic model of the robot may degrade under uncertainties associated with the dynamic model of the robot if they are not accounted for. Generally, adaptive or robust control is used to cope with these uncertainties. Adaptive control includes an online parameter adaptation mechanism to take these uncertainties into account. Adaptive control has the advantage of the online performance improvement over robust control. Therefore, adaptive control is an appealing approach for the control of the robot manipulators in the presence of disturbances and uncertainties in the dynamic model of the robot.

In this study an adaptive control algorithm proposed by an earlier study implemented on a six-DOF robot manipulator. Experiments are accomplished on a Stäubli RX160 robot manipulator. An integral term is also added to the original control law to reduce steady-state errors. Performance of the adaptive control algorithm is compared to performace of the two non-adaptive approaches for varying payloads, namely computed torque control and passivity-based control.

This thesis consists of five chapters. In the first chapter purpose of the thesis, literature review and basic information about the industrial robot system used in the experiments are given. Second chapter mentions basics of kinematic and dynamic modeling of robot manipulators as well as models for friction and spring effects. Third chapter gives theoretical background for the control algorithms implemented in the experiments. Control laws and block diagrams for computed torque control, passivity-based control and adaptive control are given. Fourth chapter gives the experimental results of this study. Position errors, velocity errors, torque values and rms position errors for varying payloads are compared to each other for the three controllers implemented in the experiments. Finally concluding remarks are given in the fifth chapter.

ALTI SERBESTLİK DERECELİ BİR ROBOT MANİPÜLATÖRÜN ADAPTİF KONTROLÜ

ÖZET

Endüstriyel robot manipülatörler endüstride tut ve yerleştir, boyama, paketleme, montaj, kaynak ve malzeme işleme gibi endüstriyel uygulamalarda kullanılır. Robotların performanslarının iyileştirilmesi, robotların kullanıldığı görevlerdeki verimi ve doğruluğu artırmak için önemli bir görevdir. Kontrol algoritması robotun doğruluk, yörünge izleme hatası, kontrol eforu gibi performas kriterlerini etkileyen önemli bir parçasıdır. Robotun dinamik modelini temel alan kontrol algoritmalarının performansı dinamik modeldeki belirsizlikler hesaba katılmadığı takdirde kötüleşebilir. Bu belirsizliklerle başa çıkmak için genellikle uyarlamalı ve dayanıklı kontrol kullanılır. Uyarlamalı kontrol bu belirsizlikleri hesaba katmak için gerçek zamanlı bir parametre uyarlama mekanizması bulundurmaktadır. Uyarlamalı kontrolün dayanıklı kontrole göre gerçek zamanlı performans iyileştirmesi avantajı bulunmaktadır. Bu yüzden robotun dinamik modelindeki bozucu etkiler ve belirsizliklerin varlığında uyarlamalı kontrol cazip bir yaklaşımdır.

Robot manipülatörlerin uyarlamalı kontrolü ile ilgili literatürde teorik çalışmalar bulunmakla birlikte bu çalışmalarda genellikle altı serbestlik dereceli robot manipülatör gibi karmaşık ve yüksek derecede nonlineer sistemler üzerinde deneysel çalışma yapılması konusunda eksiklik bulunmaktadır. Teorik tasarım, analiz ve bilgisayar simülasyonları genellikle sürtünme etkisi, modellenmemiş yüksek frekans dinamikleri ve ölçüm gürültüsü gibi kontrol sisteminin kararlılığı ve performansını etkileyen etkileri ihmal etmektedir. Bu nedenle teorik analiz ve simülasyonların sonuçlarını doğrulamak için deneysel çalışmalar gerekmektedir.

Bu çalışmada daha önceki bir çalışmada önerilen uyarlamalı kontrol algoritmasının altı serbestlik dereceli robot manipülatöre uygulanması sunulmaktadır. Ayrıca sürekli hal hatalarını azaltmak amacıyla orijinal kontrol kanununa bir integral terimi de eklenmiştir. Uygulanan uyarlamalı kontrol algoritmasının performansını analiz etmek için değişken yüklerde belli bir yörüngeyi izlemesi sağlanmıştır. Deneyler Stäubli RX160 robot manipülatörü üzerinde gerçekleştirilmiştir. Uyarlamalı kontrol algoritmasının performansı değişken yükler için hesaplanmış tork kontrolü ve pasiflik bazlı kontrol olmak üzere iki uyarlamalı olmayan kontrol yaklaşımının performasıyla karşılaştırılmıştır.

Bu tez beş bölümden oluşmaktadır. Birinci bölümde tezin amacı, literatür taraması ve deneylerde kullanılan endüstriyel robot sistemi ile ilgili temel bilgiler verilmektedir.

Robotun genel ölçüleri, çalışma uzayı, kontrolcüsü ve kontrol algoritmalarının uygulanmasına imkan tanıyan yazılım arayüzü ile ilgili temel bilgiler verilmiştir.

İkinci bölüm robot manipülatörlerin kinematik ve dinamik modellemesinin temellerinin yanı sıra sürtünme ve yay etkileri için kullanılan modellerden bahsetmektedir.

Öncelikle konum ve oryantasyonun kartezyen koordinatlarda temsil edilmesine yönelik temel bilgiler verilmiştir. Eksen dönüşümleri için kullanılan dönme matrisleri ve oryantasyonun temsil edilmesinde kullanılan Euler açıları açıklanmıştır.

Robotun ileri kinematik modeli verilen eklem açıları için robotun uç işlevcisinin konum ve oryantasyonunu veren modeldir. İleri kinematik modelin elde edilmesinde kullanılan Denavit-Hartenberg yöntemi açıklanmıştır. Ters kinematik model ise verilen robot uç işlevcisi konum ve oryantasyonu için gereken eklem açılarını veren modeldir. Ters kinematik modelin elde edilmesi için kinematik ayrıştırma yönteminden bahsedilmiştir. Hız kinematiği eklem açısal hızları ile robot uç işlevcisinin doğrusal ve açısal hızları arasındaki ilişkiyi vermektedir. Hız kinematiğine ilişkin denklemler verilmiştir.

Robotun dinamik modeli robotun hareketi ile üzerine etkiyen kuvvet ve momentlerin arasındaki bağlantıyı gösteren modeldir. Dinamik modelleme için Euler-Lagrange, Newton-Euler ve değiştirilmiş Newton-Euler yöntemleri anlatılmıştır. Euler-Lagrange yöntemi daha çok kontrol sistemi tasarımı için kullanılırken Newton-Euler yöntemi iteratif yapısı sayesinde daha hızlı hesaplama imkanı sağladığından dinamik modelin simülasyonlarda veya kontrolcüde gerçek zamanlı olarak hesaplanmasında kullanılır.

Newton-Euler yöntemi hesaplanmış tork kontrolcüsünde dinamik modelin gerçek zamanlı olarak hesaplanmasında kullanılırken pasiflik bazlı ve uyarlamalı kontrolcüde ikinci bir hız ifadesinin tanımlanmasından dolayı kullanılamamaktadır. Bu nedenle bu kontrolcülerde değiştirilmiş Newton-Euler modeli kullanılmaktadır. Değiştirilmiş Newton-Euler modeli ayrıca uyarlamalı kontrolcünün yapısında bulunan regresör matrisinin hesaplanmasında da kullanılmaktadır.

Ayrıca sürtünme ve yay etkileri için aynı robot üzerinde daha önce gerçekleştirilen bir çalışmada baz alınan model ve tanılanan parametre değerleri verilmiştir. Sürtünme için Coulomb, viskoz ve Striebeck etkilerini modelleyen bir model kullanılırken yay etkileri için kullanılan robotun karakteristiğine uygun olarak nonlineer bir model kullanılmıştır.

Üçüncü bölüm deneylerde uygulanan kontrol algoritmaları için teorik altyapıyı vermektedir. Hesaplanmış tork kontrolü, pasiflik bazlı kontrol ve adaptif kontrol için kontrol kanunları ve blok diyagramları verilmektedir.

Hesaplamalı tork kontrolü ve pasiflik bazlı kontrolcü için konum ve hız hatalarının sıfıra yakınsamasının dinamik modelin tamamıyla bilindiği varsayımına bağlı olduğu, uyarlamalı kontrolcü için bu varsayıma gerek olmadan konum ve hız hatalarının sıfıra yakınsadığı belirtilmiştir. Uyarlamalı kontrolcünün kontrol kanununa eklenen integral terimi ile birlikte Lyapunov kararlılık analizi yapılmış ve konum hatasının sıfıra yakınsaması için pozitif tanımlı köşegen kazanç matrislerinin yeterli olduğu gösterilmiştir. Pasiflik bazlı kontrol ile uyarlamalı kontrol algoritmasındaki benzerliğe dikkat çekilmiştir. İki kontrolcü yapısı arasındaki temel farkın adaptasyon kanunu ve eklenen integral terimi olduğu belirtilmiştir.

Dördüncü bölüm bu çalışmanın deneysel sonuçlarını vermektedir. Hesaplamalı tork kontrolü, pasiflik bazlı kontrol ve uyarlamalı kontrol olmak üzere üç kontrolcünün değişken yükler taşıması durumunda yörünge izleme performansları karşılaştırılmıştır. Değişken yüklerden kaynaklanan dinamik modeldeki belirsizliğin uygulanan üç kontrolcünün performansını nasıl etkilediği irdelenmiştir.

Yörüngeler eklem uzayında oluşturulmuştur. Oluşturulan yörüngeler seç ve yerleştir uygulamalarını temsil etmektedir. Taşınan değişken yüklerin dinamik modeldeki belirsizliği temsil edebilmesi sebebiyle seç ve yerleştir uygulamaları seçilmiştir. Yörüngeler robotun çalışma uzayında rastgele seçilen 15 noktanın ardışık olarak eklem uzayında beşinci mertebeden polinomlarla birleştirilmesiyle oluşturulmuştur.

Deneyler yüksüz durum ve 3 kg, 5 kg, 7 kg ve 9 kg yükün taşındığı durumlar için tekrarlanmıştır. Değişken yükler için konum hataları, hız hataları, tork değerleri ve rms konum hataları deneylerde uygulanan üç kontrolcü için karşılaştırılmıştır.

Uyarlamalı ve pasiflik bazlı kontrolcüler için konum hatalarının maksimum değerleri altıncı eklem dışında 5 dereceyi aşmazken hesaplamalı tork kontrolünde bu değer 50 dereceye kadar çıkmaktadır. Altıncı eklem için ise maksimum konum hatası değerleri pasiflik bazlı ve hesaplamalı tork kontrolcüler için 10 dereceyi bulmakta, uyarlamalı kontrolcü içinse 20 dereceye ulaşmaktadır. Pasiflik bazlı kontrolcü ile uyarlamalı kontrol arasındaki temel farkın adaptasyon kazancı olduğu düşünülürse adaptasyon kazancının ince ayarlanması uyarlamalı kontrolcünün performansını artırmaya yardımcı olabileceği düşünülebilir.

Tork değerlerine bakıldığında ise üç kontrolcü için benzer tork değerleri görülmekle birlikte uyarlamalı kontrolcüde çatırtı etkileri gözlenmektedir. Kontrol kanununa eklenen integral teriminin süreksiz yapısından dolayı bu etkiye sebep olduğu tahmini yürütülebilir.

Son olarak sonuçlara yönelik açıklamalar beşinci bölümde verilmektedir. Deneysel sonuçlar konum hatalarının rms değerlerindeki taşınan yüklere göre değişiminin altıncı eklem dışındaki eklemlerde genellikle uyarlamalı kontrolcü için diğer kontrolcülere göre daha az olduğunu göstermiştir. Hesaplamalı tork kontrolcüsünün ise yük değişimlerinden önemli ölçüde etkilendiği görülmektedir.

RMS konum hatalarındaki değişimlerin ortalama değerleri uyarlamalı kontrolcü için %43, pasiflik bazlı kontrolcü için %95, hesaplamalı tork kontrolcüsü için %318 olarak bulunmuştur. Buna göre uyarlamalı kontrolcünün yüklerdeki değişime en az duyarlı kontrolcü, hesaplamalı tork kontrolcüsünün ise en duyarlı kontrolcü olduğu görülmüştür.

Uyarlamalı kontrolcü tarafından gerçek zamanlı olarak hesaplanan dinamik parametrelere bakıldığında ise bu parametrelerin tam olarak gerçek değerlerine yakınsamadığı söylenebilir. Adaptasyon kanununun dinamik parametreleri yalnızca konum ve hız hatalarına bağlı olarak güncellediği göz önüne alındığında bu durumun beklendiği söylenebilir.

1. INTRODUCTION

According to the International Federation of Robotics, the definition of the industrial robot based on ISO 8373:2012 is an "automatically controlled, reprogrammable multipurpose manipulator, programmable in three or more axes, which can be either fixed in place or mobile for use in industrial automation applications." [1]. Industrial robots are increasingly used in industrial applications such as pick and place, painting, packaging, assembly, welding, and machining. As stated in [2], there are approximately 3 million industrial robot units in operation in 2021. The number of annual installations of industrial robot units are given in Figure 1.1.

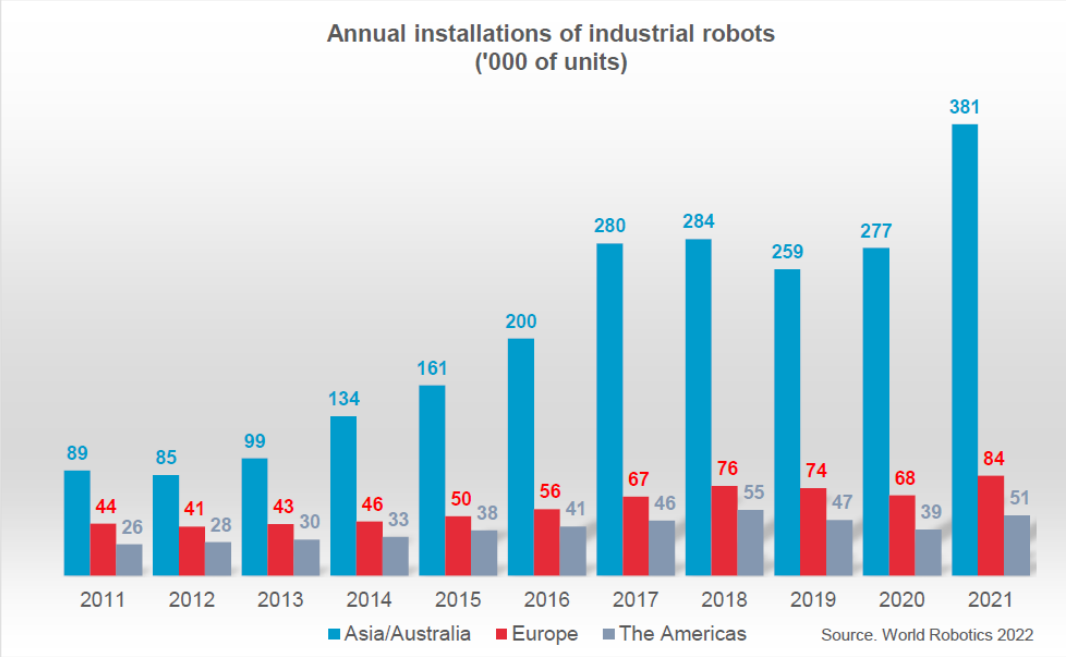

Figure 1.1 : Annual installations of industrial robot units [2].

Industrial robots are used for increasing productivity, efficiency and quality of products. Therefore improving the performance of the robots is desired for increasing efficiency. Since control algorithms implemented by the software of the robot are the key components for improving performance parameters such as accuracy, trajectory tracking error and control effort; design of different control algorithms for improving the performance of the robot manipulators is a topic of interest for researchers.

In the rest of this chapter, purpose of thesis is given followed by a literature review about the related topic and information about industrial robot system used in this study.

1.1 Purpose of Thesis

Dynamical models of serial robot manipulators are generally highly nonlinear and coupled. There are also uncertainties in the dynamic model of the robot. These uncertainties usually stem from effects such as uncertainties in the dynamic parameters of the robot, carrying an unknown load, and unknown forces acting on the robot. Therefore the performance of the control algorithms based on the dynamic model of the robot may degrade under these uncertainties when they are not accounted for. Two main approaches to take these uncertainties into account are adaptive and robust control. Adaptive control has the advantage of the online improvement of performance over robust control. In this thesis, an adaptive control algorithm proposed by Slotine and Li (1987) is implemented to improve the performance of the robot. The performance of the adaptive control algorithm is compared to the computed torque control algorithm and passivity-based control algorithm.

1.2 Literature Review

Presence of nonlinearities and uncertainties in the dynamic model of the robots makes adaptive control an attractive approach for the control of robots. Adaptive control can be categorized into three categories as model reference adaptive control, self-tuning, and gain scheduled control [3]. Several studies on adaptive control of robot manipulators are listed in [3] and [4]. One of the early studies done in this research area is (Dubowski and DesForges, 1979). Authors applied adaptive control by modifying the position and velocity gains by an adaptive update rule [5]. Tomizuka and Horowitz (1983) proposed an adaptive control algorithm that includes the dynamic model of the robot in the control law to compensate for nonlinearities and updates the dynamic model of the robot by an adaptive update rule in [6]. In [7] computed torque method is employed with an adaptation algorithm. This adaptation algorithm that updates the dynamic parameters of the robot requires the computation of the regressor matrix and inverse of the mass matrix. Slotine and Li (1987) proposed an adaptive control algorithm that consists of a feedforward part where the dynamic model of the robot is used and a feedback part which is essentially a PD control term in [8]. Dynamic

parameters of the robot are updated with an adaptation rule that eliminates the need for the inverse of the mass matrix but requires the computation of the regressor matrix of the robot. However, computation of the regressor matrix gets more complex as the degrees of freedom of the robot increases.

1.3 Industrial Robot System

Industrial robot system used in the experiments consists of Stäubli RX160 industrial robot manipulator, CS8C controller, manual control pendant and Low Level Robot Control Interface(LLI).

Stäubli RX160 industrial robot manipulator is a 6 DOF robot manipulator with 6 revolute joints. Structure of the manipulator is articulated arm with a spherical wrist as shown in Figure 1.2.

Figure 1.2 : Stäubli RX160 industrial robot manipulator [9].

RX160 robot manipulator weighs 248 kg and nominal and maximum payload capacity of the robot is 20 kg and 34 kg respectively [9].

Dimensions of the RX160 robot manipulator is given in Figure 1.3.

Workspace of the RX160 robot manipulator is given in Figure 1.4 and technical details about range, speed and angular resolution of the joints of the robot are given in Table 1.1.

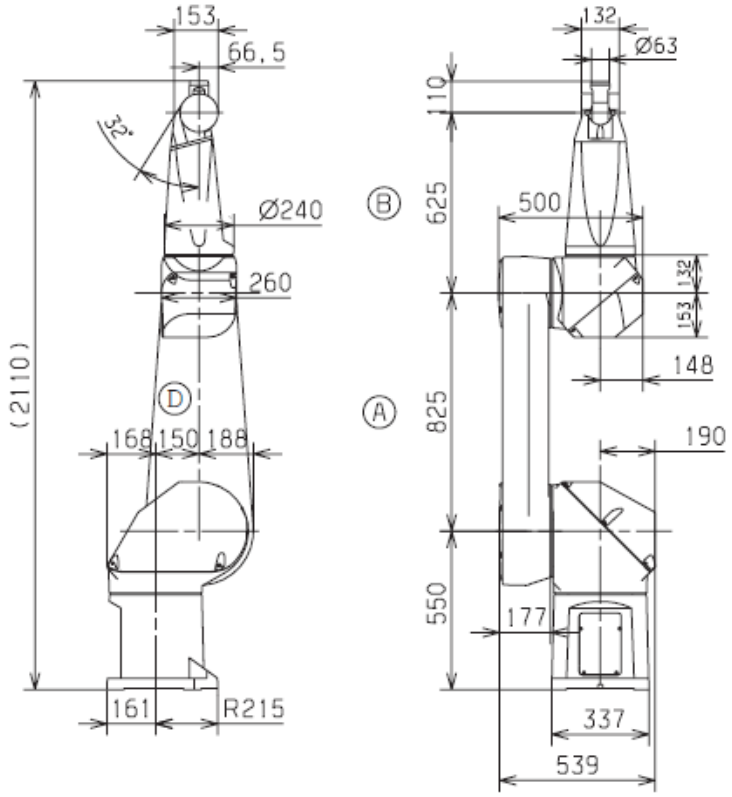

Figure 1.3 : Dimensions of Stäubli RX160 industrial robot manipulator [9].

Joints of the robot are actuated with brushless DC motors. Brushless motors move the joints through a transmission mechanism that consists of cycloidal transmission and helical gears with pre-stressed ball-bearings for the first four joints.

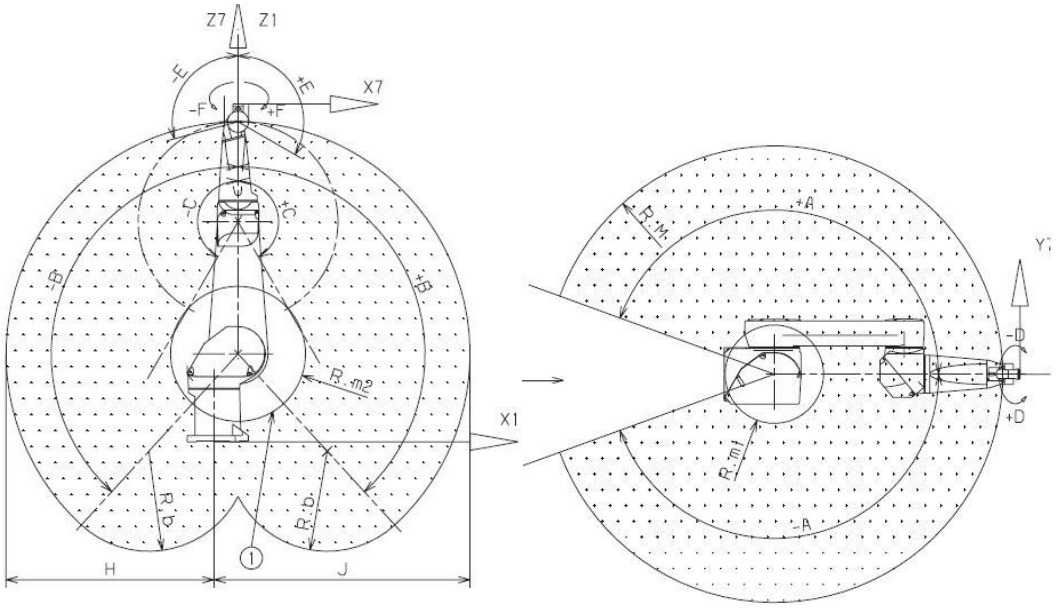

Figure 1.4 : Workspace of RX160 industrial robot manipulator [9].

Table 1.1 : Amplitude, speed and resolution of joints of RX160.

Axis	1	2	3	4	5	6
Amplitude (^{o})	320	275	300	540	225	540
Range (^{o})	±160	±137.5	±150	±270	+120,-105	±270
Nominal Speed (^{o}/s)	165	150	190	295	260	440
Maximum Speed (^{o}/s)	200	200	255	315	390	870
Angular Resolution ($^o.10^{-3}$)	0.042	0.042	0.054	0.062	0.12	0.17

Joints 5 and 6 consists of a differential coupling mechanism. RX160 is equipped with resolvers to provide angular position and velocity feedback and proprioceptive sensors to provide torque feedback.

