ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

FORCE CONTROL OF STÄUBLI RX-160 MANIPULATOR

M.Sc. THESIS

Serhat AKBAŞ

Department of Mechatronics Engineering

Mechatronics Engineering Programme

SEPTEMBER 2015

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF
SCIENCE ENGINEERING AND TECHNOLOGY

FORCE CONTROL OF STÄUBLI RX-160 MANIPULATOR

M.Sc. THESIS

Serhat AKBAŞ

518121028

Department of Mechatronics Engineering

Mechatronics Engineering Programme

Thesis Advisor : Doç. Dr. Zeki Yağız BAYRAKTAROĞLU

SEPTEMBER 2015

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

STÄUBLI RX-160 MANİPÜLATÖRÜNÜN KUVVET KONTROLÜ

YÜKSEK LİSANS TEZİ

Serhat AKBAŞ

518121028

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı : Doç. Dr. Zeki Yağız BAYRAKTAROĞLU

EYLÜL 2015

Serhat AKBAŞ, a **M.Sc.** student of ITU **Graduate School of Science Engineering and Technology**, student ID 518121028, successfully defended the **thesis** entitled "**FORCE CONTROL OF STÄUBLI RX-160 MANIPULATOR**", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Doç. Dr. Zeki Yağız BAYRAKTAROĞLU**
Istanbul Technical University

Jury Members : **Doç. Dr. Zeki Yağız BAYRAKTAROĞLU**
Istanbul Technical University

Prof. Dr. Ata MUĞAN
Istanbul Technical University

Yrd. Doç. Dr. Kadir ERKAN
Yıldız Technical University

Date of Submission : 27 August 2015
Date of Defense : 30 September 2015

FOREWORD

First of all, I thank my advisor Zeki Yağız Bayraktaroğlu for his continued support and interest throughout the thesis. I am grateful to him for the chance I got to study on a robotic manipulator.

I also thank Istanbul Technical University Mechatronics Education and Research Center personnel for providing materials I needed. I also appreciate the peaceful and comfortable workplace.

September 2015 Serhat Akbaş

TABLE OF CONTENTS

Page

FOREWORD .... **vii**
TABLE OF CONTENTS .... **ix**
ABBREVIATIONS .... **xi**
LIST OF TABLES .... **xiii**
LIST OF FIGURES .... **xv**
SUMMARY .... **xvii**
ÖZET .... **xix**
1. INTRODUCTION **1**
  1.1 Purpose of Thesis 1
  1.2 Literature Review 2
  1.3 Stäubli RX-160 Manipulator 3
    1.3.1 Technical properties 3
    1.3.2 Hardware 4
      1.3.2.1 Manipulator mechanics 4
      1.3.2.2 Actuators and encoders 6
      1.3.2.3 Controller and manual control pendant 6
      1.3.2.4 ATI force/torque sensor 7
      1.3.2.5 End-effector 9
    1.3.3 Software 10
2. MODELING AND IDENTIFICATION **13**
  2.1 Geometric Model 13
    2.1.1 Forward geometric model 13
      2.1.1.1 Coordinate frames 14
      2.1.1.2 Denavit-Hartenberg parameters 14
  2.2 Kinematic Model 17
    2.2.1 Forward kinematic model 17
    2.2.2 Inverse kinematic model 18
  2.3 Dynamic Model 19
    2.3.1 Euler-Lagrange formulation 20
    2.3.2 Newton-Euler formulation 22
  2.4 Interaction Forces 23
  2.5 Identification of Unknown Parameters 24
    2.5.1 Spring model 24
    2.5.2 Friction model 24
3. CONTROL SYSTEM DESIGN AND SIMULATIONS **27**
  3.1 Motion Control 27
    3.1.1 Trajectory generation 28
      3.1.1.1 Trapezoidal velocity profile 28
      3.1.1.2 Quintic polynomial position profile 29
      3.1.1.3 Trajectory generation in task space 30
    3.1.2 Task-space scheme 32

        3.1.2.1 Calculation of position error in task-space 33
        3.1.2.2 Simulation results 34
  3.2 Compliant Motion Control 36
    3.2.1 Impedance control 37
    3.2.2 Hybrid force/position control 41
4. IMPLEMENTATION AND EXPERIMENTS 47
  4.1 Implementation 47
  4.2 Experimental Results 49
    4.2.1 Motion control 49
    4.2.2 Impedance control 52
    4.2.3 Hybrid force/position control 60
5. CONCLUSION 65
REFERENCES 67
APPENDICES 69
CURRICULUM VITAE 123

ABBREVIATIONS

LLI **:** Low Level Interface
DOF **:** Degree of Freedom
PID **:** Proportional – Integral – Derivative Control
D-H **:** Denavit-Hartenberg

LIST OF TABLES

Page

Table 1.1 : Technical properties of RX-160 [7] 6
Table 1.2 : Calibration ranges and resolutions of ATI Delta transducers [9]. 9
Table 2.1 : D-H parameters for RX-160. 16

LIST OF FIGURES

Page

Figure 1.1 : Stäubli RX-160 manipulator [1]. 2
Figure 1.2 : The mechanical configuration of RX-160 [7]. 3
Figure 1.3 : Dimensions of RX-160 [7]. 4
Figure 1.4 : Work envelope of RX-160 [7]. 5
Figure 1.5 : CS8C robot controller [8]. 7
Figure 1.6 : Manual control pendant [8]. 8
Figure 1.7 : ATI Delta F/T sensor [9]. 8
Figure 1.8 : Frame attached to the sensor [10]. 9
Figure 1.9 : F/T controller [10]. 10
Figure 1.10 : End-effector. 10
Figure 1.11 : CS8C general architecture [11]. 11
Figure 1.12 : LLI implementation [11]. 12
Figure 2.1 : Coordinate frames of RX-160. 15
Figure 2.2 : Forces and moments acting on link i. 20
Figure 2.3 : Spring model, real data and percent error [15]. 25
Figure 2.4 : Joint friction with Stribeck effect [15]. 26
Figure 3.1 : Trapezoidal velocity profile. 28
Figure 3.2 : Position, velocity and acceleration profiles for trapezoidal velocity. 29
Figure 3.3 : Quintic polynomial position profile. 31
Figure 3.4 : Computed torque control in task-space. 33
Figure 3.5 : End-effector desired and feedback path in motion control. 35
Figure 3.6 : Translational errors in motion control. 35
Figure 3.7 : Rotational errors in motion control. 36
Figure 3.8 : Calculated joint torques in motion control. 36
Figure 3.9 : Nonlinear decoupling impedance control. 38
Figure 3.10 : Desired and feedback path in impedance control. 39
Figure 3.11 : Desired and feedback trajectories in impedance control. 39
Figure 3.12 : Translational errors in impedance control. 40
Figure 3.13 : Rotational errors in impedance control. 40
Figure 3.14 : Force applied by the end-effector in impedance control. 40
Figure 3.15 : Calculated joint torques in impedance control. 41
Figure 3.16 : Hybrid force/position control. 41
Figure 3.17 : Dynamic hybrid force/position control. 43
Figure 3.18 : Desired and feedback path in hybrid force/position control. 43
Figure 3.19 : Desired and feedback trajectories in hybrid force/position control. 44
Figure 3.20 : Translational errors in hybrid force/position control. 44
Figure 3.21 : Rotational errors in hybrid force/position control. 44
Figure 3.22 : Desired and feedback force in hybrid force/position control. 45
Figure 3.23 : Calculated joint torques in hybrid force/position control. 45
Figure 4.1 : Experimental setup. 47
Figure 4.2 : Task-space motion control diagram in experiments. 48

Figure 4.3 : Impedance control diagram in experiments. 48
Figure 4.4 : Hybrid force/position control diagram in experiments. 49
Figure 4.5 : Desired and feedback path. 50
Figure 4.6 : Desired and feedback trajectories. 50
Figure 4.7 : Translational errors. 51
Figure 4.8 : Rotational errors. 51
Figure 4.9 : Applied joint torques. 51
Figure 4.10 : Desired and feedback path. 52
Figure 4.11 : Desired and feedback positions. 53
Figure 4.12 : Desired and feedback velocities. 53
Figure 4.13 : Translational errors. 53
Figure 4.14 : Rotational errors. 54
Figure 4.15 : Feedback forces. 54
Figure 4.16 : Applied joint torques. 54
Figure 4.17 : Desired and feedback path. 55
Figure 4.18 : Desired and feedback positions. 55
Figure 4.19 : Desired and feedback velocities. 56
Figure 4.20 : Translational errors. 56
Figure 4.21 : Rotational errors. 56
Figure 4.22 : Feedback forces. 57
Figure 4.23 : Applied joint torques. 57
Figure 4.24 : Desired and feedback path. 58
Figure 4.25 : Desired and feedback positions. 58
Figure 4.26 : Desired and feedback velocities. 58
Figure 4.27 : Translational errors. 59
Figure 4.28 : Rotational errors. 59
Figure 4.29 : Feedback forces. 59
Figure 4.30 : Applied joint torques. 60
Figure 4.31 : Desired and feedback path. 61
Figure 4.32 : Desired and feedback trajectories. 61
Figure 4.33 : Translational errors. 62
Figure 4.34 : Rotational errors. 62
Figure 4.35 : Desired and feedback forces. 62
Figure 4.36 : Applied joint torques. 63

FORCE CONTROL OF STÄUBLİ RX-160 MANIPULATOR

SUMMARY

Today, humans are started to be replaced by robots in many areas. Work speed and quality of robots are increasing day by day with the developing robot technologies. Robots are becoming irreplaceable components of industrial applications. With recent developments, robots can perform surgical operations and be used in space vehicles.

Industrial manipulators are mostly used in applications like packing, welding, material handling and painting on a production line. While performing these tasks, manipulators have very limited interaction with the environment or none at all. The main purpose of these tasks is to follow a path in free space.

Manipulators can also be used in tasks where interaction is desired. Assembling, deburring, grinding and polishing are among these tasks. However, classical motion control methods are insufficient for these tasks. The position of the manipulator and the workpiece must be known with a high precision and the environment must be modeled accurately in order to execute these tasks with pure motion control. These conditions cannot always be satisfied. If we assume a rigid manipulator and environment, a small error can cause an increase to contact forces to a point where the workpiece or the manipulator can be damaged. For this reason, we need some degree of “compliance” between the manipulator and the environment.

“Compliant motion control” or “force control” can give us a better solution in cases where interaction with the environment is present. Contact forces are controlled directly or indirectly by generating proper control signals.

In this thesis, two widely known force control strategies are considered: *impedance control* and *hybrid force/position control*. Impedance control can be classified as an indirect force control where errors in the motion are related with contact forces. This relation, known as *mechanical impedance*, has an advantage that its parameters can be adjusted to ensure a desired behavior of interaction. On the other hand, hybrid force/position control, which is classified as a direct force control, requires desired values of contact forces. In return, the controller calculates the error between the desired and feedback forces and generates a proper control signal to follow desired force trajectory.

For both of the control strategies mentioned above, contact force measurements from a force/torque sensor are used. These sensor data are certainly required for hybrid force/position control. However, for impedance control, there are control schemes where force feedback is not required.

The purpose of this thesis is applying different force control algorithms on Stäubli RX-160 industrial type manipulator that resides in Istanbul Technical University Mechatronics Education and Research Center.

By using LLI (Low Level Interface) provided by Stäubli, control algorithms that are reported in literature can be embedded to manipulator controller and tested. LLI

enables us to control all aspects to control an industrial type manipulator where we can use our own kinematic and dynamic models.

This thesis consists of five chapters. In the first chapter, the purpose of thesis, technical properties of RX-160, hardware and software that are used in this study will be mentioned.

In second chapter, kinematic and dynamic model of RX-160 are derived and unknown parameters are mentioned. In order to derive dynamic model of the robot, mass properties must be known. These parameters are provided by Stäubli. The spring model and friction parameters are identified in a past work and they are used directly in this thesis.

In chapter three, some control methods are introduced and simulation results are shown. Trajectory planning is also mentioned in this chapter.

In fourth chapter, the control methods mentioned in the previous chapter will be applied to the manipulator in real-time. The data collected from sensors are analyzed and commented.

In fifth and the last chapter, all the work done throughout the thesis is summarized and suggestions are made for future works.

STÄUBLİ RX-160 MANİPULATÖRÜNÜN KUVVET KONTROLÜ

ÖZET

Günümüzde robotlar pek çok işte insanların yerini almaya başlamıştır. Her gün gelişen robot teknolojisi, robotların iş yapma hızını ve iş kalitesini artırmaktadır. Robotlar kullanıldıkları alanların vazgeçilmez unsuru olmaktadır. Bugün robotlar, cerrahi operasyonlarda ve uzay araçlarında kullanılabilmektedir.

Robotlar arasında manipülatörler, endüstriyel uygulamalarda sıklıkla karşımıza çıkmaktadır. Endüstride üretim hızının ve kalitesinin sürekli artması istendiği için, gelecekte daha fazla robotun üretim sahasında kullanılacağını söyleyebiliriz.

Endüstriyel manipülatörler, çoğunlukla üretim hattında parça taşıma, paketleme, kaynak ve boya yapma işlerinde kullanılmaktadır. Dikkat edilecek olursa, bu uygulamalarda çevre ile temas çok az veya hiç yoktur. Robot, izlediği yol boyunca herhangi bir engelle karşılaşmaz. Diğer bir deyişle, bu uygulamalarda asıl amaç çevre ile etkileşim değil, serbest uzayda planlanmış bir hareketi gerçekleştirmektir.

Endüstriyel manipülatörler çevre ile etkileşim gerektiren uygulamalarda da kullanılabilmektedir. Montaj, çapak alma, taşlama ve cilalama bu uygulamalardandır. Bu tip uygulamalarda klasik konum kontrolü yetersiz kalmaktadır. Klasik konum kontrol yöntemleriyle istenen kalitede iş yapmak için, iş parçasının konumunun son derece hassas olması, manipulatörün ve çevrenin de çok iyi modellenmiş olması gerekmektedir. Bu şartları sağlamak her zaman mümkün olmayabilir. Çevre ve manipulatörün sertliğinin yüksek olduğunu düşünürsek, bu şartların sağlanamaması durumunda, konumda ufak bir hatanın oluşması, çok büyük temas kuvvetlerinin ortaya çıkmasına sebep olmaktadır. Bu da iş parçasına veya manipülatöre zarar verebilir.

Eğer çevre ile etkileşim söz konusuysa, "uyumlu hareket kontrolü" veya "kuvvet kontrolü" bize daha iyi sonuç verecektir. Bu kontrol yönteminde, serbest konum kontrol yönteminden farklı olarak, çevre ile olan temas sonucunda oluşan kuvvetler dikkate alınır, bu kuvvetler doğrudan veya dolaylı yoldan işleme sokulur ve uygun kontrol sinyalleri üretilir.

Kuvvet kontrolü, robotik sistemlerin bilinmeyen bir çevrede öngörülemeyen durumlarda çalışabilmesi ve esnek bir davranış sergilemesinde ve akıllı tepkiler vermesinde önemli bir rol oynamakta ve insan-robot etkileşimine katkıda bulunmaktadır.

Yukarıda anlatılan çevre ile uyumlu davranış iki şekilde elde edilebilir: pasif etkileşim kontrolü ve aktif etkileşim kontrolü.

Pasif etkileşim kontrolünde, temas kuvvetleri, robotun yapısal esnekliğiyle, servo motorların esnekliğiyle veya robotun ucuna monte edilmiş özel bir uç işlevciyle kontrol altında tutulmaya çalışılır. Endüstriyel uygulamalarda genellikle üçüncü yöntem kullanılır. Bu metot basit ve ucuz olmakla birlikte, robotun çevreyle temasından doğan hatalara karşı tepkisi de (programlanmış bir kontrol algoritmasından farklı olarak) hızlıdır. Fakat saydığımız avantajlarına rağmen bu yöntem, farklı

endüstriyel uygulamalara karşı esnekliğe sahip değildir. Her uygulama için özel olarak bir uç işlevci tasarlanması gerekmektedir.

Aktif etkileşim kontrolünde, manipulatörün çevre ile uyumu özel olarak tasarlanmış bir kontrol sistemi sayesinde sağlanmaya çalışılır. Bu kontrol yöntemlerinde genellikle robotun bileğine monte edilmiş bir kuvvet sensörünün ölçtüğü temas kuvvetleri geri besleme ile işleme sokulur. Bu tezde kullanılan kuvvet sensörü, iş uzayında altı kuvvet bileşenini (üç kuvvet, üç moment) ölçebilmektedir.

Bu tezde aktif etkileşim kontrolü ele alınmıştır. Aktif etkileşim kontrolü ayrıca kendi içerisinde iki kategoriye ayrılabilir: dolaylı kuvvet kontrolü ve doğrudan kuvvet kontrolü. Dolaylı kuvvet kontrolünde, temas kuvvetleri, modifiye edilmiş bir hareket kontrol metoduyla kontrol edilir. Doğrudan kuvvet kontrolünde ise kullanıcı, manipülatöre istenen temas kuvvetlerini besler ve manipulatör de bu referans değerlerini izlemeye çalışır.

İlk kategorideki kontrol yöntemleri için *empedans kontrolü* örnek olarak verilebilir. Bu yöntemde, manipulatörün hareketi sırasında çevre ile etkileşim sebebiyle ortaya çıkan konum hataları, temas kuvvetleriyle ilişkilendirilir. Bu ilişki ise *mekanik empedans* olarak tanımlanır. Bu kontrol yönteminde, manipulatör çevre ile etkileşimi sırasında istenen mekanik davranışı göstermeye zorlanır.

İkinci kategoriye örnek olarak *hibrid kuvvet/konum kontrolü* verilebilir. Bu kontrol yönteminde iş uzayı, iki alt uzaya ayrılır. Bu alt uzaylardan birine konum kontrolü uygulanırken diğerine kuvvet kontrolü uygulanır. Aynı alt uzaya hem konum, hem de kuvvet kontrolü uygulanamaz. Kuvvet kontrolü uygulanan uzayda, doğrudan istenen temas kuvvetleri üretilir ve bunların takip edilmesi için gerekli kontrol sinyalleri üretilir. Konum kontrolünden ve kuvvet kontrolünden üretilen kontrol sinyalleri eklem torklarına (veya kuvvetlerine) dönüştürüldükten sonra bu değerler toplanır ve manipülatöre gönderilir.

Bu tezde, İstanbul Teknik Üniversitesi Mekatronik Eğitim ve Araştırma Merkezi bünyesindeki Stäubli RX-160 endüstriyel manipulatörünün üzerinde farklı kuvvet kontrol algoritmaları uygulanması amaçlanmaktadır.

Stäubli RX-160 endüstriyel robotik manipulatör, boya yapmak veya iş parçalarını üretim hattında bir noktadan diğerine taşımak amacıyla tasarlanmıştır. Robot orta boyutta olup, nominal hızlarda 20 kg yük taşıyabilmektedir.

Stäubli firmasının sağlamış olduğu LLI programlama arayüzü sayesinde, manipulatör üzerinde, literatürde önerilen kontrol algoritmaları, manipulatör kontrolcüsüne gömülüp test edilebilmektedir. LLI, kullanıcıya manipülatörü her yönüyle ele alıp kontrol edebilmesini sağlamaktadır. Bu da akademik çalışmalar için bize avantaj sağlamaktadır. Literatürde gösterilen kontrol şemaları, robot üzerinde denenebilmekte ve bu şemaların performansları ölçülebilmektedir. LLI aynı zamanda birtakım güvenlik fonksiyonları da içermektedir. Bu da, bir ölçüye kadar, robot üzerinde yapılan denemelerde meydana gelen hataların robota veya çevreye zarara vermesini önlemektedir.

Bu tez beş ana bölümden oluşup, ilk bölümde tezin amacına, Stäubli RX-160 manipulatörün başlıca teknik özelliklerine ve manipulatörün donanım ve yazılımına değinilmiştir.

İkinci bölümde, manipulatörün kinematik ve dinamik modelleri çıkarılıp, manipulatörün bilinmeyen parametrelerinin tanınmasından bahsedilmektedir. Bu modeller kontrol için gerekmektedir. Dinamik modelin çıkarılması için gereken kütle parametreleri, Stäubli tarafından sağlanmıştır. Geçmişte bu robot üzerinde yapılmış tanımlama çalışmasından elde edilen sürtünme ve yay modeli bu tezde doğrudan kullanılmıştır.

Üçüncü bölümde, ilk olarak yörünge planlamasından bahsedilmiştir. Yörünge planlaması robotun izlemesi gereken yolun zamana bağlı denklemini verir. Bu bölümde daha sonra, hareket kontrol şeması gösterilmiştir. Ardından, kuvvet kontrol şemaları gösterilmiş ve bunların benzetim ortamında elde edilen sonuçları verilmiştir. Benzetim sonuçları, kontrol şemalarının gerçek zamanlı uygulanmasından önce robotun hareketi ve eklemlere uygulanması gereken kuvvetler hakkında ipuçları vermiştir. Fakat benzetim sonuçları ile gerçek zamanlı uygulama sonuçları arasında bir karşılaştırma yapılması amaçlanmamıştır. Bunun nedeni, ileri dinamik robot modeli ve kontrol şemasında kullanılan ters dinamik robot modelinin benzetimlerde birebir aynı olmasıdır. Bu da, robotun mükemmel bir şekilde modellendiğini ileri sürer ki böyle bir varsayım gerçekçi değildir.

Dördüncü bölümde, ilk olarak deney ortamından bahsedilmiş, deneyler sırasında ortaya çıkan sorunlara değinilmiştir. Daha sonra, bir önceki bölümde bahsedilen kontrol şemaları, gerçek zamanlı olarak robota uygulanmış ve elde edilen sonuçlar gösterilmiştir.

Beşinci ve son bölümde, yapılan çalışmalar özetlenip gelecekte yapılabilecek çalışmalar için öneriler sunulmuştur.

1. INTRODUCTION

Robotics is a multidisciplinary engineering area that is interested in designing machines that support humans in variety of tasks. Mainly, mechanical engineering, electrical engineering, computer science and mathematics are needed to have a good understanding in robotics.

The term “robot” was started to be used after a story named *Rossum’s Universal Robots* written by Karel Capek in 1920. In this story, the word “robota”, meaning “work” in Czech was used.

There are many machines that can be classified as a robot today. Rovers, exoskeletal robots, humanoid robots and unmanned aerial vehicles (UAVs) are among them. In this thesis we will be concerned only with serial type industrial robotic manipulators. Throughout the thesis, the term “manipulator” will be used to refer industrial robotic manipulators. The manipulator that is used in this study is shown in Figure 1.1.

Manipulators are essentially a mechanical arm with serially connected links. Today, they are used in various applications. Material transfer, welding, painting and assembling are among them. Manipulators have high work speed, accuracy and they reduce production costs in industry. These attributes caused humans to be replaced by manipulators in some tasks. Also, work place may be too dangerous for a human to work in. The manipulators are employed in those situations to ensure human safety.

In order to control a manipulator with a model based approach, we have to derive its mathematical model first. These models are grouped as kinematic model and dynamic model. These models are then used in a chosen control algorithm. The performance of the control algorithm depends on an accurate and detailed model of the manipulator.

1.1 Purpose of Thesis

Today, some industrial applications require interaction of the robot with the environment. Examples for such applications are, polishing, grinding, assembly, and machining. While the manipulator is following a desired path, interaction forces must be controlled.

Figure 1.1 : Stäubli RX-160 manipulator [1].

During interaction, the environment sets constraints on the paths that can be followed by the end-effector. This situation is referred to as “constrained motion”. If we want to use pure motion control to limit interaction forces, we need an accurate model of robot as well as the environment. A manipulator can be modeled accurately to a certain degree. However an accurate model of the environment is generally difficult to obtain. Even small errors may be intolerable and cause rise to contact forces that can damage the environment or the manipulator. For this reason, some degree of “compliance” is expected from the manipulator.

The purpose of this thesis is the adaptation of Stäubli RX-160 manipulator that resides in Istanbul Technical University Mechatronics Education and Research Center to force control applications. For this purpose, control algorithms are chosen from literature and tested in the simulation environment and then applied to the manipulator in real-time.

1.2 Literature Review

Much work has been carried out in the force control area. Active stiffness control in Cartesian coordinates was proposed in [2]. Original idea of mechanical impedance model used for controlling interaction between the manipulator and the environment is presented in [3]. The original hybrid position/force control concept was introduced in [4]. The explicit inclusion of the manipulator dynamic model was presented in [5]. Kinematic instability issue in hybrid position/force control is pointed out and a correct formulation is proposed in [6].

1.3 Stäubli RX-160 Manipulator

Stäubli RX-160 is a medium payload, rigid, highly precise articulated arm with six degrees of freedom. It is designed mainly for machine tending, in which the manipulator is used to load and unload machine tools [1].

In this section, the components of Stäubli RX-160 manipulator will be mentioned.

1.3.1 Technical properties

The manipulator's links are interconnected serially by six joints (Figure 1.2). Links are named as: the base (A), the shoulder (B), the arm (C), the elbow (D), the forearm (E), and the wrist (F). Dimensions of the manipulator are shown in Figure 1.3.

Motion of joints are generated by six brushless motors that are equipped with resolvers. Also, every motor is supported with a parking brake.

Working temperature of the manipulator is between $+5^{\circ}\mathrm{C}$ and $+40^{\circ}\mathrm{C}$. Humidity of the work environment should be between $30\% - 95\%$. Maximum work altitude is 2000 m. Manipulator weighs 248 kg. The work envelope of the manipulator is shown in Figure 1.4.

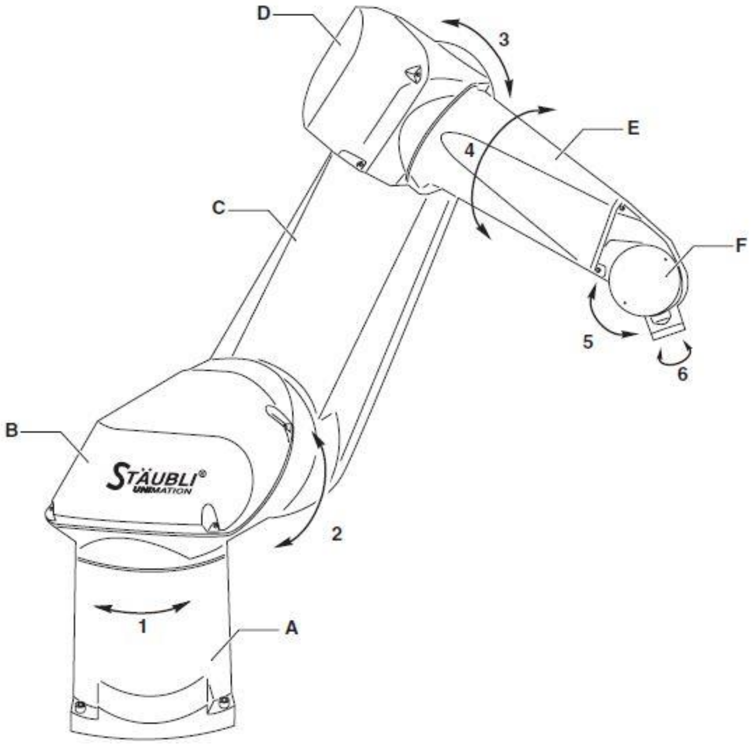


Figure 1.2 : The mechanical configuration of RX-160 [7].

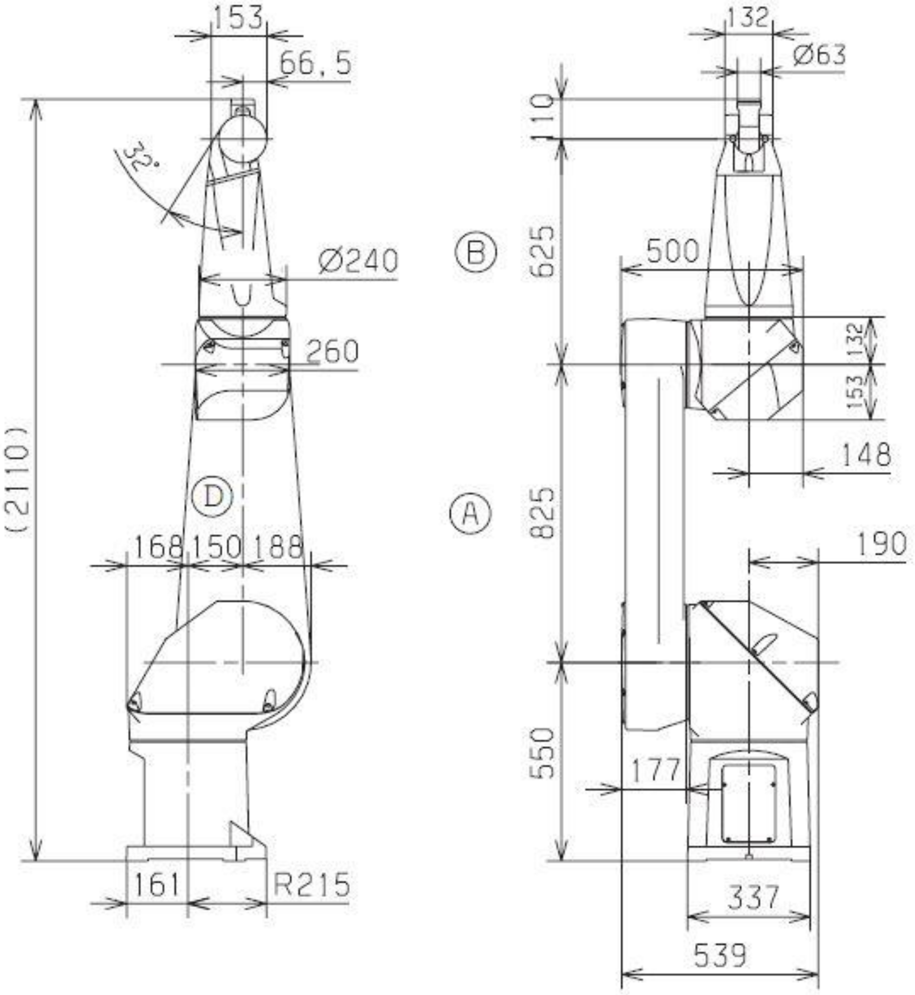


Figure 1.3 : Dimensions of RX-160 [7].

In Table 1.1, joints' amplitudes, working ranges, nominal speeds, maximum speeds and angular resolutions are given.

The manipulator's load capacity is 20 kg at nominal speeds in all configurations while taking maximum inertias into account.

1.3.2 Hardware

In this section, we will introduce equipment of the robot, force/torque sensor and the end-effector.

1.3.2.1 Manipulator mechanics

The robot arm assembly contains the motorization, brakes, motion transmission mechanisms, cable bundles, pneumatic and electric circuits for the user and the counterbalance system [7].

The RX-160 arm assembly consists of a rigid and encased structure to protect it against external aggressions. Its design is based on transmission modules: JSC (Stäubli Combined Joint) used on first four axes. The wrist consists of axis 5 and 6.

Arm balance is performed by an integrated spring system. The arm has built-in spring counterbalance system giving an attractive low-weight system [7].

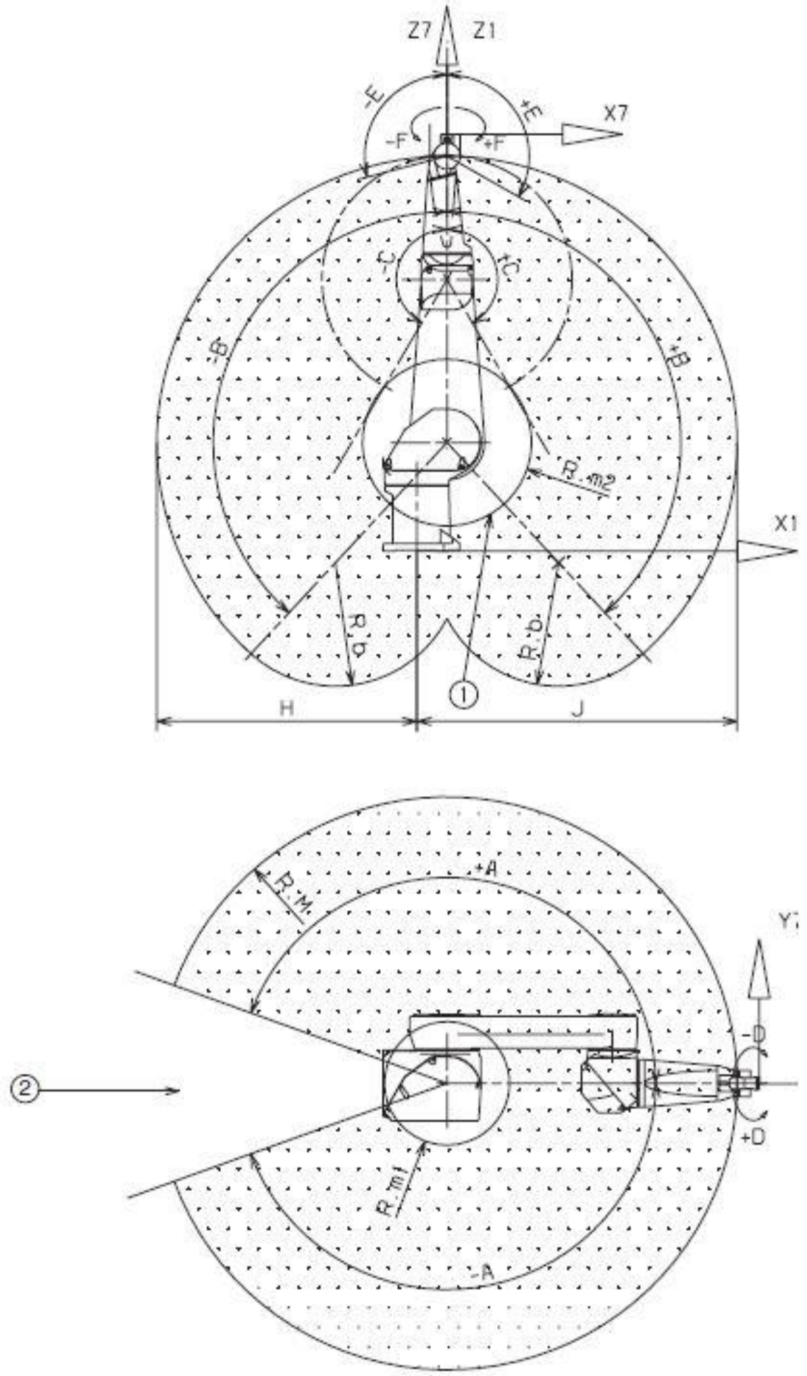


Figure 1.4 : Work envelope of RX-160 [7].

Table 1.1 : Technical properties of RX-160 [7].

Axis	Amplitude (°)	Working range distribution (°)	Nominal speed $(°/s)$	Maximum speed $(°/s)$	Angular resolution $(°.10^{-3})$
1	320	A ± 160	165	200	0.042
2	275	B ± 137.5	150	200	0.042
3	300	C ± 270	190	255	0.054
4	540	D ± 270	295	315	0.062
5	225	E + 120 - 105	260	390	0.12
6	540	F ± 270	440	870	0.17

1.3.2.2 Actuators and encoders

Every axis on the robot is actuated by 3-phase brushless DC motor over a gearbox. Position and velocity of joints are measured by encoders on every axis.

1.3.2.3 Controller and manual control pendant

In Figure 1.5, the controller named “CS8C” is shown. The CS8C controller is made up of a processor (5), the intelligent part of the installation. The processor controls the robot via digital power amplifiers (1) dedicated to each axis of the arm. The electrical power is converted by PSM (7) power section, the RPS (2) power supply, and the ARPS (3) power supply which supplies to each of the above elements the voltage required for correct operation from main voltage delivered by the electrical network. The functions required for electrical safety are grouped together on the RSI (4) board. The main power supply switch is shown with (6). The controller has two serial links and two ethernet ports for communication. For fieldbus, DeviceNet, Profibus, CANopen and ModBus TCP client are available.

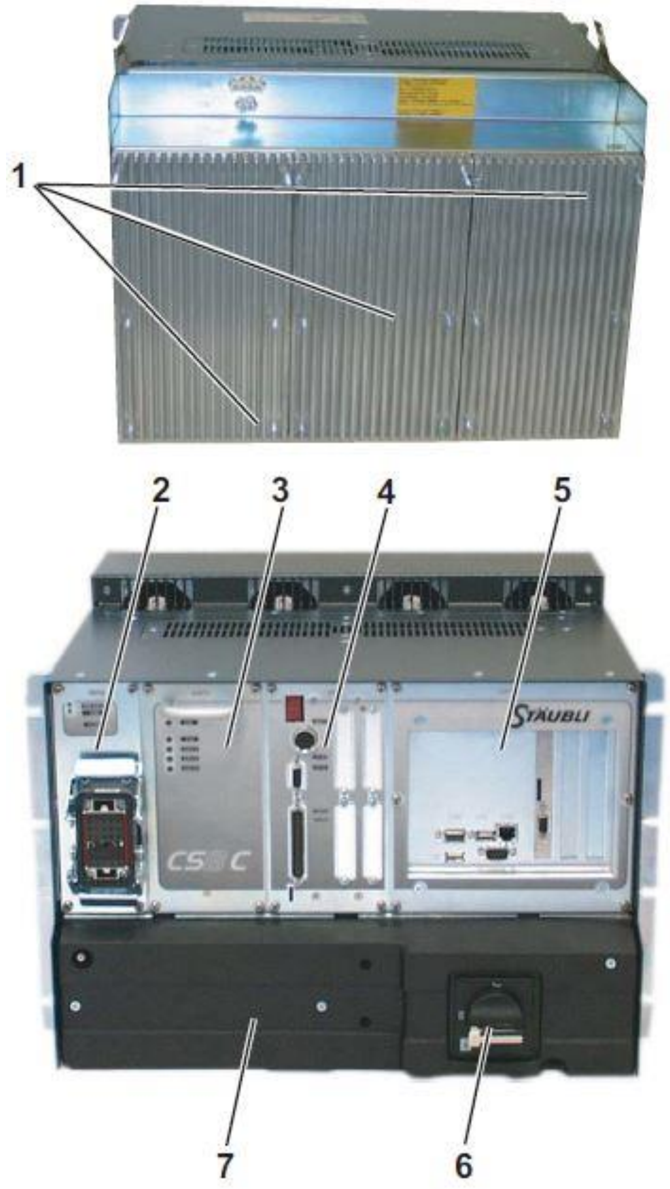


Figure 1.5 : CS8C robot controller [8].

Communication between the operator and the controller is performed by the “manual control pendant” (MCP) shown in Figure 1.6. In our study, MCP is not used for control purposes but kept close in order to use emergency stop button in situations like unexpected arm movements.

1.3.2.4 ATI force/torque sensor

In our applications, contact forces on the end-effector must be measured and fed back in order to realize force control of the manipulator. For this purpose, ATI Delta force/torque sensor is used (Figure 1.7). This sensor can measure all six components of force and torque in Cartesian space.

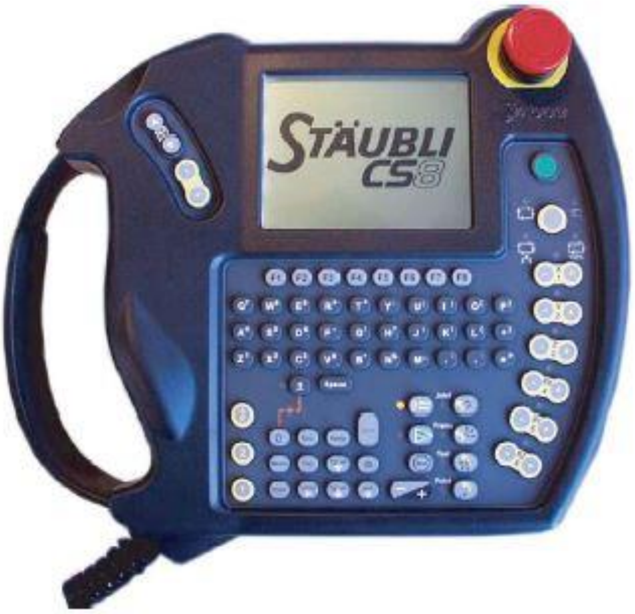


Figure 1.6 : Manual control pendant [8].

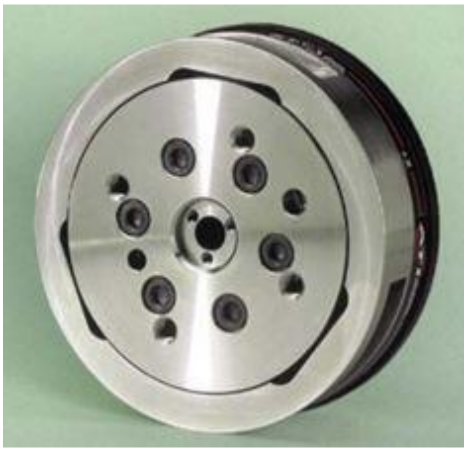


Figure 1.7 : ATI Delta F/T sensor [9].

The default frame attached to the sensor is shown in Figure 1.8. This frame has to be converted to the end-effector frame, in order to get correct measurements.

The calibrations ranges that can be used with Delta model transducers are shown in Table 1.2. Our transducer is calibrated within SI-660-60.

The controller of the force/torque sensor system is shown in Figure 1.9. The controller transforms strain gage data to Cartesian force/torque components. The analog output of the transducer controller is connected to a WAGO terminal. Analog data can be read from WAGO terminal using ModBus TCP/IP protocol. The sensor's controller is also connected to a PC with serial connection to access controllers interface. We send commands (sensor biasing, tool frame creation etc.) via this interface.

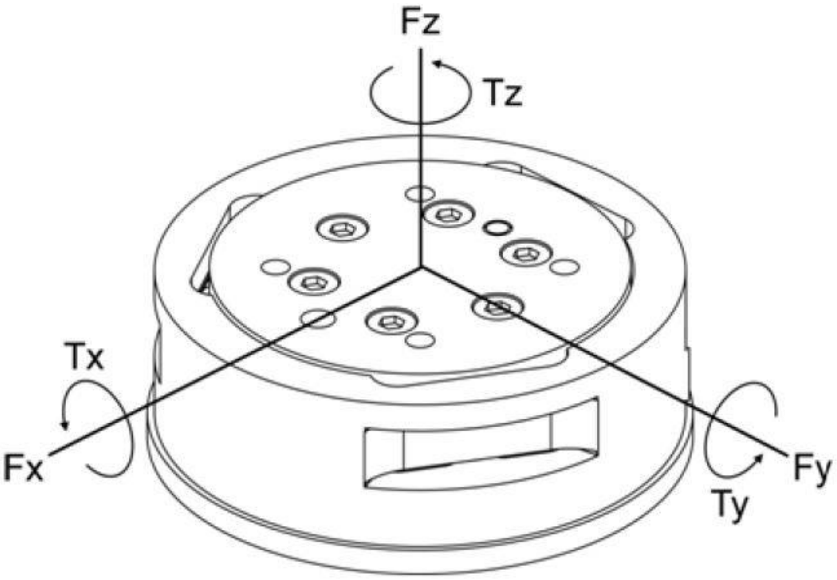


Figure 1.8 : Frame attached to the sensor [10].

Table 1.2 : Calibration ranges and resolutions of ATI Delta transducers [9].

	Sensing Ranges				Resolution			
Calibration	F_x, F_y	F_z	T_x, T_y	T_z	F_x, F_y	F_z	T_x, T_y	T_z
SI-165-15	165 N	495 N	15 N.m	15 N.m	1/32 N	1/16 N	1/528 N.m	1/528 N.m
SI-330-30	330 N	990 N	30 N.m	30 N.m	1/16 N	1/8 N	5/1333 N.m	5/1333 N.m
SI-660-60	660 N	1980 N	60 N.m	60 N.m	1/8 N	1/4 N	10/1333 N.m	10/1333 N.m

1.3.2.5 End-effector

The end-effector, shown in Figure 1.10 is attached to the wrist in our study. Since we made our tests on a stiff surface, this end-effector made our tests safer. Also, the ball wheel on the tip is free to rotate in all directions. Therefore, the contact friction between surface and the tip of the robot is lowered.

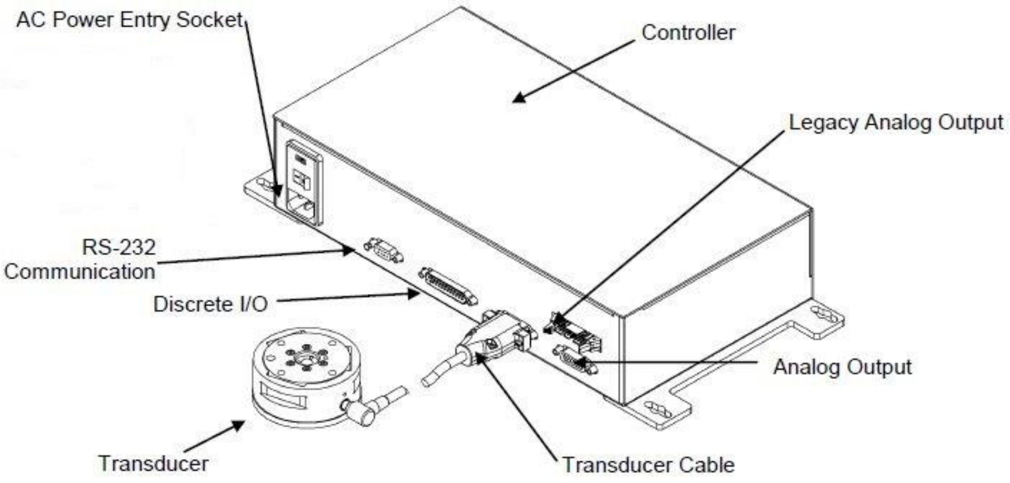


Figure 1.9 : F/T controller [10].

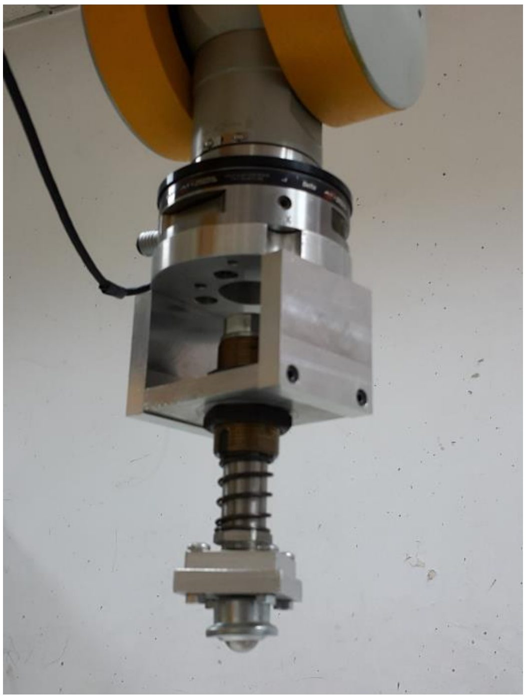


Figure 1.10 : End-effector.

1.3.3 Software

There are two different programming interfaces that are provided by Stäubli to program the manipulator. The first one is called “VAL3” which uses a high level programming language that is suitable for industrial applications. The second one is called “LLI (Low Level Interface)” which contains library functions written in C language. In our study, all control algorithms are written and embedded in LLI.

Using robots in engineering schools, universities and applied research centers is a plus, both for students and teachers [11].

LLI gives user a complete freedom who has to manage all aspects of the robot control: its geometry, its kinematics and its dynamics.

The CS8C controller integrates a PENTIUM® arithmetic unit and the VxWorks® real-time operating system. The general architecture is shown in Figure 1.11. The C or C++ programming will then be made with Windriver®'s Tornado® environment of development.

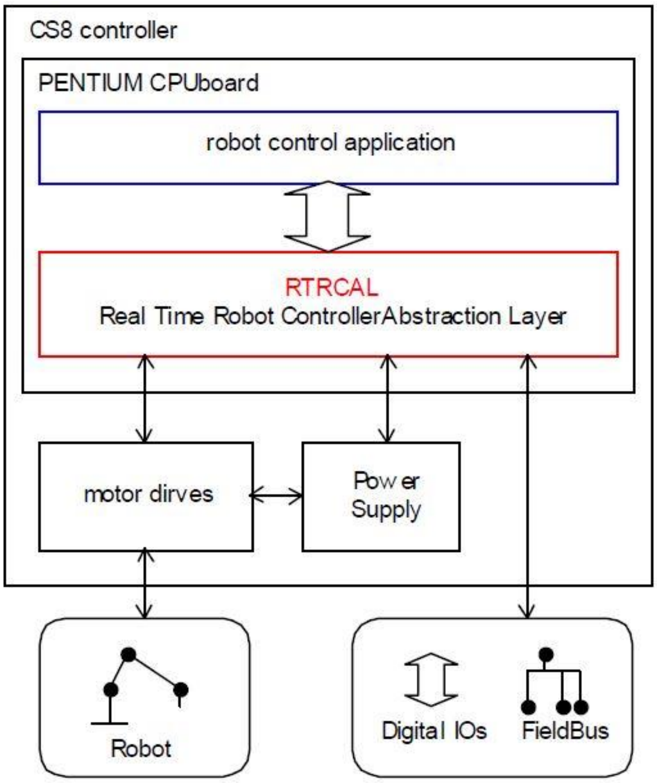


Figure 1.11 : CS8C general architecture [11].

At every cycle time, the robot receives command inputs (cmd) and provide sensory feedback information (fbk). The working principle of LLI is shown in Figure 1.12. The feedback information consists of position, velocity, torque and position error at joint level. The command is represented by the position, velocity, velocity anticipation and torque anticipation.

There are two working modes in LLI: position-velocity mode and torque mode. In position-velocity mode, position and velocity are updated according to user specified trajectory at every cycle. The controller calculates the torques that must be supplied to joints in order to follow the desired trajectory. In torque mode, the user must calculate the torques needed and send them to drives. In our study, we tested our control algorithms in torque mode.

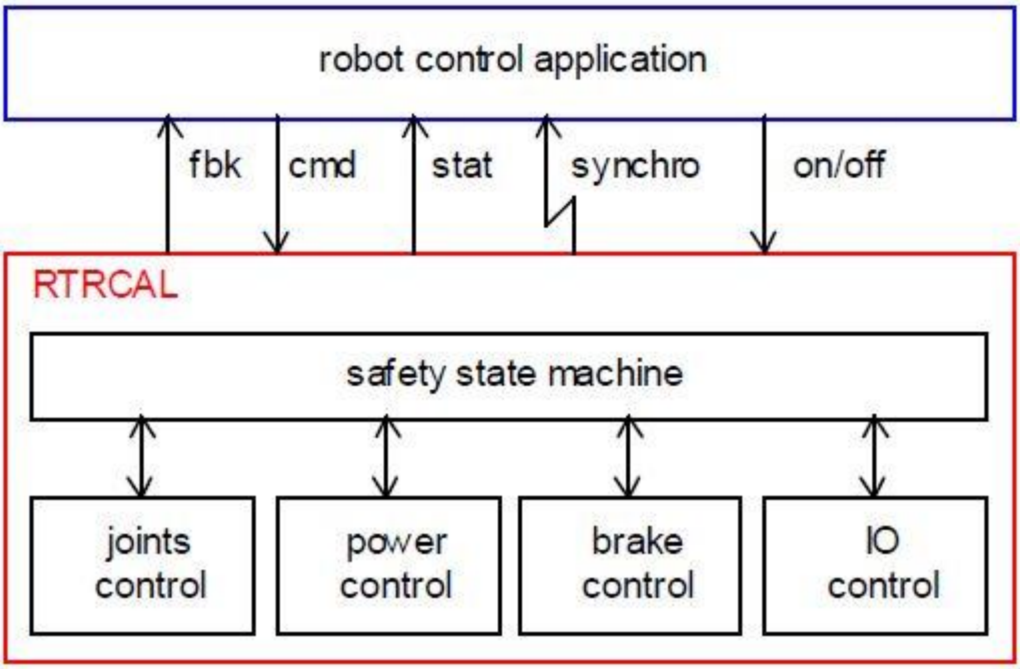


Figure 1.12 : LLI implementation [11].

2. MODELING AND IDENTIFICATION

In robotics, modeling generally separated into two main groups: kinematic model and dynamic model. Kinematic model describes the motion of the manipulator without considering the forces that cause the motion. On the other hand a dynamic model gives us forces or moments for a desired motion.

In this section we will generate both kinematic and dynamic model of the manipulator. Also, we will mention identification of spring and joint frictions.

2.1 Geometric Model

Geometric model gives the relation between joint positions and the position and orientation of a point on the manipulator (end-effector tip point in our case). In this section, forward geometric model of the manipulator will be generated which gives us the position and orientation in terms of joint variables. Inverse geometric model is not needed in our study so it will not be discussed here.

2.1.1 Forward geometric model

In a serial manipulator, a link is connected to its adjacent links with one joint. So a manipulator with n joints will have $n+1$ links. Joint i connects link $i-1$ to link i. Link 0 is fixed and referred as the "base link".

Generally, manipulators have joints with a single DOF (degree of freedom) which are revolute and prismatic joints. The angle of rotation in revolute joints and linear displacement in prismatic joints are called "joint variables" and denoted as "q".

$$q_i = \begin{cases} \theta_i & if\ joint\ i\ is\ revolute \\ d_i & if\ joint\ i\ is\ preismatic \end{cases} \tag{2.1}$$

Stäubli RX-160 have six joints which are all revolute type. Therefore, from now on, all models will be generated for revolute joints and prismatic joints will not be mentioned.

2.1.1.1 Coordinate frames

First, we have to attach coordinate frames on links in order to find geometric model. For this purpose, we will use Denavit and Hartenberg (D-H) convention for selecting frames of reference. Frames are attached according to the following procedure:

- z_i-axis is aligned with the axis of rotation of joint $i+1$. The direction can be chosen arbitrarily.
- x_i-axis is chosen along the common normal of z_{i-1} and z_i pointing from z_{i-1} to z_i. x_0-axis can be chosen arbitrarily. If z_{i-1} and z_i coincides, the direction of x_i can be chosen in the direction of $z_{i-1} \times z_i$ or in the opposite direction. If z_{i-1} and z_i are parallel, there is infinite number of common normal. In this situation x_i-axis can be determined in a convenient manner.

Attached coordinate frames is shown in Figure 2.1. The origin of the last frame (frame 6) is at the tip of the end-effector which is not shown in this figure.

2.1.1.2 Denavit-Hartenberg parameters

We use homogeneous transformation matrices to express a coordinate frame with respect to another. These matrices have the form,

$$ {}^{i-1}A_i = \begin{bmatrix} {}^{i-1}R_i & {}^{i-1}p_i \\ 0 & 1 \end{bmatrix} \tag{2.2}$$

${}^{i-1}A_i$ is a 4×4 homogeneous matrix, that expresses the position and orientation of frame i with respect to frame $i-1$. ${}^{i-1}R_i$ is the 3×3 rotation part and ${}^{i-1}p_i$ is the 3×1 translation part of ${}^{i-1}A_i$ (from now on superscript will be omitted). In order to find the position and the orientation of the end-effector, we have to express the end-effector frame with respect to the base frame. This can be done by multiplying transformation matrices sequentially as in equation (2.3).

$$ {}^{0}T_n = A_1 A_2 \dots A_{n-1} A_n \tag{2.3}$$

According to D-H method, one can find A_i with four successive transformations.

$$ A_i = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \tag{2.4}$$

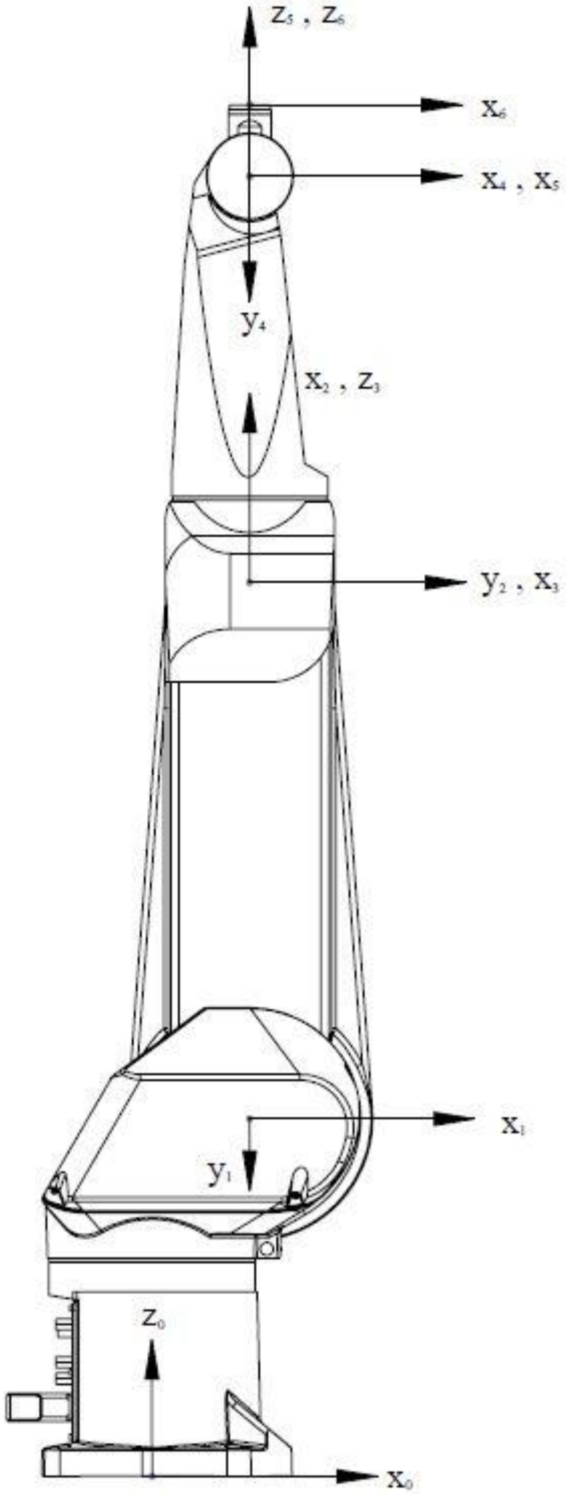


Figure 2.1 : Coordinate frames of RX-160.

Terms in equation (2.4) can be written in matrix form as,

$$Rot_{z,\theta_i} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

$$Trans_{z,d_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.6}$$

$$Trans_{x,a_i} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

$$Rot_{x,\alpha_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.8}$$

$$A_i = \begin{bmatrix} c\theta_i & -s\theta_i\, c\alpha_i & s\theta_i\, s\alpha_i & a_i\, c\theta_i \\ s\theta_i & c\theta_i\, s\alpha_i & -c\theta_i\, s\alpha_i & a_i\, s\theta_i \\ 0 & s\alpha_i & s\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

In equations (2.5) through (2.9), θ_i, d_i, a_i and α_i are called as “D-H parameters”. Although there are four parameters, three of them are constant for a single DOF joint. In our case d_i, a_i, and α_i are constant and θ_i is the joint variable. D-H parameters for RX-160 are given in Table 2.1.

Table 2.1 : D-H parameters for RX-160.

Joint	θ_i (rad)	d_i (mm)	a_i (mm)	α_i (rad)
1	θ_1	550	150	$-\pi/2$
2	$\theta_2 - \pi/2$	0	825	0
3	$\theta_3 + \pi/2$	0	0	$\pi/2$
4	θ_4	625	0	$-\pi/2$
5	θ_5	0	0	$\pi/2$
6	θ_6	320	0	0

2.2 Kinematic Model

In this section, we will consider joint and link velocities and their relations. We will introduce Jacobian matrix for kinematic analysis of a manipulator. In robotics, Jacobian matrix has many uses including, differential motions, relating joint velocities with link velocities, singular configuration analysis, derivation of dynamic model and static model analysis.

2.2.1 Forward kinematic model

In forward kinematic model, we write end-effector velocity $\dot{X}_e$ in terms of joint velocities $\dot{q}$ as

$$\dot{X}_e = J(q)\dot{q} \tag{2.10}$$

where $J(q)$ is called "Jacobian matrix" and it can be seen that it is configuration dependent which means that for a given set of joint velocities, the end-effector velocity will change according to joint positions.

This matrix can also be used to find differential displacements as

$$dX = J(q)dq \tag{2.11}$$

Here, the vector $q = (q_1 \dots q_n)^T$ gives joint variables. $J(q)$ is a $6 \times n$ matrix and obtained by taking [12] as reference.

Equation (2.10) can be divided into two equations as in equation (2.12) and (2.13).

$$\dot{p}_e = J_p(q)\dot{q} \tag{2.12}$$

$$\omega_e = J_o(q)\dot{q} \tag{2.13}$$

Here, $\dot{p}_e$ and ω_e are linear and angular velocity of the end-effector respectively. J_p is the upper $3 \times n$ part and J_o is the lower $3 \times n$ part of J. We are going to consider J_p and J_o seperately and build them column by column.

The i-th column of J_p is written as

$$J_p^i = z_{i-1} \times p_n^{i-1} \tag{2.14}$$

and the i-th column of J_o is written as

$$J_o^i = z_{i-1} \tag{2.15}$$

Here, z_{i-1} is the z-axis of frame $i-1$ and p_n^{i-1} is the position vector that points from the origin of the $i-1$ to the origin of the last frame and is expressed in the base frame.

The terms that are found from equations (2.14) and (2.15) are placed in the Jacobian matrix as in equation (2.16).

$$J = \begin{bmatrix} J_p^1 & \cdots & J_p^n \\ J_o^1 & \cdots & J_o^n \end{bmatrix} \tag{2.16}$$

This Jacobian matrix is referred as “geometrical” Jacobian to distinguish it from “analytical” Jacobian which is obtained by taking partial derivatives of forward geometrical model with respect to joint variables.

We may also want to know end-effector acceleration in terms of joint variables. This is known as “second order kinematic model” which can be easily obtained by taking derivative of both sides of equation (2.10).

$$\ddot{X}_e = \dot{J}(q, \dot{q})\dot{q} + J(q)\ddot{q} \tag{2.17}$$

2.2.2 Inverse kinematic model

Inverse kinematic model gives joint velocities $\dot{q}$ for a desired end-effector velocity $\dot{X}$.

If we solve equation (2.10) for $\dot{q}$, we obtain

$$\dot{q} = J^{-1}\dot{X} \tag{2.18}$$

The above equation has solution only if inverse of J exists. However, in singular configurations, the inverse of Jacobian matrix does not exist. Still, throughout the thesis, this analytical solution will be used and singular configurations will be avoided. Singular configurations can be found by solving $\det(J) = 0$.

In Appendix D, an easy way to calculate inverse of Jacobian matrix is shown. The method is known as inverse by partitioning and is an efficient way to calculate inverse of large matrices. It can also be used to find singular configurations of the manipulator by taking the inverse symbolically.

Second order inverse kinematic model can be found by solving equation (2.17) for $\ddot{q}$.

$$\ddot{q} = J^{-1}(\ddot{X} - \dot{J}\dot{q}) \tag{2.19}$$

2.3 Dynamic Model

In previous section, we studied kinematics of a manipulator which describes the motion without considering forces and moments acting on the manipulator. At this point, we can find position, velocity or acceleration of any point of interest on the manipulator with a given set of joint position, velocity and acceleration. In this section we will take a look at the dynamics of manipulators.