Controller of the robot is Stäubli CS8C controller which is part of the CS8 controller series. CS8C controller consists of a processor, digital power amplifiers, servo drivers and digital I/O interfaces. CS8C controller uses a PENTIUM® arithmetic unit as the processor and VxWorks® real-time operating system as the operating system. CS8C controller is shown in Figure 1.5 and general architecture of the controller is given in Figure 1.6.

Figure 1.5 : CS8C Controller [9].

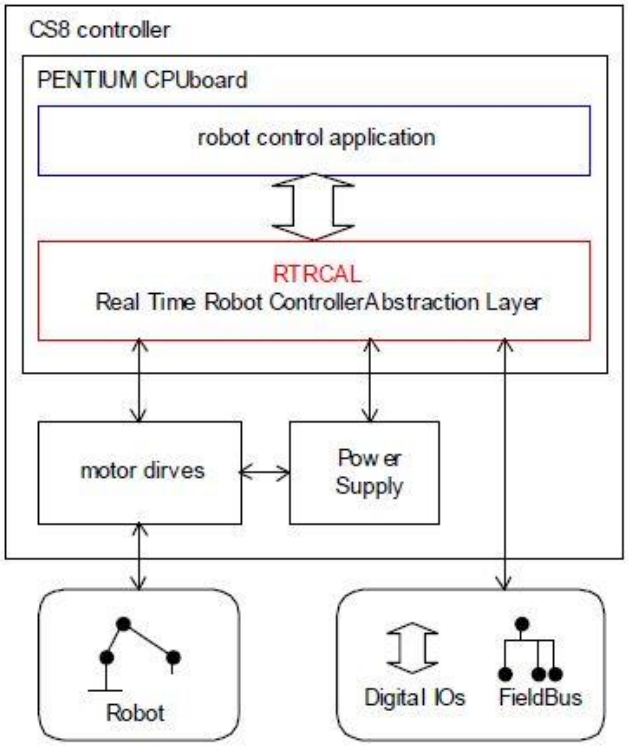

Figure 1.6 : General architecture of CS8 controller [10].

Manual control pendant (MCP) is also available for the robot used for controlling the movements of the robot with provided buttons by hand. MCP is shown in Figure 1.7.

Figure 1.7 : Manual Control Pendant [9].

Tasks such as implementation of the designed control algorithm, data acqusition from the sensors of the robot and use of external sensors are handled through software interface of the robot. LLI is the robot control interface provided by the manufacturer to handle these tasks. C/C++ is used as the programming language in the LLI.

Structure of the control system of a joint is shown in Figure 1.8. There are two control modes for the robot as default defined as ‘Position-Velocity’ mode and ‘Torque’ mode.

In Position-Velocity mode only position and velocity references are given and embedded control law in the controller computes the required torque value while the control law that computes the required feedforward torque is implemented by the user in the torque mode.

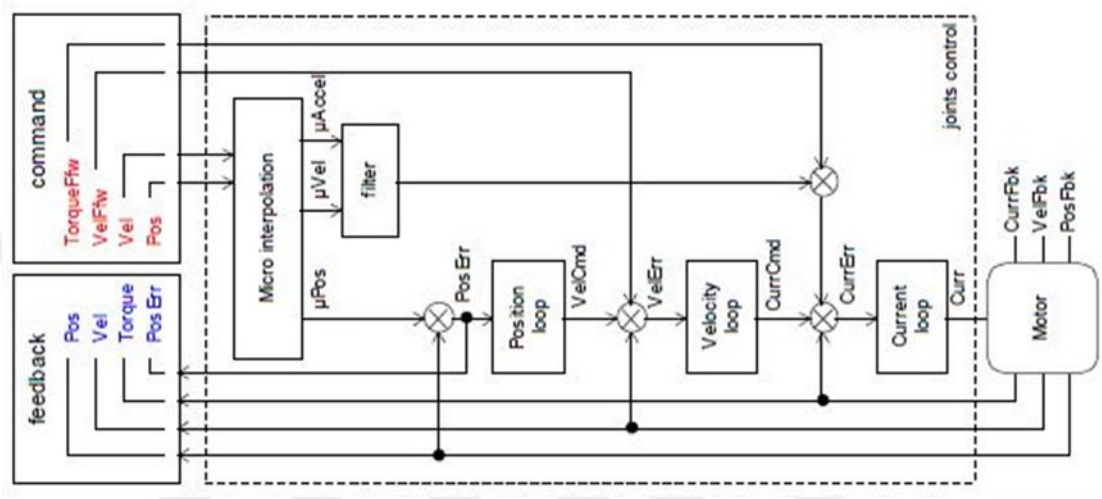

Figure 1.8 : Structure of the joint control system[10].

Microsoft® Visual Studio® IDE is used to edit the code used in the robot. MATLAB® is used for analysis and design of the control system.

Rest of the thesis is organized as follows. In the second chapter kinematic and dynamic modeling of robot manipulators are introduced. Theoretical background on computed torque control, passivity-based control and adaptive control are given in the third chapter. Experimental results and conclusions are given in fourth and fifth chapter respectively.

2. MATHEMATICAL MODELING

In this chapter firstly, kinematic modeling of the robot manipulators is explained. Then different algorithms used in dynamic modeling of the robot are introduced. Finally complete dynamic model of the robot is given.

2.1 Kinematic Modeling

Kinematic modeling is used for describing the motion of the robot. Torques or forces exerted on the robot are not taken into account in kinematic modeling. Firstly, basic information about position and orientation representation is given. Secondly, forward kinematics which gives the end effector position and orientation as a function of joint values is explained. Then inverse kinematics that determines the required joint angles for given end effector position and orientation is explained. Finally velocity kinematics that describes the relationship between joint variables and linear and angular velocities of the end effector is introduced.

2.1.1 Representation of position and orientation

End effector position of the robot is expressed with respect to a coordinate frame as shown in Figure 2.1. Position of the end effector can be represented using Cartesian, Cylindirical and Spherical coordinates. Generally, Cartesian coordinates are used for defining robot tasks and written as follows:

$$\boldsymbol{p} = \begin{bmatrix} \boldsymbol{p_x} \\ \boldsymbol{p_y} \\ \boldsymbol{p_z} \end{bmatrix} \tag{2.1}$$

Orientatiton of the end effector can be represented in different forms. One way of representing the orientation is rotation matrix. A rotated coordinate frame is shown in Figure 2.2. Rotation matrix maps a vector expressed in the rotated frame into Frame 0.

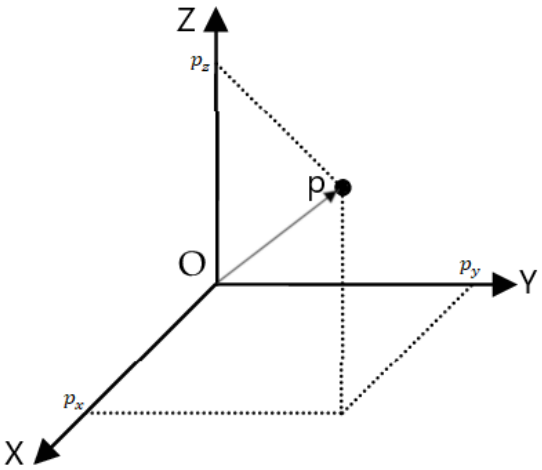

Figure 2.1 : Representation of Position in Cartesian Coordinates.

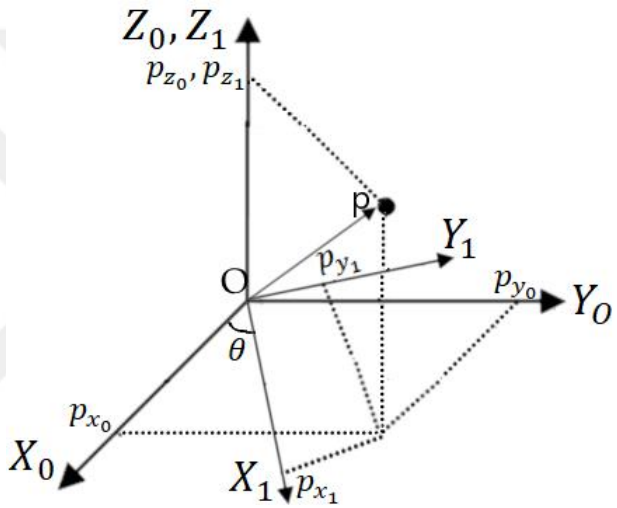

Figure 2.2 : Coordinates of a vector w.r.t. Two Coordinate Frames.

$$^0p = R_1^0\ ^1p \tag{2.2}$$

The rotation matrix R_1^0 in equation 2.2 can be found by writing $p_{x_0}, p_{y_0}, p_{z_0}$ in terms of $p_{x_1}, p_{y_1}, p_{z_1}$. Any arbitrary rotation can be written as a combination of elementary rotation matrices given in equation 2.3.

Another way of representing orientation is Euler angles. Euler angles parameterize the rotations with 3 independent parameters. Considering the rotated frame shown in Figure 2.3 orientation of the rotated frame can be parameterized by three successive rotations about X,Y' and Z" axes respectively.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$
$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \quad (2.3)$$
$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Euler angles ϕ,θ and ψ represents the orientation of the frame and since rotations occur in X,Y',Z" order it is called XYZ Euler angles. There are other Euler angle sets such as ZYZ, ZXZ, ZYX named according to order of rotation.

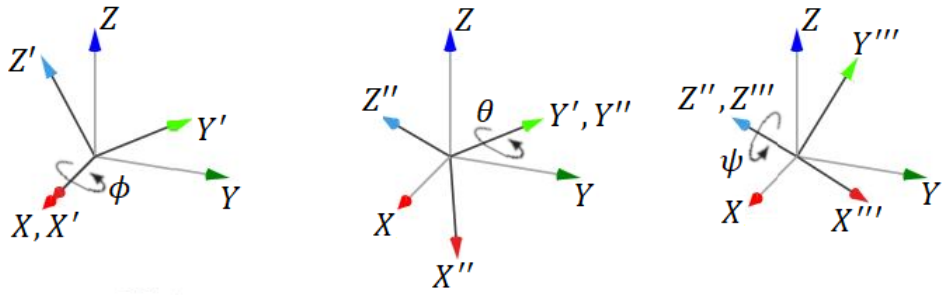

Figure 2.3 : Euler Angles.

Corresponding rotation matrix for the given Euler angles is given in equation 2.5 as

$$R_{XYZ} = R_x(\phi)R_y(\theta)R_z(\psi) \quad (2.4)$$

$$R_{XYZ} = \begin{bmatrix} \mathrm{c}(\theta)\,\mathrm{c}(\psi) & -\mathrm{c}(\theta)\,\mathrm{s}(\psi) & \mathrm{s}(\theta) \\ \mathrm{c}(\phi)\,\mathrm{s}(\psi) + s(\phi)s(\theta)c(\psi) & \mathrm{c}(\theta)\,\mathrm{c}(\psi) - s(\phi)s(\theta)s(\psi) & -\mathrm{c}(\theta)\,\mathrm{s}(\phi) \\ \mathrm{s}(\theta)\,\mathrm{s}(\psi) - c(\phi)s(\theta)c(\psi) & \mathrm{s}(\phi)\,\mathrm{c}(\psi) + c(\phi)s(\theta)s(\psi) & \mathrm{c}(\theta)\,\mathrm{c}(\phi) \end{bmatrix} \quad (2.5)$$

where $\mathrm{c}(.)$ and $\mathrm{s}(.)$ are $\cos(.)$ and $\sin(.)$.

If the rotation matrix is known then Euler angles can be determined as follows:

$${}^0p = {}^0d + R_1^0\,{}^1p \quad (2.7)$$

If there is a long sequence of rotations and translations equation 2.7 becomes complex. The homogeneous transformation matrix represents this motion in a more compact way as in equation 2.8. Equation 2.7 is rewritten with the homogeneous transformation matrix as in equation 2.9.

$$T_1^0 = \begin{bmatrix} R_1^0 & {}^0d \\ 0 & 1 \end{bmatrix} \quad (2.8)$$

$$\begin{bmatrix} {}^{0}p \\ 1 \end{bmatrix} = T_1^0 \begin{bmatrix} {}^{1}p \\ 1 \end{bmatrix} \tag{2.9}$$

2.1.2 Forward kinematics

Forward kinematic model of the robot gives the end effector position and orientation as output for given joint variables. Forward kinematic model of the robot is found by attaching coordinate frames to each link and using homogeneous transformation matrices to represent the position and orientation of the frames moving with the links. Denavit-Hartenberg convention is used to obtain the forward kinematic model which is widely used in robotics. Denavit-Hartenberg convention can be explained as follows:

- z_i axis is placed as the axis of revolution of joint i+1.
- x_i axis is placed in the direction of any common normal between z_{i-1} and z_i if they are coplanar or parallel. If z_{i-1} and z_i intersects each other then x_i axis is placed in the direction that is normal to the plane formed by z_{i-1} and z_i axes.
- y_i axis is placed so that right hand rule is satisfied.
- After frame assignment, 4 parameters are defined to specify the homogeneous transformation matrix T_i^{i-1}: $a_i, \alpha_i, d_i, \theta_i$. a_i is the length of the common normal line, α_i is the angle between z_{i-1} and z_i about x_{i-1}, d_i is the distance between x_{i-1} and x_i along z_{i-1} and θ_i is the angle between x_{i-1} and x_i about z_{i-1}. A table of the DH parameters is prepared for the robot.
- Homogeneous transformation matrices are computed with the DH parameters as follows:
- $T_i^{i-1} = Trans(z_{i-1}, d_i)Rot(z_{i-1}, \theta_i)Trans(x_{i-1}, a_i)Rot(x_{i-1}, \alpha_i)$.

Forward kinematic model of the robot is obtained by multiplying the homogeneous transformation matrices.

Coordinate frames assigned to RX160 robot are shown in Figure 2.4.

DH table for the RX160 robot is given in Table 2.1.

DH parameters shown in the DH table are used to compute homogeneous transformation matrices. Then forward kinematic model of the robot is computed as in equation 2.10.

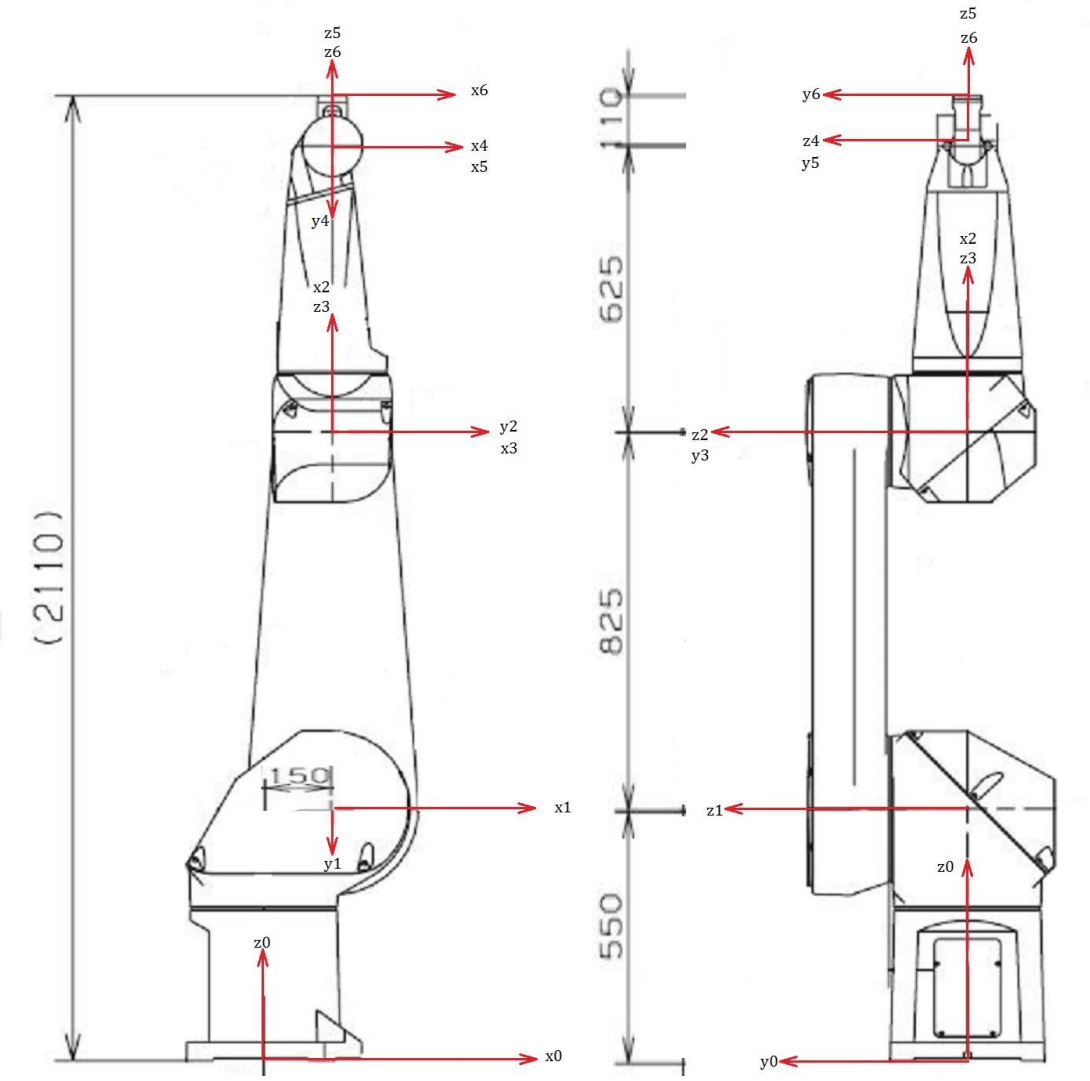

Figure 2.4 : Coordinate Frame Assignment to RX160 Robot.

Table 2.1 : DH Table of RX160.

i	d[m]	θ[rad]	a[m]	α[rad]
1	0.55	q_1	0.15	$-\pi/2$
2	0	$q_2 - \pi/2$	0.825	0
3	0	$q_3 + \pi/2$	0	$\pi/2$
4	0.625	q_4	0	$-\pi/2$
5	0	q_5	0	$\pi/2$
6	0.11	q_6	0	0

$$T_6^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 \tag{2.10}$$

First three element of the fourth column of the T_6^0 represents the position of the end effector and upper left 3x3 part of the T_6^0 represents the orientation of the end effector.

2.1.3 Inverse kinematics

Inverse kinematic model of the robot gives possible values of joint variables as output for given end effector position and orientation. Finding an analytical solution to inverse kinematic problem gets more complex as the degrees of freedom of the robot increases since it generally involves solving a set of nonlinear equations. One approach to solve the inverse kinematic problem for a 6 DOF serial robot manipulator with a spherical joint at the end effector is kinematic decoupling. In this method, inverse kinematic problem is solved by treating the end effector position and orientation independently from each other. Firstly; since rotation of last three joints doesn't effect the position of the intersection of the last three axes which is called as wrist center, first three joint angles are found from the desired position of the wrist center. Then, last three joint angles are found from the desired orientation of the wrist.

2.1.4 Velocity kinematics

Velocity kinematics describes how joint velocities and end effector velocities are related to each other. Forward velocity kinematics gives the function that outputs linear and angular velocities of the end effector for given joint velocities. This relationship is given by jacobian matrix as shown in equation 2.11.

$$\dot{x}_e = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} J_T(q) \\ J_R(q) \end{bmatrix} \dot{q} = J(q)\dot{q} \tag{2.11}$$

Details on calculating the jacobian matrix are given in [11].

Inverse velocity kinematics outputs the joint velocities for given position and orientation of the end effector as shown in equation 2.12.

$$\dot{q} = J^{-1}(q)\dot{x}_e \tag{2.12}$$

2.2 Dynamic Modeling

Dynamic modeling gives a mathematical description for the robot motion by taking forces and torques into account. There are two widely used methods used for deriving the dynamic model of the robot manipulators: Euler – Lagrange equations and Newton – Euler formulation. While Euler – Lagrange equations are generally used in design of control algorithms, Newton – Euler formulation is generally used for numerical

calculations in the real time implementation of the control algorithm and simulation. There is also a modified version of the Newton – Euler method developed for implementation of passivity-based control algorithms which can't be implemented with the standard Newton - Euler formulation. This method is discussed in detail in [12]. There are other dynamic effects such as friction and gravity counter-balance spring. Euler – Lagrange method, Newton – Euler method and other dynamic effects are explained in the rest of the chapter.

2.2.1 Euler – Lagrange method

Euler – Lagrange equations that gives the equation of motion for the n degrees of freedom mechanical systems are given as equation 2.13 where Lagrangian $L = K - P$ is defined as difference between kinetic and potential energy of the system.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_k}\right) - \frac{\partial L}{\partial q_k} = \tau_k \,, \quad k = 1, \dots, n \tag{2.13}$$

General form of the equation of the motion for the robot manipulators are given as equation 2.14 where $M(q)$ is defined as inertia matrix, $C(q, \dot{q})$ is the centrifugal – coriolis matrix and $g(q)$ is the gravity vector.

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \tag{2.14}$$

Details on derivation of the equations and computation of the matrices are given in [11]. Here the computation of the matrices are given as equation 2.15, 2.16 and 2.17 where m_i and I_i are the mass and inertia tensor of each link, respectively.

$$M(q) = \sum_{i=1}^{n} \left(m_i {J_T}_i^T J_{T_i} + {J_R}_i^T R_i^0 I_i {R_i^0}^T J_{R_i} \right) \tag{2.15}$$

$$C_{kj} = \sum_{i=1}^{n} \left(\frac{\partial M_{kj}}{\partial q_i} - \frac{1}{2}\frac{\partial M_{ij}}{\partial q_k} \right) \dot{q}_i \,, \quad k = 1, \dots, n \,, j = 1, \dots, n \tag{2.16}$$

$$g_k = \frac{\partial P}{\partial q_k} \,, \quad k = 1, \dots, n \tag{2.17}$$

2.1.2 Newton - Euler Method

In Newton – Euler method equations of motion for each link of the robot written indivudually. Which results in recursive equations to calculate forces and torques that

produces given trajectory q(t). Detailed derivation of the equations are given in [11]. Pseudocode for the recursive Newton – Euler algorithm is given as follows:

$$for\ i = 1, \dots, n$$

$$\omega_i^{(0)} = \omega_{i-1}^{(0)} + \dot{q}_i z_i \tag{2.18}$$

$$\dot{\omega}_i^{(0)} = \dot{\omega}_{i-1}^{(0)} + \dot{q}_i \omega_{i-1}^{(0)} \times z_i + \ddot{q}_i z_i \tag{2.19}$$

$$\mu_i^{(0)} = \mu_{i-1}^{(0)} + \dot{\omega}_{i-1}^{(0)} \times r_{i-1,i} + \omega_{i-1}^{(0)} \times \omega_{i-1}^{(0)} \times r_{i-1,i} \tag{2.20}$$

$$for\ i = n, \dots, 1$$

$$f_i^{(0)} = f_{i+1}^{(0)} + m_i \mu_i^{(0)} + m_i \dot{\omega}_i^{(0)} \times r_{i,ci} + m_i \omega_i^{(0)} \times \omega_i^{(0)} \times r_{i,ci} \tag{2.21}$$

$$n_i^{(0)} = n_{i+1}^{(0)} + r_{i,i+1} \times f_{i+1}^{(0)} + m_i \mu_i^{(0)} \times r_{i,ci} + I_i \dot{\omega}_i^{(0)} + \omega_i^{(0)} \times \left(I_i \omega_i^{(0)} \right) \tag{2.22}$$

$$\tau_i^{(0)} = z_i^T n_i^{(0)} \tag{2.23}$$

Where initial conditions are given as $\omega_0^{(0)} = \dot{\omega}_0^{(0)} = \mu_i^{(0)} = f_{n+1}^{(0)} = n_{n+1}^{(0)} = 0_{3\times1}$, $r_{i-1,i}$ is the distance between coordinate frames i-1 and i and $r_{i,ci}$ is the distance between coordinate frame and center of mass of link i. Superscript $.^{(0)}$ indicates that the vector is expressed in the base frame.