Dynamic model of a manipulator describes the relationship between forces acting on the manipulator and motion of the manipulator. Dynamic model plays an important role in design, simulation and control of manipulators.

Robot dynamics can be categorized into two problems: forward dynamics and inverse dynamics. In forward dynamics, joint torques/forces, are known and joint accelerations are calculated. Forward dynamic model is used for computer simulations.

$$\ddot{q} = f(q, \dot{q}, \tau) \tag{2.20}$$

Inverse dynamics concerns with finding necessary torques/forces that must be applied on joints to follow a desired trajectory. It is used in control algorithms.

$$\tau = g(q, \dot{q}, \ddot{q}) \tag{2.21}$$

There are two general types of methodology that are used to obtain dynamic model of manipulators: Euler-Lagrange and Newton-Euler formulations. In our study, we saw that the control algorithm that uses Euler-Lagrange dynamic model cannot finish its calculations in one cycle time. Euler-Lagrange dynamic model has more calculation burden than Newton-Euler formulation and this is a well-known problem about Lagrange-Euler formulation. On the other hand, Euler-Lagrange can be expressed in a closed form which enables us to analyze the properties of the dynamic model with respect to mechanical parameters. For these reasons, Euler-Lagrange formulation is used is simulation environment while Newton-Euler formulation is used in real experiments which is known to be more suitable for real-time implementation.

We will discuss these two methodology briefly here. Reader can refer to [12] and [13] for more detailed analysis of Euler-Lagrange and Newton-Euler dynamic model respectively.

2.3.1 Euler-Lagrange formulation

The “Lagrangian” of the system is defined as the difference between the kinetic and potential energy of the mechanical system.

$$L = K - U \tag{2.22}$$

Assuming no external forces and no friction, we can write Euler-Lagrange equation of motion as

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \tau_i \quad for\ i = 1,2,\dots,n \tag{2.23}$$

The order n of the system is determined by the number of “generalized coordinates” that are required to describe the evolution of the system. The n Denavit-Hartenberg joint variables can serve as a set of generalized coordinates for an n-link rigid manipulator [14].

The forces and moments acting on a typical link i of a serial manipulator is shown on Figure 2.2.

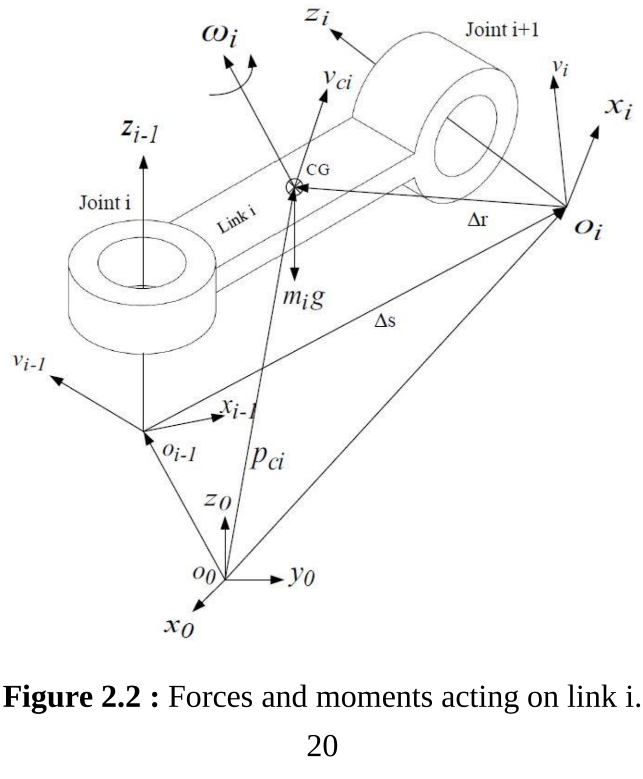


Figure 2.2 : Forces and moments acting on link i.

The kinetic energy of link i can be written in matrix form as

$$K_i = \frac{1}{2} m_i {v_{ci}}^T v_{ci} + \frac{1}{2} {\omega_i}^T I_i \omega_i \tag{2.24}$$

Here, v_{ci} is the linear velocity of the center of mass of link i and ω_i is the angular velocity of link i. They are expressed in the base frame. Linear and angular velocity of a link can be calculated with the same method as we used in section 2.2. The only difference is the point of which we want to calculate linear and angular velocity.

In equation (2.24), I_i is the inertia tensor of link i, taken about the center of mass CG. This tensor must also be expressed in the base frame. Let ${}^{i}I_i$ be the inertia tensor of link i, taken about the center of mass and expressed in the link frame i. I_i and ${}^{i}I_i$ is related as

$$I_i = {}^{0}R_i \; {}^{i}I_i \left({}^{0}R_i \right)^T \tag{2.25}$$

Equation (2.25) is a similarity transformation. Since frame i moves with the link i, the elements of ${}^{i}I_i$ are constant values. On the other hand, the elements of I_i are functions of joint variables.

Total kinetic energy of an n-link manipulator can be written as

$$K = \frac{1}{2} \sum_{i=1}^{n} \left(m_i {v_{ci}}^T v_{ci} + {\omega_i}^T I_i \omega_i \right) \tag{2.26}$$

In case of rigid dynamics, the only source of potential energy is gravity [14]. The potential energy of an n-link manipulator can be written as

$$U = -\sum_{i=1}^{n} m_i g^T p_{ci} \tag{2.27}$$

where p_{ci} is the position vector of the center of mass of link i and g is the gravitational acceleration vector. Both of them are expressed in the base frame.

Substituting equations (2.26) and (2.27) into (2.23) will give the closed form equation of motions in matrix form as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{2.28}$$

The first term in equation (2.28) accounts for the inertia of the manipulator. $M(q)$ is the $n \times n$ inertia matrix of the manipulator and depends on the joint positions. The second term represents Coriolis and centrifugal forces and is called as “velocity coupling vector”. In this vector, terms involving ${\dot{q}_i}^2$ represent centrifugal effect and terms involving $q_i q_j$ where $i \neq j$ represent Coriolis effect. The last is the compensation of gravitational forces.

2.3.2 Newton-Euler formulation

Newton-Euler formulation is an efficient solution for inverse dynamics problem which is based on equation (2.29), Newton’s second law of motion and equation (2.30), Euler’s second law of motion.

$$F = m_i \dot{v}_{ci} \tag{2.29}$$

$$N = I_i \dot{\omega}_i + \omega_i \times (I_i \omega_i) \tag{2.30}$$

In this recursive formulation, serial structure of the manipulator is exploited. Equations of motion for each link are obtained by two recursions: forward recursion and backward recursion. In forward recursion, linear and angular velocities of each link are calculated with respect to given joint positions and velocities starting from the base to end-effector. In backward recursion, forces and moments are obtained with respect to linear and angular velocity of each link that are calculated from the forward recursion. For revolute joints forward recursion is performed with the following equations.

$$\omega_i = \omega_{i-1} + \dot{q}_i z_{i-1} \tag{2.31}$$

$$\dot{\omega}_i = \dot{\omega}_{i-1} + \ddot{q}_i z_{i-1} + \omega_{i-1} \times (\dot{q}_i z_{i-1}) \tag{2.32}$$

$$v_i = v_{i-1} + \omega_i \times \Delta s_i \tag{2.33}$$

$$\dot{v}_i = \dot{v}_{i-1} + \dot{\omega}_i \times \Delta s_i + \omega_i \times (\omega_i \times \Delta s_i) \tag{2.34}$$

and for $i = 0$

$$\omega_0 = 0 \tag{2.35}$$

$$\dot{\omega}_0 = 0 \tag{2.36}$$

$$v_0 = 0 \tag{2.37}$$

$$\dot{v}_0 = -g \tag{2.38}$$

By equation (2.38), we added the effects of gravity. This effect is propagated through the arm while performing forward recursion.

Once velocities and accelerations are obtained, we begin our backward recursion, starting from the end-effector frame and moving backward to the base. This is performed by writing force and moment balances as in the following equations.

$$f_i = f_{i+1} + m_i[\dot{v}_i + \dot{\omega}_i \times \Delta r_i + \omega_i \times (\omega_i \times \Delta r_i)] \tag{2.39}$$

$$n_i = n_{i+1} + (\Delta s_i + \Delta r_i) \times f_i - \Delta r_i \times f_{i+1} + I_i\dot{\omega}_i + \omega_i \times (I_i\omega_i) \tag{2.40}$$

In equation (2.39), the term in square brackets represents linear acceleration of the center of mass of link i. The inertia tensor in equation (2.40) is calculated as in equation (2.25). For revolute joints, moment n_i will cause a torque about axis z_{i-1}.

$$\tau_i = (n_i)^T z_{i-1} \tag{2.41}$$

Since we start with the end-effector frame, for $i = n = 6$, f_7 and n_7 is environmental force and moment respectively that are acting on the end-effector (which are equal to zero if manipulator is moving freely and carrying no load).

$$f_{i+1} = f_7 = -f^{environment} \tag{2.42}$$

$$n_{i+1} = n_7 = -n^{environment} \tag{2.43}$$

2.4 Interaction Forces

When the end-effector of a manipulator makes contact with the environment, forces and moments are generated at the interaction point (from now on “force” will be used to refer both force and moment). These forces are then transferred to joints as "interaction torques". By using Jacobian matrix of the manipulator, we can calculate joint torques as

$$\tau = J^T F_e \tag{2.44}$$

Here $F_e = \left[F_x, F_y, F_z, M_x, M_y, M_z\right]^T$ is the "force wrench" that is applied by the manipulator on the environment and is expressed in the base frame. By adding interaction torques, equation (2.28) is modified as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + J^T F_e = \tau \tag{2.45}$$

2.5 Identification of Unknown Parameters

In previous sections, dynamic model and interaction torque of an n-link manipulator are obtained. There are two terms that must be added to equation (2.45) in order to have a complete model of the manipulator. These terms are spring and friction models. In past, the author of [15] has made a detailed identification on RX-160 and reported the results. In this thesis, those results are used to complete the dynamic model.

2.5.1 Spring model

In RX-160, a spring is attached between second and third joints in order to support the second joint that is carrying the second link and links that come after the second link.

The torque, applied only on the second joint by the spring is a function of only second joint's position. This torque is equal to

$$\tau_s = -46.56{q_2}^3 + 8.07{q_2}^2 + 471.09q_2 + 66.29 \tag{2.46}$$

Spring model, measured spring data and percent error is shown in Figure 2.3.

2.5.2 Friction model

Joint frictions has to be considered in order to have a complete model of the manipulator. If frictions are not compensated, it may cause significant tracking errors. Friction arises in the bearings, transmissions and seals where two surfaces are in relative motion and in contact [16].

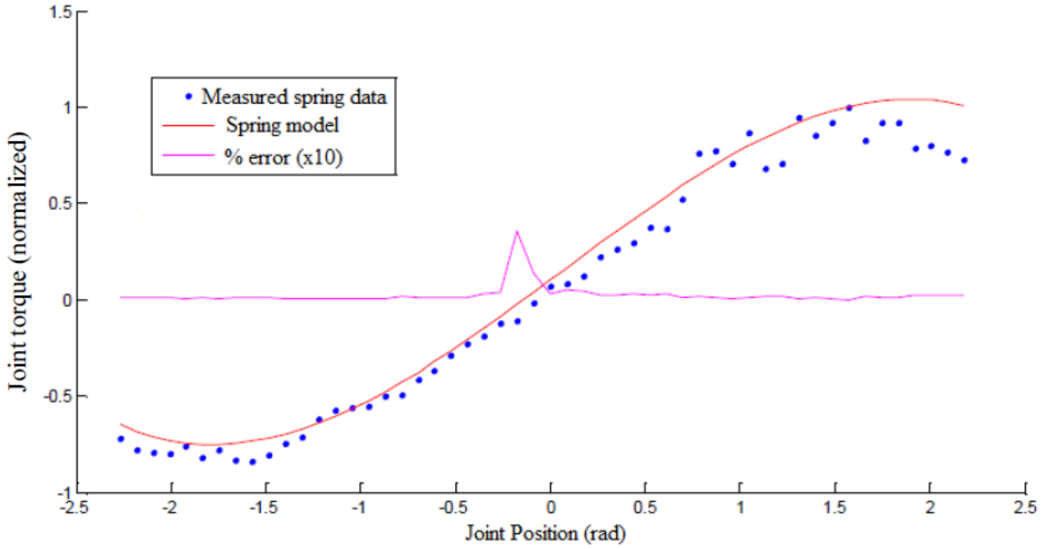


Figure 2.3 : Spring model, real data and percent error [15].

Generally, three kinds of frictions are considered in the analysis of joint frictions of manipulators: Coulomb friction, static friction and viscous friction [17].

Coulomb friction assumes a constant friction component that is independent of the magnitude of the velocity. The static friction is the torque necessary to initiate motion from rest. It is often greater than the Coulomb friction. The viscous friction is generally represented as being proportional to the velocity, but in low velocities, a phenomena called as “Stribeck effect” arises. Among researchers, most accepted friction model is given by the equation (2.47).

$$\tau_f = sign(\dot{q})\left[\tau_{f,C}\left(\tau_{f,s} - \tau_{f,C}\right)e^{-\left|\dot{q}/\dot{q}^{(s)}\right|^{\delta^{(s)}}}\right] + c^{(v)}\dot{q} \qquad (2.47)$$

Here, τ_f is the friction torque, $\tau_{f,C}$ is Coulomb friction, $\tau_{f,C}$ is static friction, $c^{(v)}$ is viscous friction coefficient and $\dot{q}^{(s)}$ is Stribeck velocity. The term $\delta^{(s)}$ is changing between 0.5 and 1 depending on various materials. Friction model described by equation (2.47) is shown in Figure 2.4.

We complete our dynamic model by adding spring torque and joint frictions to equation (2.45) to obtain our final dynamic model as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau_s(q) + \tau_f(\dot{q}) + J^T F_e = \tau \qquad (2.48)$$

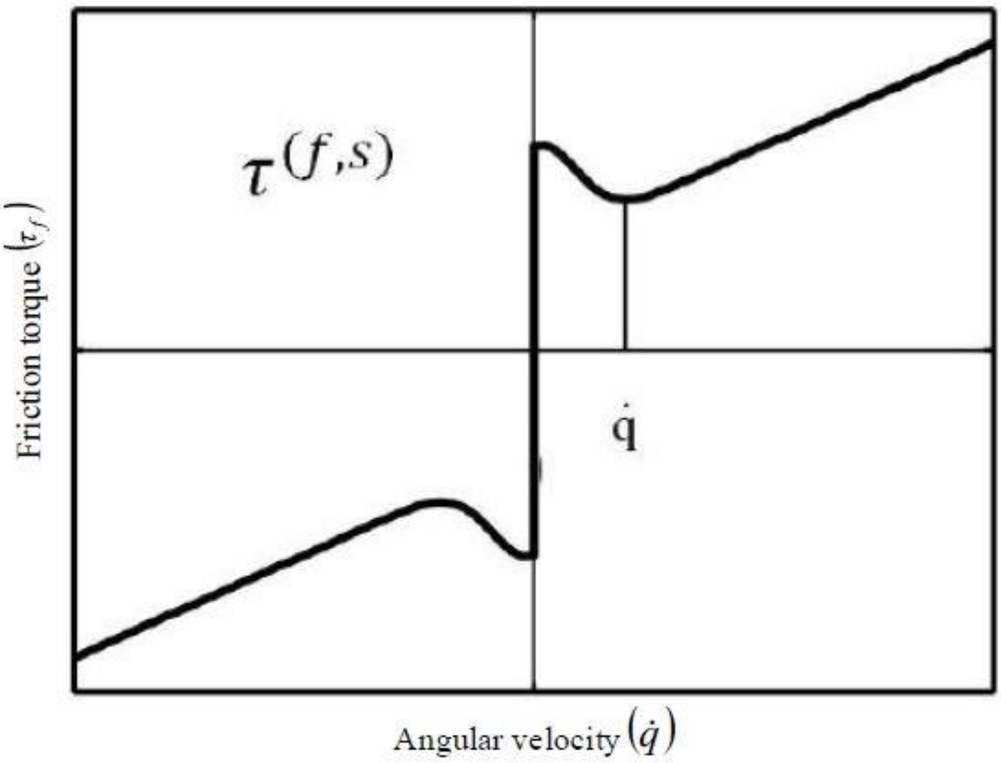


Figure 2.4 : Joint friction with Stribeck effect [15].

3. CONTROL SYSTEM DESIGN AND SIMULATIONS

The control problem of a manipulator can be described as determining joint forces and torques with respect to a commanded task. We can talk about two different kinds of tasks: motion in free space and compliant motion. We may want the manipulator to follow a desired trajectory in free space, meaning there won't be any significant interaction between the robot and the environment. Pick and place, welding and painting operations can be given as examples for such tasks.

Second kind of task is that we may want the manipulator to interact with the environment with some degree of compliance, meaning forces that arise from interaction should be controlled while following a desired trajectory. Grinding, deburring, polishing, assembling are this kind of task.

In this chapter, we will try to apply different kinds of control schemes on the robot in MATLAB®'s Simulink® environment. First we will briefly explain motion control in task space. We will consider only one control scheme for motion control since it is not the main topic of this thesis. Then we will explain different control schemes for compliant motion control and give simulation results.

3.1 Motion Control

In previous chapter, we derived the dynamic model of the robot which has the general form as in equation (2.48). The last term in the left-hand side of this equation drops out in free motion control since the motion is performed without interaction with the environment.

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau_s(q) + \tau_f(\dot{q}) = \tau \tag{3.1}$$

The control problem is computing joint torques for trajectory tracking or point-to-point motion. In this thesis, only trajectory tracking problem is considered.

3.1.1 Trajectory generation

We have to obtain reference inputs to motion control system to execute a specific manipulator task. This input is the "trajectory" which is a "path" and a "time history" along the path [17]. Trajectory generation consists of obtaining a time sequence of the values attained by an interpolating function (typically a polynomial) of the desired trajectory [14]. In this section, we will discuss point-to-point trajectories.

3.1.1.1 Trapezoidal velocity profile

In this trajectory, motion is separated into three pieces with different velocity profiles. First the velocity increases linearly to a maximum value followed by a constant velocity at this maximum value. In the last part, velocity decreases linearly to zero at the end of the motion (Figure 3.1).

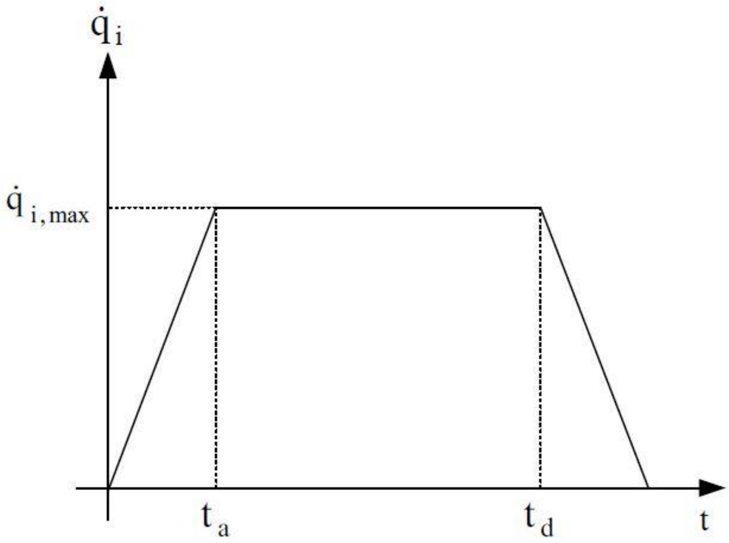


Figure 3.1 : Trapezoidal velocity profile.

Let $q_i^{initial}$, q_i^{final} and D_i be the initial position, final position and distance traveled for joint i respectively. The position of joint i is represented by a piecewise function as in equation (3.2). Position, velocity and acceleration with respect to time is plotted in Figure 3.2. Acceleration, deceleration and total motion time (t_a, t_d, t_f respectively) can be calculated with respect to maximum joint velocity and accelerations. This will ensure to obtain minimum time to travel. It is not considered here however in [17], it is shown how to calculate minimum motion time and synchronize all joints to this motion time.

$$q_i(t) = \begin{cases} q_i^{initial} + \dfrac{D_i t^2}{2t_a t_d} & for\ 0 \le t < t_a \\ q_i^{initial} - \dfrac{D(t_a - 2t)}{2t_d} & for\ t_a \le t < t_d \\ q_i^{final} - \dfrac{D\left(t - t_f\right)^2}{2t_a t_d} & for\ t_d \le t \le t_f \end{cases} \tag{3.2}$$

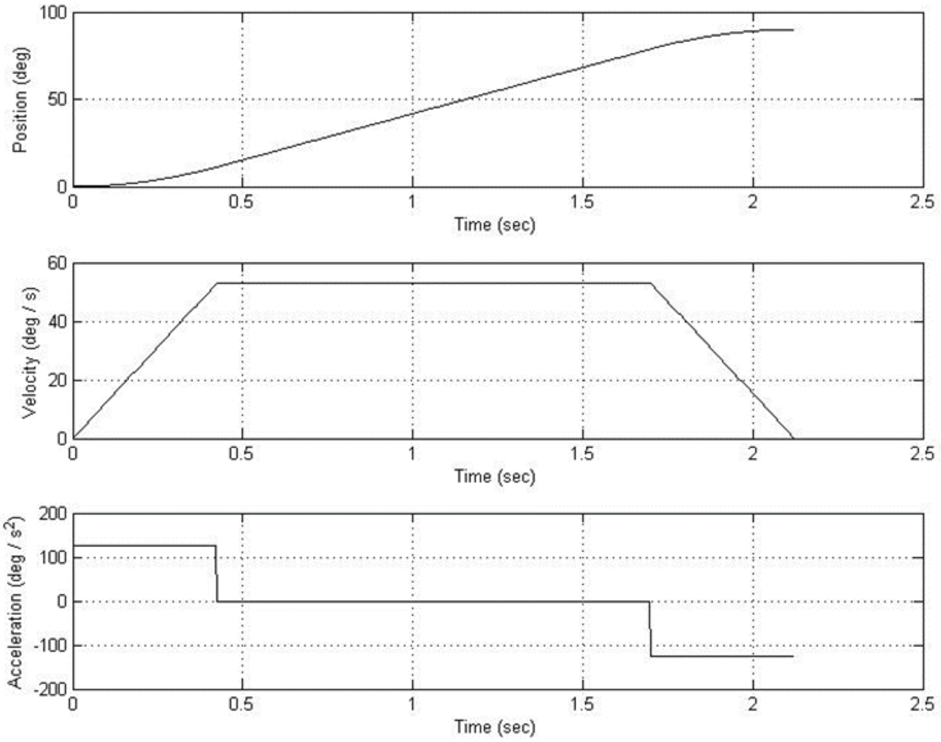


Figure 3.2 : Position, velocity and acceleration profiles for trapezoidal velocity.

3.1.1.2 Quintic polynomial position profile

When motion in high speed is desired or when a robot is handling heavy or delicate objects, it is worth ensuring the continuity of accelerations, in order to avoid exiting resonances in the mechanics [17]. Unlike linear or cubic polynomial interpolation or trapezoidal velocity profile, quintic polynomial interpolation ensures continuous acceleration.

Trajectory between $q_i^{initial}$ and q_i^{final} can be written as

$$q_i(t) = q_i^{initial} + r(t)D_i \tag{3.3}$$

Function $r(t)$ is a fifth order polynomial written as

$$r(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{3.4}$$

The coefficients of the polynomial $r(t)$ is found by the following boundary conditions.

$$\begin{gathered} r(0) = 0 \\ r(t_f) = 1 \\ \dot{r}(0) = 0 \\ \dot{r}(t_f) = 0 \\ \ddot{r}(0) = 0 \\ \ddot{r}(t_f) = 0 \end{gathered} \tag{3.5}$$

where t_f is total motion time. If we solve $r(t)$ for the boundary conditions in equation (3.5), we obtain the following interpolation function.

$$r(t) = 10\left(\frac{t}{t_f}\right)^3 - 15\left(\frac{t}{t_f}\right)^4 + 6\left(\frac{t}{t_f}\right)^5 \tag{3.6}$$

Position, velocity and acceleration for this trajectory are plotted with respect time in Figure 3.3.

3.1.1.3 Trajectory generation in task space

In previous section, we showed trajectory generation between initial and final joint positions with free path. However, the input to our control algorithms will be positions in task-space, not in joint-space. Trajectory generation in task-space is similar to trajectory generation in joint-space.

Recalling from section 2.1.1, initial and final positions of the end-effector can be represented by homogeneous transformation matrices. Now we have initial and final position matrices as $X^{initial}$ and X^{final}. In order to generate a trajectory between these two points, we will analyze translation and rotation seperately.

Let ${}^{0}T_n^{i}$ and ${}^{0}T_n^{f}$ be the homogeneous transformation matrices of initial and final position of the end-effector.

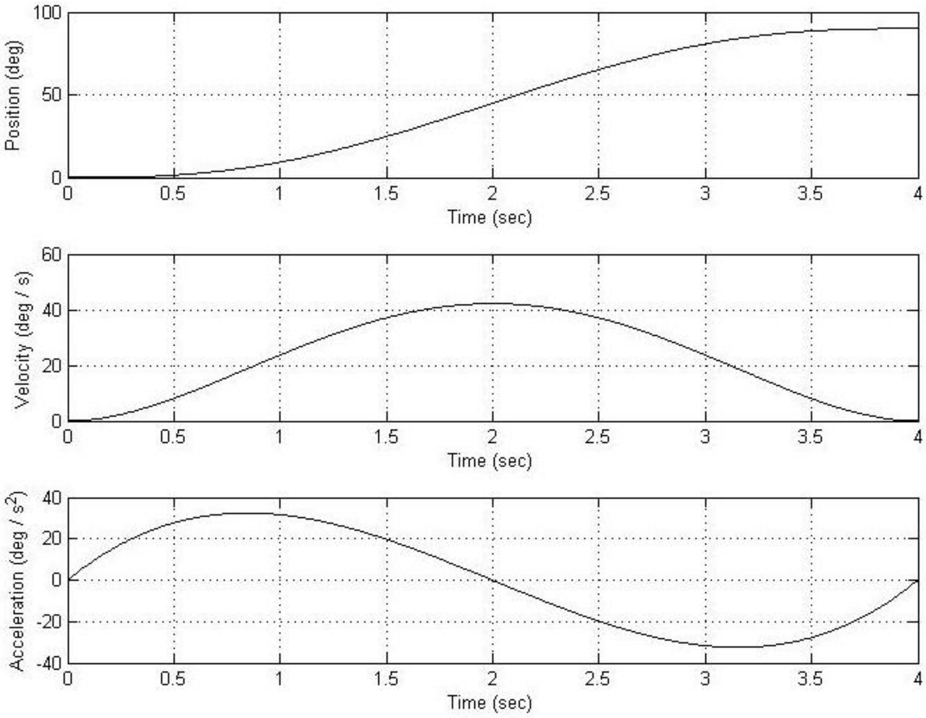


Figure 3.3 : Quintic polynomial position profile.

$$ {}^0T_n^i = \begin{bmatrix} {}^0R_n^i & {}^0p_n^i \\ 0 & 1 \end{bmatrix} \tag{3.7}$$

$$ {}^0T_n^f = \begin{bmatrix} {}^0R_n^f & {}^0p_n^f \\ 0 & 1 \end{bmatrix} \tag{3.8}$$

The end-effector should perform a translation between the origins of ${}^0T_n^i$ and ${}^0T_n^f$. The distance to travel is written as

$$ D = {}^0p_n^f - {}^0p_n^i \tag{3.9}$$

Trajectory of the end-effector for translational part can be written as

$$ p(t) = {}^0p_n^i + r(t)D \tag{3.10}$$

Trajectory for the orientational part of the motion is harder to define. In translation, we have three DOF of motion and we can use three Cartesian coordinates to represent it. In orientation however, we again have three DOF of motion but the representation of orientation with homogeneous transformation matrix have nine parameters which is redundant.

In literature, most common ways to represent orientation with less parameters are Euler angles, angle-axis, and quaternions. In this thesis, we will use angle-axis representation.

In angle axis representation, an angle θ and a unit vector $\boldsymbol{u}$ can represent orientation of a frame with respect to another. Rotation matrix and angle-axis representation is related as

$$\begin{aligned} &rot(\boldsymbol{u}, \alpha) \\ &= \begin{bmatrix} {u_x}^2(1 - c\alpha) + c\alpha & u_x u_y(1 - c\alpha) - u_z s\alpha & u_x u_z(1 - c\alpha) + u_y s\alpha \\ u_y u_x(1 - c\alpha) + u_z s\alpha & {u_y}^2(1 - c\alpha) + c\alpha & u_y u_z(1 - c\alpha) - u_x s\alpha \\ u_z u_x(1 - c\alpha) - u_y s\alpha & u_z u_x(1 - c\alpha) + u_x s\alpha & {u_z}^2(1 - c\alpha) + c\alpha \end{bmatrix} \end{aligned} \tag{3.11}$$

where $\boldsymbol{u}$ is expressed in the base frame.

For the rotation part of the motion we can write

$$rot(\boldsymbol{u}, \alpha)\ {}^{0}R_n^i = {}^{0}R_n^f \tag{3.12}$$

Since $\left({}^{0}R_n^i\right)^{-1} = \left({}^{0}R_n^i\right)^T$, equation (3.12) leads to

$$rot(\boldsymbol{u}, \alpha) = {}^{0}R_n^f\left({}^{0}R_n^i\right)^T \tag{3.13}$$

Trajectory of the end-effector for rotation can be written as

$$R(t) = rot(\boldsymbol{u}, r(t)\alpha)\ {}^{0}R_n^i \tag{3.14}$$

3.1.2 Task-space scheme

The control method known as “computed torque control” uses inverse dynamic model of the robot. Unlike independent joint control, this method takes coupling effects between joints into account. It also linearizes and decouples the dynamic model of the manipulator.

In [17], this control scheme is shown both for joint-space and task-space. We will only use task-space scheme which is also known as “resolved acceleration control”.

In section 2.3, for free motion, the dynamic model is obtained and written as

$$\tau = M\ddot{q} + H \tag{3.15}$$

where

$$H = C(q,\dot{q}) + G(q) + \tau_s(q) + \tau_f(\dot{q}) \tag{3.16}$$

By using second order kinematic model, equation (3.15) can be rewritten as

$$\tau = MJ^{-1}\left(\ddot{X} - \dot{J}\dot{q}\right) + H \tag{3.17}$$

Let $\widehat{M}$ and $\widehat{H}$ be the estimated terms of $M(q)$ and $H(q,\dot{q})$. Assuming noiseless position and velocity measurements, we can define control law as

$$\tau = \widehat{M}J^{-1}\left(w - \dot{J}\dot{q}\right) + \widehat{H} \tag{3.18}$$

where

$$w = \ddot{X}^d + K_d\left(\dot{X}^d - \dot{X}\right) + K_p\left(X^d - X\right) \tag{3.19}$$

In the above equation, X^d is desired position and X is the feedback position. The corresponding block diagram is shown in Figure 3.4.

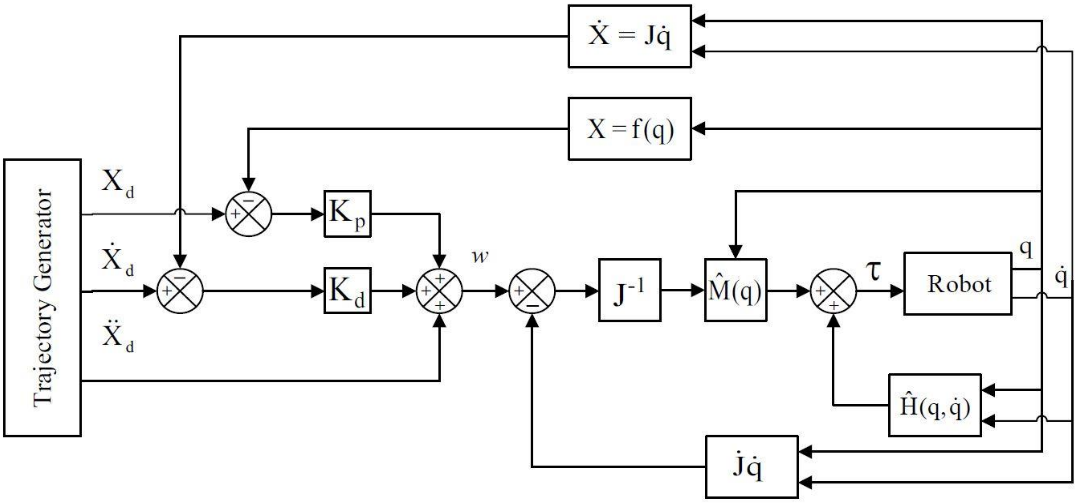


Figure 3.4 : Computed torque control in task-space.

3.1.2.1 Calculation of position error in task-space

Right now, we are calculating desired and feedback task-space positions with homogeneous transformation matrices. However, in order to use position error in equation (3.19), we need a vector with six elements. So we need to extract a 6 × 1 error vector from homogeneous transformation matrices. For this purpose, the method in [17] is used.

Let ${}^{0}T_{n}^{des}$ and ${}^{0}T_{n}^{fbk}$ be desired and feedback task-space position matrix respectively.

$$ {}^{0}T_{n}^{des} = \begin{bmatrix} {}^{0}R_{n}^{d} & {}^{0}p_{n}^{d} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^{0}s_{n}^{d} & {}^{0}n_{n}^{d} & {}^{0}a_{n}^{d} & {}^{0}p_{n}^{d} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.20} $$

$$ {}^{0}T_{n}^{fbk} = \begin{bmatrix} {}^{0}R_{n}^{f} & {}^{0}p_{n}^{f} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^{0}s_{n}^{f} & {}^{0}n_{n}^{f} & {}^{0}a_{n}^{f} & {}^{0}p_{n}^{f} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.21} $$

Translational position and orientation error can be separated as

$$ e = \begin{bmatrix} e_p \\ e_o \end{bmatrix} \tag{3.22} $$

The translational position error is obtained as

$$ e_p = {}^{0}p_{n}^{f} - {}^{0}p_{n}^{d} \tag{3.23} $$

and the orientation error is obtained with

$$ e_o = \boldsymbol{u}\alpha \tag{3.24} $$

where $\boldsymbol{u}$ and α are obtained by solving $rot(\boldsymbol{u}, \alpha)\ {}^{0}R_{n}^{f} = {}^{0}R_{n}^{d}$.

3.1.2.2 Simulation results

The end-effector is commanded to follow a circle in XY-plane with a radius of $150\ mm$ while maintaining its orientation in 20 seconds. The results are shown in Figure 3.5 to Figure 3.8.

As it can be seen from figures, the end-effector can follow the desired trajectory perfectly. The reason for this is, the same robot model is used in both forward dynamics and in control algorithm. So this means, the robot is modeled perfectly which is not the case in reality. Some error should be introduced to the robot model that is used in the control structure for more realistic results.

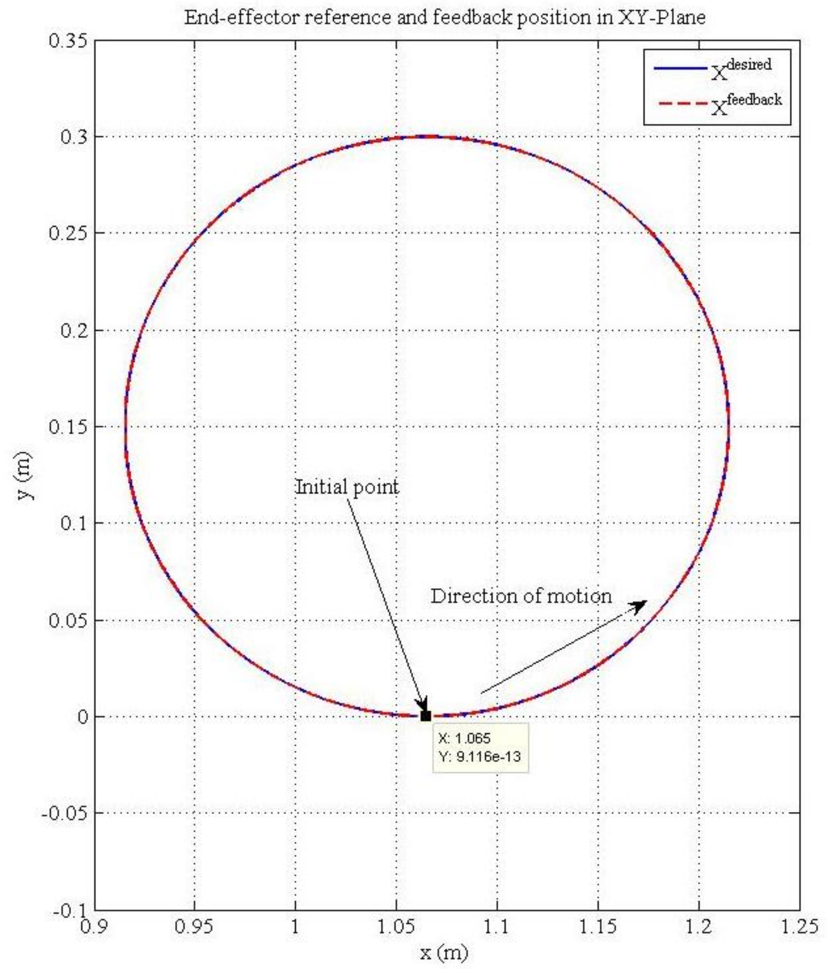


Figure 3.5 : End-effector desired and feedback path in motion control.

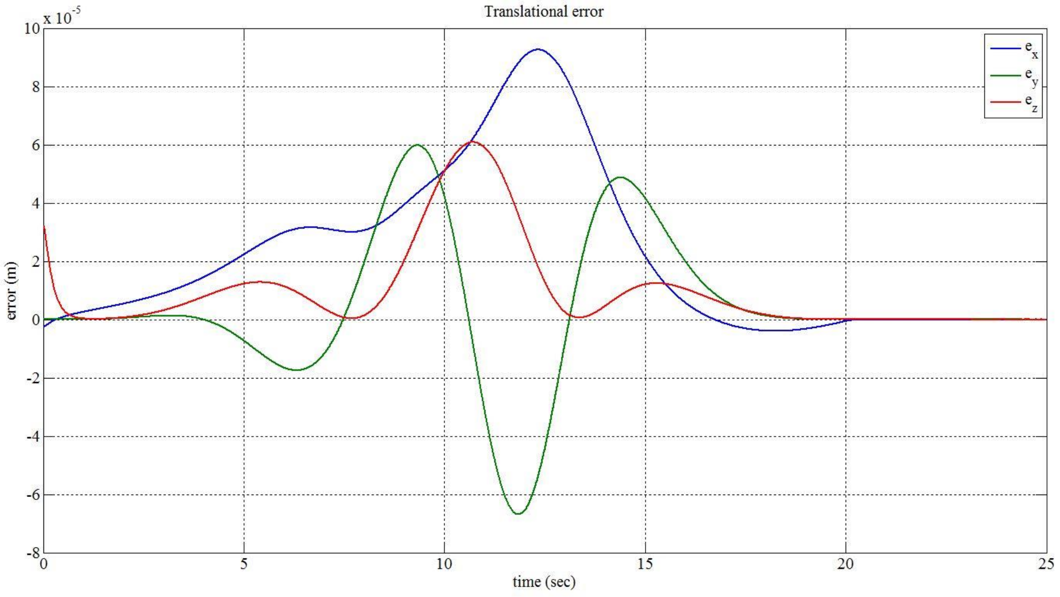


Figure 3.6 : Translational errors in motion control.

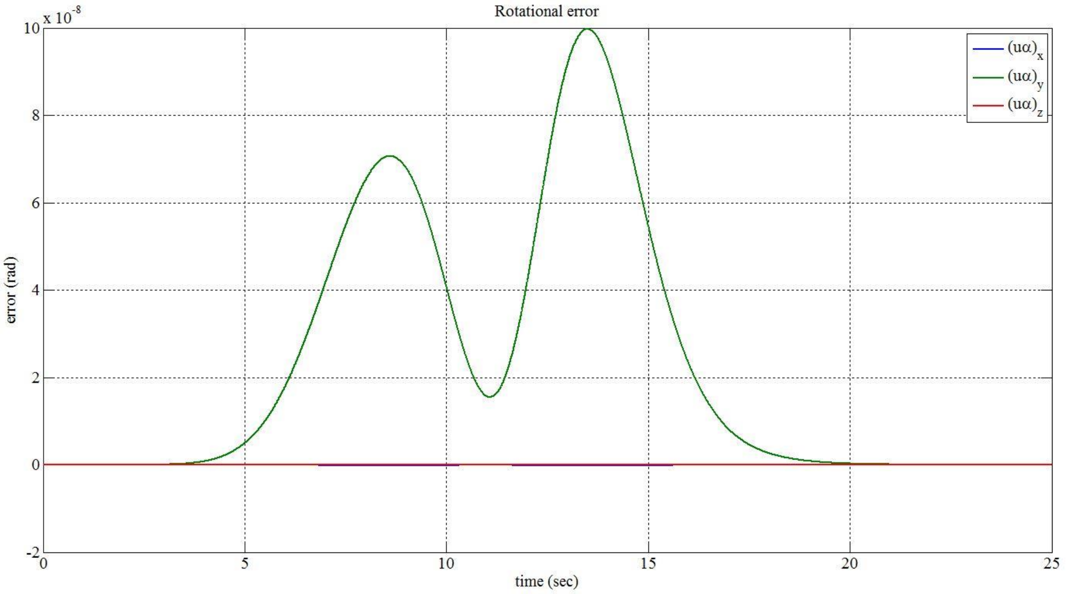


Figure 3.7 : Rotational errors in motion control.

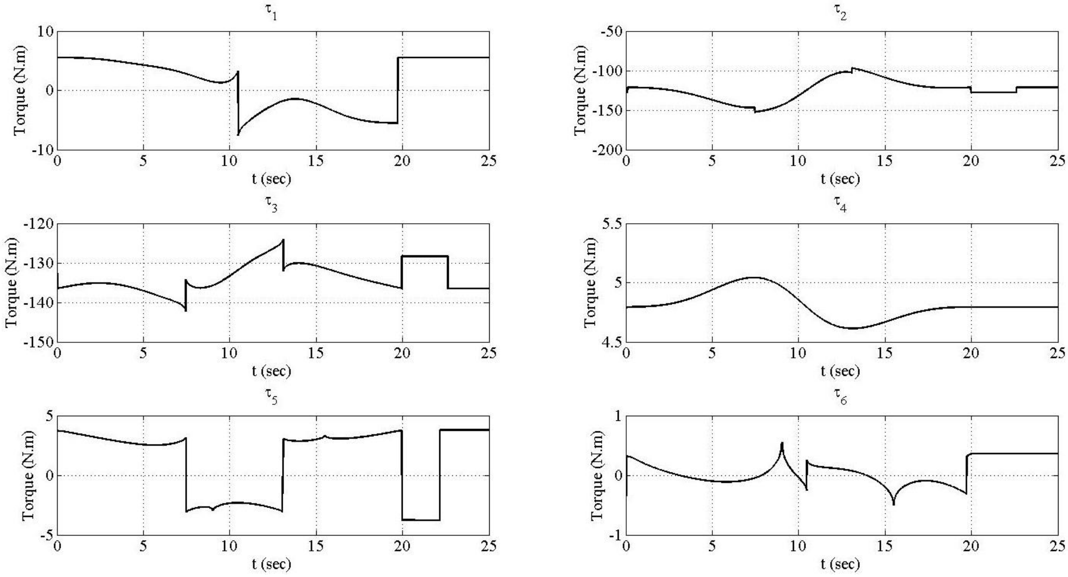


Figure 3.8 : Calculated joint torques in motion control.

3.2 Compliant Motion Control

In certain industrial applications, the manipulator must interact with its environment. Unlike in free motion, the manipulator is constrained by the environment and interaction forces arise where the manipulator is in contact with the environment. In order to control these interaction forces with purely position-based systems, a precise model of the manipulator and knowledge of the exact location and stiffness of the environment are required [17]. Another approach is to control these contact forces directly or indirectly. In this section, we will discuss two widely used manipulator force control algorithms and give simulation results.

3.2.1 Impedance control

Mechanical impedance relates forces with velocities acting on a mechanical system. The relation between applied force and the end-effector velocity can be written as

$$F(s) = Z(s)\dot{X}(s) \tag{3.25}$$

or in terms of position

$$F(s) = sZ(s)X(s) \tag{3.26}$$

The objective of impedance control is to make the end-effector behave like a desired mechanical system along each direction of the task-space and achieve a desired impedance. If we chose desired impedance as mass-spring-damper system, the transfer function can be written as

$$\frac{F(s)}{X(s)} = sZ(s) = M_d s^2 + B_d s + K_d \tag{3.27}$$

where M_d, B_d and K_d represent the desired inertia, damping and stiffness matrix respectively. The values of these matrices can be chosen to obtain a desired performance by the following statements [17]:

- "High values are given to M_d in the directions where contact is exptected in order to limit the dynamics of the robot."
- "High values are given to B_d in the directions where it is necessary to dissipate the kinetic energy and therefore damp the response."
- "The stiffness K_d effects the accuracy of the position control: along the force control directions, the stiffness should be small enough to limit the contact forces. On the other hand, along a position controlled direction, the user should set high stiffness to obtain accurate positioning of the end-effector."

The dynamic model of a manipulator that is in contact with the environment is found to be

$$\tau = M(q)\ddot{q} + H(q,\dot{q}) + J^T F_e \tag{3.28}$$

Desired behavior of the end-effector in equation (3.27) can be written as

$$F_e = M_d\left(\ddot{X}^d - \ddot{X}\right) + B_d\left(\dot{X}^d - \dot{X}\right) + K_d(X^d - X) \tag{3.29}$$

If we solve equation (3.29) for $\ddot{X}$ we obtain

$$\ddot{X} = \ddot{X}^d + {M_d}^{-1}\left[B_d\left(\dot{X}^d - \dot{X}\right) + K_d(X^d - X) - F_e\right] \tag{3.30}$$

where $\ddot{X}^d$ is the desired trajectory. Equation (3.30) ensures that the manipulator will behave as described by equation (3.29) while following a desired trajectory.

Next, we will consider resolved acceleration control in equation (3.18) in which $w(t)$ is replaced by the equation (3.30) and the wrench exerted by the manipulator on the environment $J^T F_e$ is taken into account.

$$\begin{aligned}\tau = \widehat{M}J^{-1}\{\ddot{X}^d + {M_d}^{-1}\left[B_d\left(\dot{X}^d - \dot{X}\right) + K_d(X^d - X) - F_e\right] \\ - \dot{J}\dot{q}\} + \widehat{H} + J^T F_e\end{aligned} \tag{3.31}$$

The block diagram of the above equation is shown in Figure 3.9.

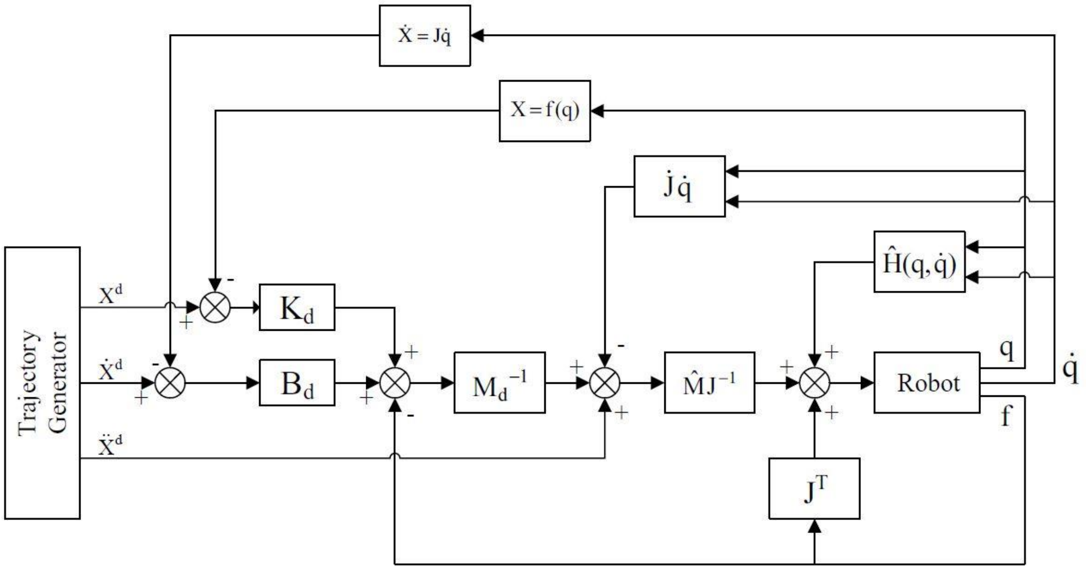


Figure 3.9 : Nonlinear decoupling impedance control.

The behavior of the impedance control is simulated by constraining the motion of the manipulator in only z-direction with a flat surface with a stiffness of $4000\ N/m$. The end-effector is commanded to follow a circular path in XZ-plane with no information about the constraint in its path. Desired inertia, damping and stiffness are set to be $M_d = diag(5, 5, 11.5, 2.5, 2.5, 2.5)$, $B_d = diag(125, 125, 70, 125, 125, 125)$, $K_d = diag(350, 350, 200, 500, 500, 500)$. The results are as follows.

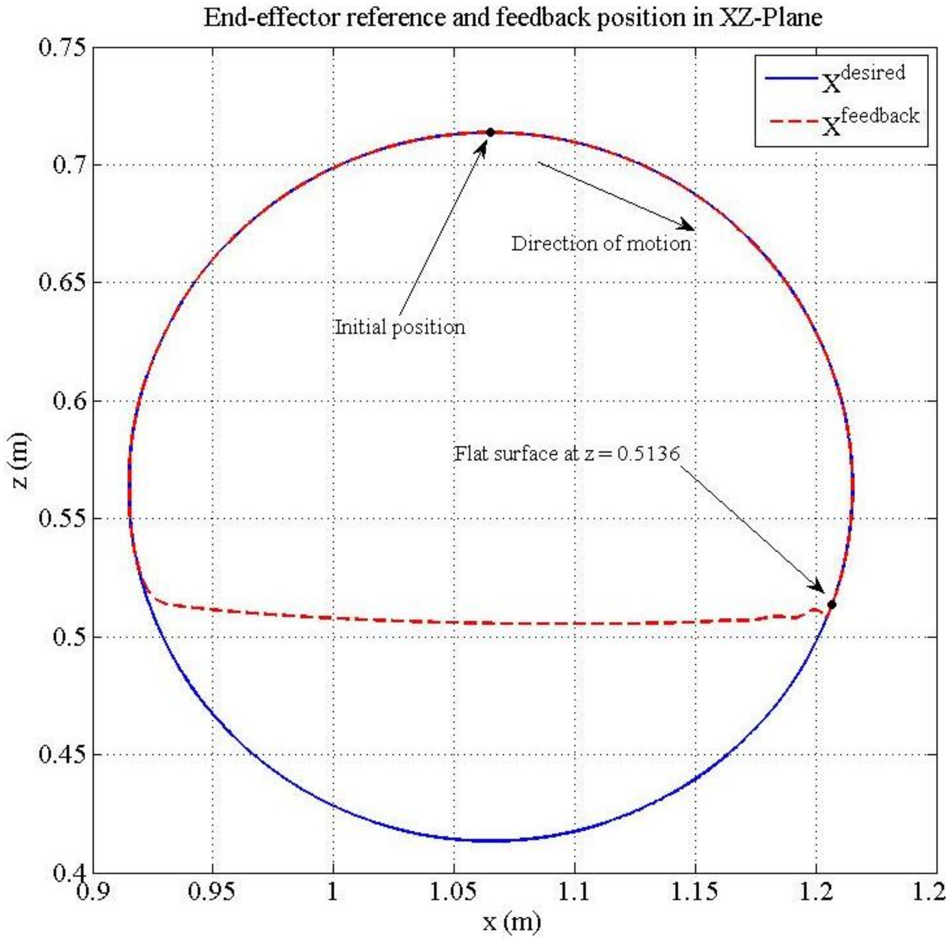


Figure 3.10 : Desired and feedback path in impedance control.

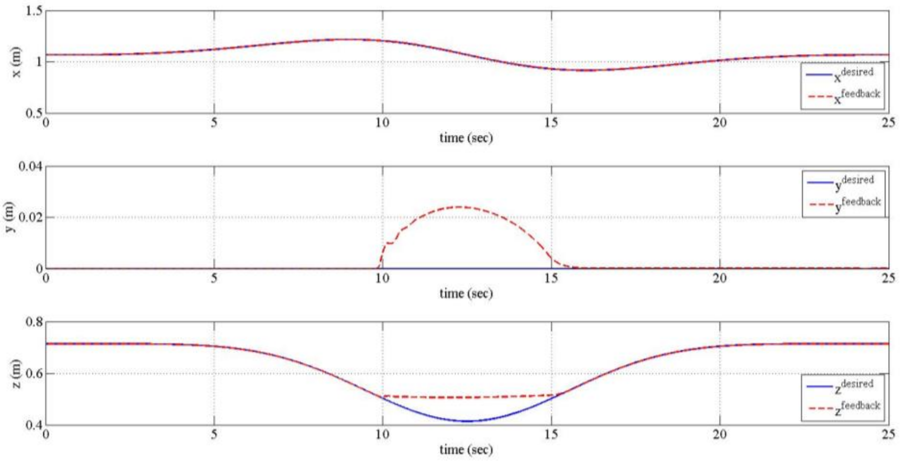


Figure 3.11 : Desired and feedback trajectories in impedance control.

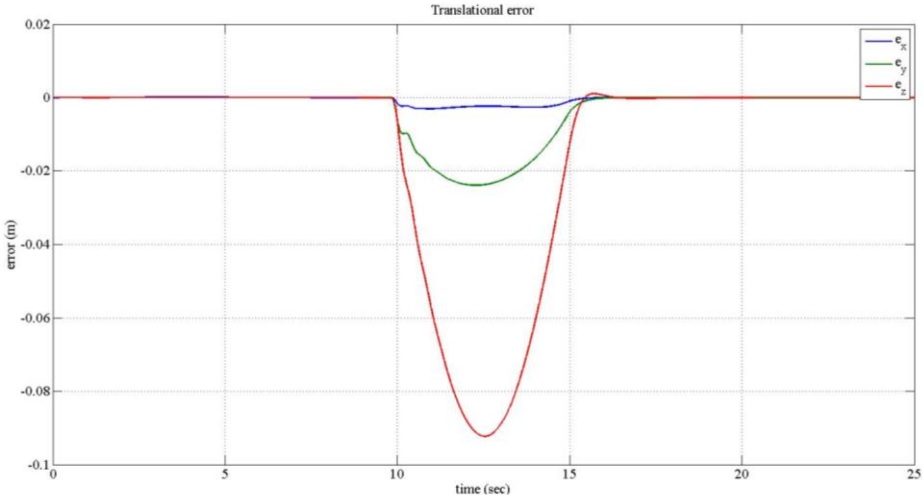


Figure 3.12 : Translational errors in impedance control.

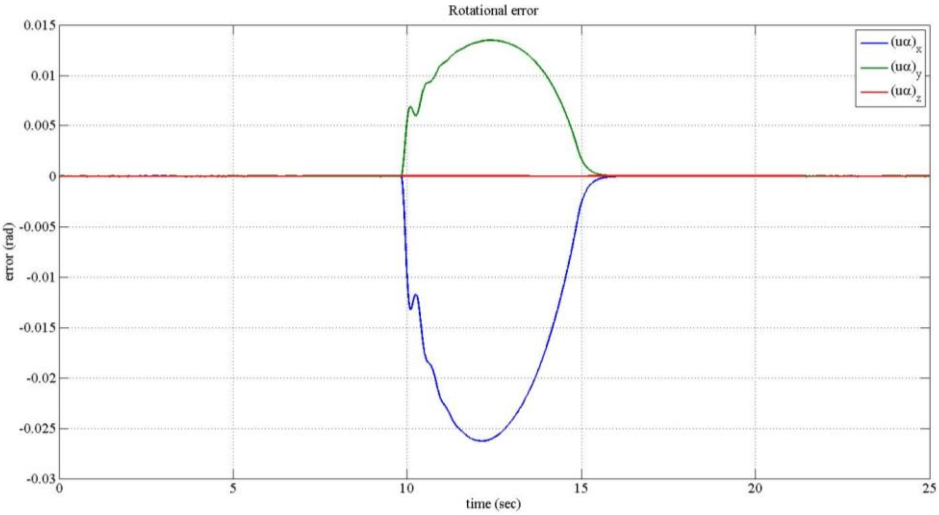


Figure 3.13 : Rotational errors in impedance control.

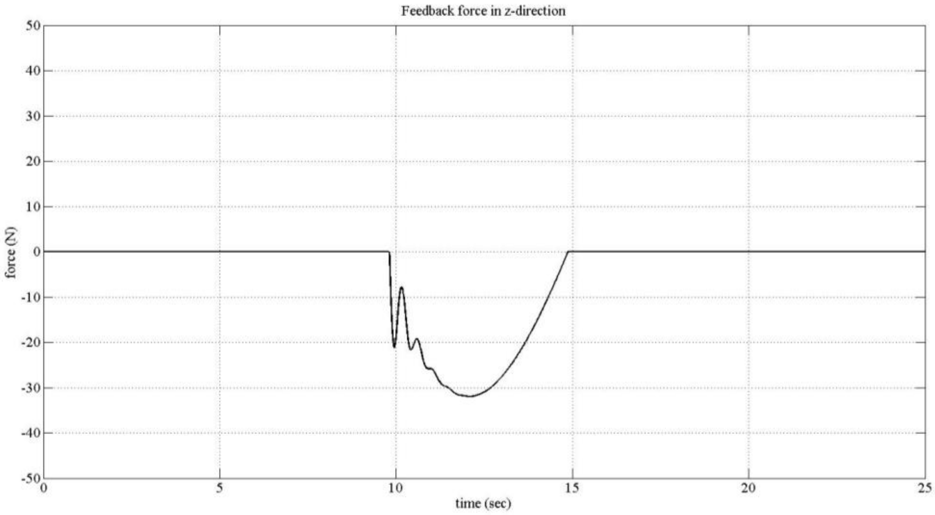


Figure 3.14 : Force applied by the end-effector in impedance control.

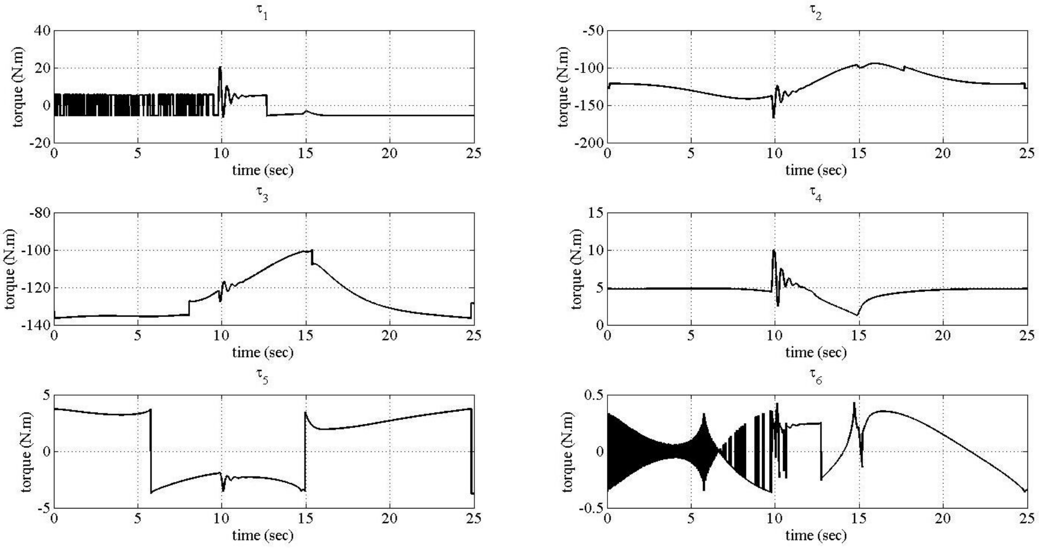


Figure 3.15 : Calculated joint torques in impedance control.

3.2.2 Hybrid force/position control

Using impedance control, we can specify a desired dynamic behavior of the manipulator, however we cannot specify a force value to be tracked. In impedance control force is controlled indirectly. The aim of hybrid force/position control is to split up simultaneous control of both end-effector motion and contact forces into separate decoupled subproblems [18].

The advantage of hybrid control is that the position and force information are analyzed independently to take advantage of well-known control techniques for each and are combined only at the final stage when both have been converted to joint torques [6].

This method was first proposed by Craig and Raibert [4] and the block diagram of the original control scheme is shown in Figure 3.16.

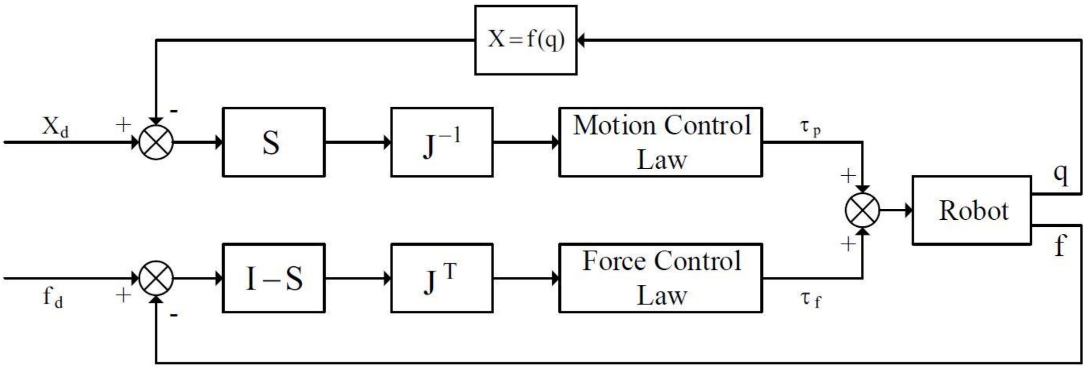


Figure 3.16 : Hybrid force/position control.

Each degree of freedom of the task-space is controlled by either position or force loop through the use of a *compliance selection matrix* which is denoted as S and diagonal such that

$$S = diag(s_1, \dots, s_6) \tag{3.32}$$

where $s_i = 1$ if i-th degree of freedom in task-space is position controlled or $s_i = 0$ if it is force controlled. In this control scheme, resolved acceleration control can be used for the motion control loop and PID can be used for force control loop. The total control law can be written as

$$\tau = \tau_p + \tau_f + \widehat{H} \tag{3.33}$$

where

$$\tau_p = \widehat{M} J^{-1} S\left[\ddot{X}^d + K_p(X^d - X) + K_d\left(\dot{X}^d - \dot{X}\right) - \dot{J}\dot{q}\right] \tag{3.34}$$

$$\tau_f = J^T(I - S)\left[f^d + K_{fp}\left(f^d - f\right) - K_{fd}\dot{X} + K_{fI}\int_{t_0}^{t}\left(f^d - f\right)dt\right] \tag{3.35}$$

Note that that, due to the noise of force sensors, task-space velocity is used instead of derivative of force error to damp the response in the force control loop. The corresponding block diagram is shown in Figure 3.17.