Pseudocode for the modified Newton – Euler algorithm presented in [12] is given as follows:

$$for\ i = 1, \dots, n$$

$$\omega_i^{(0)} = \omega_{i-1}^{(0)} + \dot{q}_i z_i \tag{2.24}$$

$$\omega_{r_i}^{(0)} = \omega_{r_{i-1}}^{(0)} + \dot{q}_{r_i} z_i \tag{2.25}$$

$$\dot{\omega}_i^{(0)} = \dot{\omega}_{i-1}^{(0)} + \dot{q}_i \omega_{r_{i-1}}^{(0)} \times z_i + \ddot{q}_{r_i} z_i \tag{2.26}$$

$$\mu_i^{(0)} = \mu_{i-1}^{(0)} + \dot{\omega}_{i-1}^{(0)} \times r_{i-1,i} + \omega_{i-1}^{(0)} \times \omega_{r_{i-1}}^{(0)} \times r_{i-1,i} \tag{2.27}$$

$$for\ i = n, \dots, 1$$

$$f_i^{(0)} = f_{i+1}^{(0)} + m_i\mu_i^{(0)} + m_i\dot{\omega}_i^{(0)} \times r_{i,ci} + m_i\omega_i^{(0)} \times \omega_{r_i}^{(0)} \times r_{i,ci} \quad (2.28)$$

$$n_i^{(0)} = n_{i+1}^{(0)} + r_{i,i+1} \times f_{i+1}^{(0)} + m_i\mu_i^{(0)} \times r_{i,ci} + I_i\dot{\omega}_i^{(0)} + \omega_{r_i}^{(0)} \times \left(I_i\omega_i^{(0)}\right) \quad (2.29)$$

$$\tau_i^{(0)} = z_i^T n_i^{(0)} \quad (2.30)$$

2.1.3 Friction and Spring Model

Friction and gravity counter-balance spring models are used in the dynamic model to compansate the torques caused by friction and spring effects. There are several models in the literature that can be used to model friction and spring effects for robot manipulators. A friction and spring model is identified on the robot used in the experiments of this study in [13]. Friction model identified in [13] is a nonlinear friction model that includes coulomb, viscous and Striebeck effects. Friction models for joint 5 and 6 are different than other joints due to coupling mechanism between these joints. Structure of the joints are given in Figure 2.5. Spring model is also a nonlinear model that accounts for the torques caused by gravity counter-balance springs in the second joint. Identifed friction model in [13] is given as follows.

$$\tau_{f_i} = \tau_i^{(a,BL)} exp\left(-\left(\frac{\dot{q}_i}{\dot{q}_i^{(s)}}\right)^{\delta_i^{(a)}}\right) + c_i^{(v)}\dot{q}_i^{1-\delta_i^{(v)}} \quad (2.31)$$

$$\tau_{f_a} = \tau_5^{(a,BL)} exp\left(-\left(\frac{\dot{q}_5}{\dot{q}_5^{(s)}}\right)^{\delta_5^{(a)}}\right) + c_5^{(v)}\dot{q}_5^{1-\delta_5^{(v)}} \quad (2.32)$$

$$\tau_{f_b} = \tau_6^{(a,BL)} exp(-\left(\frac{\dot{q}_6}{\dot{q}_6^{(s)}}\right)^{\delta_6^{(a)}}) + \tau_6^{(v,l)}\left(1 - \exp\left(\frac{\dot{q}_6}{\dot{q}_6^{(l)}}\right)\right) \quad (2.33)$$

$$\tau_{f_c} = \tau_7^{(a,BL)} exp(-\left(\frac{\dot{q}_7}{\dot{q}_7^{(s)}}\right)^{\delta_7^{(a)}}) + c_7^{(v)}\dot{q}_7^{1-\delta_7^{(v)}} \quad (2.34)$$

$$\tau_{f_5} = \tau_{f_a} + \tau_{f_c} \quad (2.35)$$

$$\tau_{f_6} = \tau_{f_b} + \tau_{f_c} \quad (2.36)$$

Where $\dot{q}_7 = \dot{q}_5 + \dot{q}_6$. Friction model parameters identified in [13] are given in Table 2.2 and Table 2.3.

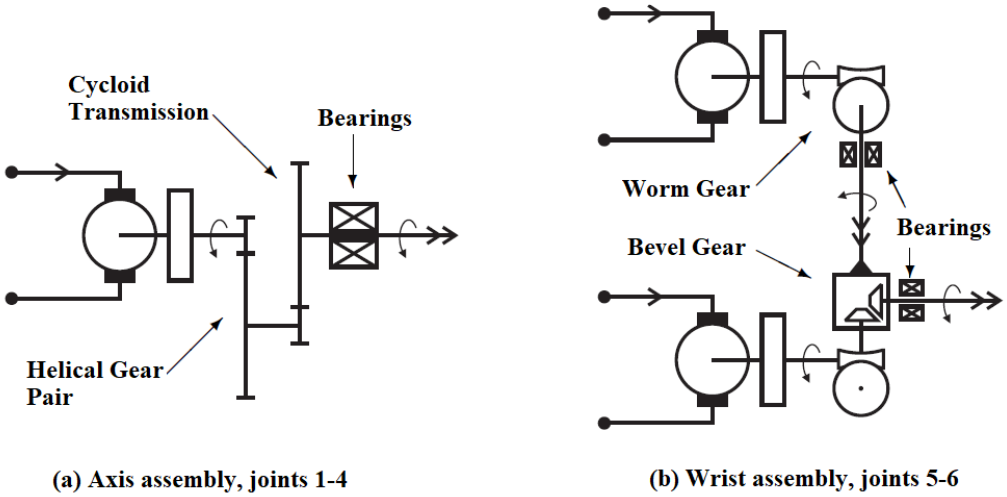

Figure 2.5 : Joint Structures of RX160 Robot Manipulator. [13]

Table 2.2: Friction Model Parameters Identified in [13] for Positive Direction.

Model	$\tau^{(a,BL)}$	$\dot{q}^{(s)}$	$\delta^{(a)}$	$c^{(v)}$	$\delta^{(v)}$	$\tau^{(v,l)}$	$\dot{q}^{(l)}$
1	35	0.26	0.8	150	0.24	-	-
2	70	0.078	0.28	132	0.352	-	-
3	15	0.09	0.35	68	0.48	-	-
4	6.36	0.06	0.57	20.738	0.56	-	-
a	5.016	0.06	0.57	24.53	0.41	-	-
b	0.5	1.6	0.6	-	-	1.16	2
c	0.48	0.04	0.1	9.04	0.25	-	-

Structure of the gravity counter-balance spring system on the second link of the robot is shown in Figure 2.6. Spring model is given as follows:

Let $r = 0.08$ m, $L = 0.728$ m and $|AC| = r + L$. Then

$$\tau_{s_2}(q_2) = (k(|BC| - L) + P_c)|AC|\sin(b) \tag{2.37}$$

Table 2.3: Friction Model Parameters Identified in [13] for Negative Direction.

Model	$\tau^{(a,BL)}$	$\dot{q}^{(s)}$	$\delta^{(a)}$	$c^{(v)}$	$\delta^{(v)}$	$\tau^{(v,l)}$	$\dot{q}^{(l)}$
1	-36.95	0.26	0.4	-140	0.24	-	-
2	-85.95	0.3	0.21	-118.5	0.28	-	-
3	-36	0.9	0.465	-58	0.3	-	-
4	-6.41	0.72	0.5	-17.5	0.46	-	-
a	-4.1	0.072	0.34	-22	0.3	-	-
b	-0.52	0.4	0.8	-	-	-0.94	0.8
c	-1.61	0.5	0.04	-18.4	0.38	-	-

Structure of the gravity counter-balance spring system on the second link of the robot is shown in Figure 2.6. Spring model is given as follows:

Let $r = 0.08$ m, $L = 0.728$ m and $|AC| = r + L$. Then

$$\tau_{s_2}(q_2) = (k(|BC| - L) + P_c)|AC|\sin(b) \tag{2.37}$$

Identified values of the spring model parameters are given in [13] as $k = 23950\ N/m$, $P_c = 5314\ N$ and $a_0 = -0.8458\ deg$.

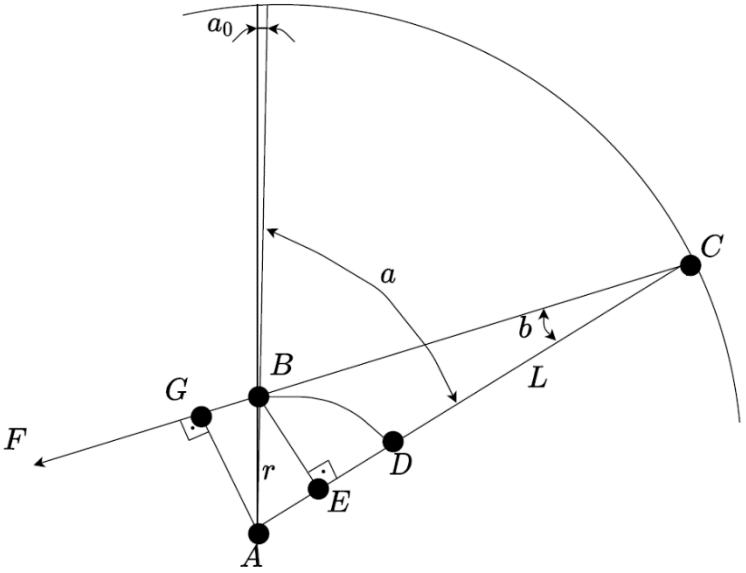

Figure 2.6 : Gravity Counter-Balance Spring System.

Hence complete dynamic model of the system is given as follows.

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) + \tau_f(\dot{q}) + \tau_s(q) = \tau \tag{2.38}$$

3. MOTION CONTROL

The objective of motion control in robotics can be defined as the design of a control law that assures the tracking of the desired trajectory. Generation of the desired trajectory depends on the task executed by the robot. Trajectories used in this study are explained in this chapter. There are several approaches to control of robot manipulators in the literature. Later, two non-adaptive controllers and an adaptive controller used in this study are explained. Non-adaptive approaches are well-known computed torque controller and passivity-based controller presented in [14]. Controllers implemented in this study are in joint space. Which means trajectory tracking is accomplished for joint variables.

3.1 Trajectory Planning

Trajectory planning is required for the execution of a specific task by the robot. A trajectory is a time sequence of points in the joint space or task space. The desired trajectory is generated as a sequence of points so that the robot executes the specified task. Details on trajectory planning methods are given in [11]. Quintic polynomial trajectories and trajectories with trapezoidal velocity profile are the trajectories used in this study.

3.2 Computed Torque Control

Computed torque control is essentially a nonlinear control technique called feedback linearization designed for the control of robot manipulators. The basic idea of the computed torque control is constructing a control law that linearizes the system by canceling out the nonlinearities. Let $\tilde{q} = q - q_d$. Then control law for the computed torque control is given as in equation 3.1. The structure of the computed torque control is shown in Figure 3.1.

$$\tau = M\left(\ddot{q}_d - K_D\dot{\tilde{q}} - K_P\tilde{q}\right) + C\dot{q} + g + \tau_f + \tau_s \tag{3.1}$$

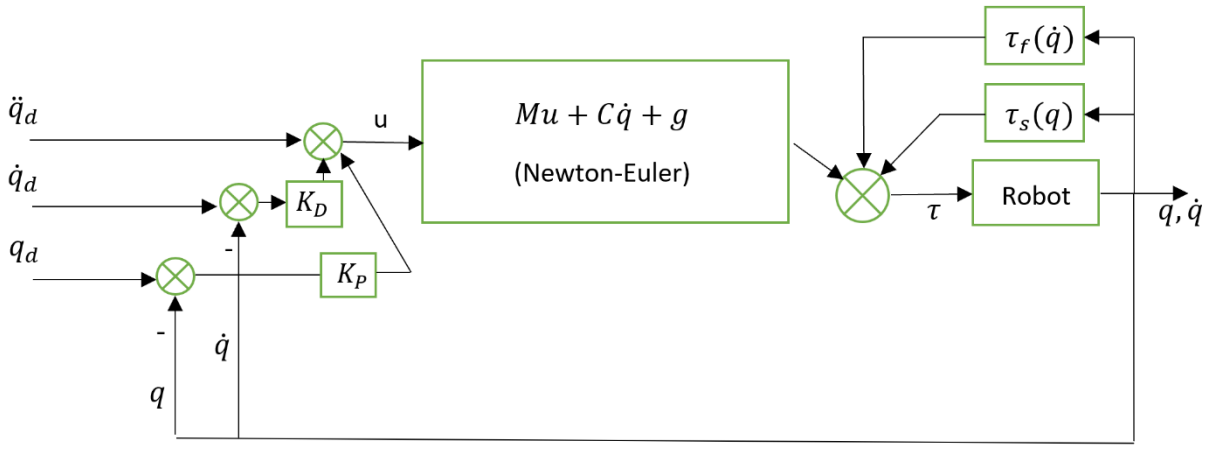

Figure 3.1 : Block Diagram of the Computed Torque Control.

Assumption of exact cancellation of nonlinearities allows one to write the following error equation.

$$\ddot{\tilde{q}} + K_D\dot{\tilde{q}} + K_P\tilde{q} = 0 \tag{3.2}$$

Which means position error of the system converges to zero for positive definite diagonal K_P and K_D matrices.

3.3 Passivity-based Control

In this section, a controller based on the skew symmetry and passivity properties of the dynamic model of the robot is considered. Details on skew symmetry and passivity properties and the passivity-based controller discussed here are given in [11]. Block diagram of the passivity-based controller is given in Figure 3.2.

Consider the following control law

$$\tau = M\ddot{q}_r + C\dot{q}_r + g + \tau_f + \tau_s - K_D s \tag{3.3}$$

where $\dot{q}_r = \dot{q}_d - \Lambda\tilde{q}$ and $s = \dot{q} - \dot{q}_r = \dot{\tilde{q}} + \Lambda\tilde{q}$. Dynamic parameters of the robot are assumed to be known and constant.

Following Lyapunov function candidate is considered for stability analysis.

$$V(t) = \frac{1}{2}s^T Ms + \tilde{q}^T\,\Lambda K_D\tilde{q} \tag{3.4}$$

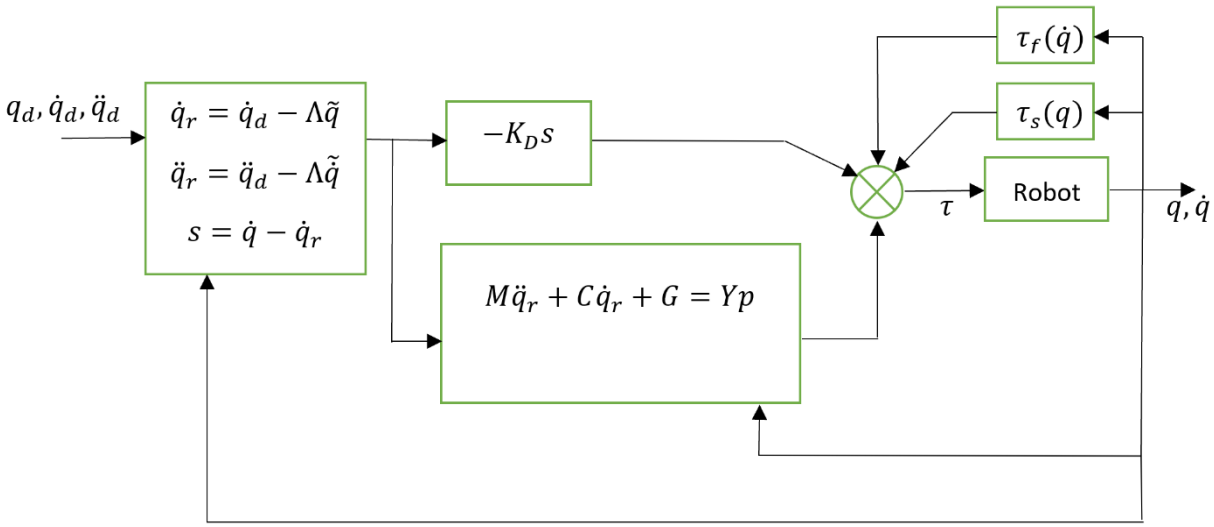

Figure 3.2 : Block Diagram of the Passivity-based Control.

Derivative of $V(t)$ with respect to time is given as follows.

$$\dot{V}(t) = s^T(M\ddot{q} - M\ddot{q}_r) + \frac{1}{2}s^T\dot{M}s + 2\dot{\tilde{q}}^T\Lambda K_D\tilde{q} \tag{3.5}$$

Equation 3.5 can be rearranged by using equation 2.38 and skew symmetry property of $\dot{M} - 2C$ as shown in equation 3.6.

$$\dot{V}(t) = s^T\left(\tau - Yp - \tau_f - \tau_s\right) + 2\dot{\tilde{q}}^T\Lambda K_D\tilde{q} \tag{3.6}$$

Equation 3.6 can be rearranged by substituting the control law in equation 3.3 as follows.

$$\dot{V}(t) = -s^T K_D s + 2\dot{\tilde{q}}^T\Lambda K_D\tilde{q} = -\dot{\tilde{q}}^T K_D\dot{\tilde{q}} - \tilde{q}^T\Lambda^T K_D\Lambda\tilde{q} \tag{3.7}$$

Then position and velocity errors converge to zero for positive definite diagonal K_D and Λ matrices that results in negative semi-definite $\dot{V}(t)$.

3.4 Adaptive Control

Controllers presented in the previous sections do not take uncertainties in the dynamic model of the robot into account. In this section, adaptive control algorithm proposed in [8] is explained.

This adaptive controller takes uncertainties in the dynamic parameters of the robot into account by updating the dynamic parameters with an online adaptation mechanism. Estimated dynamic model of the robot can be written shown in equation 3.8 by using the linearity in the parameters property of the dynamic model.

$$\widehat{M}\ddot{q}_r + \hat{C}\dot{q}_r + \hat{g} = Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)\hat{p} \tag{3.8}$$

Control and adaptation laws proposed in [8] are given as follows.

$$\tau = \widehat{M}\ddot{q}_r + \hat{C}\dot{q}_r + \hat{g} - K_D s \tag{3.9}$$

$$\dot{\hat{p}} = -\Gamma Y^T s \tag{3.10}$$

However, this control law neglects the torques caused by friction and spring effects. Therefore, aforementioned friction and spring models identified in [13] are added to the control law. An integral term is also added to the control law to reduce the steady state errors.

Let $S_{sq} = diag(sgn(s \odot \int \tilde{q} dt))$ where $\odot$ denotes element-wise multiplication. Then modified control law is given as

$$\tau = Y\hat{p} + \hat{\tau}_f + \hat{\tau}_s - K_D s - K_I S_{sq} \int \tilde{q} dt. \tag{3.11}$$

Block diagram of the adaptive control algorithm with the modified control law is given in Figure 3.3.

Following Lyapunov function candidate is considered for stability analysis.

$$V(t) = \frac{1}{2} s^T M s + \frac{1}{2} \tilde{p}^T \, \Gamma^{-1} \, \tilde{p} \tag{3.12}$$

Derivative of $V(t)$ with respect to time is given as follows.

$$\dot{V}(t) = s^T (M\ddot{q} - M\ddot{q}_r) + \frac{1}{2} s^T \dot{M} s + \dot{\hat{p}}^T \Gamma^{-1} \tilde{p} \tag{3.13}$$

Equation 3.13 can be rearranged by using equation 2.38 and skew symmetry property of $\dot{M} - 2C$ as shown in equation 3.14.

$$\dot{V}(t) = s^T \left(\tau - Yp - \tau_f - \tau_s\right) + \dot{\hat{p}}^T \Gamma^{-1} \tilde{p} \tag{3.14}$$

Equation 3.14 can be rearranged by substituting control and adaptation laws in equation 3.11 and equation 3.10 as follows.

$$\dot{V}(t) = -s^T K_D s - s^T K_I S_{sq} \int \tilde{q} dt \tag{3.15}$$

In equation 3.15 errors on friction and spring models are neglected. Equation 3.15 shows that positive definite diagonal K_D and K_I matrices results in negative semi-definite $\dot{V}(t)$ which guarantees the convergence of s to zero. As stated in [15], convergence of s to zero also means that $\tilde{q}$ and $\dot{\tilde{q}}$ converges to zero.

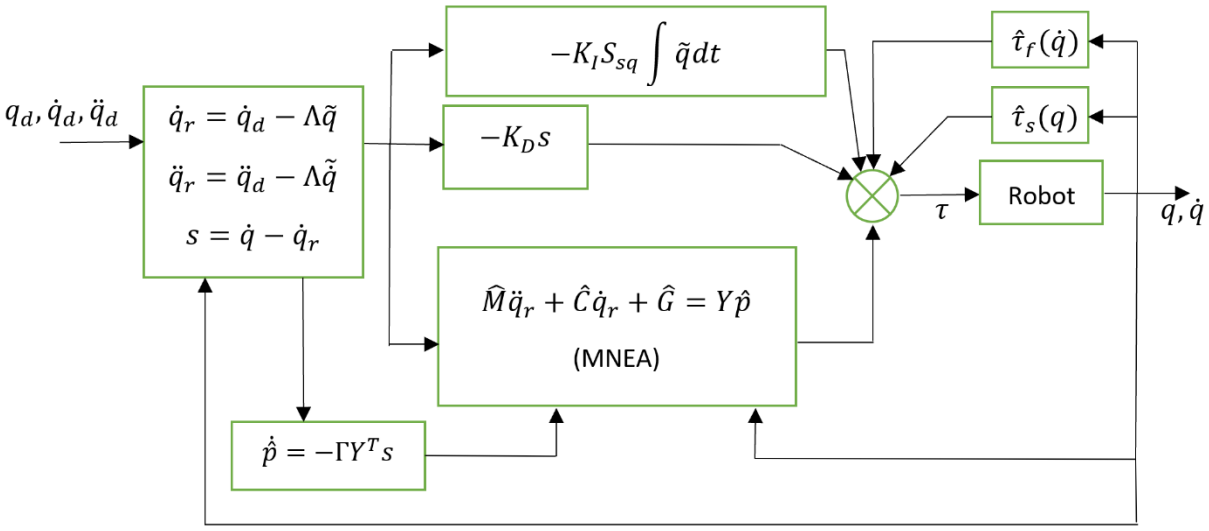

Figure 3.3 : Block Diagram of the Adaptive Control.

Since computation of dynamic model in control law includes two velocity terms as $\dot{q}$ and $\dot{q}_r$ it can't be computed with standard Newton-Euler algorithm [16]. Therefore, aforementioned modified Newton-Euler algorithm [12] is used to compute the dynamic model in the control law. As seen from Figure 3.2 and 3.3, differences between adaptive control algorithm explained here and passivity-based control algorithm are online parameter adaptation mechanism and integral term. Which indicates that adaptive control algorithm explained here essentially has the same structure as passivity-based control algorithm for zero adaptation and integral gains.

4. EXPERIMENTAL RESULTS

In this chapter, experimental results obtained by using Stäubli RX160 robot manipulator are presented. Trajectory tracking performances of the aforementioned adaptive, passivity – based and computed torque controllers are compared to each other for varying payloads. Validation trajectories in the experiments are generated so that they are representative of pick and place applications. Pick and place applications are chosen since varying payloads carried in the pick and place applications have the ability to represent uncertainties in the dynamic model of the robot. In [17], validation trajectories that are representative of pick and place applications are generated so that the end effector of the robot moves between 20 randomly chosen points in the workspace of the robot. Similarly, validation trajectories in this study are chosen so that robot moves between 16 randomly chosen points in the workspace of the robot with different distance and speed values between the points. Quintic trajectory is used between the points. Configurations of the points in the joint space, distance, and maximum speed values between the points during the trajectories are given in Table 4.1. Validation trajectories are shown in Figure 4.2. The path followed by the robot in task space is shown in Figure 4.3. End effector velocities are shown in Figure 4.4. Experiments run for the validation trajectories shown in Figure 4.2 for no payload and payloads of 3, 5, 7 and 9 kg. Payloads used in the experiments are shown in Figure 4.1.

Figure 4.1 : Payload of 3 kg.

Additional masses are added to the end of the payload.

Table 4.1: Waypoints used in the Validation Trajectories.

Dist. btw. Pts.(m)	Max. Speed	Config.	$q_1(°)$	$q_2(°)$	$q_3(°)$	$q_4(°)$	$q_5(°)$	$q_6(°)$
		1	90	0	90	0	90	90
	%10	2	110	0	56	55	34	48
0.5	%20	3	93	-37	95	24	72	0
	%30	4	87	-11	40	-15	44	-30
	%30	5	-119	-11	60	-30	15	15
0.75	%10	6	44	-43	119	0	35	49
	%20	7	11	-52	86	27	54	10
	%20	8	0	7	74	45	30	-20
1.0	%30	9	-90	-11	109	15	60	-35
	%10	10	-41	23	85	0	75	-45
	%10	11	65	-40	115	32	43	0
1.25	%20	12	136	45	65	-18	100	30
	%30	13	75	12	78	-34	74	-30
	%30	14	38	49	-110	41	23	56
1.5	%10	15	34	-57	-43	100	60	0
	%20	16	-51	-17	-79	62	53	-34

Maximum speed shows the percentage of maximum joint speed during the trajectories between two consecutive points with respect to the limits of the joint speeds. Distance between points shows the distance between two consecutive points in the robot workspace.

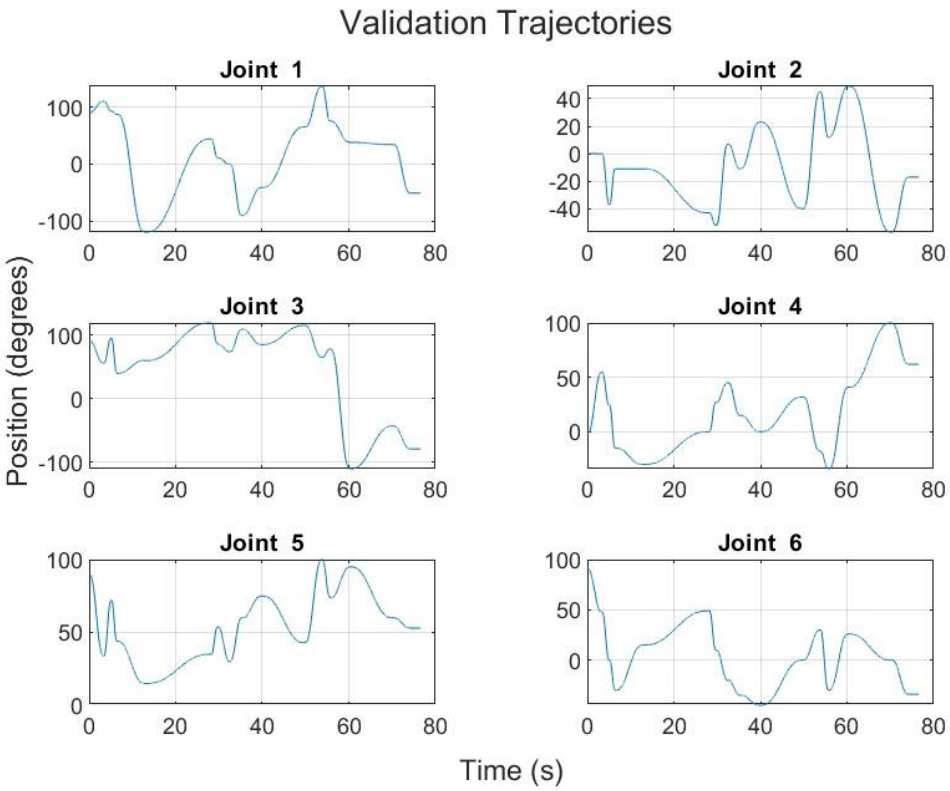

Figure 4.2 : Validation Trajectories.

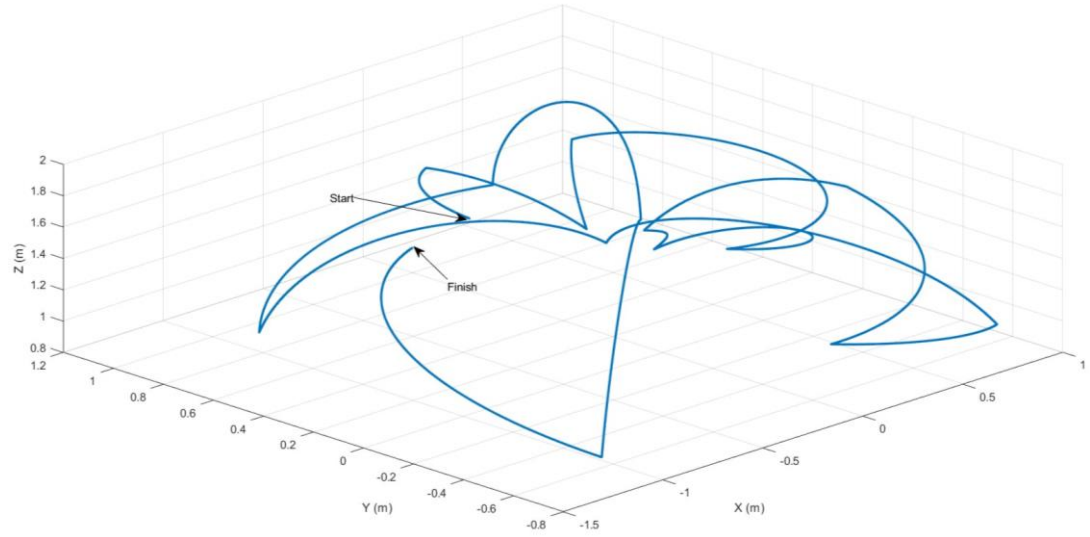

Figure 4.3 : Path Followed by the Robot end-effector.

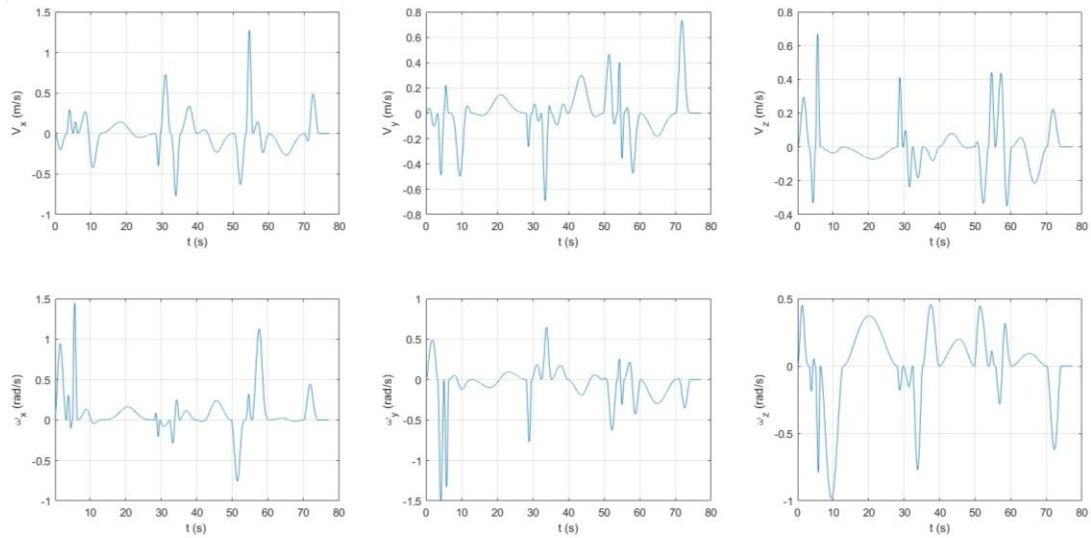

Figure 4.4 : End-effector Velocities.

4.1 Comparison of Controller Performances

In this section firstly, position errors and velocity errors during trajectory tracking for adaptive and passivity-based controllers are compared to each other. Then, same comparisons are presented for adaptive and computed torque controllers. Torque values are compared to each other for all controllers. Experiments are repeated for varying payload weights. Finally, rms values of position errors and percent change in rms position errors with respect to the rms position errors of the experiments with no payload are presented for all controllers.

Controller gain matrices used in the controllers are diagonal matrices. Diagonal elements of the controller gain matrices used in the experiments are given in Tables 4.2 and 4.3. Controller gains are tuned by trial and error. Position errors during trajectory tracking for no payload and payload of 3 kg, 5 kg, 7 kg and 9 kg for adaptive and passivity-based controllers are shown in Figures 4.5, 4.6, 4.7, 4.8 and 4.9 respectively. These figures show that maximum position errors for both adaptive and passivity-based controllers in joints 1 and 2 do not exceed 2 degrees, and in joints 3, 4 and 5 do not exceed 5 degrees. In joint 6, while the position error of the passivity-based controller does not exceed 10 degrees, maximum position error of the adaptive controller reaches to 20 degrees.

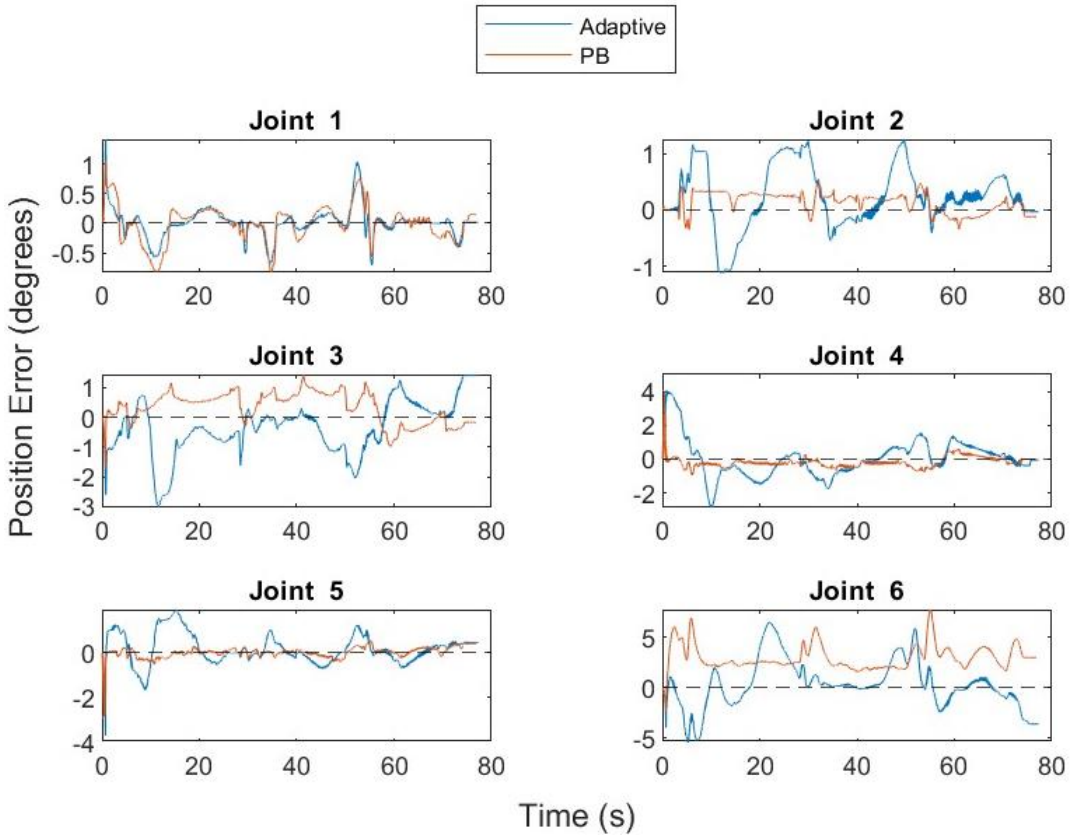

Figure 4.5 : Position Errors for No Payload(Adaptive-PB).

Table 4.2: Adaptive and Passivity-based Controller Gains.

Gains/ Contr.	$\boldsymbol{\lambda}$	$\boldsymbol{K_D}$	$\boldsymbol{K_I}$	$\boldsymbol{\Gamma}$
Adaptive	5-5-10-8-18-2	500-1200-150-100-30-20	500-5000-500-100-100-100	0.1
PB	5-5-10-8-18-2	500-1200-150-100-30-20	0-0-0-0-0-0	-

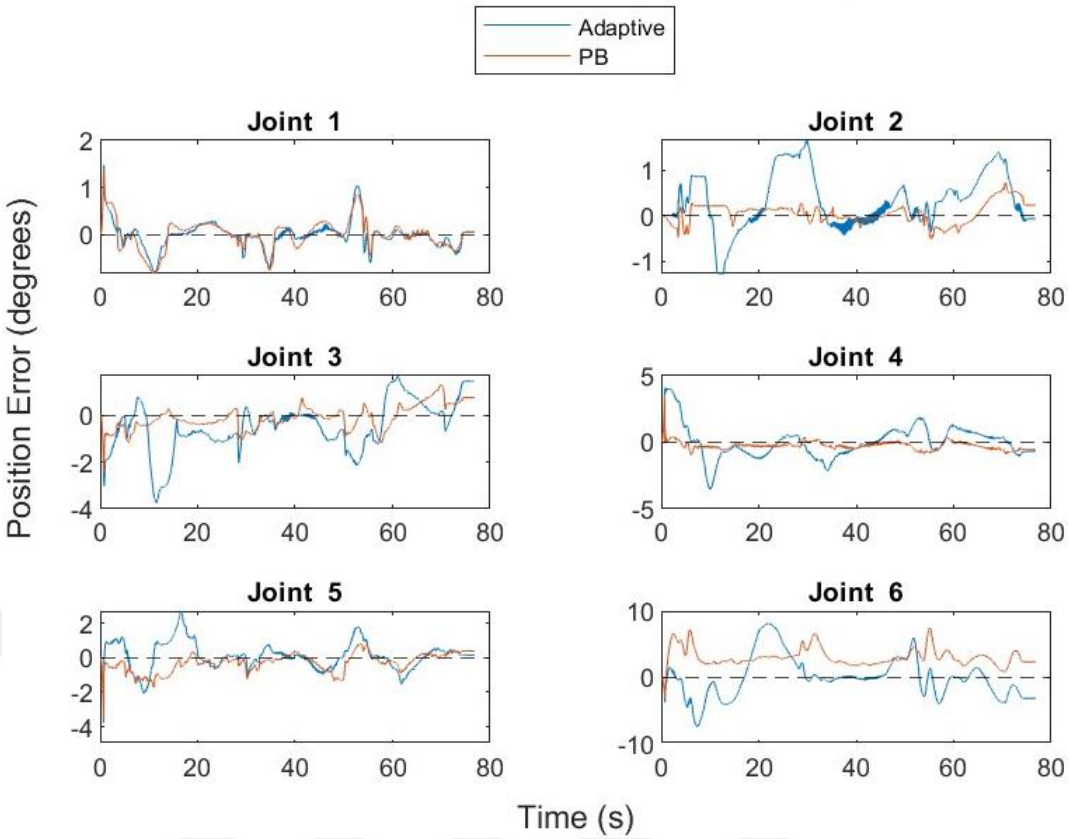

Figure 4.6 : Position Errors for Payload of 3 kg(Adaptive-PB).

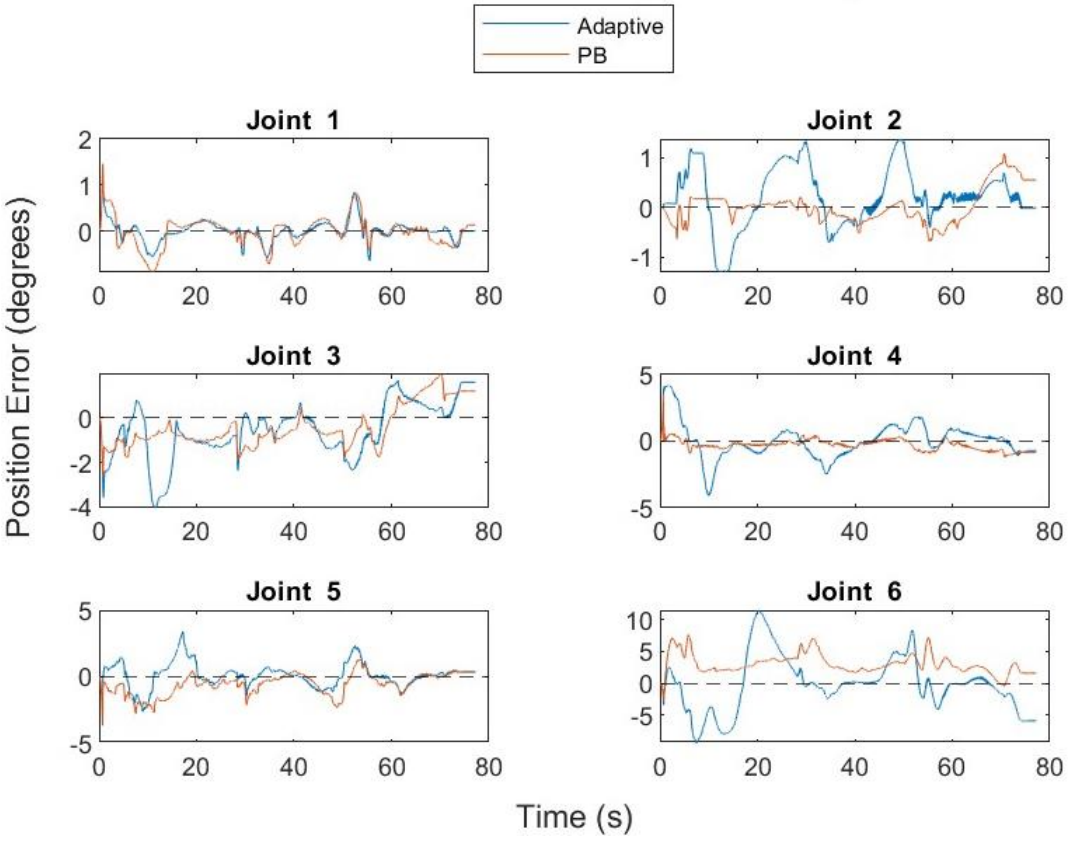

Figure 4.7 : Position Errors for Payload of 5 kg(Adaptive-PB).

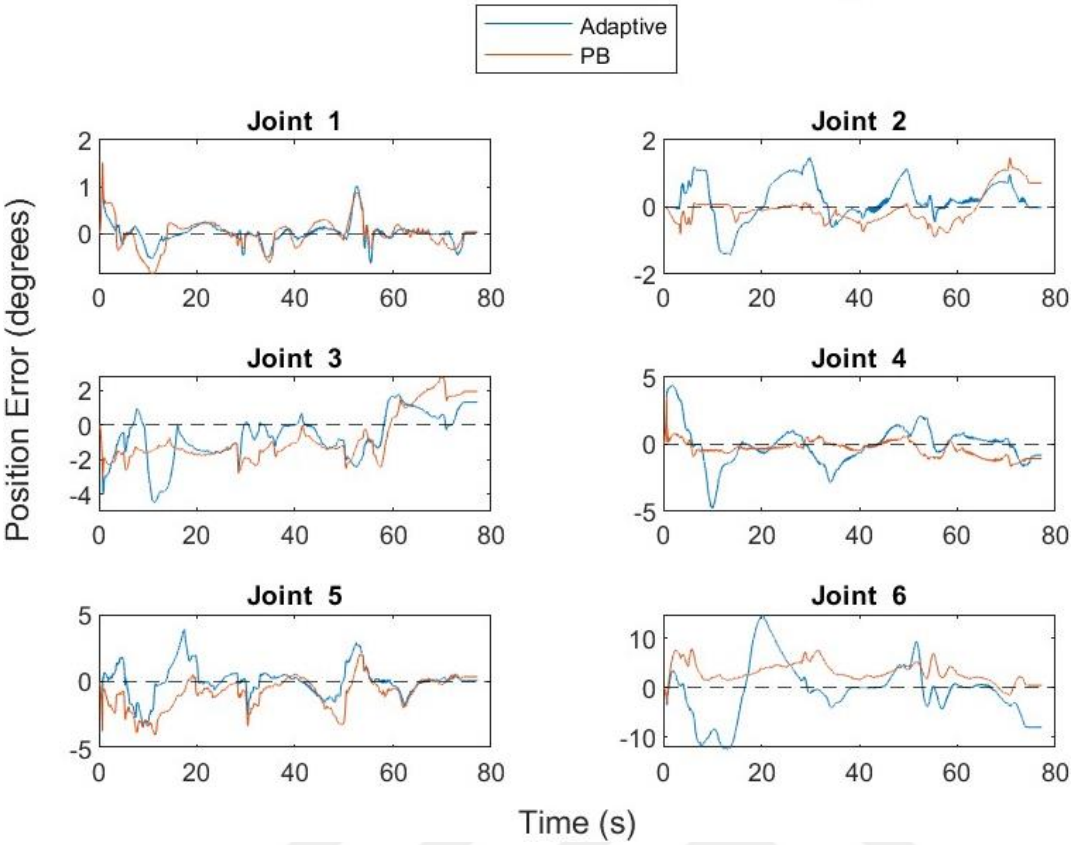

Figure 4.8 : Position Errors for Payload of 7 kg(Adaptive-PB).