Simulations are performed with the same environment model in the impedance control. Manipulator end-effector starts its motion while in contact with the environments. First, manipulator is given 10 seconds to settle to a constant force of -30 N. Then, by keeping the force constant, the end-effector is commanded to move in a circular path with a radius of 150 mm in XY-plane. Results are shown in Figure 3.18 to Figure 3.23.

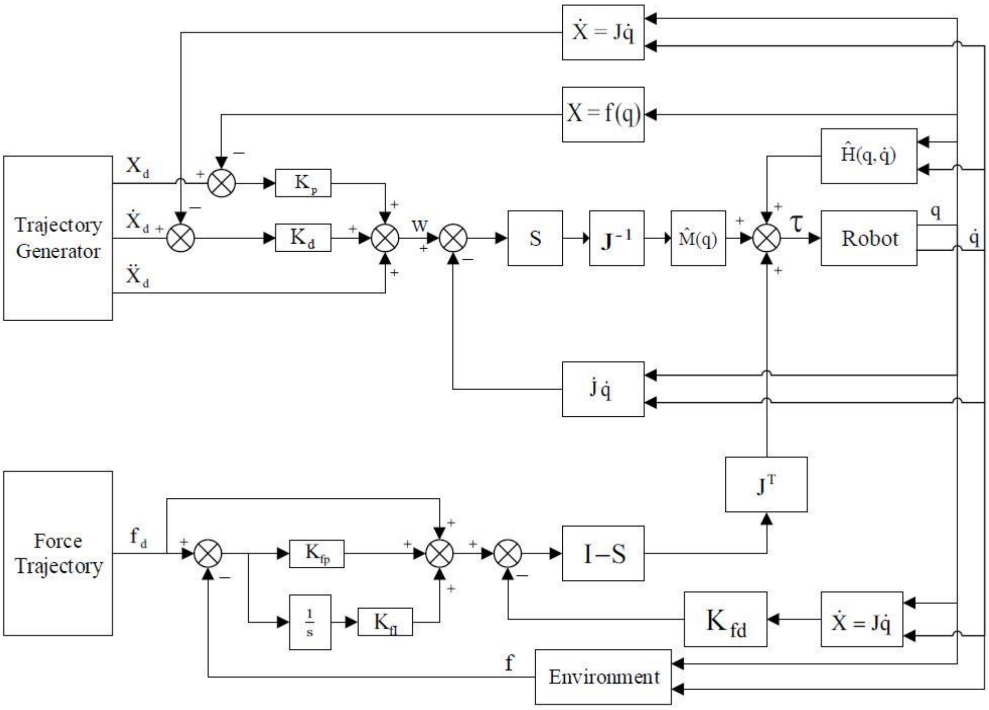


Figure 3.17 : Dynamic hybrid force/position control.

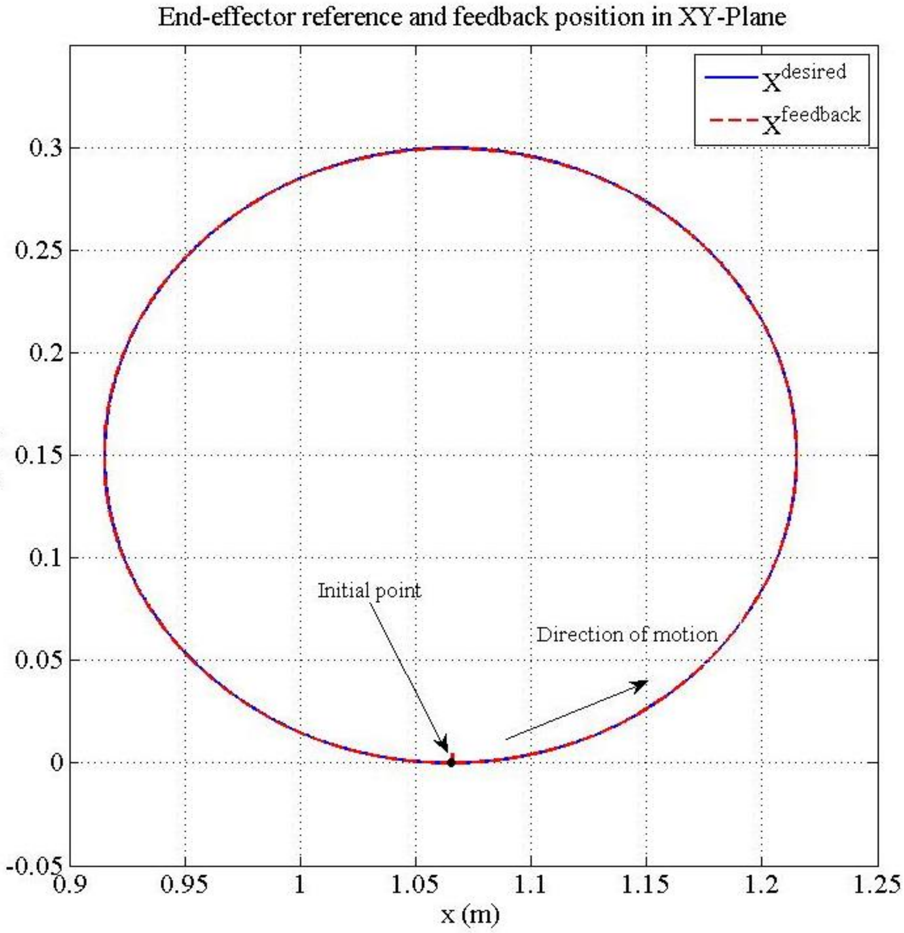


Figure 3.18 : Desired and feedback path in hybrid force/position control.

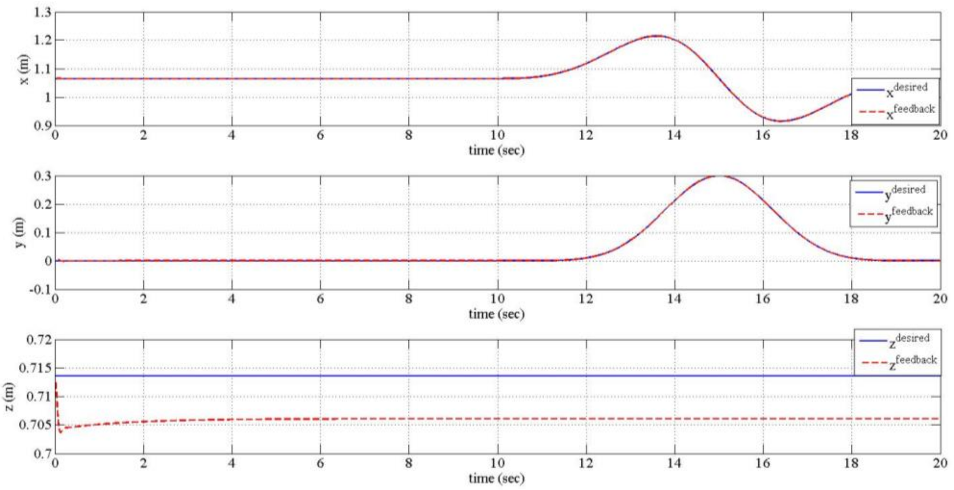


Figure 3.19 : Desired and feedback trajectories in hybrid force/position control.

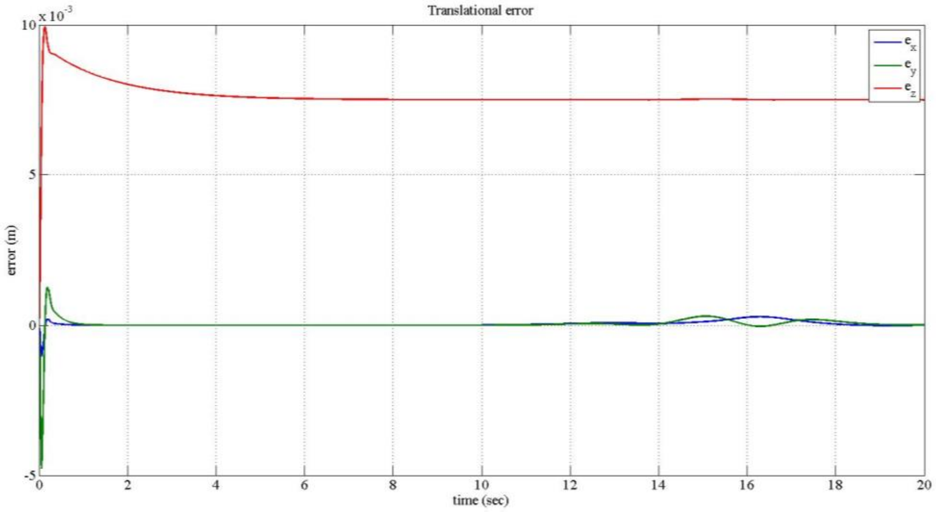


Figure 3.20 : Translational errors in hybrid force/position control.

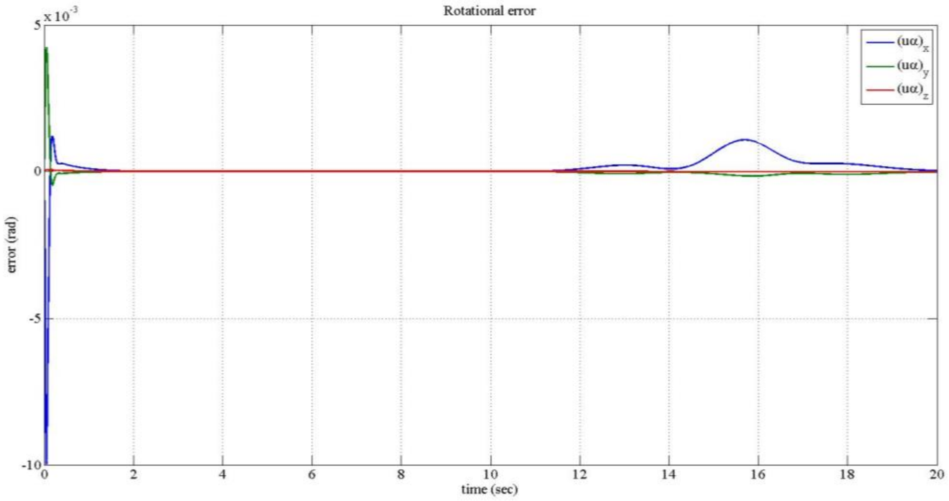


Figure 3.21 : Rotational errors in hybrid force/position control.

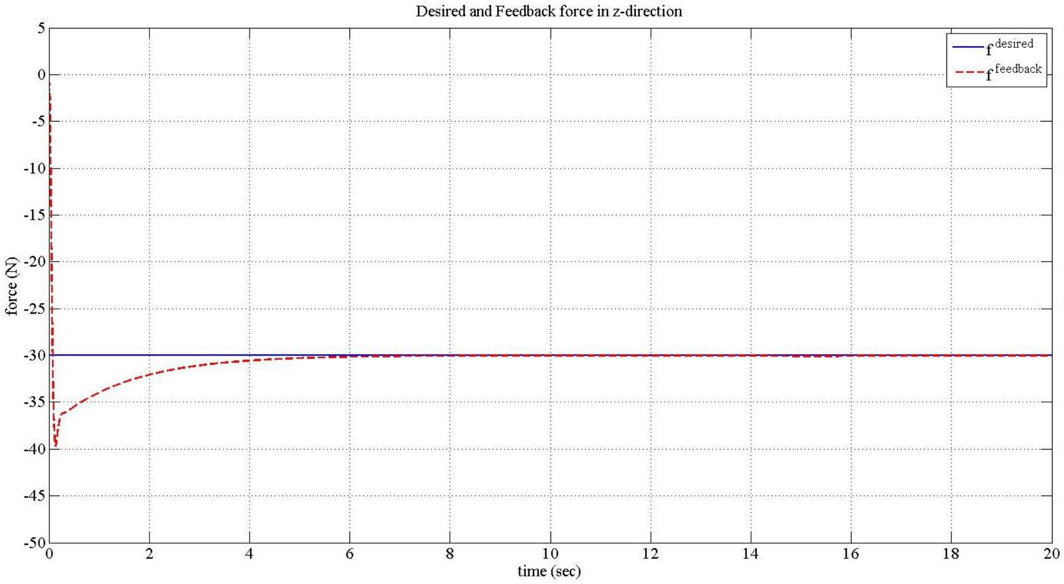


Figure 3.22 : Desired and feedback force in hybrid force/position control.

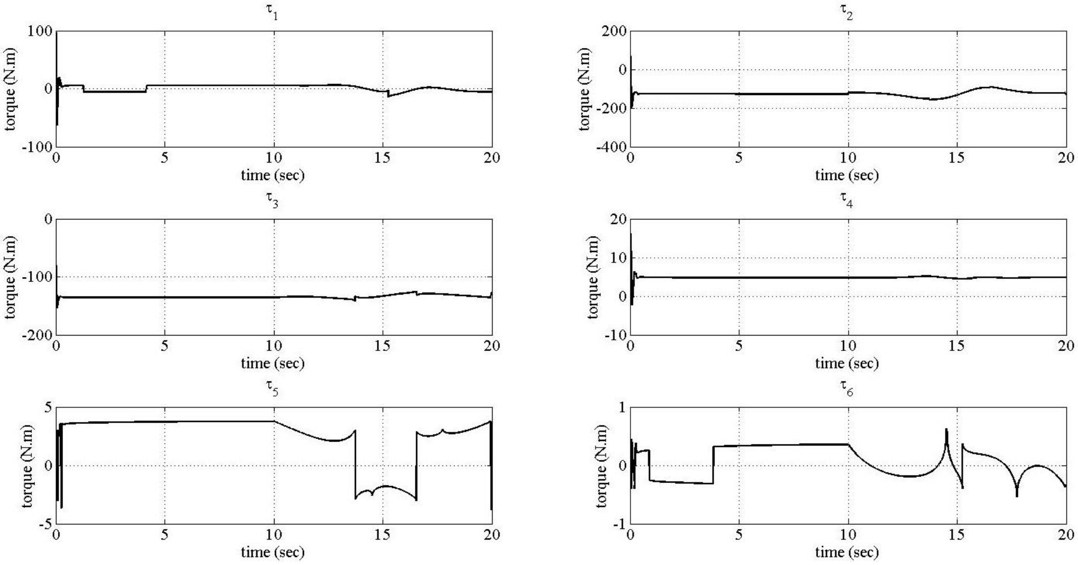


Figure 3.23 : Calculated joint torques in hybrid force/position control.

4. IMPLEMENTATION AND EXPERIMENTS

In this chapter, control schemes considered in the previous chapter are implemented on the robot and real-time experimental results are shown.

4.1 Implementation

Experimental setup is shown in Figure 4.1. The surface is flat and rigid so that the tip point of the end-effector cannot penetrate it. Since there is a metal ball wheel at the tip, the friction between the surface and tip point is ignored. So the end-effector is free to move along the tangent of the surface (XY-plane of the base frame), but constrained along the normal of the surface (z-direction of the base frame).

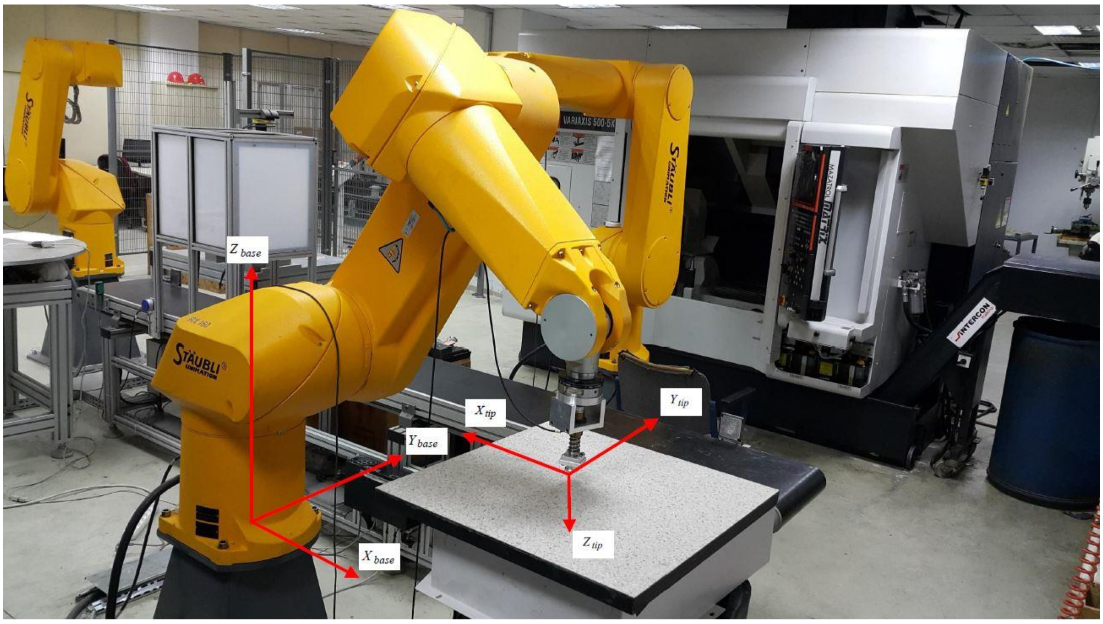


Figure 4.1 : Experimental setup.

The analog output signals from the force/torque sensor had to be filtered to remove unwanted high-frequency steps (discontinuities) that occur when the output is updated [10]. To ensure the fastest output possible, the F/T controller does not filter its analog outputs. In our experiments, we used a basic one-dimensional Kalman filter and embedded it in LLI. The six components of force/torque data are filtered independently before they are used in the control algorithm. The C code we used is given in Appendix C.

In our experiments, we saw that Lagrange-Euler equations of motion could not be calculated in one cycle time (0.004 secs). For this reason, we implemented Newton-Euler equations in the control algorithms. Motion control, impedance control and hybrid force/position control diagrams are shown in Figure 4.2, Figure 4.3 and Figure 4.4 respectively. It is not shown in the below diagrams, however joint velocity measurements are also filtered with the same filter applied on force measurements.

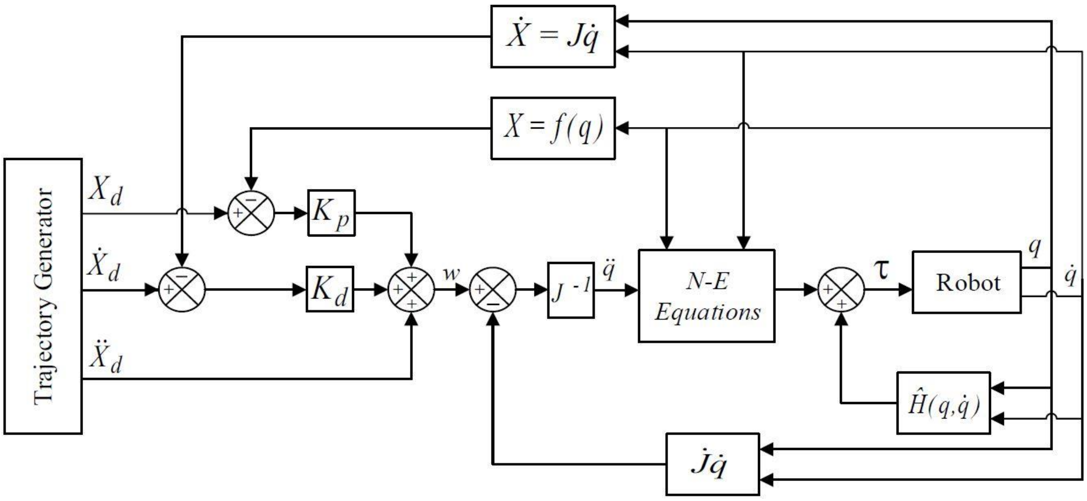


Figure 4.2 : Task-space motion control diagram in experiments.

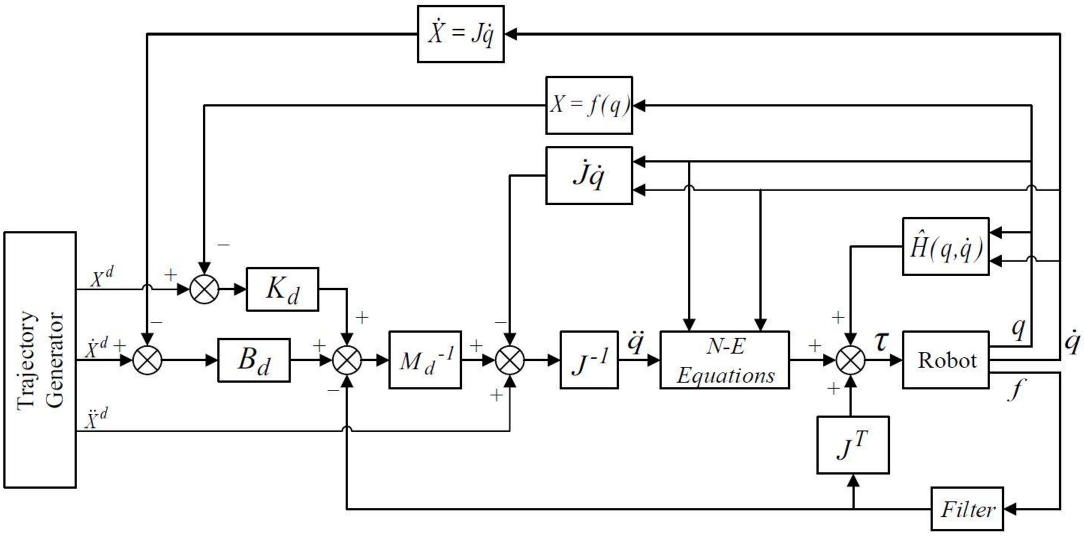


Figure 4.3 : Impedance control diagram in experiments.

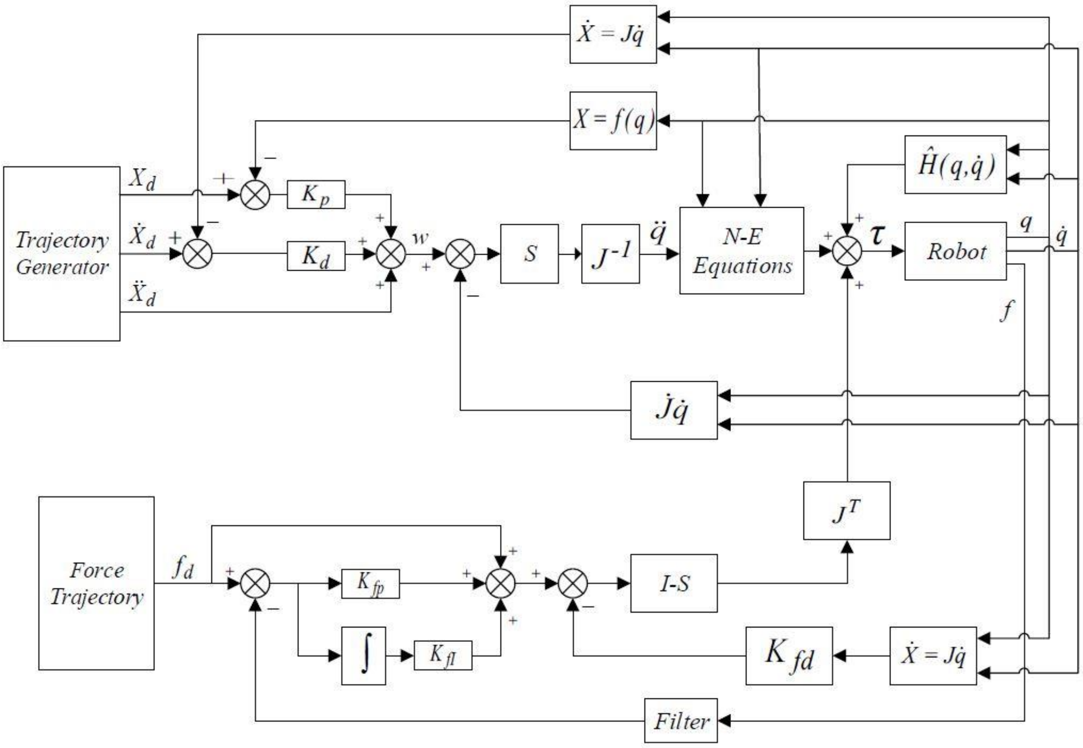


Figure 4.4 : Hybrid force/position control diagram in experiments.

In these diagrams, $\widehat{H}(q,\dot{q})$ does not consist the gravity compensation term. Gravity effects are dealt with in Newton-Euler equations.

$$\widehat{H}(q,\dot{q}) = \tau_s(q) + \tau_f(\dot{q}) \tag{4.1}$$

4.2 Experimental Results

Experimental results for each control scheme is given in this section. More experimental results for impedance control and hybrid force/position control can be found in Appendix A and Appendix B respectively.

4.2.1 Motion control

In motion control experiments, the end-effector is commanded to follow a circular path in XY-plane with 150 mm radius while maintaining its orientation and complete its motion in 20 seconds. The results are shown in the following figures.

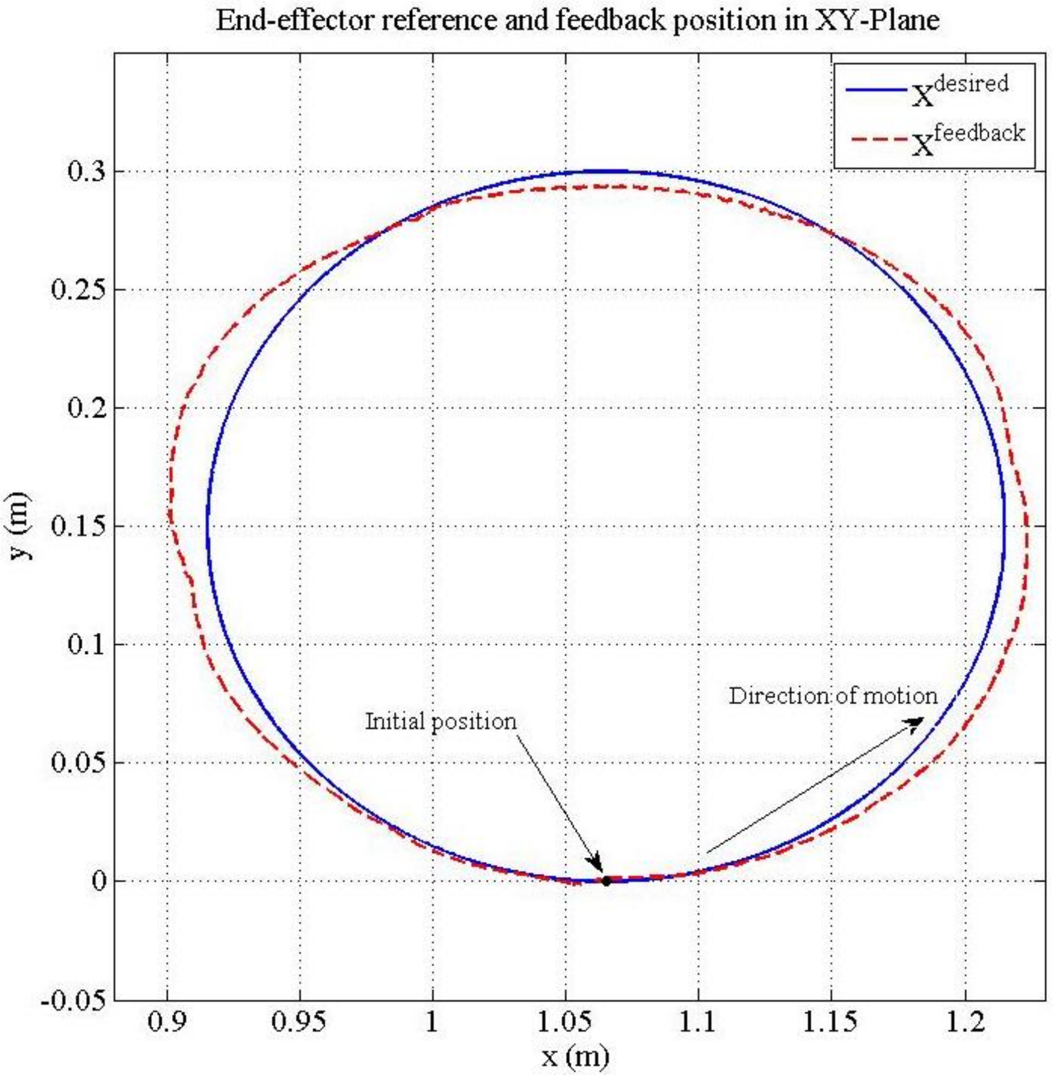


Figure 4.5 : Desired and feedback path.

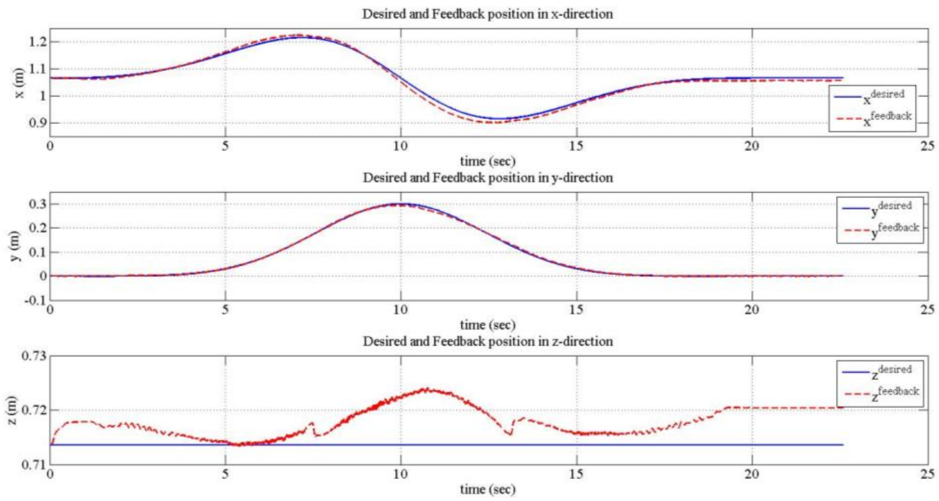


Figure 4.6 : Desired and feedback trajectories.

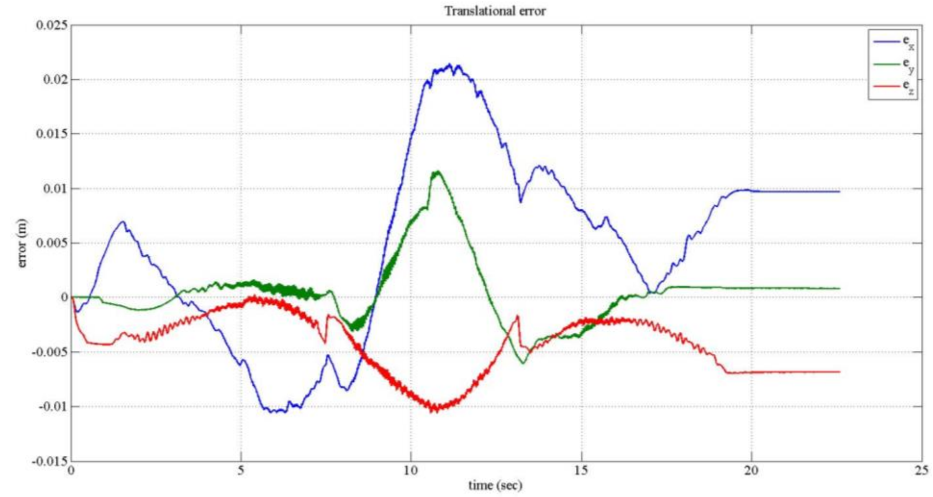


Figure 4.7 : Translational errors.