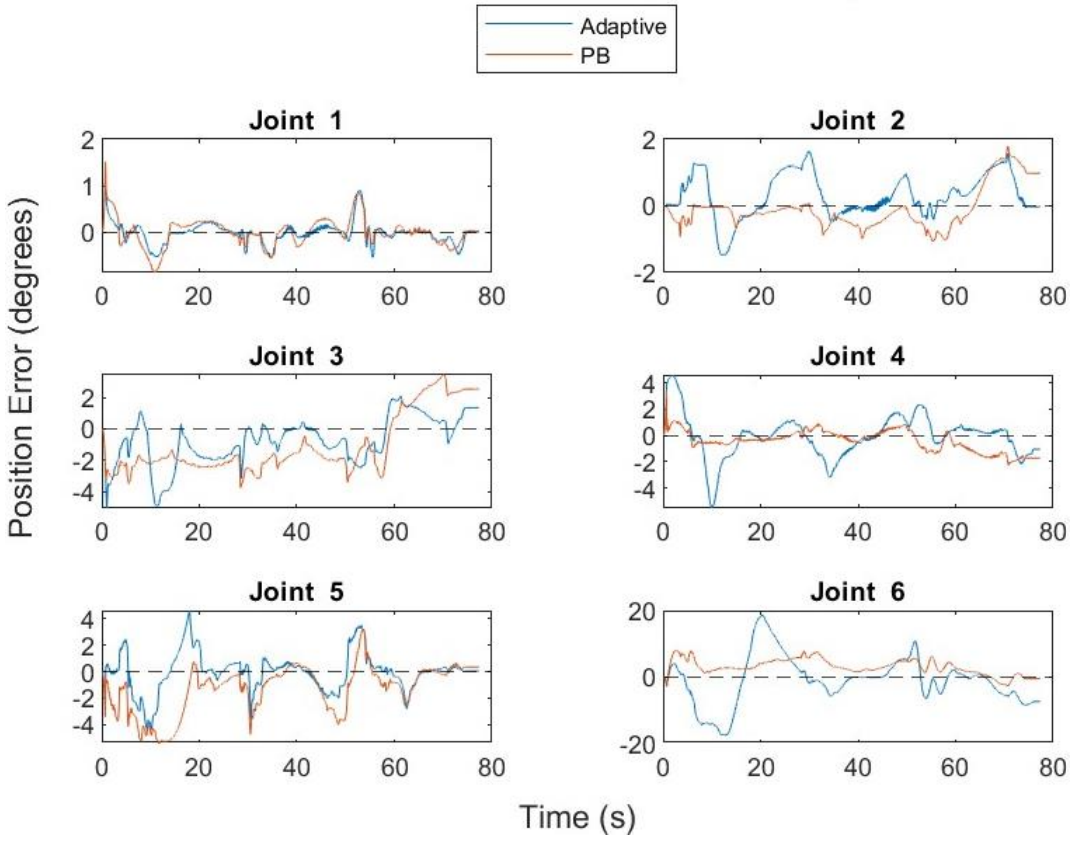

Figure 4.9 : Position Errors for Payload of 9 kg(Adaptive-PB).

Position errors during trajectory tracking for no payload and payload of 3 kg, 5 kg, 7 kg and 9 kg for adaptive and computed torque controllers are shown in Figure 4.10, 4.11, 4.12, 4.13 and 4.14 respectively. These figures show that while maximum position errors for computed torque controller in joints 1 and 2 do not exceed 5 degrees, it reaches to 50 degrees in joint 5. Similarly, maximum position error of computed torque controller takes values as high as 40 degrees in joint 3 and 20 degrees in joints 4 and 6. It is seen that while position error of computed torque controller increases significantly with increasing payload weight, increase in position error with varying payload weight is relatively low for adaptive and passivity-based controllers.

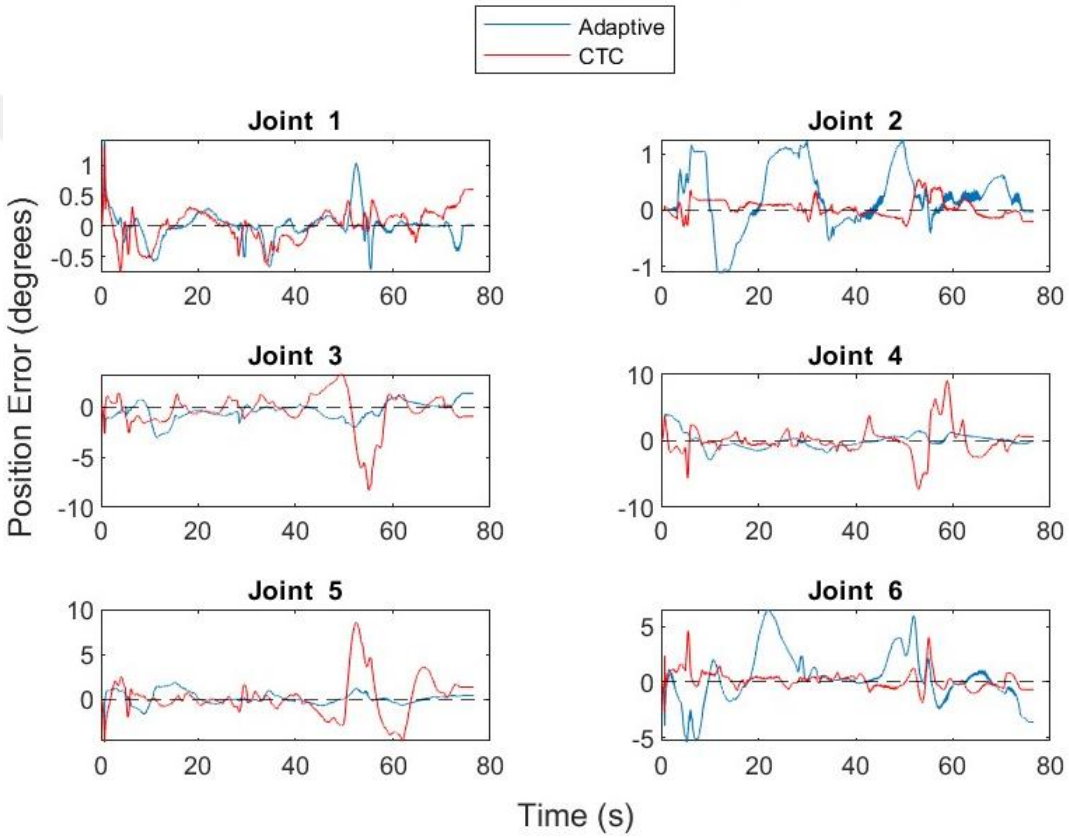

Figure 4.10 : Position Errors for No Payload(Adaptive-CTC).

Table 4.3: Computed Torque Controller Gains.

Gains/ Contr.	K_P	K_D	K_I
CTC	300-300-150-1500-11000-1500	25-50-40-20-30-20	0-0-0-1000-5000-100

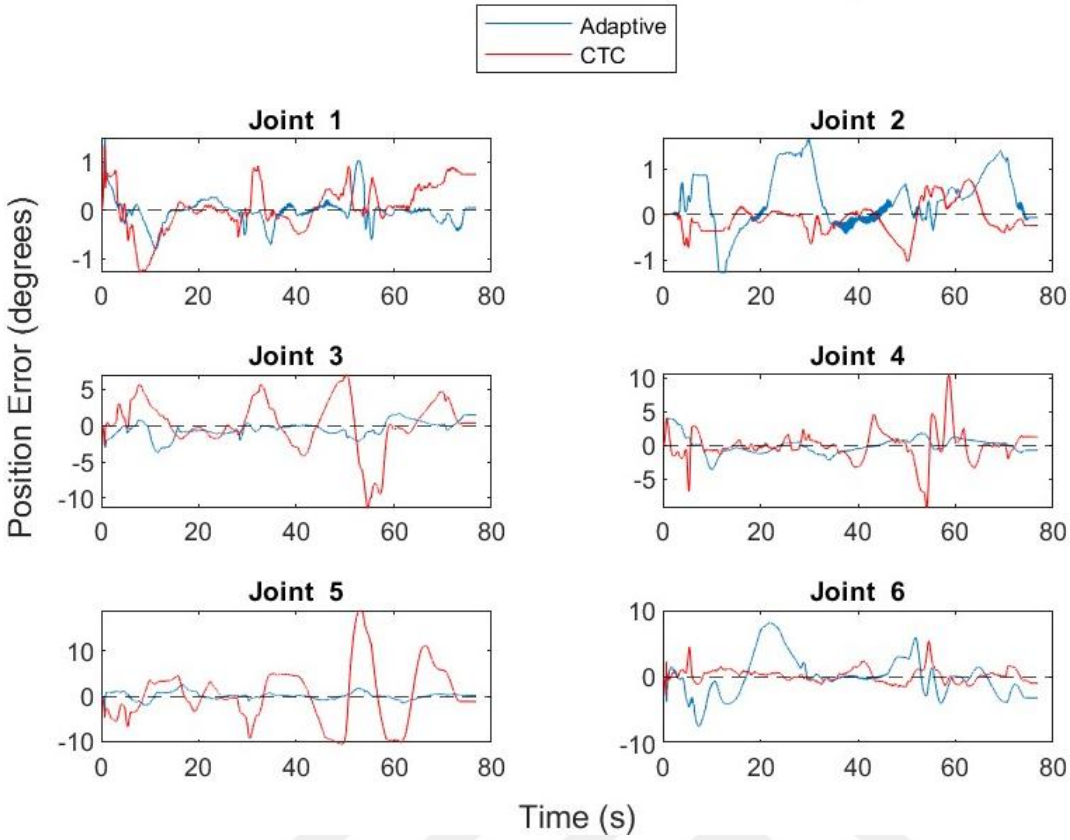

Figure 4.11 : Position Errors for Payload of 3 kg(Adaptive-CTC).

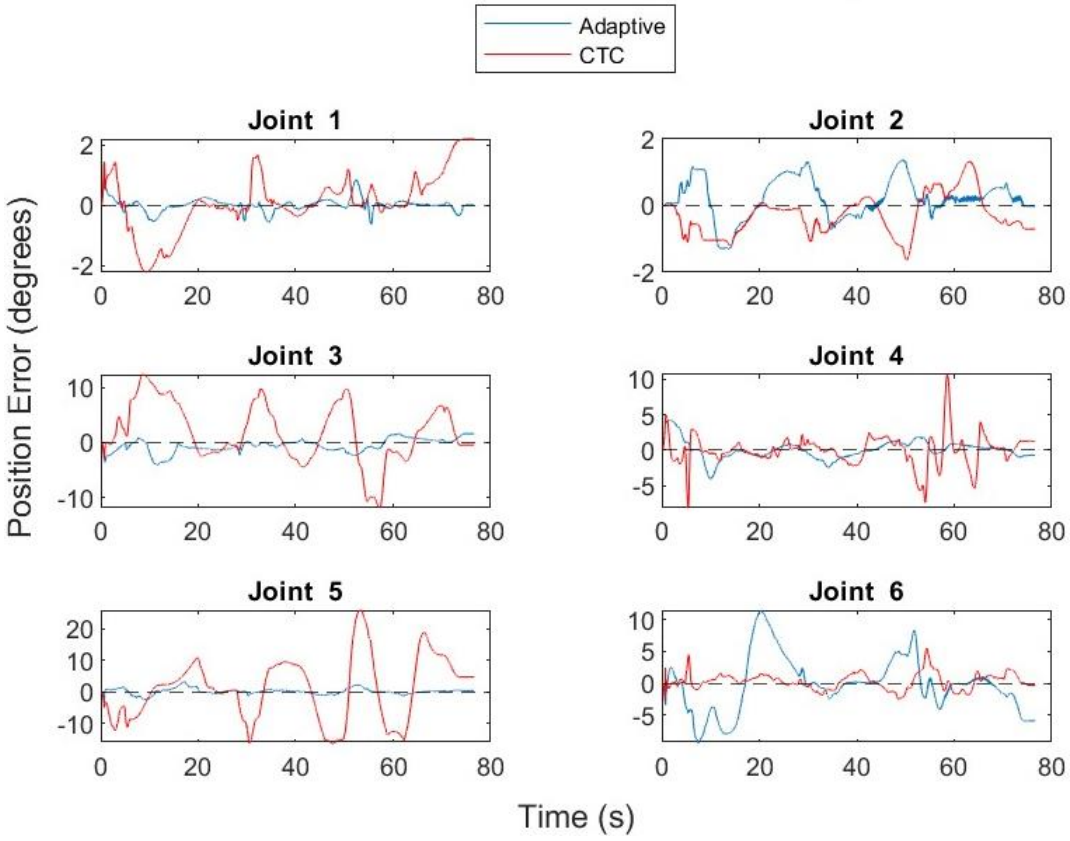

Figure 4.12 : Position Errors for Payload of 5 kg(Adaptive-CTC).

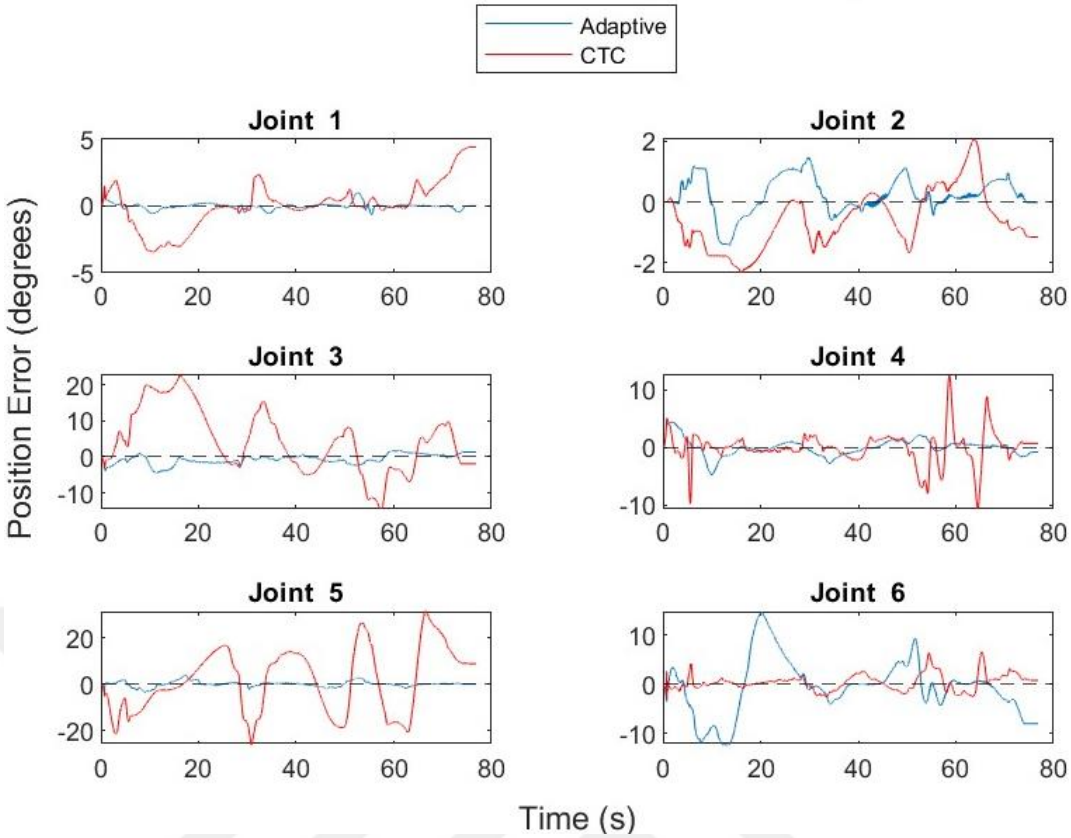

Figure 4.13 : Position Errors for Payload of 7 kg(Adaptive-CTC).

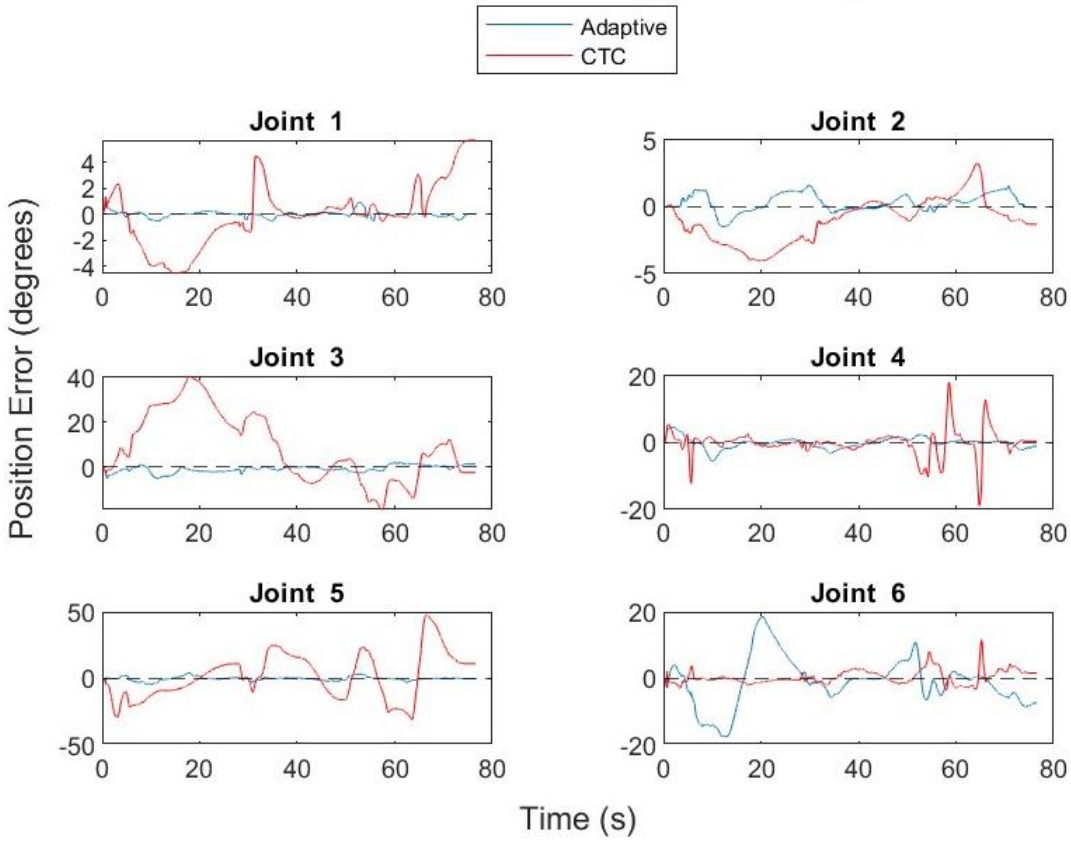

Figure 4.14 : Position Errors for Payload of 9 kg(Adaptive-CTC).

Velocity errors during trajectory tracking for no payload and payload of 3 kg, 5 kg, 7 kg and 9 kg for adaptive and passivity-based controllers are shown in Figure 4.15, 4.16, 4.17, 4.18 and 4.19 respectively. It is seen that maximum velocity errors doesn't change dramatically with varying payload weight for adaptive and passivity-based controllers.

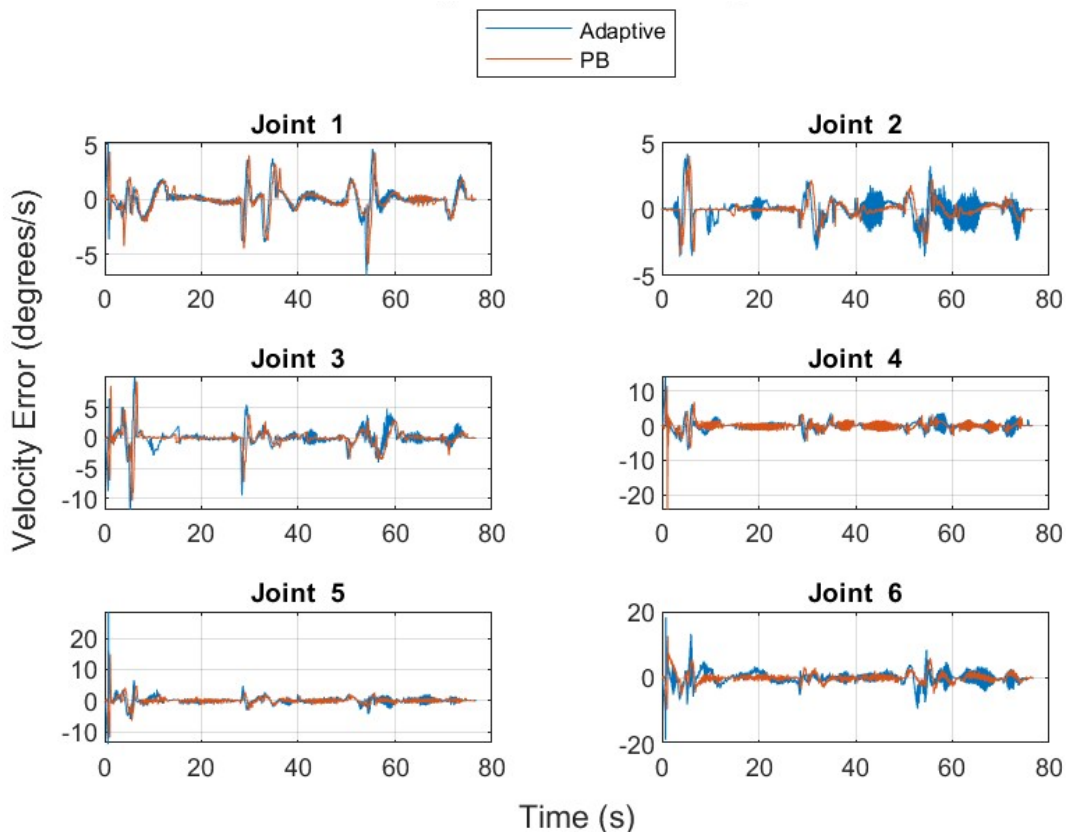

Figure 4.15 : Velocity Errors for No Payload(Adaptive-PB).

Velocity errors during trajectory tracking for no payload and payload of 3 kg, 5 kg, 7 kg and 9 kg for adaptive and computed torque controllers are shown in Figures 4.20, 4.21, 4.22, 4.23 and 4.24 respectively. These figures show that velocity error of computed torque controller is more sensitive to changes in payload weight than velocity error of adaptive and passivity-based controllers.

Torque values during trajectory tracking for no payload and payload of 3 kg, 5 kg, 7 kg and 9 kg for all three controllers are shown in Figures 4.25, 4.26, 4.27, 4.28 and 4.29 respectively. It is seen that all three controllers generates similar torque values in general. It is also seen that adaptive controller displays chattering behaviour. Integral term added to the adaptive controller may be the cause of this behaviour due to discontinious structure of the added integral term.

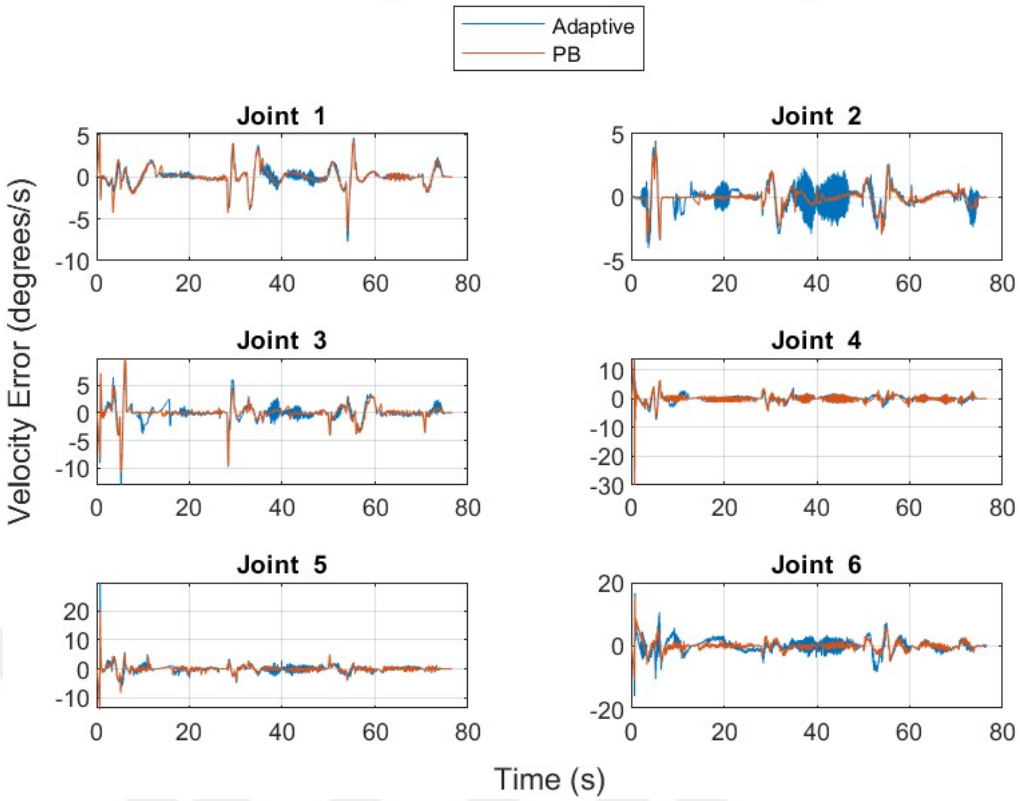

Figure 4.16 : Velocity Errors for Payload of 3 kg(Adaptive-PB).

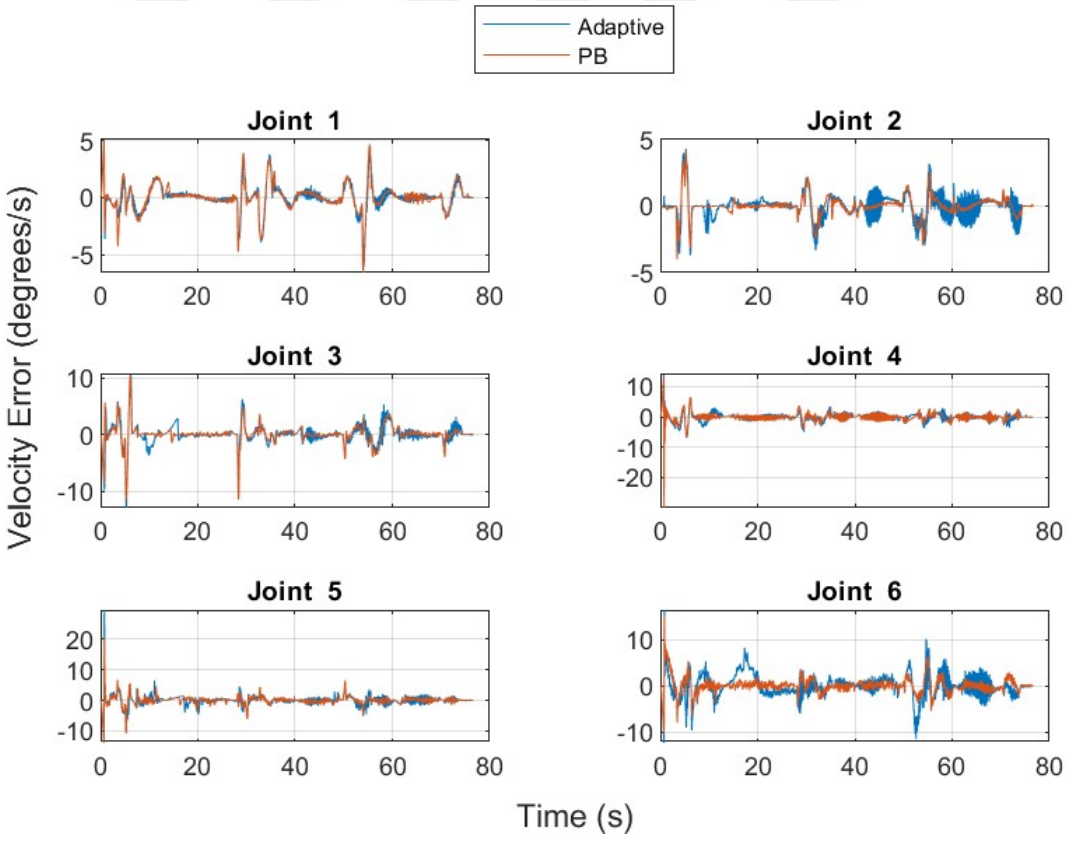

Figure 4.17 : Velocity Errors for Payload of 5 kg.

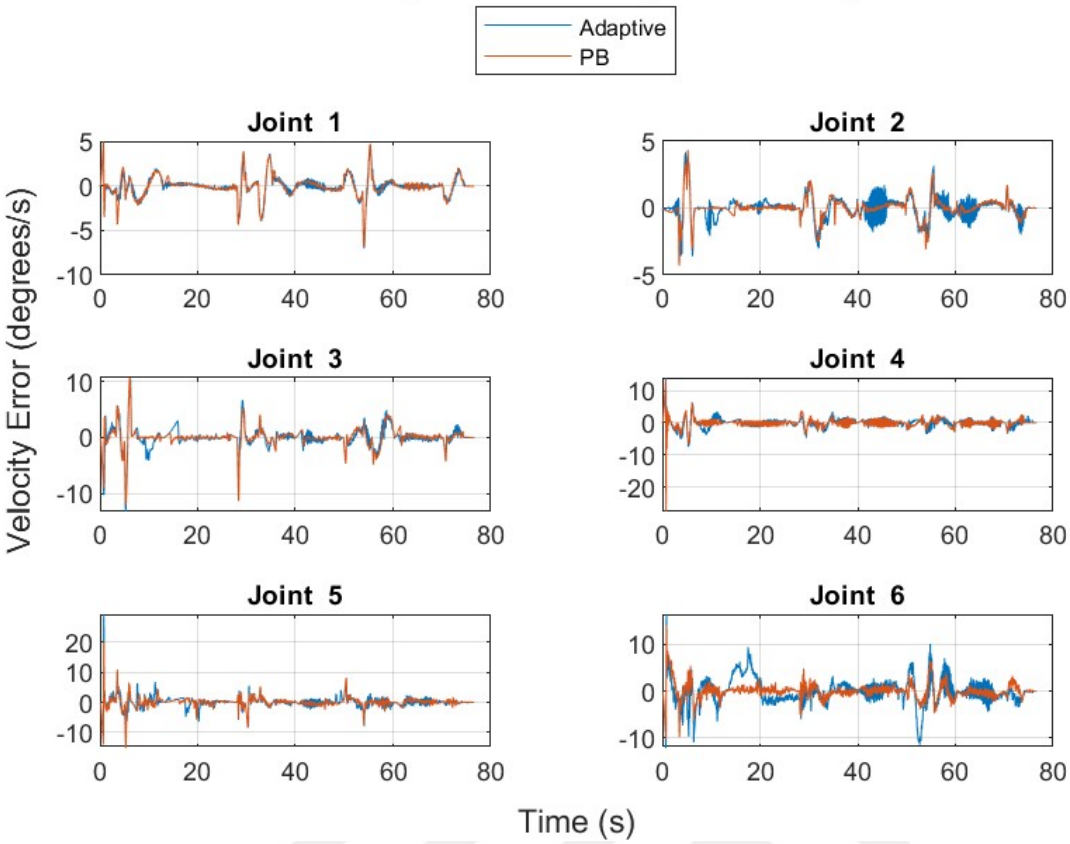

Figure 4.18 : Velocity Errors for Payload of 7 kg(Adaptive-PB).

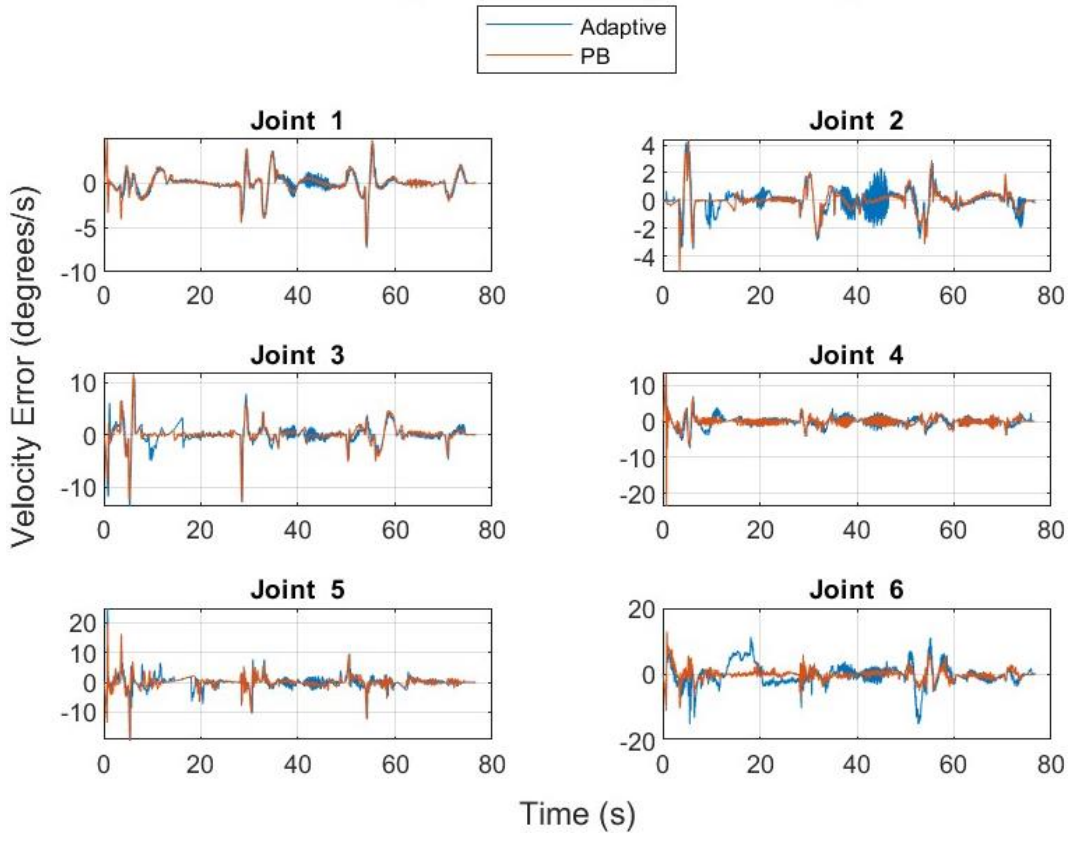

Figure 4.19 : Velocity Errors for Payload of 9 kg(Adaptive-PB).

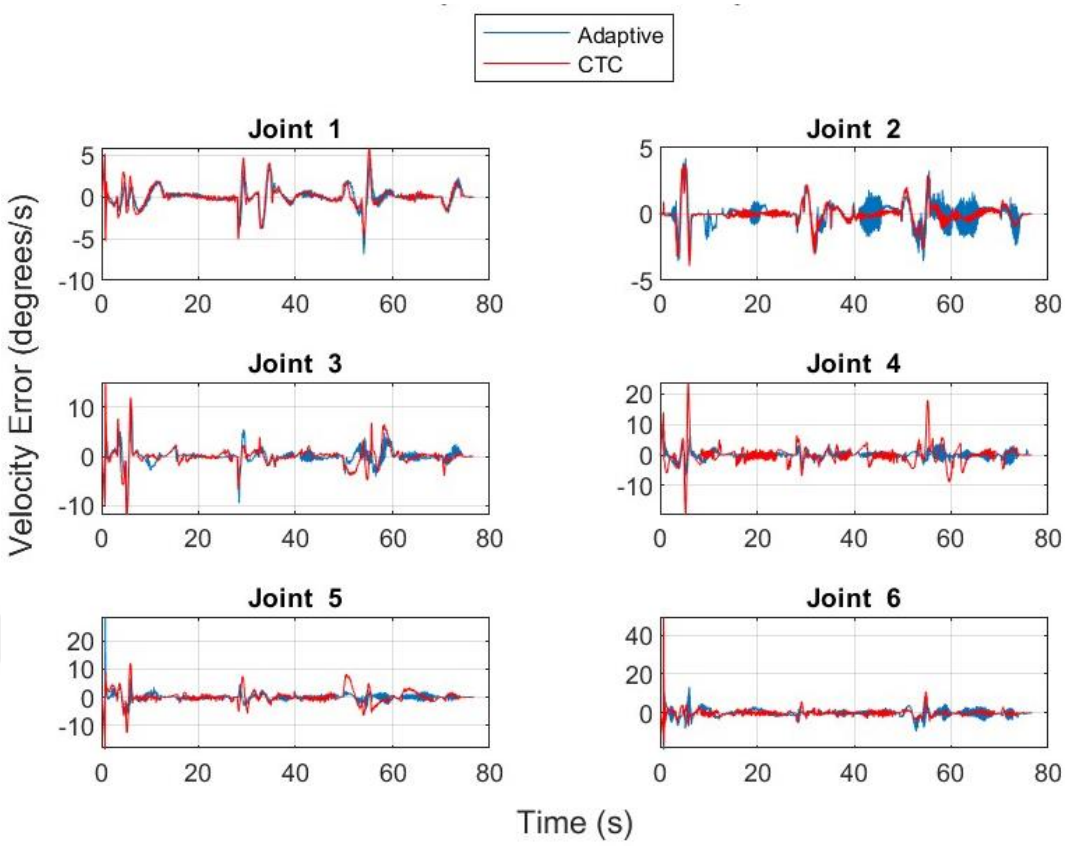

Figure 4.20 : Velocity Errors for No Payload(Adaptive-CTC).

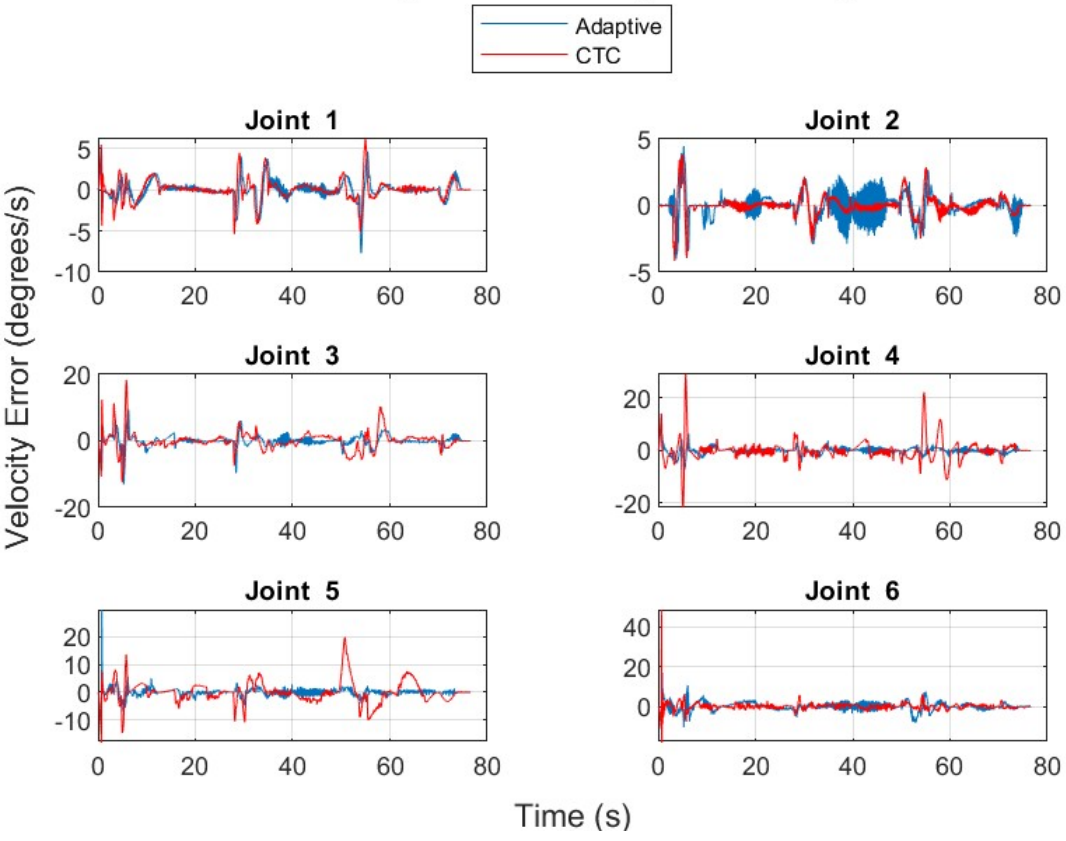

Figure 4.21 : Velocity Errors for Payload of 3 kg(Adaptive-CTC).

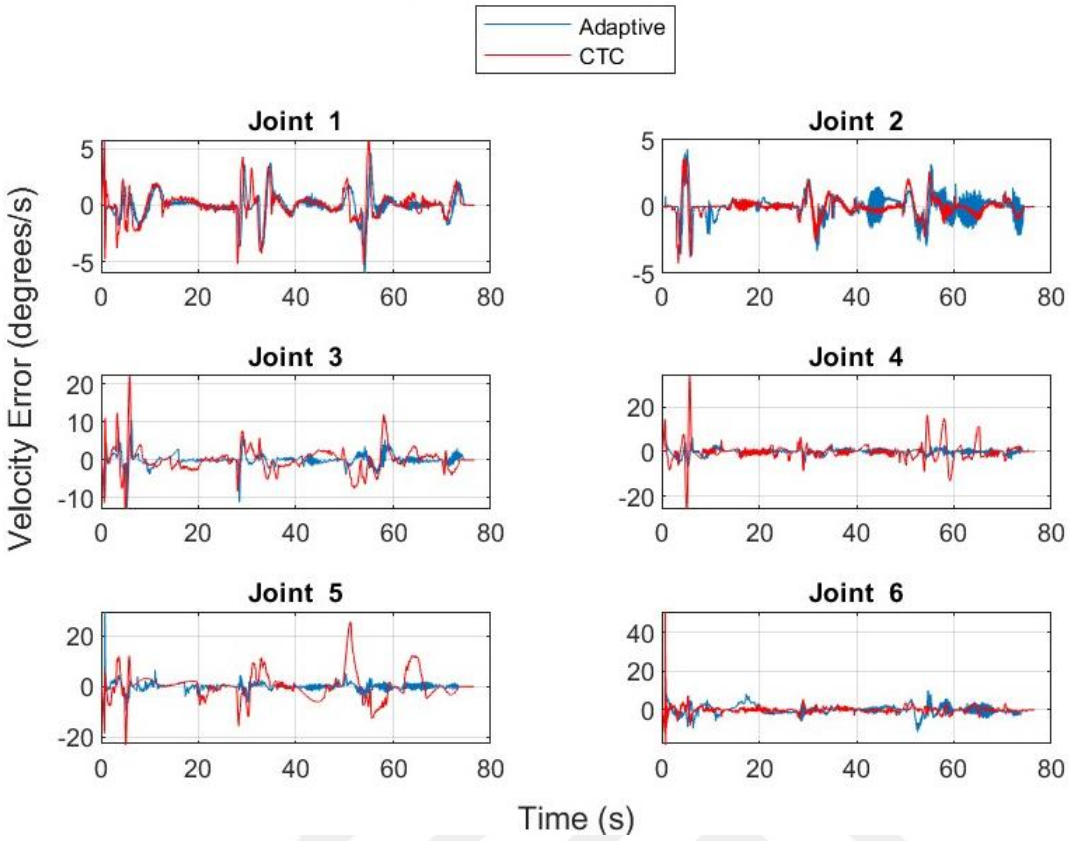

Figure 4.22 : Velocity Errors for Payload of 5 kg(Adaptive-CTC).

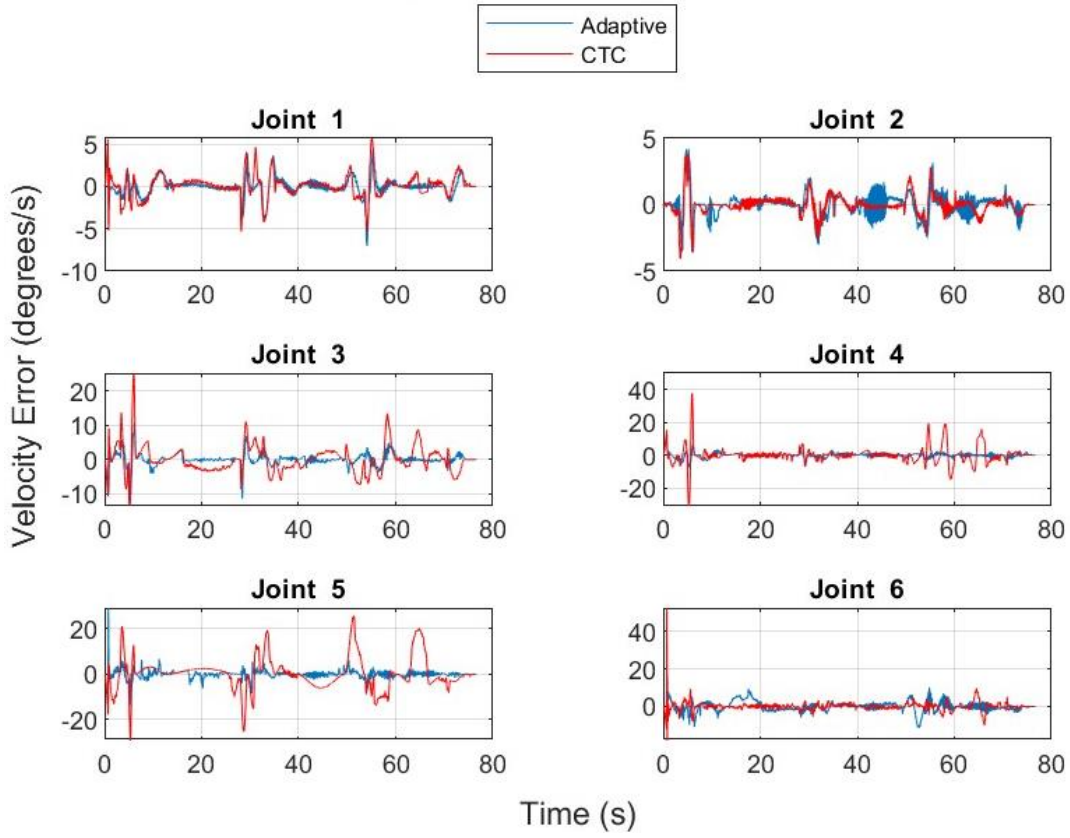

Figure 4.23 : Velocity Errors for Payload of 7 kg(Adaptive-CTC).

RMS values of position errors for varying payload weight are shown in Figure 4.30 and percent changes in rms values of position errors with respect to rms values of position errors for no payload are shown in Figure 4.31. Figure 4.30 shows that rms position errors of the adaptive and passivity-based controller are generally close to each other and lower than rms position errors of the computed torque controller except for joint 6. Figure 4.31 shows that rms position error of the computed torque controller is increased significantly with increasing payload weight. Also, rms position error of the adaptive controller is not significantly affected to varying payload weight except for joint 6, and rms position error of the passivity-based controller is significantly affected by varying payload weight in joints 4 and 5.