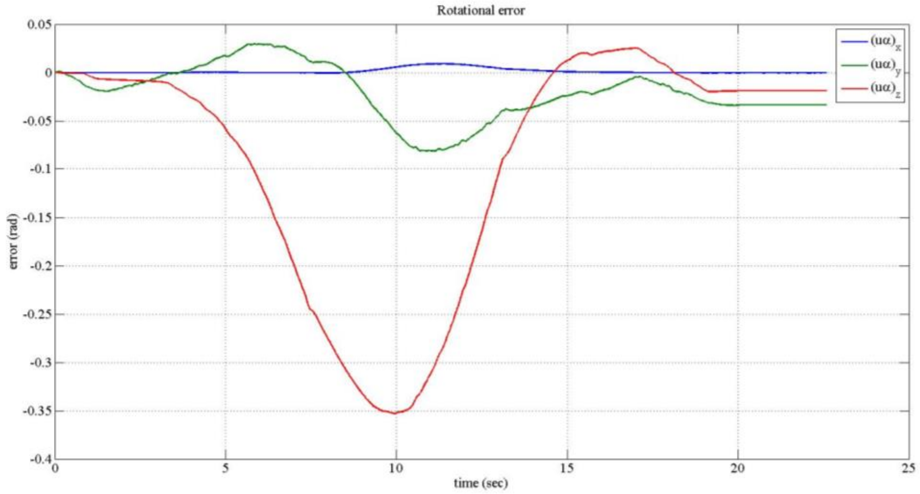


Figure 4.8 : Rotational errors.

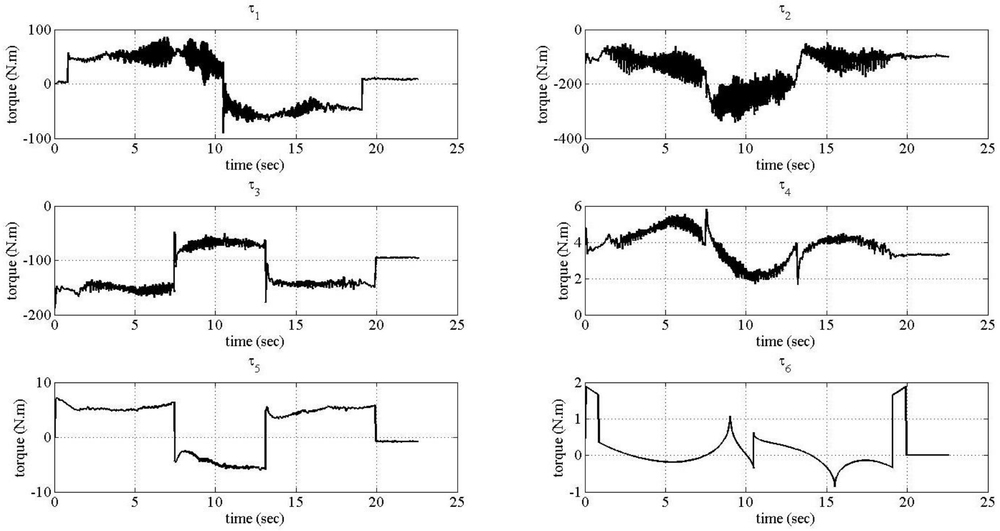


Figure 4.9 : Applied joint torques.

4.2.2 Impedance control

In impedance control experiments, first the end-effector is commanded to search for a contact in z-direction with free motion control. When contact is detected, the control algorithm is switched to impedance control. The desired position in z-direction is given as 25 mm under the surface to maintain contact with the surface. The manipulator waited 5 seconds to settle and then moved 120 mm in x-direction while keeping contact with the surface for 15 seconds. At the last part of the motion, the control switched back to free motion control and the end-effector is commanded to move in positive z-direction to break the contact in 5 seconds.

Multiple experiments are performed with different M_d and K_d. Desired damping B_d is kept constant and chosen as large as possible. We saw large values for B_d induced too much vibration in the motion.

In the following experiment, the manipulator could not keep contact with the surface. The end-effector started to bounce on the surface when the motion started. Desired values for M_d, K_d and B_d are as follows

- $M_d = diag(2, 2, 17.5, 1.5, 1.5, 1.5)$
- $K_d = diag(2000, 2000, 1000, 1000, 1000, 1000)$
- $B_d = diag(150, 150, 250, 100, 100, 100)$

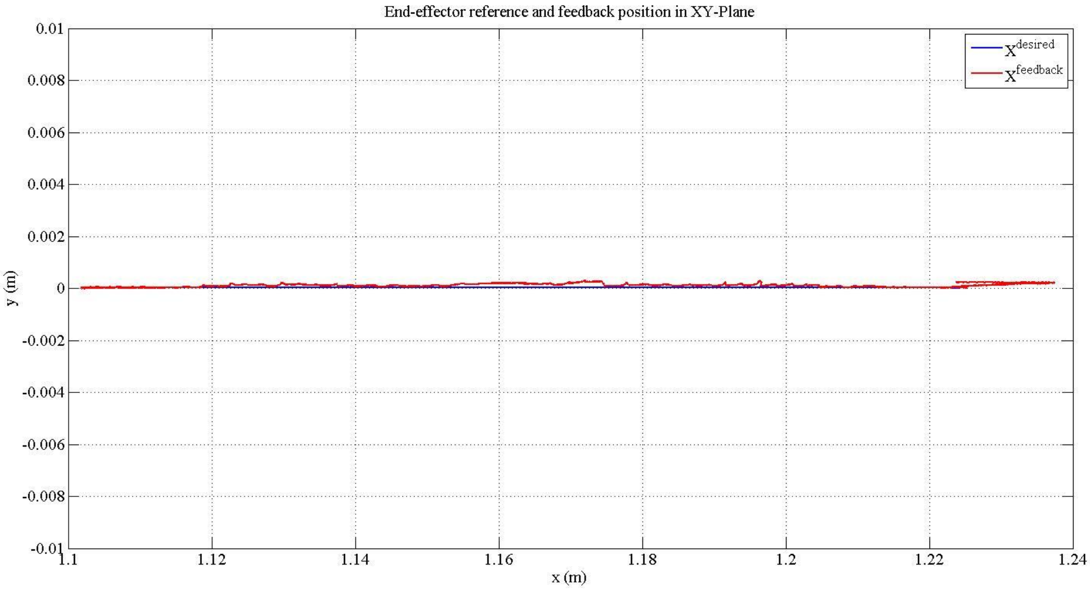


Figure 4.10 : Desired and feedback path.

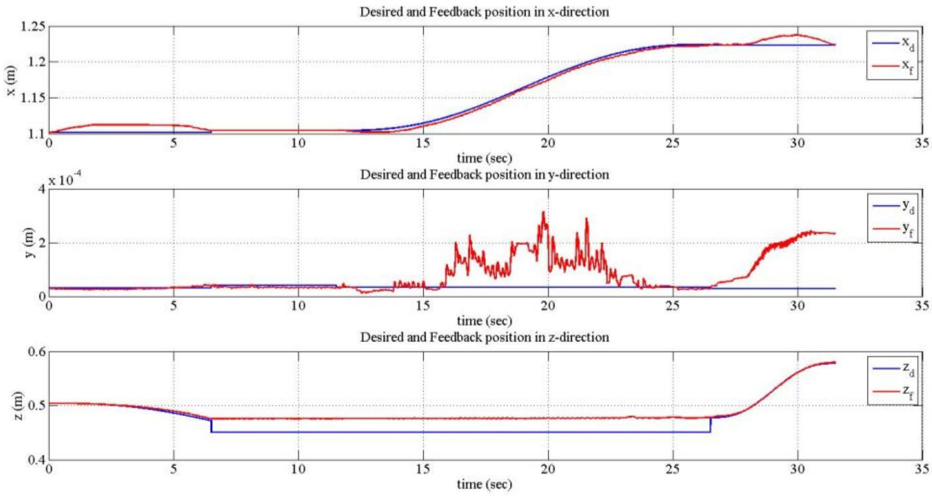


Figure 4.11 : Desired and feedback positions.

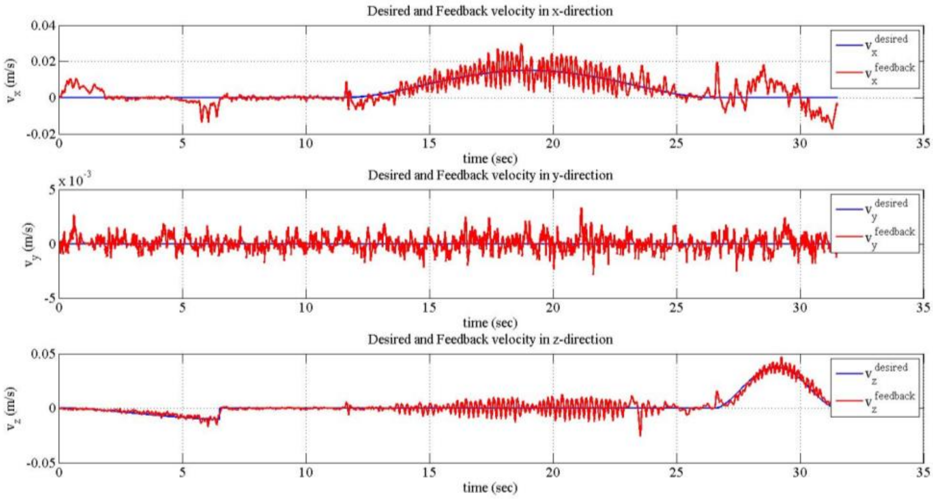


Figure 4.12 : Desired and feedback velocities.

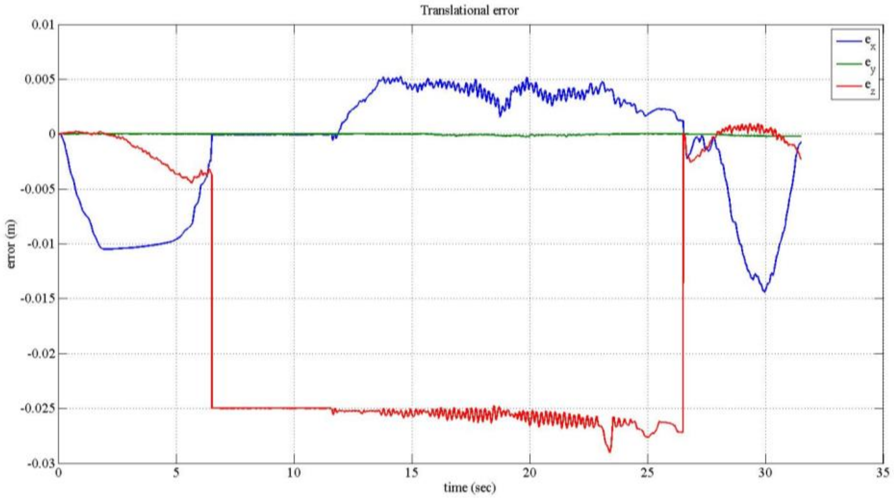


Figure 4.13 : Translational errors.

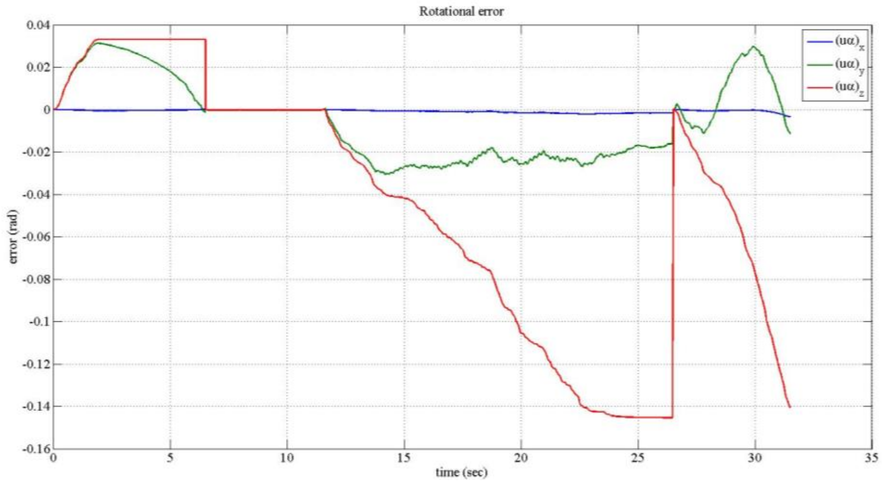


Figure 4.14 : Rotational errors.

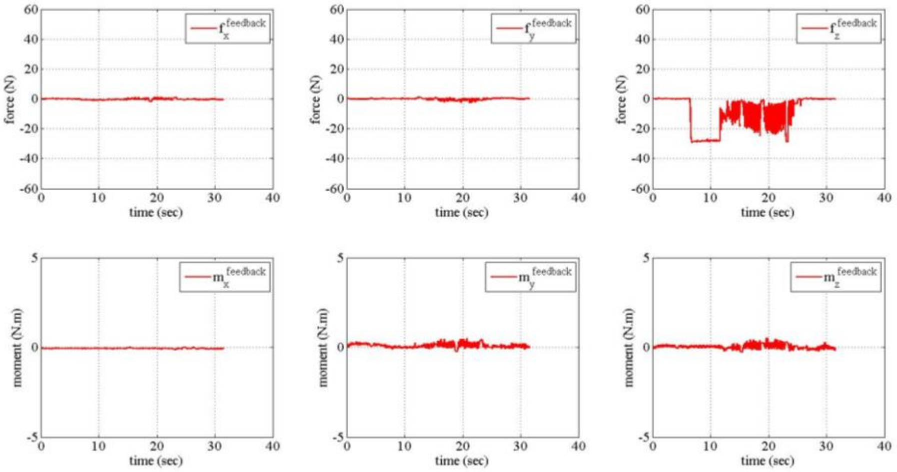


Figure 4.15 : Feedback forces.

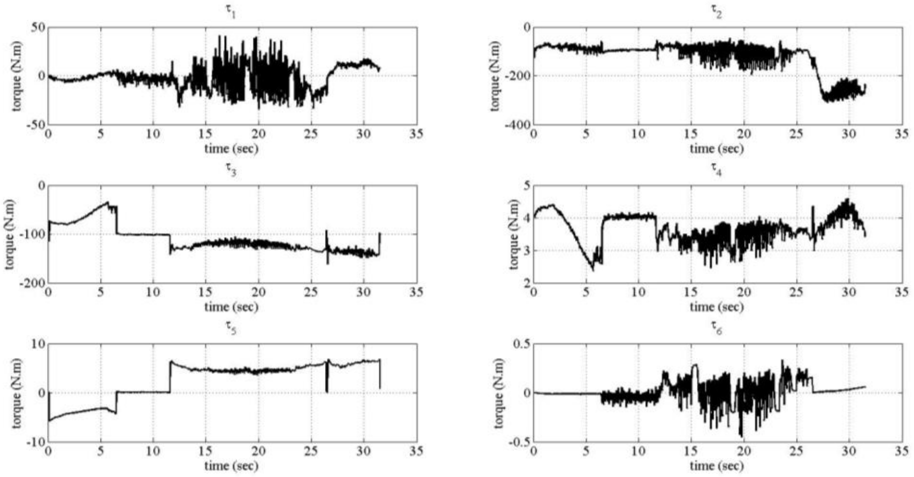


Figure 4.16 : Applied joint torques.

In the next experiment, K_d value in the z-direction is increased. This time, the end-effector could keep its contact with the surface. Also, force applied on the surface is increased.

- $M_d = diag(2, 2, 17.5, 1.5, 1.5, 1.5)$
- $K_d = diag(2000, 2000, 2500, 1000, 1000, 1000)$
- $B_d = diag(150, 150, 250, 100, 100, 100)$

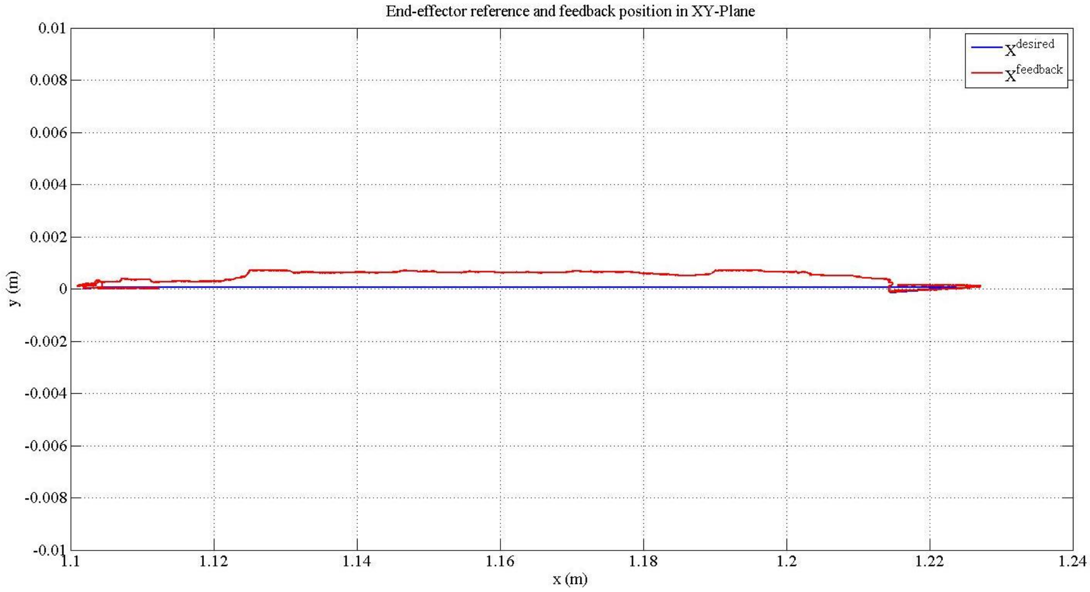


Figure 4.17 : Desired and feedback path.

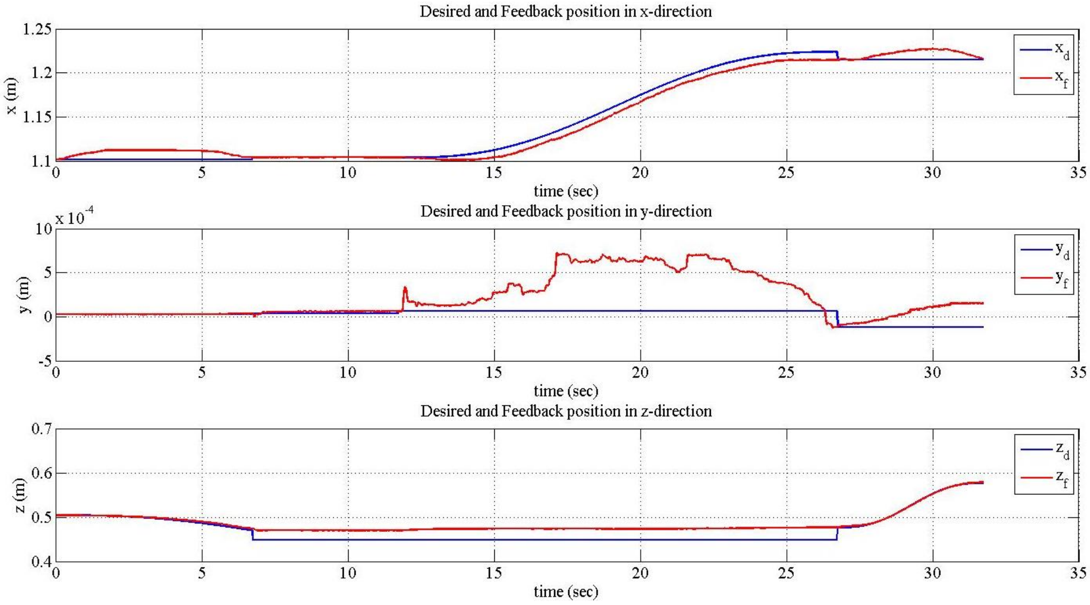


Figure 4.18 : Desired and feedback positions.

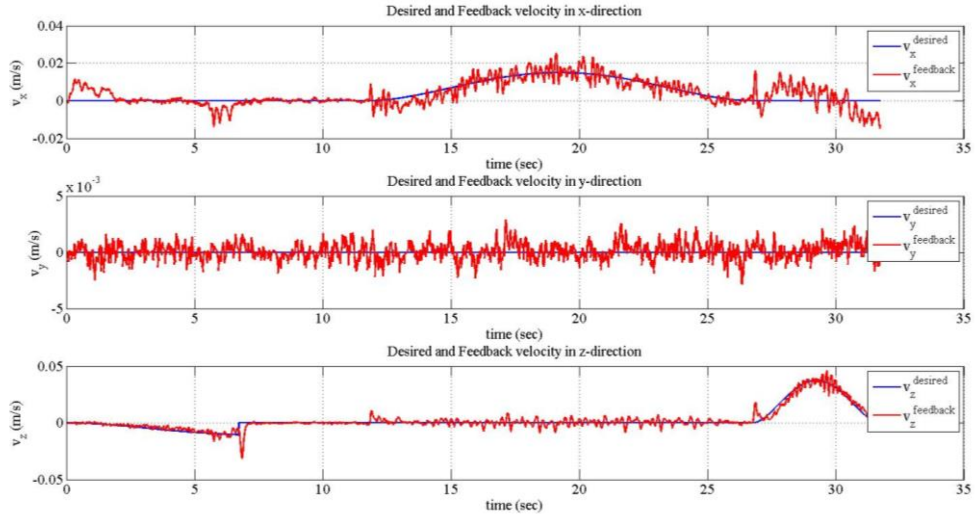


Figure 4.19 : Desired and feedback velocities.

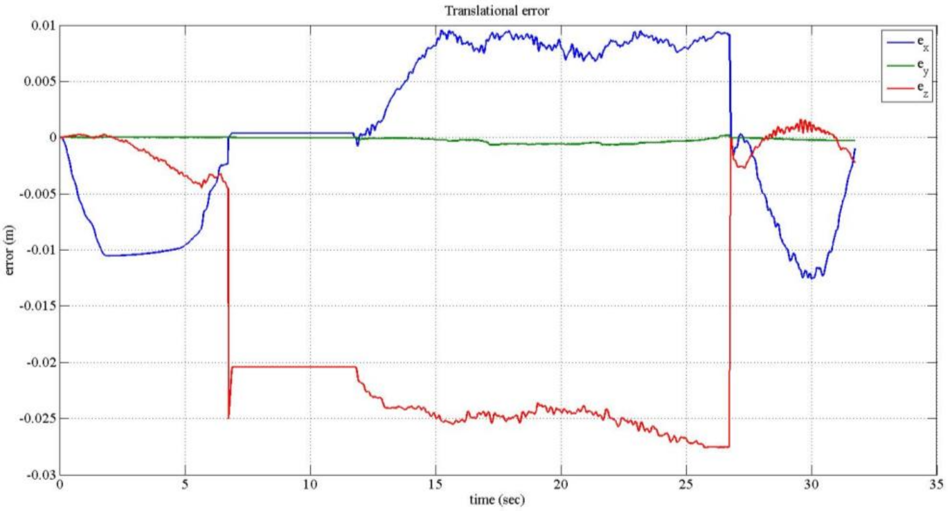


Figure 4.20 : Translational errors.

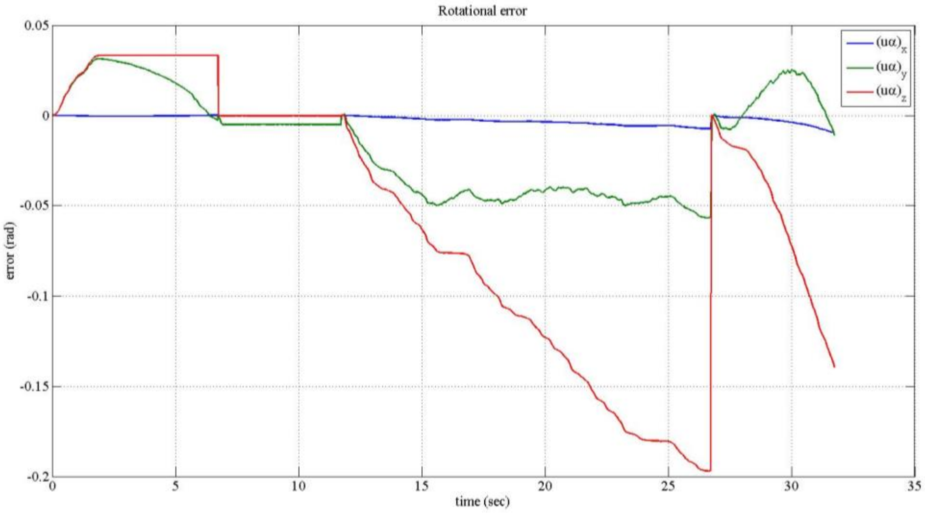


Figure 4.21 : Rotational errors.

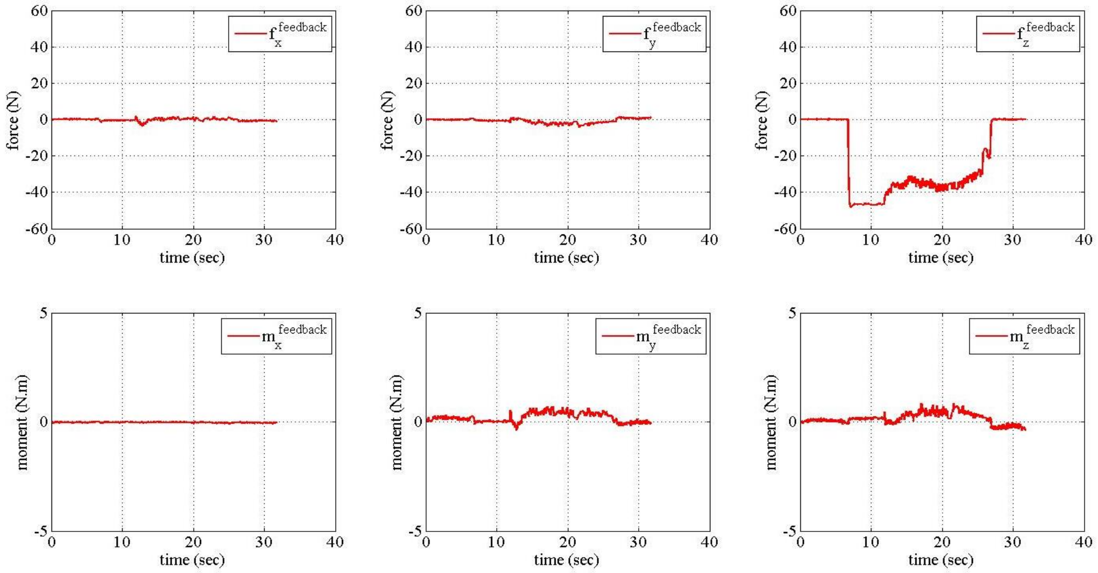


Figure 4.22 : Feedback forces.

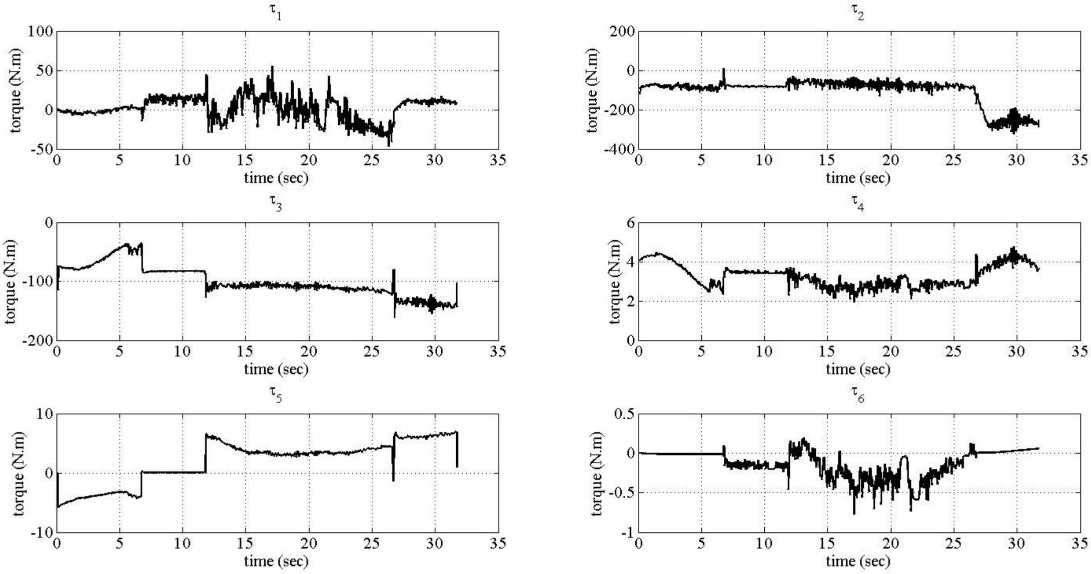


Figure 4.23 : Applied joint torques.

In the last experiment, K_d values kept constant and M_d value in z-direction is decreased. The system responded faster to position error which caused a force overshoot in z-direction.

- $M_d = diag(2, 2, 7.5, 1.5, 1.5, 1.5)$
- $K_d = diag(2000, 2000, 2500, 1000, 1000, 1000)$
- $B_d = diag(150, 150, 250, 100, 100, 100)$

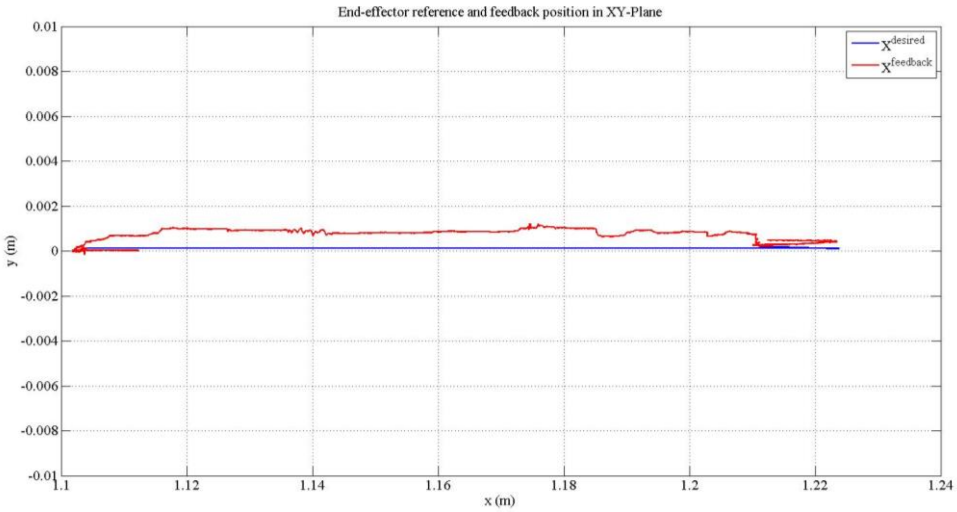


Figure 4.24 : Desired and feedback path.

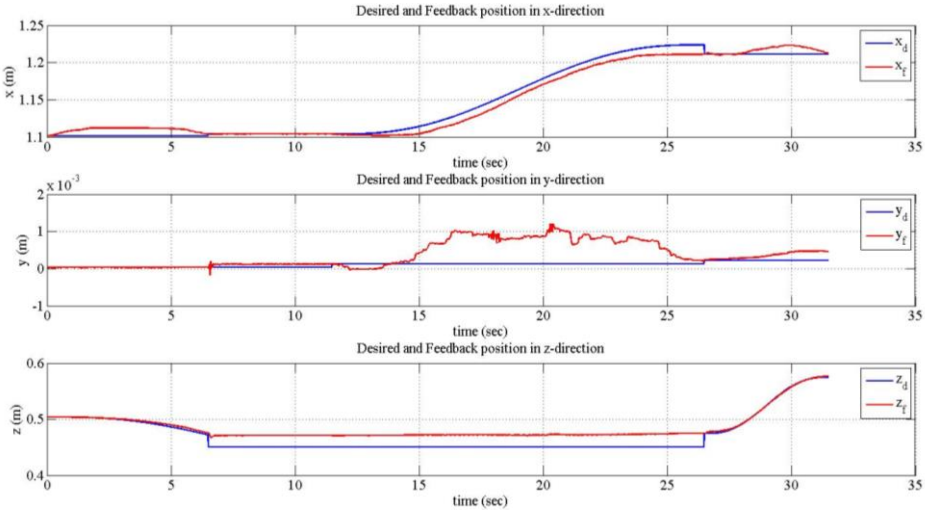


Figure 4.25 : Desired and feedback positions.

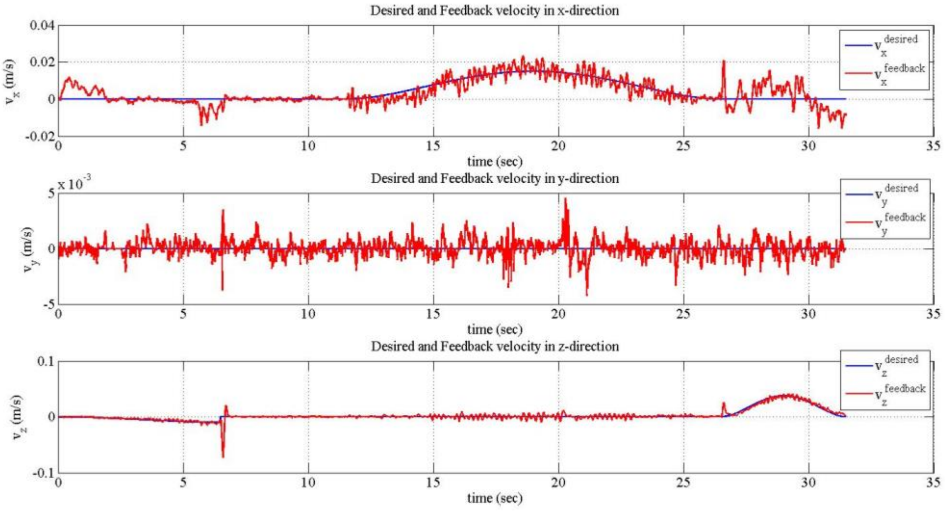


Figure 4.26 : Desired and feedback velocities.

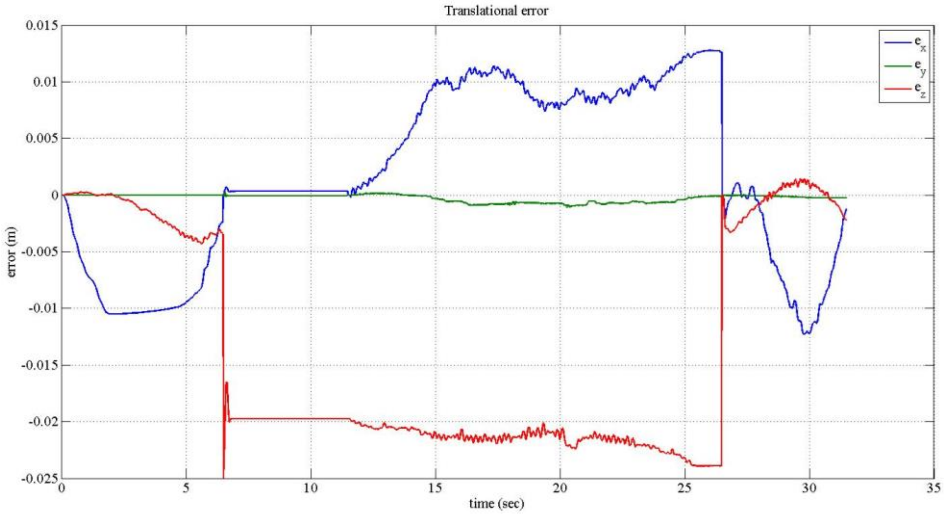


Figure 4.27 : Translational errors.

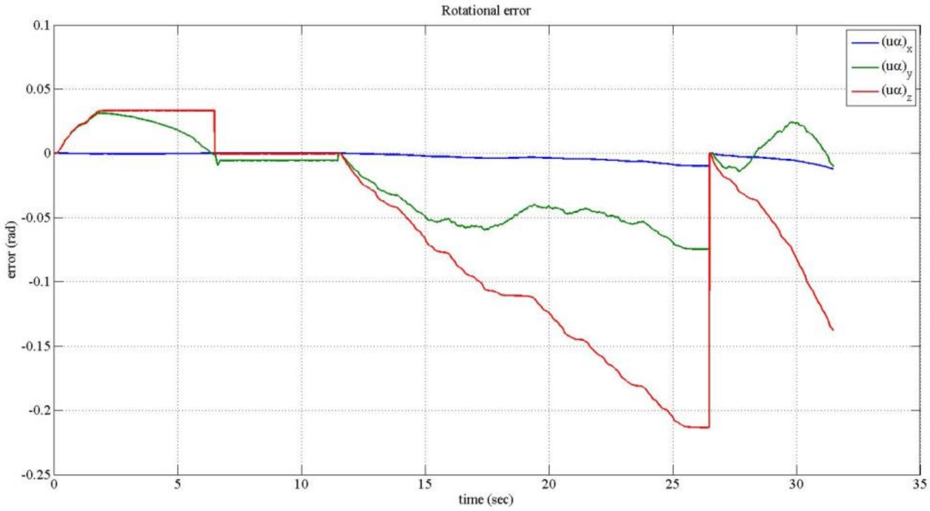


Figure 4.28 : Rotational errors.

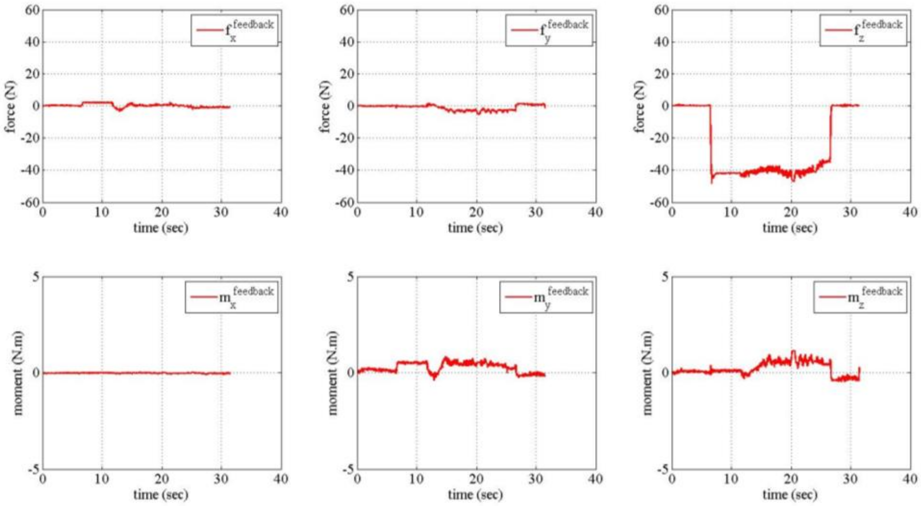


Figure 4.29 : Feedback forces.

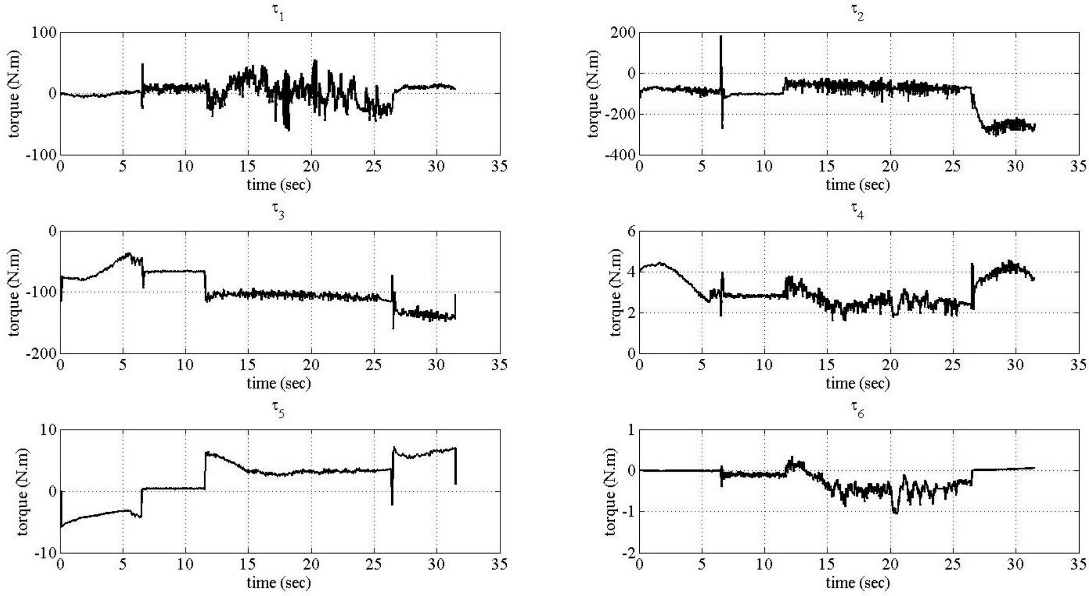


Figure 4.30 : Applied joint torques.

4.2.3 Hybrid force/position control

In hybrid force/position control experiments, the end-effector is commanded to move with free motion control in negative z-direction until it contacts with the surface. The contact force threshold is set as -15 N. When contact is detected, the controller is switched from free motion control to hybrid force/position control. Then 10 seconds is given to the manipulator to settle the force to a desired value. After 10 seconds has past, the end-effector is commanded to move on the surface while applying a constant force. At the end of the motion, the manipulator is commanded to break contact with the surface by moving in the positive z-direction for 5 seconds.

In the following experiment, the end-effector path is a straight line with displacements of 120 mm in both x and y-directions. The constant force to be applied on the surface equals to -35 N. Force control gains are set as

- $K_{fp} = diag(4.5, 4.5, 4.5, 4.5, 4.5, 4.5)$
- $K_{fd} = diag(200, 200, 200, 200, 200, 200)$
- $K_{fI} = diag(2.0, 2.0, 2.0, 2.0, 2.0, 2.0)$

In these experiments, when the controller is switched to hybrid force/position control from free motion control and to free motion control from hybrid force/position control, current feedback position is equated to initial position. This is why there are jumps in translational and rotational errors. At the start of each control phase, initial desired value is equal to feedback value, hence zero error. This is also the case for impedance control experiments.

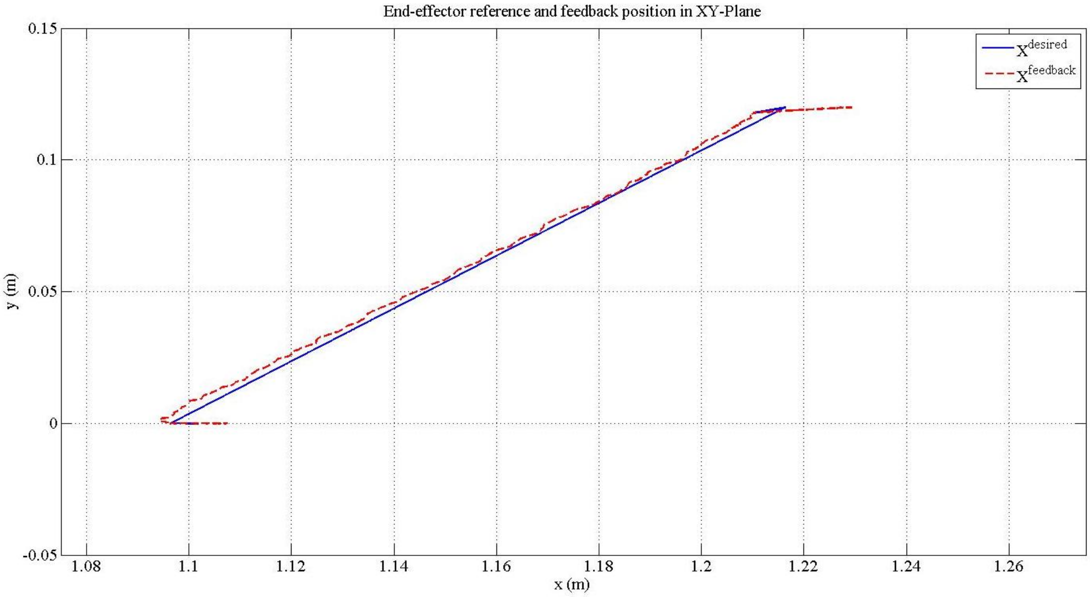


Figure 4.31 : Desired and feedback path.

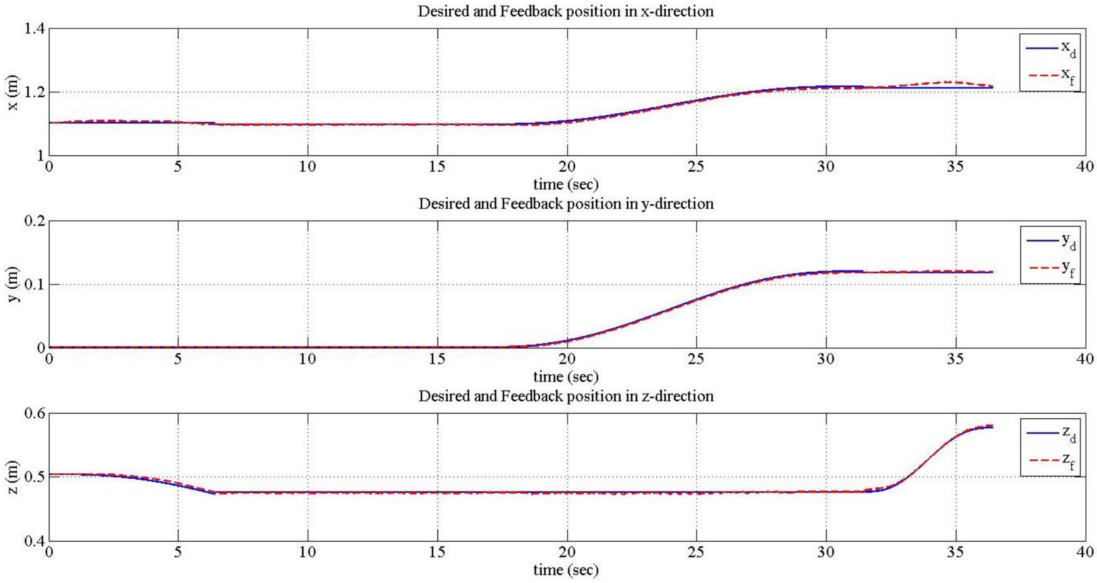


Figure 4.32 : Desired and feedback trajectories.

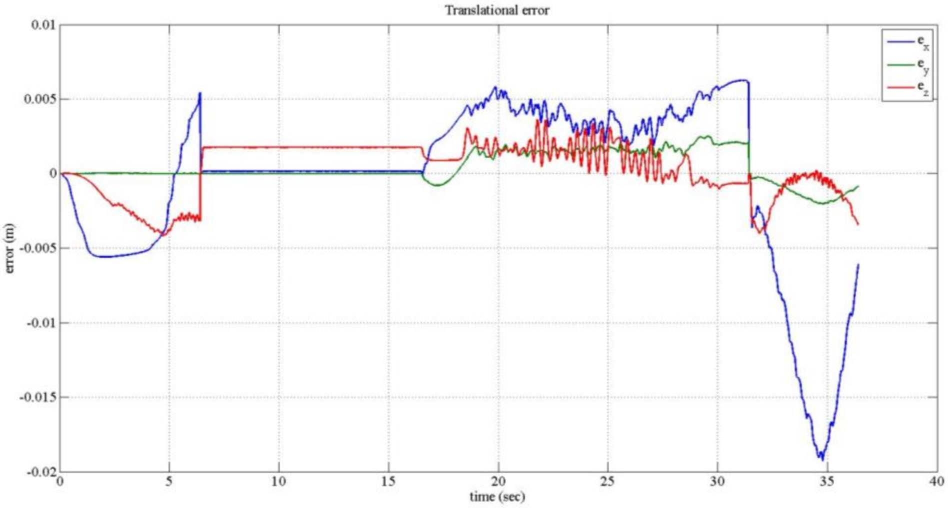


Figure 4.33 : Translational errors.

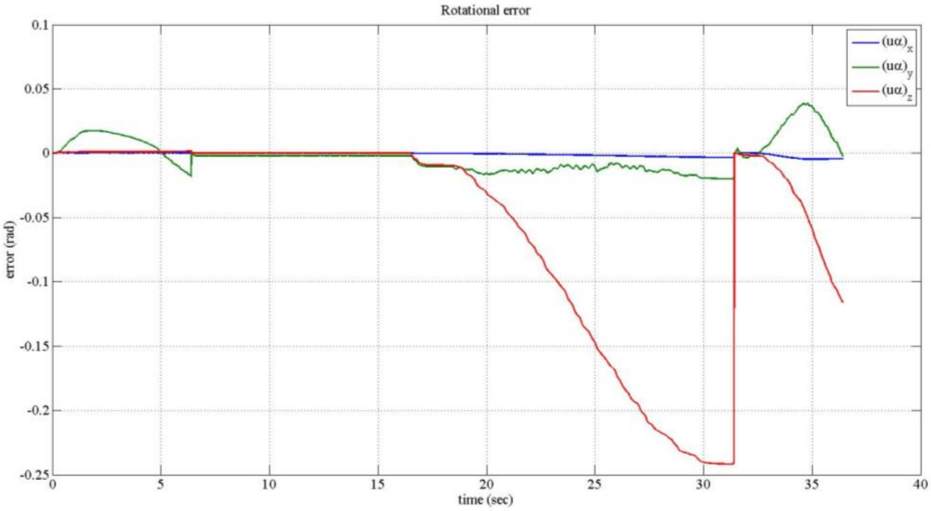


Figure 34 : Rotational errors.

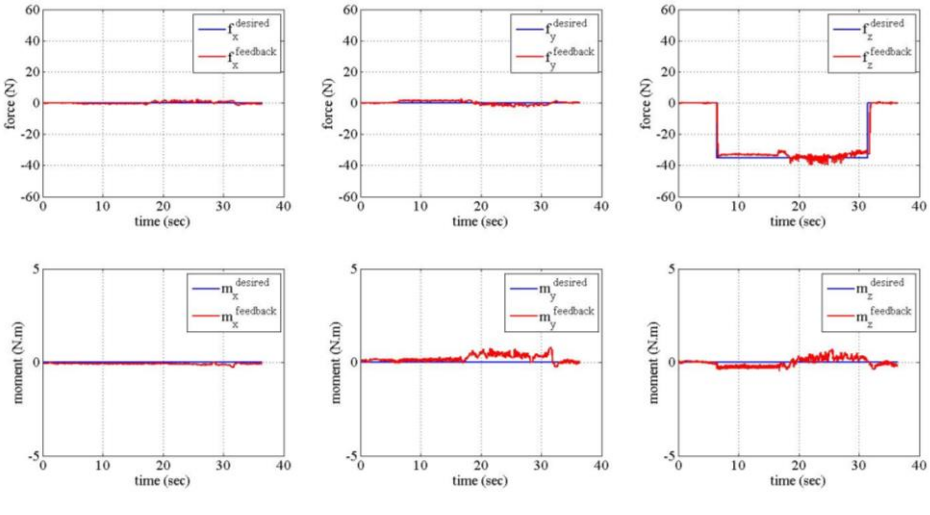


Figure 4.35 : Desired and feedback forces.

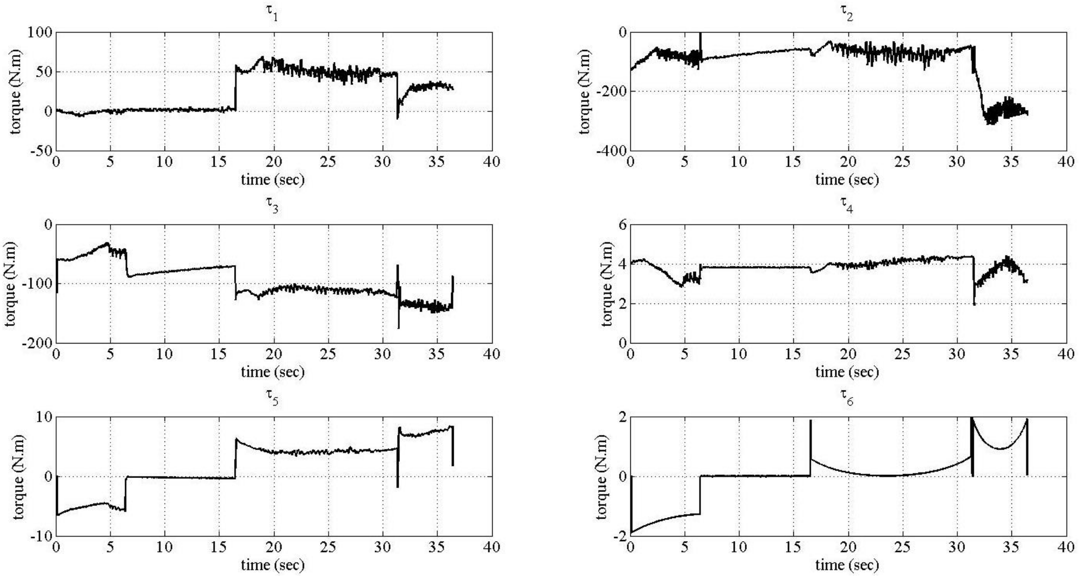


Figure 4.36 : Applied joint torques.

5. CONCLUSION

The purpose of this thesis was the implementation of force control algorithms to Stäubli RX-160 manipulator and achievement of tasks where the robot end-effector interacts with its environment. First, we modeled the robot both with Euler-Lagrange and Newton-Euler formulations. The former was used in simulations only, since it introduced too much computational delay inacceptable for real-time operations. So we switched to Newton-Euler formulation in real-time experiments.

By using LLI, we wrote programs for control schemes proposed in literature. Two force control schemes were considered: impedance control and hybrid force/position control. The main difficulty in experiments was tuning the feedback gains for these control schemes. In hybrid force/position control, since we already had PD gains from free motion control algorithm, we only had to tune the PID gains in force control part of this scheme. However, for impedance control, gains are different from free motion control. Nevertheless, impedance control gains (M_d, K_d, B_d) represent a well known mechanical system. So instead of relying on trial and error only, these parameters can be adjusted with respect to each other by pointing out the relation between them and the behavior of the system with respect to them.

Experimental results showed that the manipulator cannot follow desired orientation very well. This error is mostly caused by fifth and six axis of the manipulator which have big effects on the orientation of the end-effector. The reason maybe the friction model used for these two joints, which has a dominant effect on the dynamic model of these joints. Inertial effects are small compared to friction at these joints. In order to get better results on orientation, a more precise friction model for the fifth and sixth joints can be identified.

We also used Kalman filter to eliminate noises in the force/torque measurements. However this introduced some delay in feedback data which was ignored in our studies. In future studies, gains we used in our filter can be adjusted to have better results.

REFERENCES

[1] **Url-1** <http://www.staubli.com/en/robotics/>, date retrieved 03.05.2015.

[2] **J. K. Salisbury** (1980). Active stiffness control of a manipulator in Cartesian coordinates, 19th IEEE Conf. Decis. Contr., Albuquerque, pp 95-100.

[3] **N. Hogan** (1985). Impedance control: an approach to manipulation: parts I-III, ASME J. Dyn. Syst. Meas. Contr, 107, 1-24.

[4] **M. H. Raibert, and J.J. Craig** (1981). Hybrid position/force control of manipulators, ASME J. Dyn. Syst. Meas. Contr., 103, 126-133.

[5] **O. Khatib** (1987). A unified approach for motion and force control of robot manipulators: the operational space formulation, IEEE J. Robot. Autom., 3, 43-53.

[6] **Fisher W., and Mujtaba M.S.** (1992). Hybrid position/force control: a correct formulation, The Int. J. of Robotics Research, Vol 11(4), p. 299-311.

[7] **Stäubli Favarges** (2008). Arm – RX series 160 family, Instruction Manual.

[8] **Stäubli Favarges** (2008). CS8C Controller, Instruction Manual.

[9] **Url-2** <http://www.ati-ia.com/products/ft/ft_models.aspx?id=Delta>, date retrieved 03.05.2015.

[10] F/T Controller (CTL/CTLJ/CON) Six-Axis Force/Torque Sensor System Installation and Operation Manual, January 2014.

[11] **F. Pertin, and J.-M. Bonnet-des-Tuves** (2004). Real Time Robot Controller Abstraction Layer, In Proc. Int. Symposium on Robots (ISR), Paris, France, March 2004.

[12] **Lung-Wen Tsai** (1999). Robot Analysis, The Mechanics of Serial and Parallel Manipulators.

[13] **Robert J. Schilling** (1990). Fundamentals of Robotics.

[14] **Mark W. Spong, Seth Hutchinson, and M. Vidyasagar.** Robot Modeling and Control, First Edition.

[15] **Egemen Zengin** (2013). Stäubli RX-160 Manipulatörünün Modellenmesi, Tanınması ve Kontrolü.

[16] **Robert Rens Waiboer** (2007). Dynamic Modelling, Identification and Simulation of Industrial Robots, For Off-Line Programming of Robotised Laser Welding.

[17] **W. Khalil, and E. Dombre** (2002). Modeling, Identification and Control of Robots.

[18] **Bruno Siciliano, and Oussama Khatib** (2008). Handbook of Robotics.

[19] **William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery** (1992). Numerical Recipes in C, The Art of Scientific Computing, Second Edition.

APPENDICES

APPENDIX A : Impedance control experimental results.
APPENDIX B : Hybrid force/position control experimental results.
APPENDIX C : C code for Kalman filter.
APPENDIX D : Matrix inversion by partitioning.

APPENDIX A

In this section, further experimental results are given for impedance control.

Experiment 1:

- Draw a 120 mm long line in x-direction.
- Motion time is 15 seconds in impedance control phase.
- $M_d = diag(2, 2, 17.5, 1.5, 1.5, 1.5)$
- $K_d = diag(2000, 2000, 1500, 1000, 1000, 1000)$
- $B_d = diag(150, 150, 250, 100, 100, 100)$

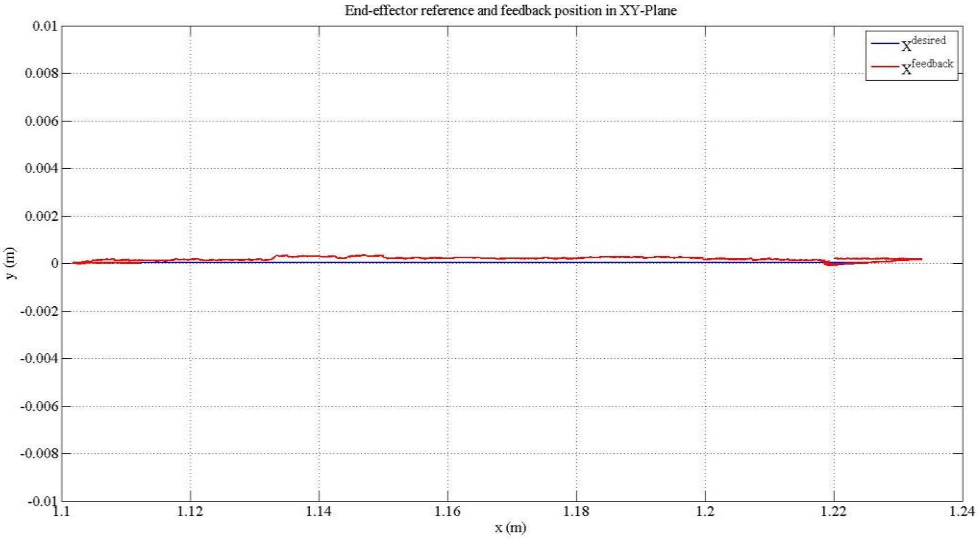


Figure A.1 : Desired and feedback path.

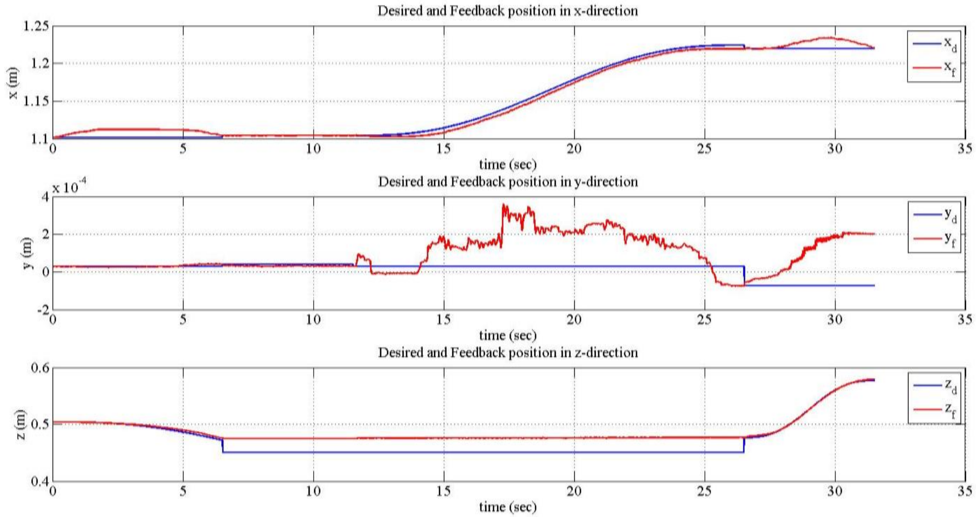


Figure A.2 : Desired and feedback positions.

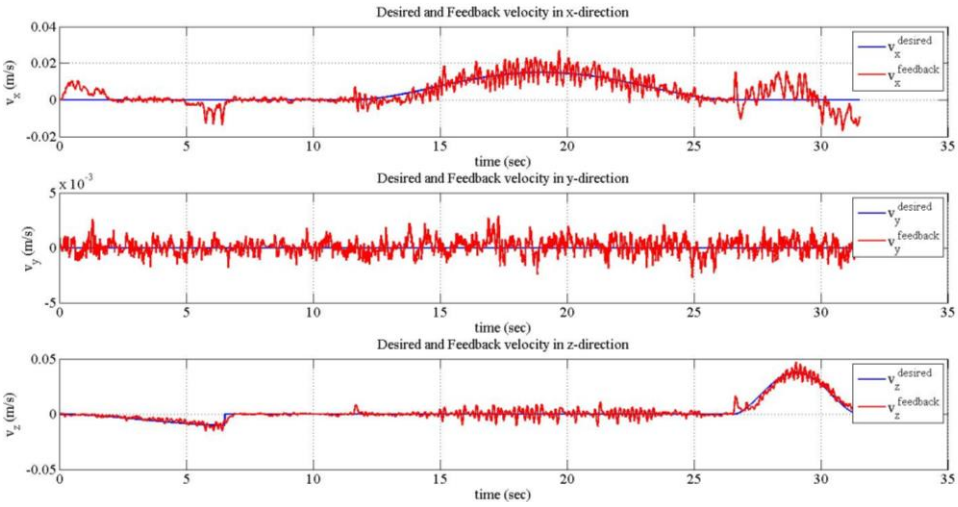


Figure A.3 : Desired and feedback velocities.