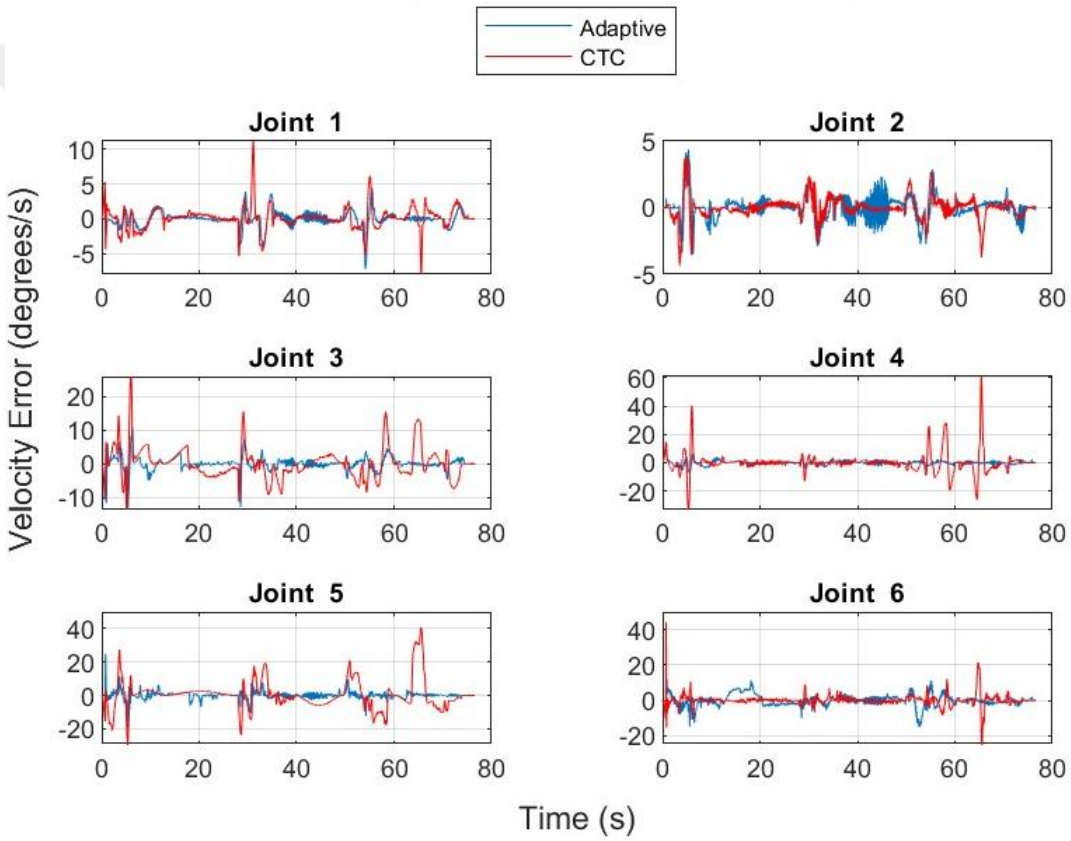

Figure 4.24 : Velocity Errors for Payload of 9 kg(Adaptive-CTC).

Dynamic parameter values provided by the manufacturer are entered as initial values of the dynamic parameter vector in the adaptive controller. Same dynamic parameter values are used for computed torque and passivity-based controllers except for the inertia parameter of link 6. The inertia parameter for link 6 is not provided by the manufacturer. Therefore, an identified value of 0.3 is used for the non-adaptive controllers. Details on the identification procedure can be found in [13]. The identified value was not suitable to be used in the adaptive controller. Therefore, the initial value

of the estimated inertia parameter of the link 6 is entered as zero in the adaptive controller. This difference may be the cause of the performance degradation of the adaptive controller in joint 6.

4.2 Parameter Convergence of the Adaptive Controller

The parameter adaptation mechanism used in the adaptive controller implemented in the experiments is based on the variable s introduced in equation 3.3 which is a combination of position and velocity errors. Therefore, parameters change in a way that makes position and velocity errors converge to zero. Experimental results show that dynamic parameters of the robot do not necessarily converge to true values of the parameters.

In the experiments, dynamic parameter values provided by the manufacturer have been entered as the initial values of parameters for adaptive control.

Figure 4.32, 4.33, 4.34, 4.35 and 4.36 shows dynamic parameters of the robot as a function of time during trajectory tracking for no payload and payload of 3 kg, 5 kg, 7 kg and 9 kg.

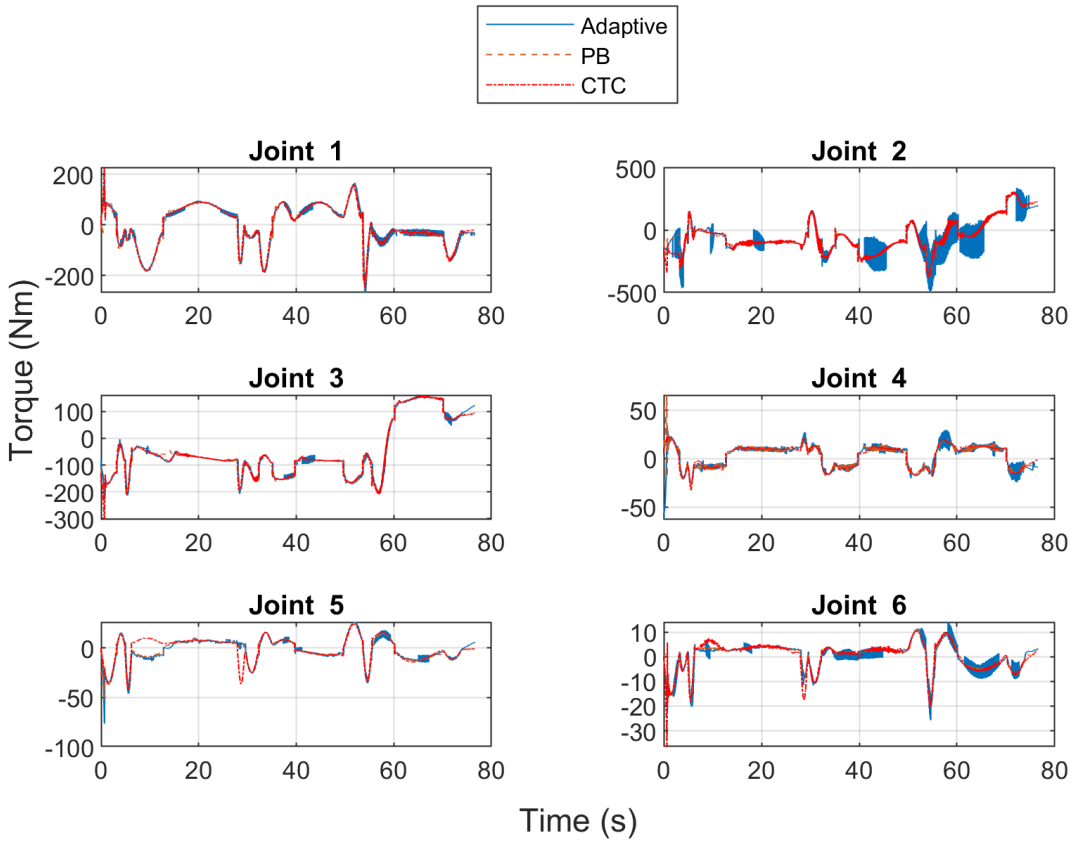

Figure 4.25 : Torque Values for No Payload.

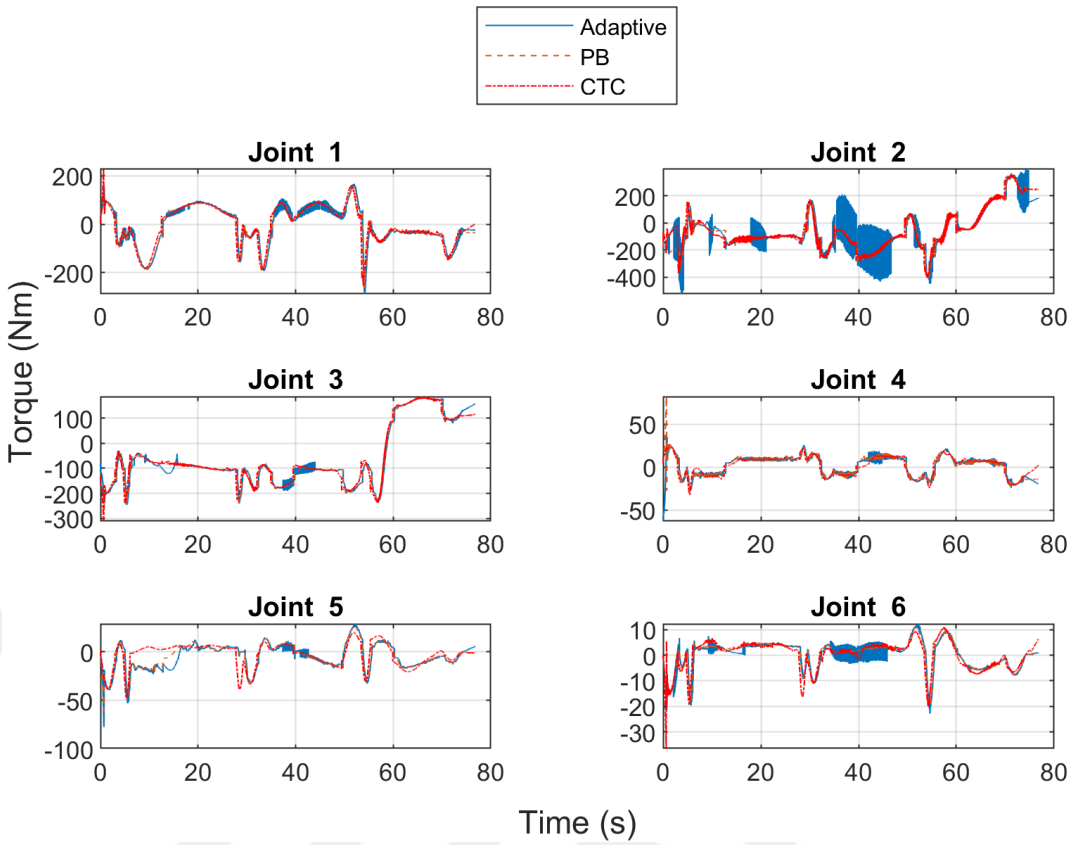

Figure 4.26 : Torque Values for Payload of 3 kg.

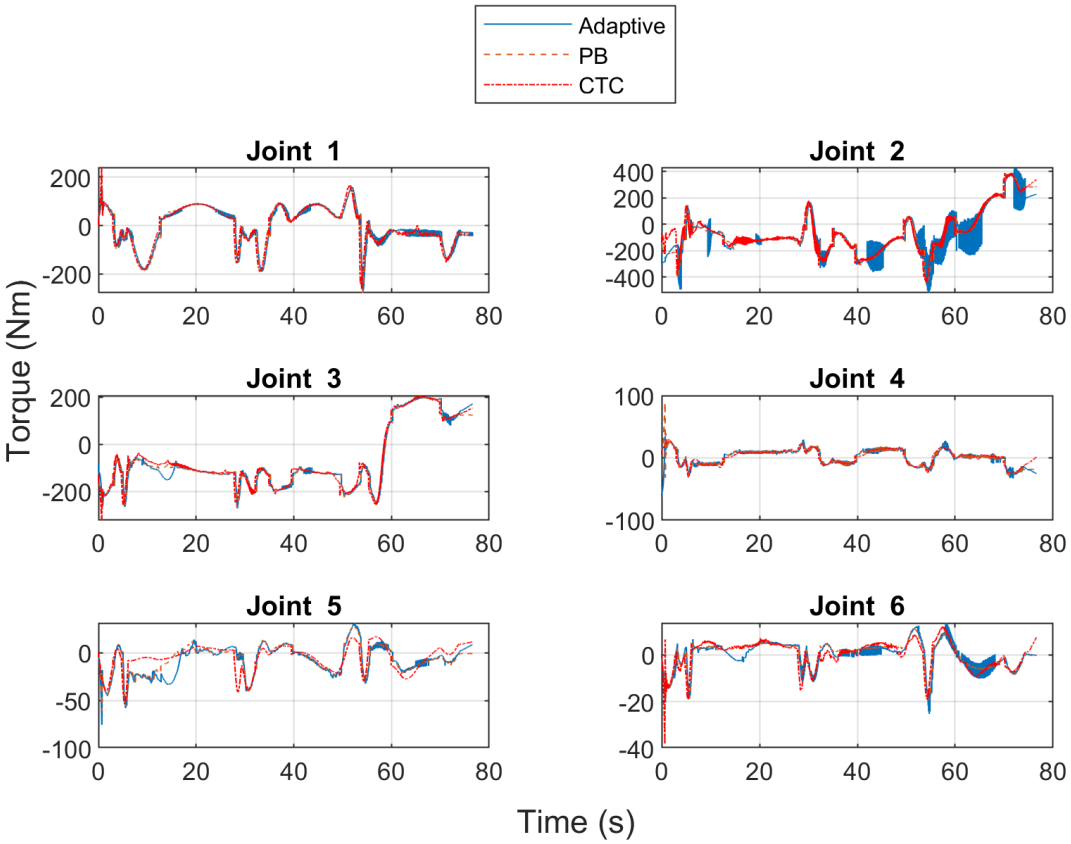

Figure 4.27 : Torque Values for Payload of 5 kg.

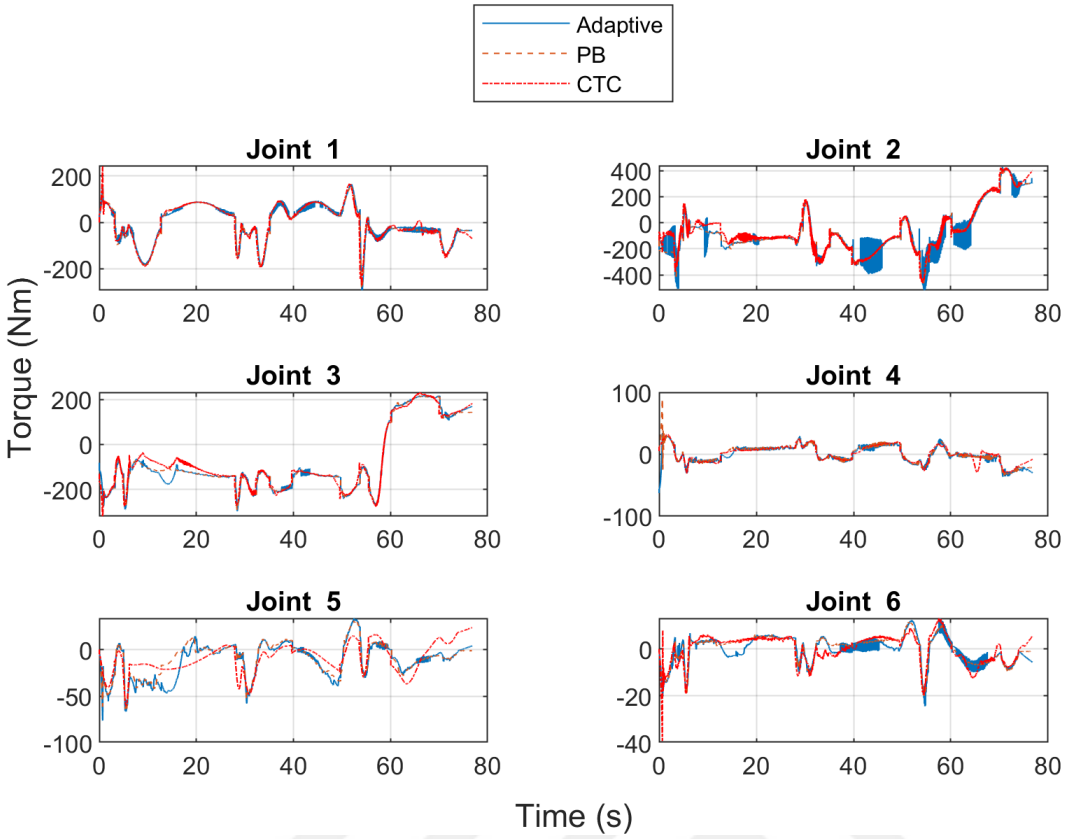

Figure 4.28 : Torque Values for Payload of 7 kg.

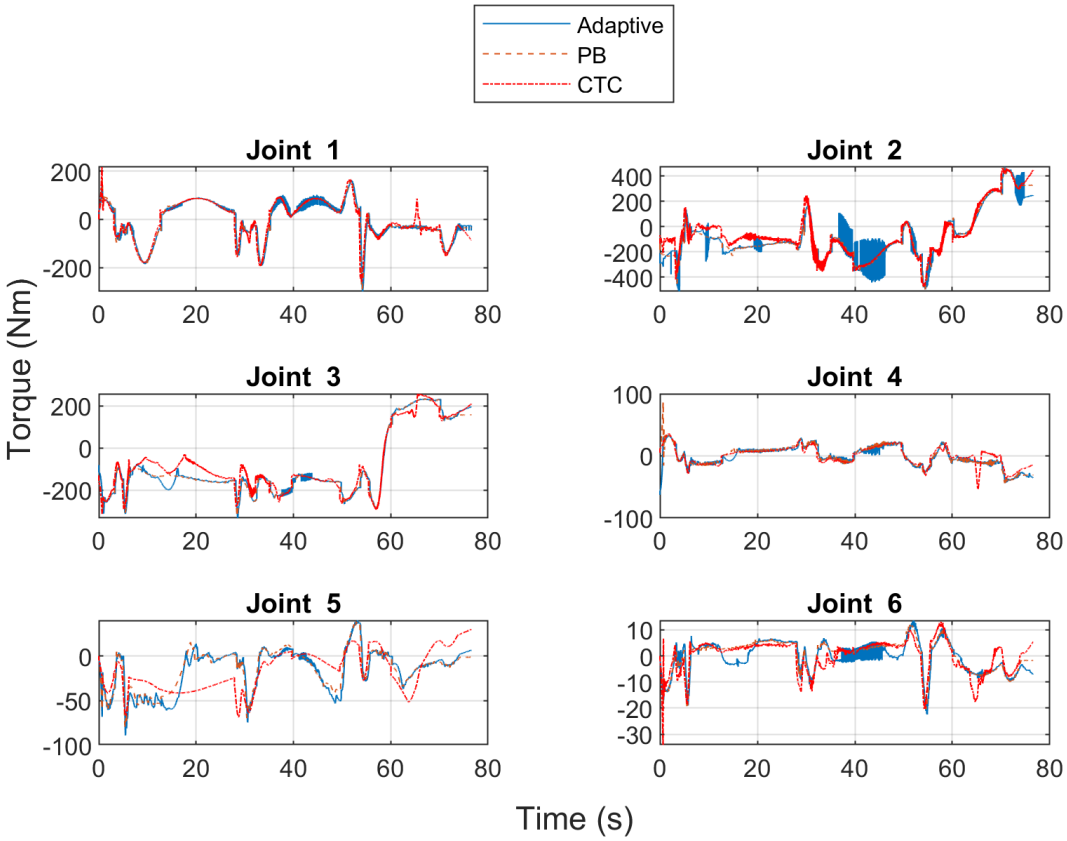

Figure 4.29 : Torque Values for Payload of 9 kg.

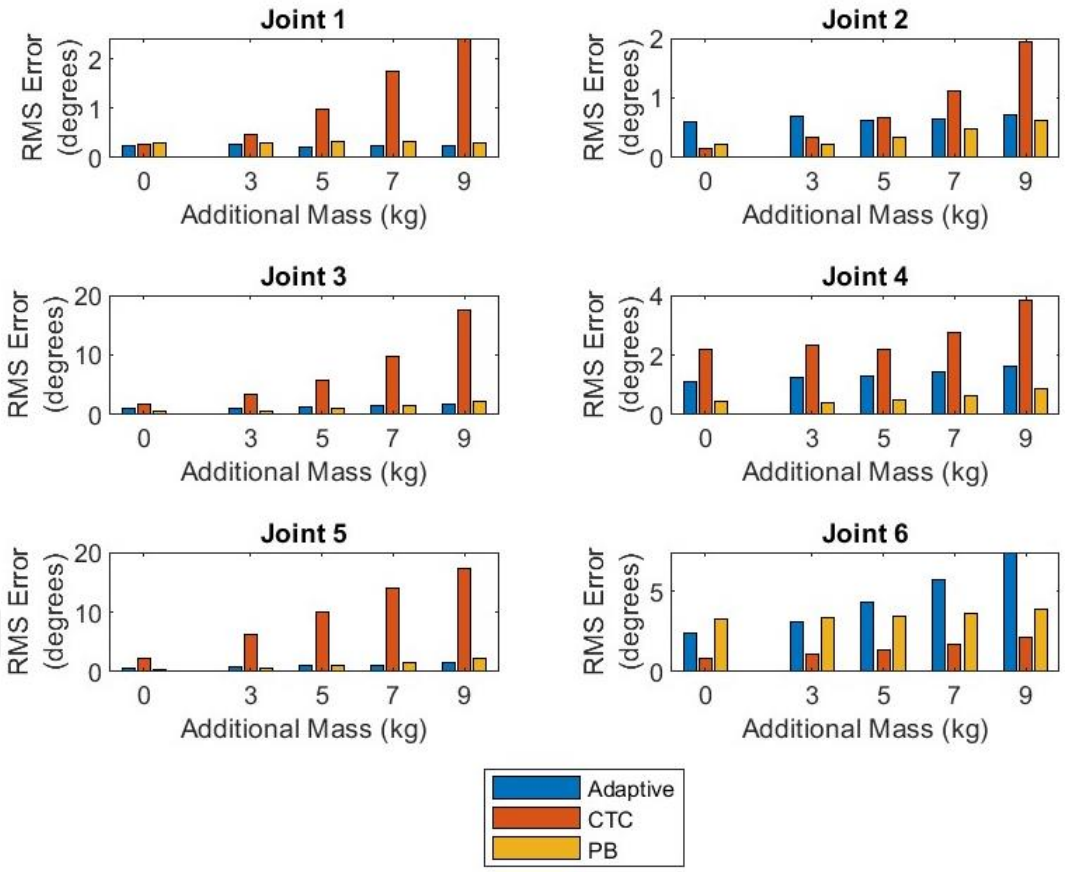

Figure 4.30 : RMS Values of Position Errors.

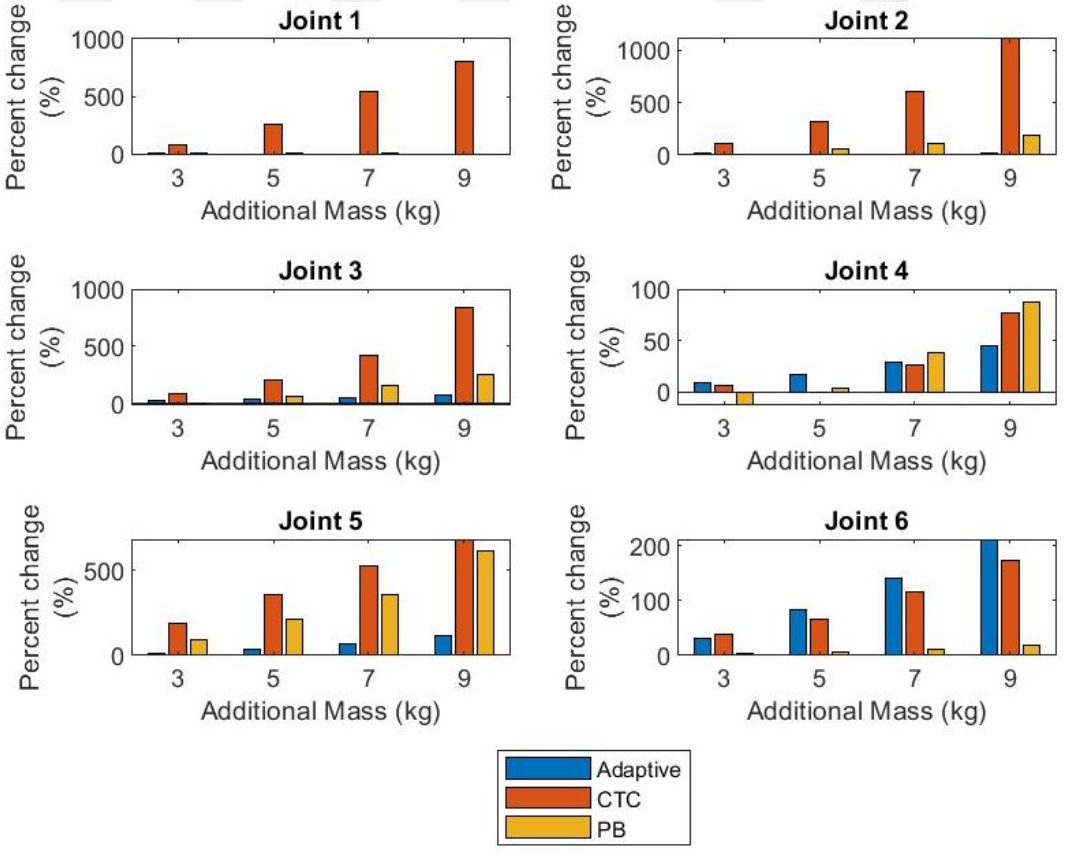

Figure 4.31 : Percent Change in RMS Values of Position Errors.

As seen from these figures dynamic parameters for link 1 and 2 remain almost constant during trajectory tracking for all payload weights. It is observed that increase in payload weight results in a significant increase in estimated mass parameters for link 5 and 6. Nevertheless, these parameters do not converge to true values of parameters.

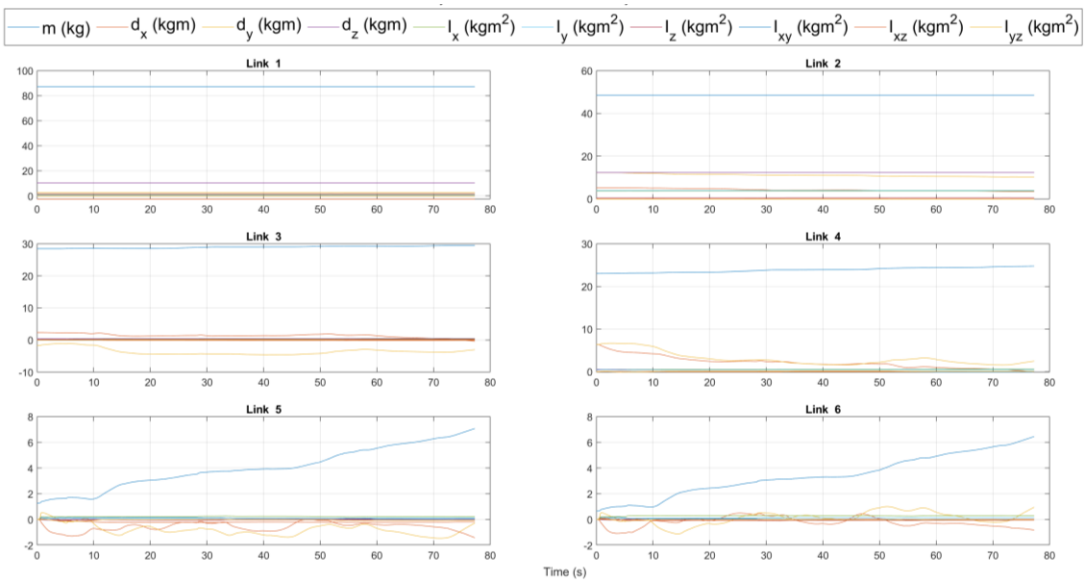

Figure 4.32 : Dynamic Parameter Values for No Payload.

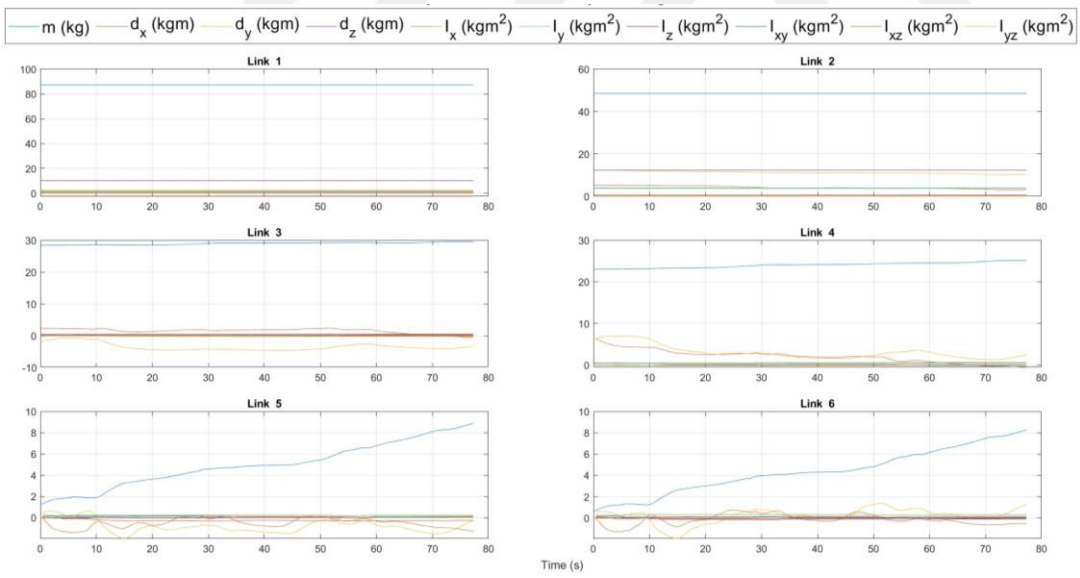

Figure 4.33 : Dynamic Parameter Values for Payload of 3 kg.

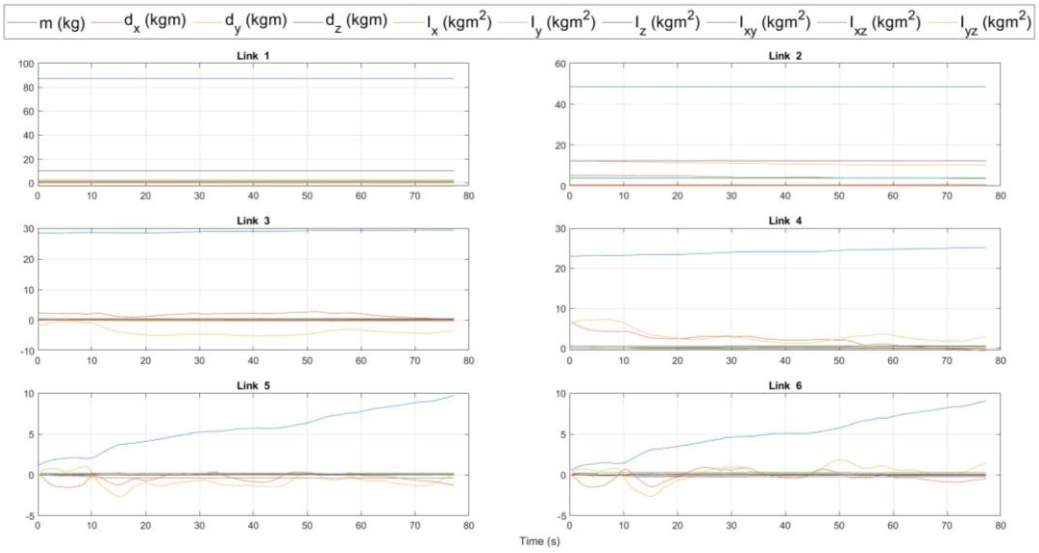

Figure 4.34 : Dynamic Parameter Values for Payload of 5 kg.

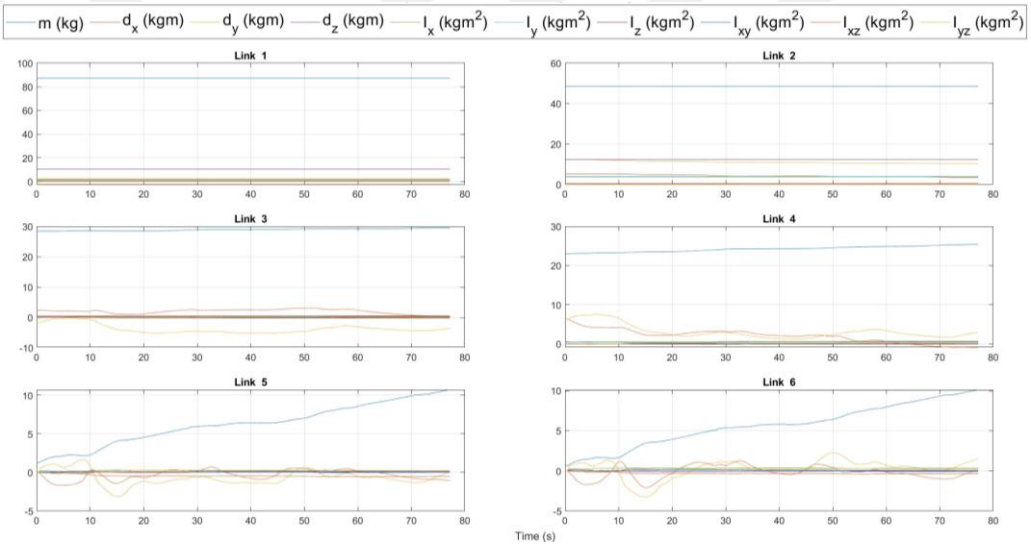

Figure 4.35 : Dynamic Parameter Values for Payload of 7 kg.

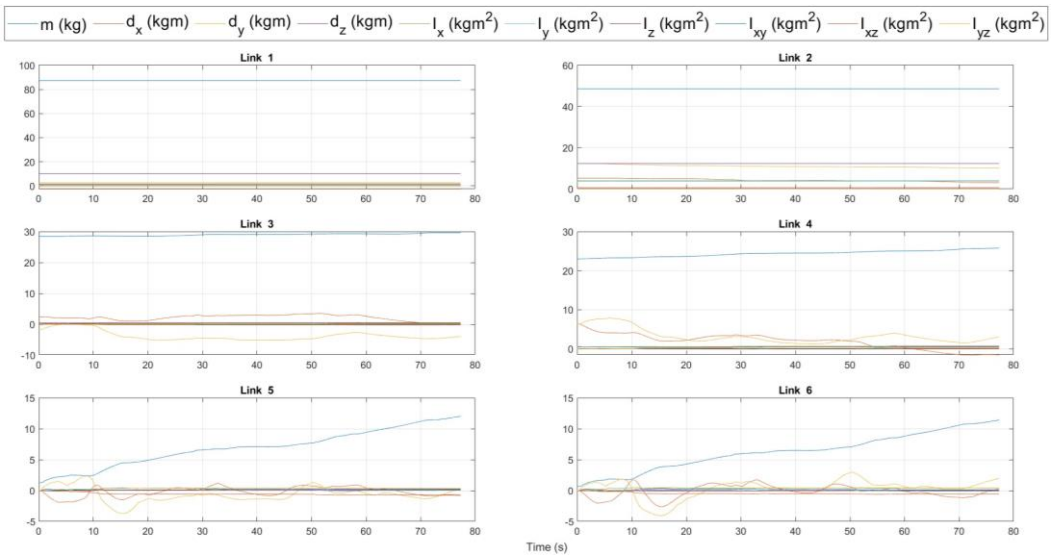

Figure 4.36 : Dynamic Parameter Values for Payload of 9 kg.

5. CONCLUSION

In this thesis, implementation of an adaptive control algorithm for the motion control of a six DOF robot manipulator is presented.

Firstly, kinematic and dynamic modeling, which is required for implementation of the robot controllers, are introduced. The Modified Newton – Euler algorithm [12] is explained for the computation of the regressor matrix and dynamic model in adaptive control.

Then theoretical background on computed torque control, passivity-based control, and adaptive control are given. It is seen that theoretically, while the computed torque and passivity-based control assume perfect knowledge of the dynamical model for the convergence of position error to zero, the adaptive control does not require that assumption.

In the adaptive control, the dynamic parameters of the robot are estimated online and may be different than true dynamic parameters of the robot. The adaptive control takes uncertainties into account through the online parameter adaptation mechanism. These uncertainties are not taken into account in computed torque and passivity-based control. Therefore, adaptive control takes a more comprehensive approach than computed torque and passivity-based control since the dynamic parameters of the robot are never exactly known.

In the experiments, trajectory tracking errors of the three controllers are compared to each other. The validation trajectories are generated in joint space. Generated trajectories are representative of pick and place applications. Pick and place applications are chosen since varying payloads are carried in the pick and place applications that cause uncertainties in the dynamic model of the robot. Point-to-point motions between 16 random points in the robot workspace with quintic trajectories are applied successively to obtain the validation trajectories. The validation trajectories consist of a variety of point-to-point trajectories with different distances between points and different maximum velocities along the trajectories. Implemented controllers follow the validation trajectories for varying payload weights to compare

the performances of the implemented controllers in the presence of uncertainties in the dynamic model of the robot.

Experimental results show that position errors for the computed torque controller are significantly affected by the varying payload weight. Increase in payload weight results in a significant increase in position errors for all joints.

On the other hand, the passivity-based controller displays more robust performance than the computed torque controller against payload weight variations. Even though the passivity-based controller shows percent changes in rms position errors for joints 4 and 5 similar to the computed torque controller, it has much lower percent changes in rms position errors than the computed torque controller for other joints.

The adaptive controller have larger values for the rms position errors than the computed torque and passivity-based controller for joints 2 and 6. However, the percent changes in rms position errors with varying payload weight are mostly lower for the adaptive controller than computed torque and passivity-based controller with joint 6 as an exception.

Average values for the percent change in rms position errors are 43% for the adaptive controller, 95% for the passivity-based controller, and 318% for the computed torque controller.

As a result adaptive controller seems to be the least sensitive controller to changes in payload weight while computed torque controller is the most sensitive controller.

It is also seen that the estimated dynamic parameters of the robot do not converge to the true values of the dynamic parameters, which is expected since the parameter adaptation mechanism is directed by position and velocity errors. Estimated mass values for link 5 and 6 increase as payload weight increases. Increase in the estimated mass value for link 6 is expected since payload is directly connected to link 6. Increase in estimated mass value for link 5 may be explained by the coupling effect between joints 5 and 6.

Further studies may include adaptation of joint friction and gravity counter balance spring model parameters along with the dynamic parameters of the robot. Also a modification in the dynamic model to take coupling effects in joints 5 and 6 can be useful for reducing the errors in joint 6.

REFERENCES

[1] *Industrial Robots*. (n.d.). Retrieved December 30, 2022, from https://ifr.org/industrial-robots

[2] *World Robotics 2022 Industrial Robots*. (2022).

[3] **Zhang, D., & Wei, B.** (2017). A review on model reference adaptive control of robotic manipulators. In *Annual Reviews in Control* (Vol. 43, pp. 188–198). Elsevier Ltd. https://doi.org/10.1016/j.arcontrol.2017.02.002

[4] **Hsia, T. C.** (1986). *Adaptive Control of Robot Manipulators - A Review*. 183–189. https://doi.org/10.1109/robot.1986.1087696

[5] **Dubowsky, S., & DesForges, D. T.** (1979). The Application of Model-Referenced Adaptive Control to Robotic Manipulators. In *Journal of Dynamic Systems* (Vol. 101, Issue 193). http://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/101/3/193/5517000/193_1.pdf

[6] **Tomizuka, M., & Horowitz, R.** (1983). Model Reference Adaptive Control of Mechanical Manipulators. *IFAC Adaptive Systems in Control and Signal Processing.*, 27–32.

[7] **Craig, J. J., Hsu, P., & Sastry, S. S.** (1986). *Adaptive Control of Mechanical Manipulators*. 190–195. https://doi.org/10.1109/robot.1986.1087661

[8] **Slotine, J.-J. E., & Weiping Li.** (1987). On the Adaptive Control of Robot Manipulators. *The International Journal of Robotics Research*, *6*(3), 49–59. https://doi.org/10.1177/027836498700600303

[9] **Faverges, S.** (2008). *Arm- RX series 160 Family Instruction Manual (D28068504A - 01/2008)*.

[10] **Pertin, F., & Bonnet-des-tuves, J.-M.** (2004). Real Time Robot Controller Abstraction Layer. *Proceedings of the Third International Symposium on Robotics*.

[11] **Spong, M. W., Hutchinson, S., & Vidyasagar, M.** (2020). *Robot Modeling and Control* (2nd ed.). John Wiley & Sons, Inc.

[12] **Ferrajoli, L., & De Luca, A.** (2009). A modified newton-euler method for dynamic computations in robot fault detection and control. *Proceedings - IEEE International Conference on Robotics and Automation*, 3359–3364. https://doi.org/10.1109/ROBOT.2009.5152618

[13] **Akbulut, M. A.** (2020). *Modelling, Identification and Passivity-Based Control of 6 DOF Industrial Robot*.

[14] **Takegaki, M., & Arimoto, S.** (1981). A New Feedback Method for Dynamic Control of Manipulators. *Journal of Dynamic Systems, Measurement, and Control*, *103*(2), 119–125. https://doi.org/10.1115/1.3139651

[15] **Slotine, J. J. E., & Li, W.** (1988). Adaptive Manipulator Control: A Case Study. *IEEE Transactions on Automatic Control*, *33*(11), 995–1003. https://doi.org/10.1109/9.14411

[16] **Luh, J. Y. S., Walker, M. W., & Paul, R. P. C.** (1980). *On-Line Computational Scheme for Mechanical Manipulators*. http://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/102/2/69/5696694/69_1.pdf

[17] **Swevers, J., Verdonck, W., & De Schutter, J.** (2007). Dynamic Model Identification for Industrial Robots. *IEEE Control Systems*, *27*(5), 58–71. https://doi.org/10.1109/MCS.2007.904659

APPENDICES

APPENDIX A: MATLAB Code for Modified Newton-Euler Algorithm (MNEA)

APPENDIX A

```
function torque = RX160_MNEA(qp, qv, qv_r, qa_r, g, p)
    R01=[cos(qp(1)) 0 -sin(qp(1));sin(qp(1)) 0 cos(qp(1));0 -1 0];
    R12=[sin(qp(2))  cos(qp(2)) 0; -cos(qp(2)) sin(qp(2)) 0;0 0 1];
    R23=[-sin(qp(3)) 0 cos(qp(3));cos(qp(3)) 0 sin(qp(3));0 1 0];
    R34=[cos(qp(4)) 0 -sin(qp(4));sin(qp(4)) 0 cos(qp(4));0 -1 0];
    R45=[cos(qp(5)) 0 sin(qp(5));sin(qp(5)) 0 -cos(qp(5));0 1 0];
    R56=[cos(qp(6)) -sin(qp(6)) 0;sin(qp(6)) cos(qp(6)) 0;0 0 1];

    R02=R01*R12;
    R03=R02*R23;
    R04=R03*R34;
    R05=R04*R45;
    R06=R05*R56;
    R  = zeros(3,3,6);
    R0  = zeros(3,3,6);
    R(:,:,1) = R01; R(:,:,2) = R12; R(:,:,3) = R23; R(:,:,4) = R34;
    R(:,:,5) = R45; R(:,:,6) = R56;
    R0(:,:,1) = R01;R0(:,:,2) = R02;R0(:,:,3) = R03;
    R0(:,:,4) = R04;R0(:,:,5) = R05;R0(:,:,6) = R06;

    z1 = R01*[0;0;1]; z2 = R02*[0;0;1]; z3 = R03*[0;0;1];
    z4 = R04*[0;0;1]; z5 = R05*[0;0;1]; z6 = R06*[0;0;1];
    z = [z1 z2 z3 z4 z5 z6];

    r0 = [0;0;0.374];
    r1 = R01*[0.15; -0.176; 0.126];
    r2 = R02*[-0.825;0;0];
    r3 = R03*[0;-0.126;0.105];
    r4 = R04*[0;0;0.520];
    r5 = R05*[0;0;0];
    r6 = R06*[0;0;0.11];
    r = [r1 r2 r3 r4 r5 r6];
    % Initial Values
    w_0 = zeros(3,1); w_r0 = zeros(3,1); wdot_0 = zeros(3,1); mu_0 = -1*g; f_n1 = zeros(3,1); n_n1 = zeros(3,1);
    % Variables
    w = zeros(3,6);w_r = zeros(3,6);wdot = zeros(3,6);mu = zeros(3,6);f = zeros(3,6);n = zeros(3,6);
    psi_n = zeros(3,10,6); psi_f = zeros(3,10,6);
    torque = zeros(6,1);
    % Forward Iteration
    mu(:,1) = (mu_0 + (S(wdot_0) + (S(w_0)*S(w_r0))) * r0);
    w(:,1) = (w_0 + qv(1) * z1);
    w_r(:,1) = (w_r0 + qv_r(1) * z1);
    wdot(:,1) = (wdot_0 + ((S(R01.' * w_r0)*qv(1)+ eye(3)*qa_r(1))) * z1);
    for i = 2:6
        mu(:,i) = (mu(:,i-1) + (S(wdot(:,i-1)) + (S(w(:,i-1))*S(w_r(:,i-1)))) * r(:,i-1));
        w(:,i) = (w(:,i-1) + qv(i) * z(:,i));
        w_r(:,i) = (w_r(:,i-1) + qv_r(i) * z(:,i));
        wdot(:,i) = (wdot(:,i-1) + ((S(w_r(:,i-1))*qv(i)+ eye(3)*qa_r(i))) * z(:,i));
    end
    % Backward Iteration
    psi_n(:,:,6) = [zeros(3,1) S(mu(:,6)).' L(wdot(:,6))+(S(w_r(:,6))*L(w(:,6)))];
    psi_f(:,:,6) = [mu(:,6) S(wdot(:,6))+(S(w(:,6))*S(w_r(:,6))) zeros(3,6)];
    psi_ni = squeeze(psi_n(:,:,6));
    psi_fi = squeeze(psi_f(:,:,6));
    n(:,6) = psi_ni*p(51:60,1) + eye(3) * (S(r6)*f_n1 + n_n1);
    f(:,6) = psi_fi*p(51:60,1) + eye(3) * (f_n1);
    torque(6) = z(:,6).' * n(:,6);
    for i = 5:-1:1
        psi_n(:,:,i) = [zeros(3,1) S(mu(:,i)).' L(wdot(:,i))+(S(w_r(:,i))*L(w(:,i)))];
        psi_f(:,:,i) = [mu(:,i) S(wdot(:,i))+(S(w(:,i))*S(w_r(:,i))) zeros(3,6)];
        psi_ni = squeeze(psi_n(:,:,i));
        psi_fi = squeeze(psi_f(:,:,i));
        n(:,i) = psi_ni*p((10*(i-1)+1):(10*i),1) + (S(r(:,i))*f(:,i+1) + n(:,i+1));
        f(:,i) = psi_fi*p((10*(i-1)+1):(10*i),1) + f(:,i+1);
        torque(i) = z(:,i).' * n(:,i);
    end
end
```

Figure A.1 : MATLAB Code for MNEA

CURRICULUM VITAE

Name Surname : Muhammet Umut DANIŞ

EDUCATION :

- **B.Sc. :** 2020, Istanbul Technical University, Mechanical Engineering Faculty, Mechanical Engineering Department

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2022 – Present, Research Assistant at Istanbul Technical University, Mechanical Engineering Department