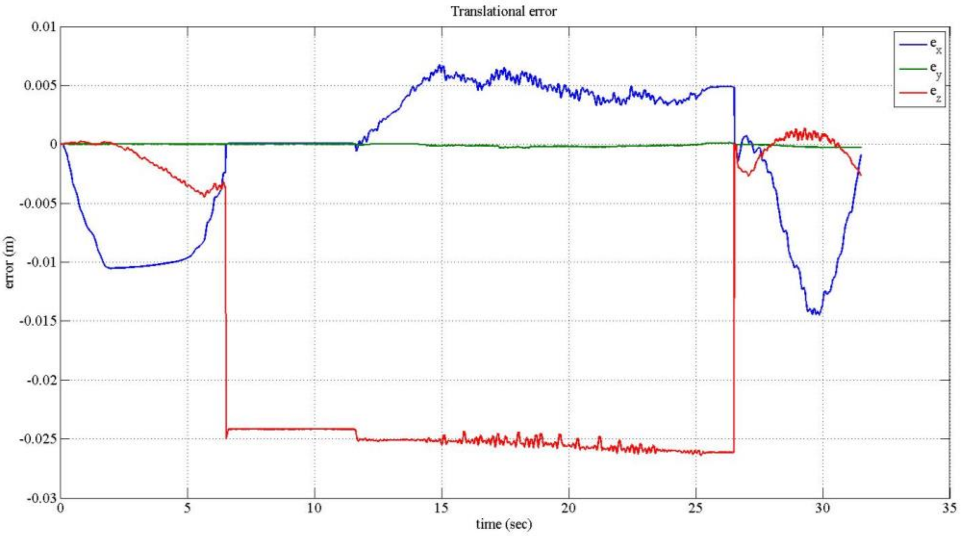


Figure A.4 : Translational errors.

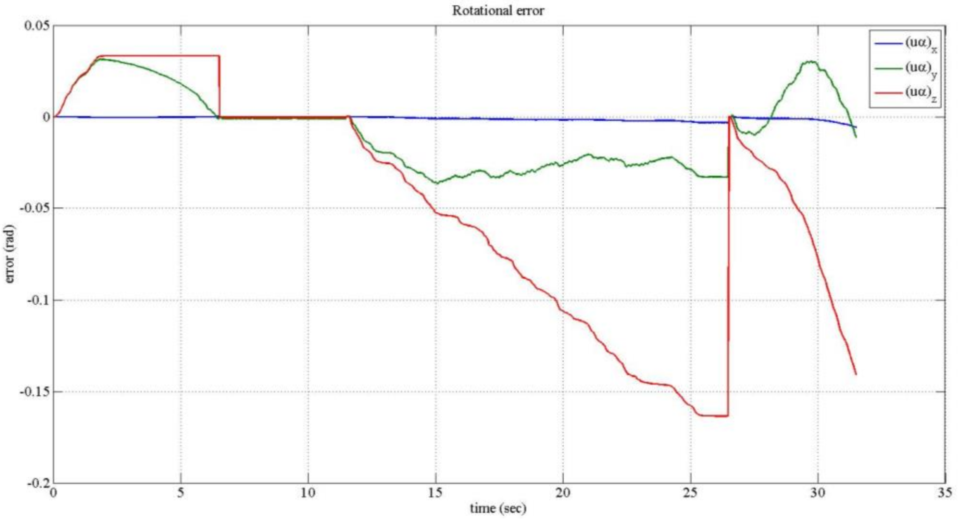


Figure A.5 : Rotational errors.

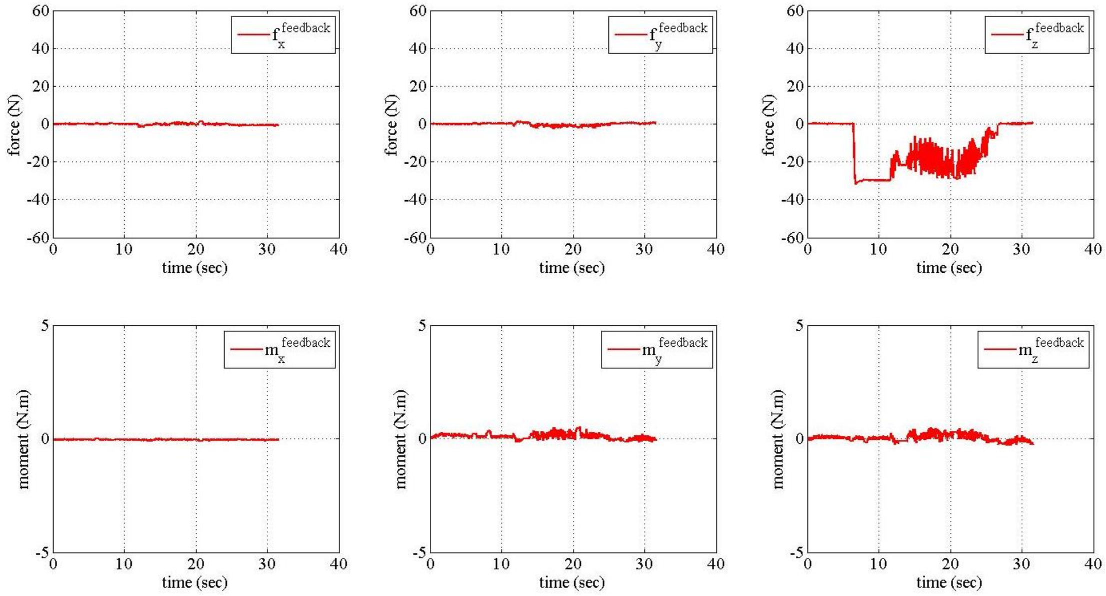


Figure A.6 : Feedback forces.

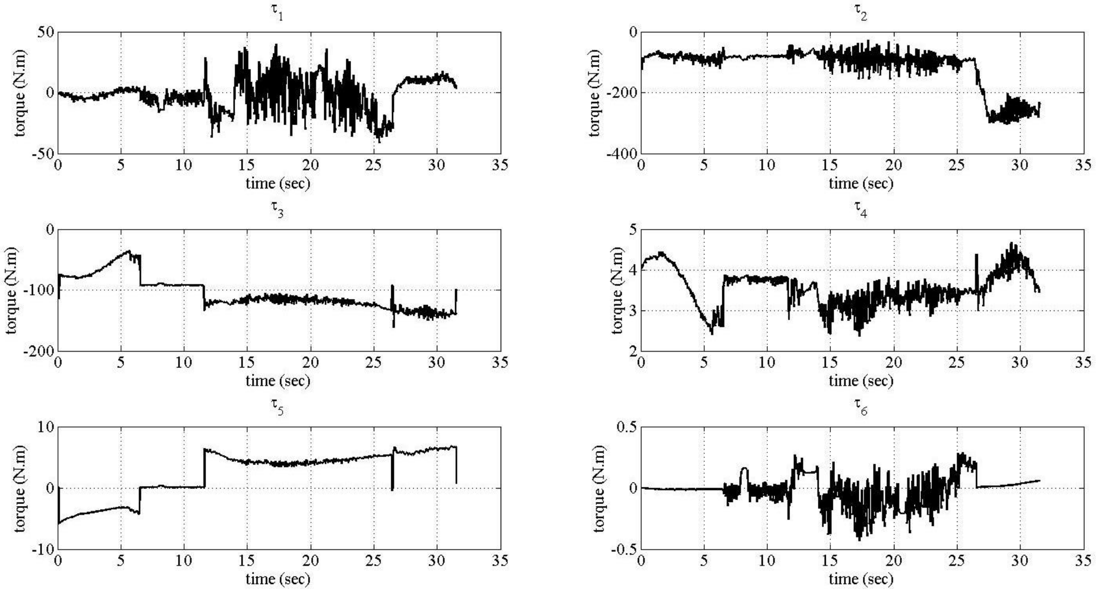


Figure A.7 : Applied joint torques.

Experiment 2:

- Draw a 120 mm long line in x-direction.
- Motion time is 15 seconds in impedance control phase.
- $M_d = diag(2, 2, 17.5, 1.5, 1.5, 1.5)$
- $K_d = diag(2000, 2000, 2000, 1000, 1000, 1000)$
- $B_d = diag(150, 150, 250, 100, 100, 100)$

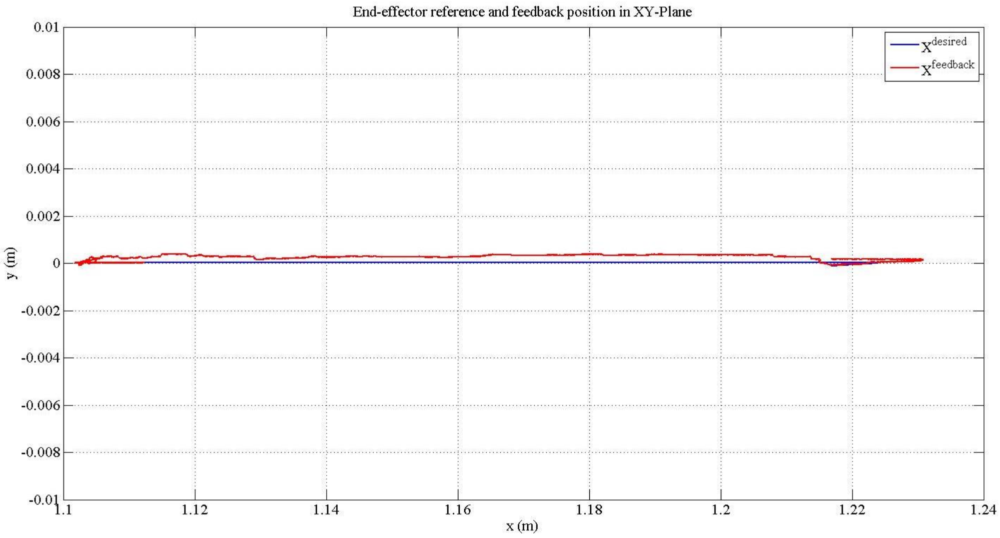


Figure A.8 : Desired and feedback path.

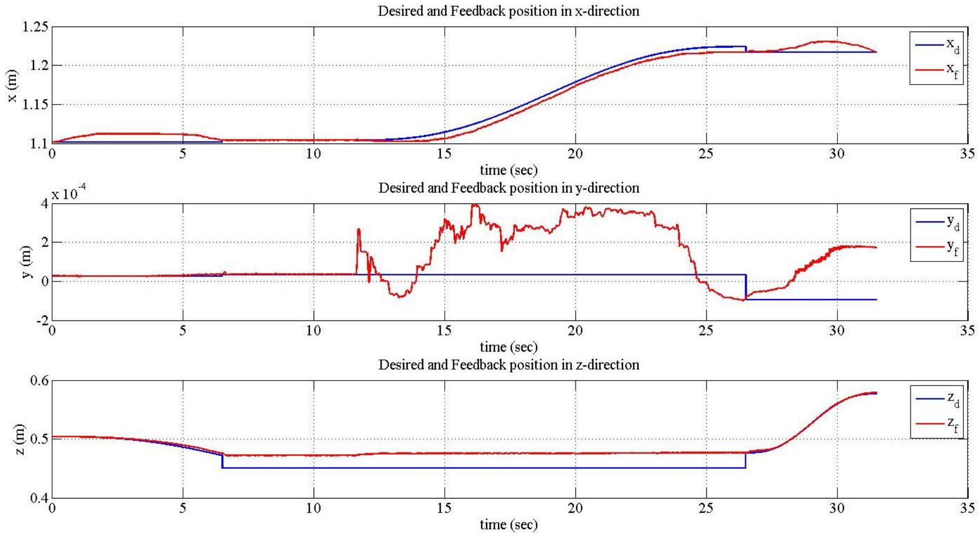


Figure A.9 : Desired and feedback positions.

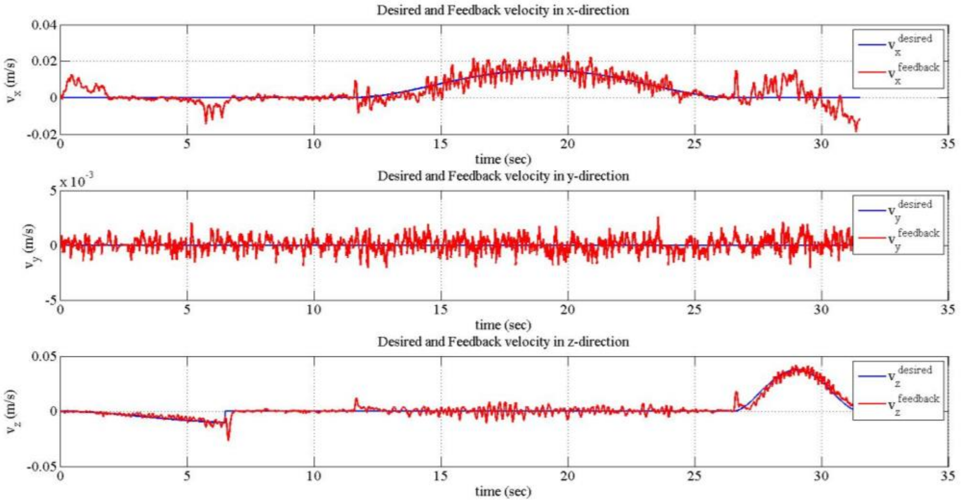


Figure A.10 : Desired and feedback velocities.

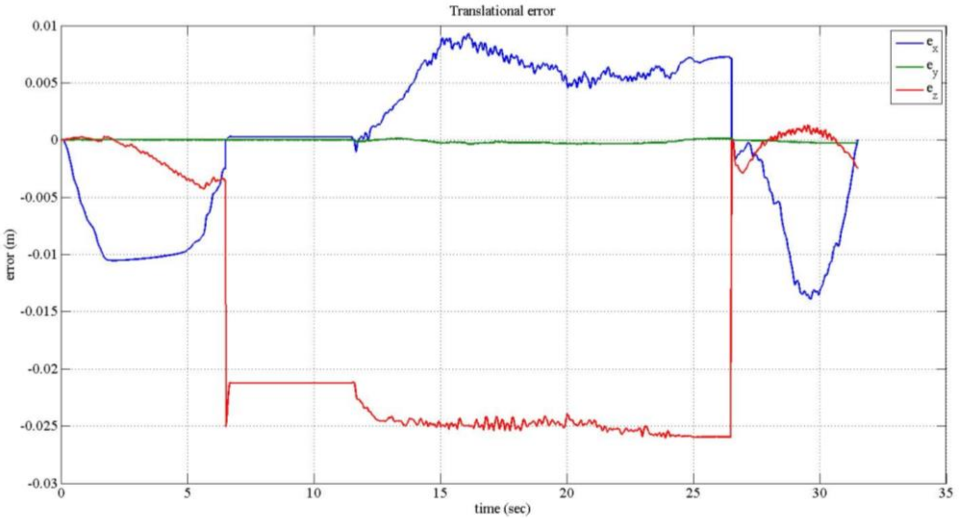


Figure A.11 : Translational errors.

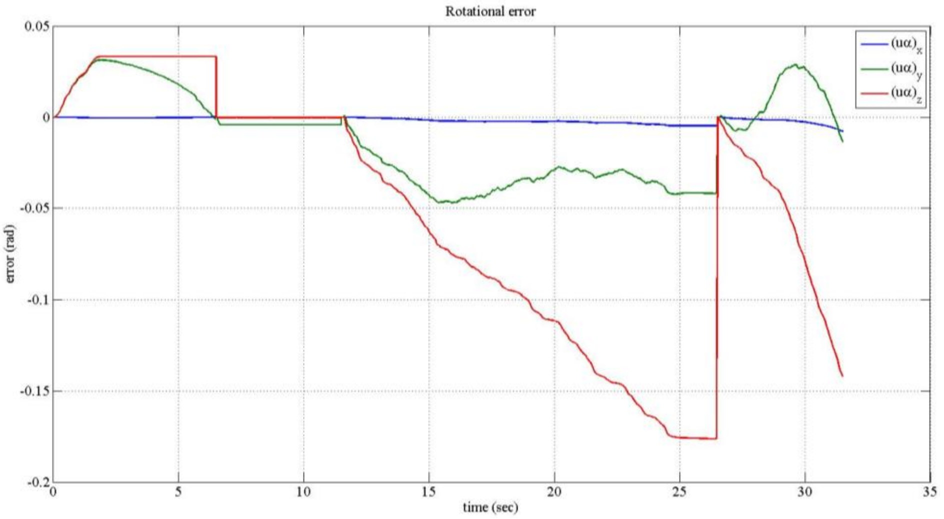


Figure A.12 : Rotational errors.

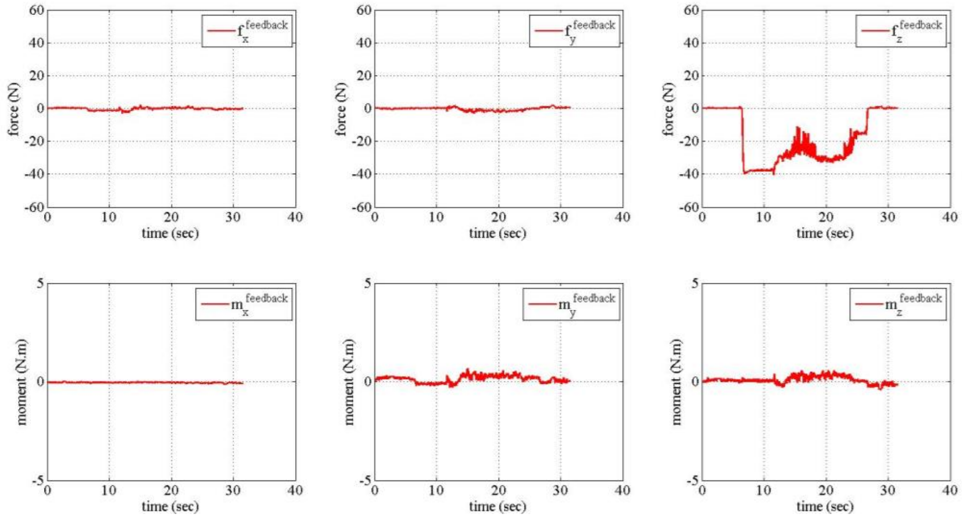


Figure A.13 : Feedback forces.

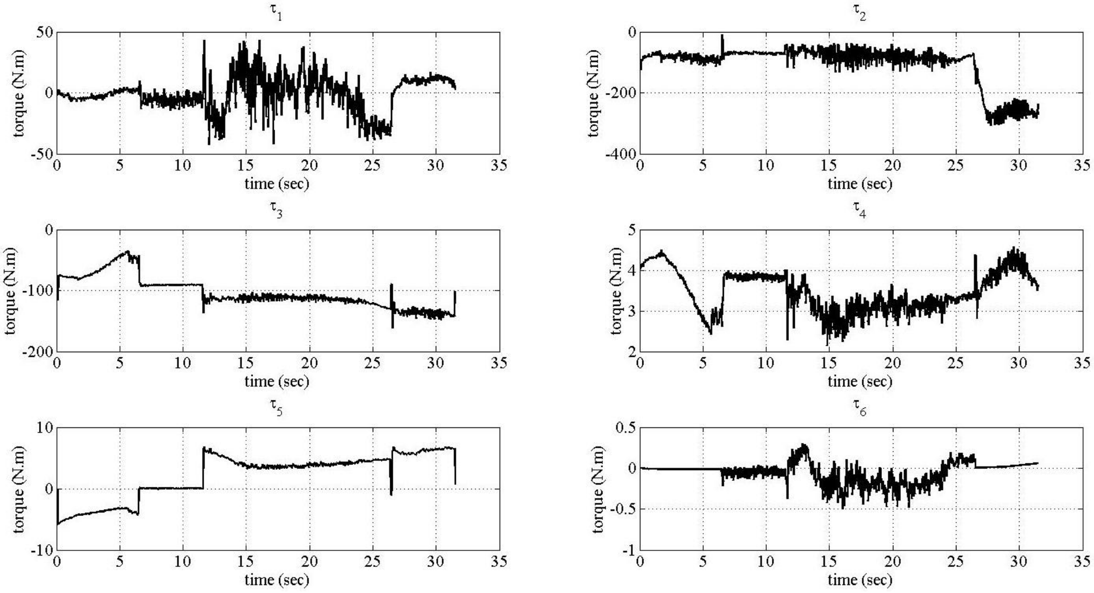


Figure A.14 : Applied joint torques.

Experiment 3:

- Draw a 120 mm long line in x-direction.
- Motion time is 15 seconds in impedance control phase.
- $M_d = diag(2, 2, 15, 1.5, 1.5, 1.5)$
- $K_d = diag(2000, 2000, 2500, 1000, 1000, 1000)$
- $B_d = diag(150, 150, 250, 100, 100, 100)$

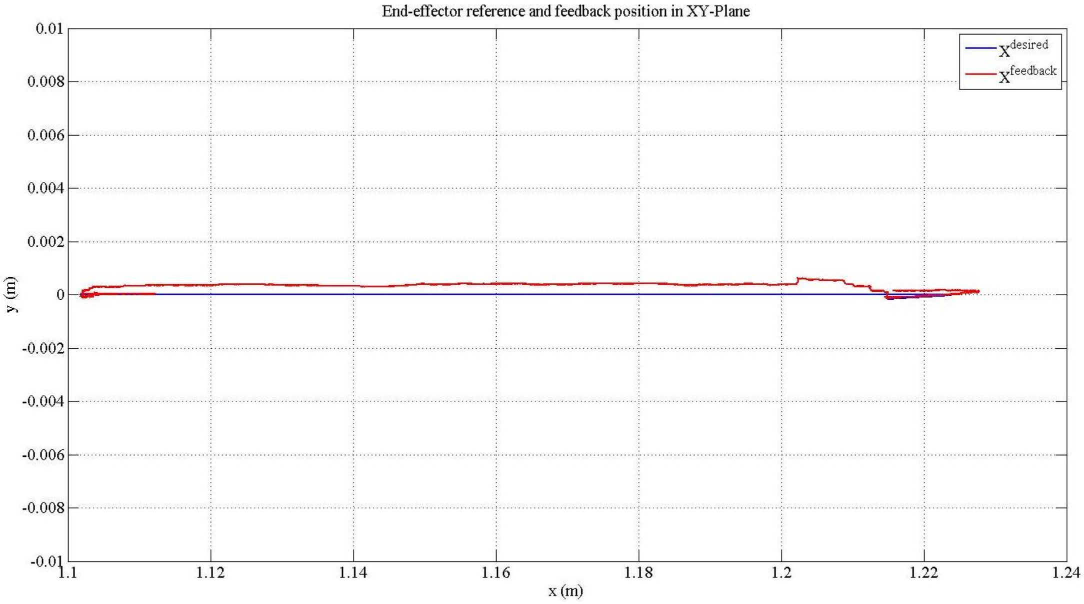


Figure A.15 : Desired and feedback path.

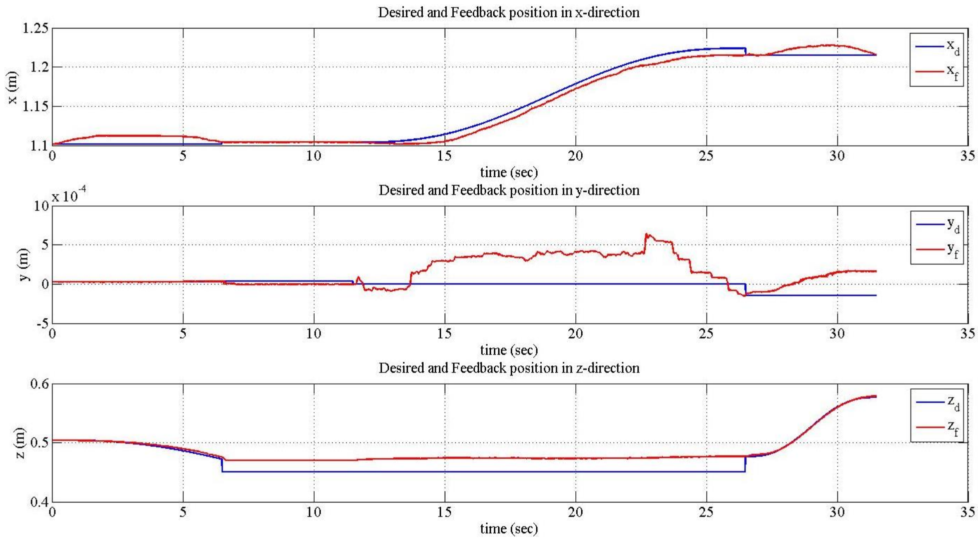


Figure A.16 : Desired and feedback positions.

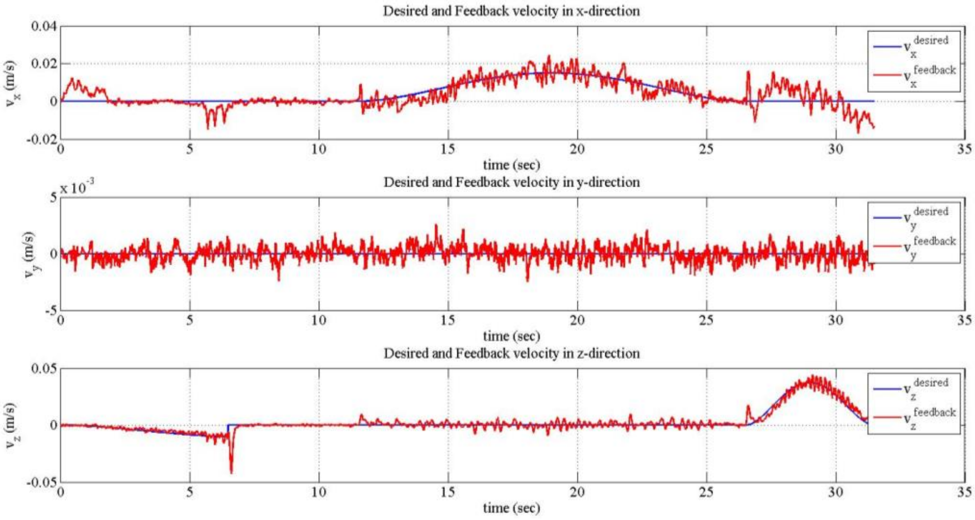


Figure A.17 : Desired and feedback velocities.

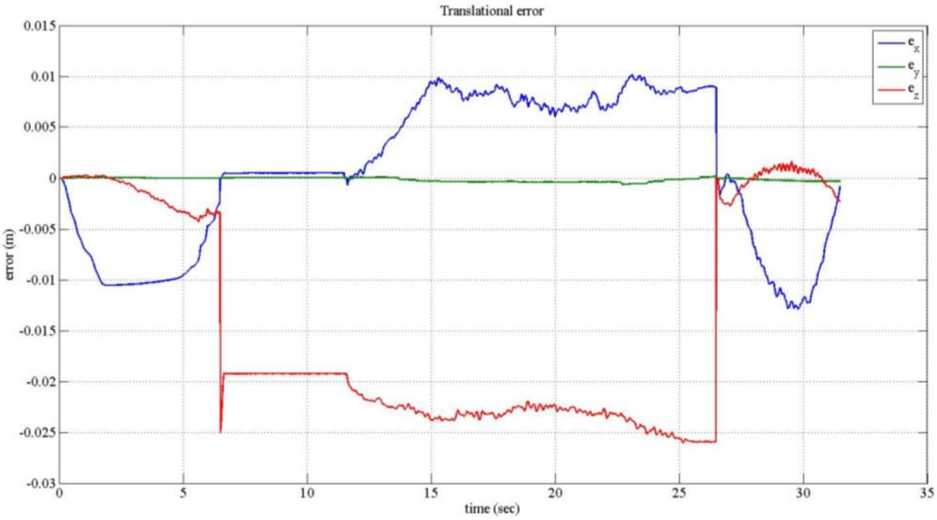


Figure A.18 : Translational errors.

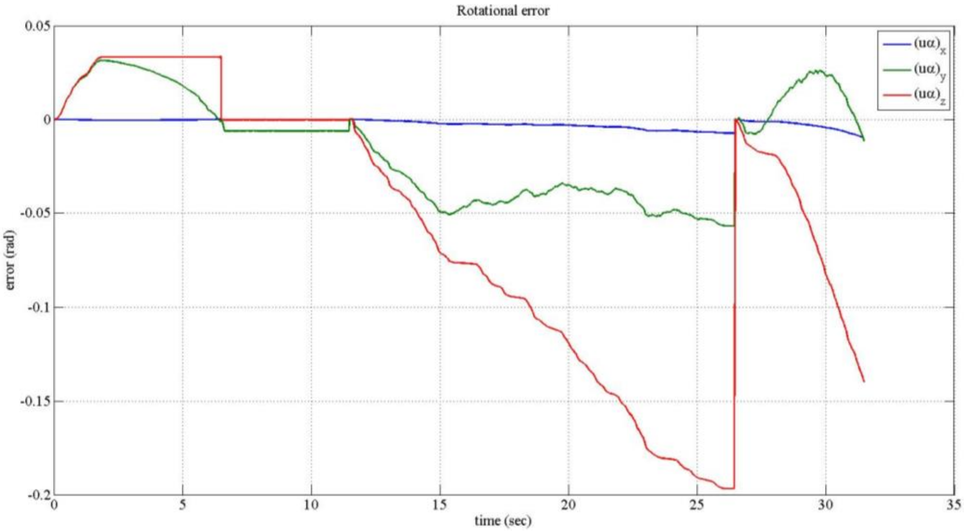


Figure A.19 : Rotational errors.

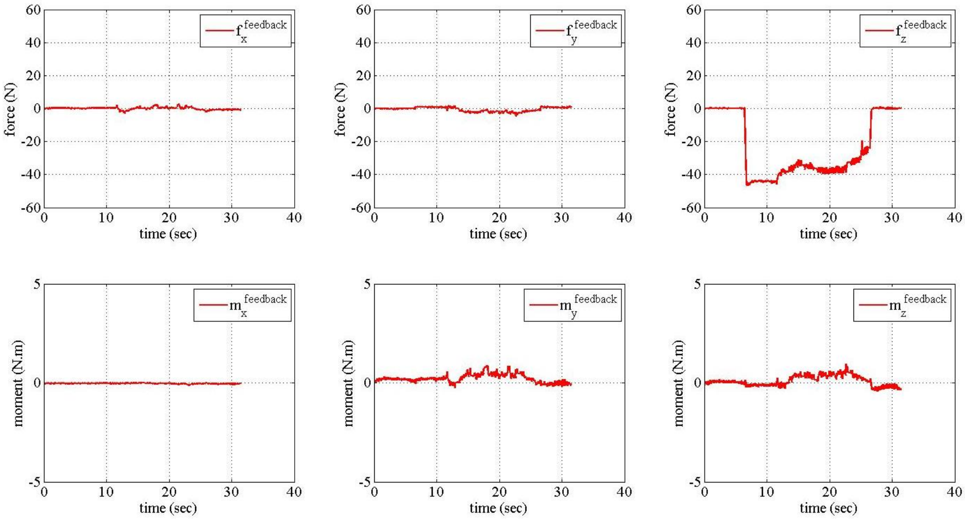


Figure A.20 : Feedback forces.

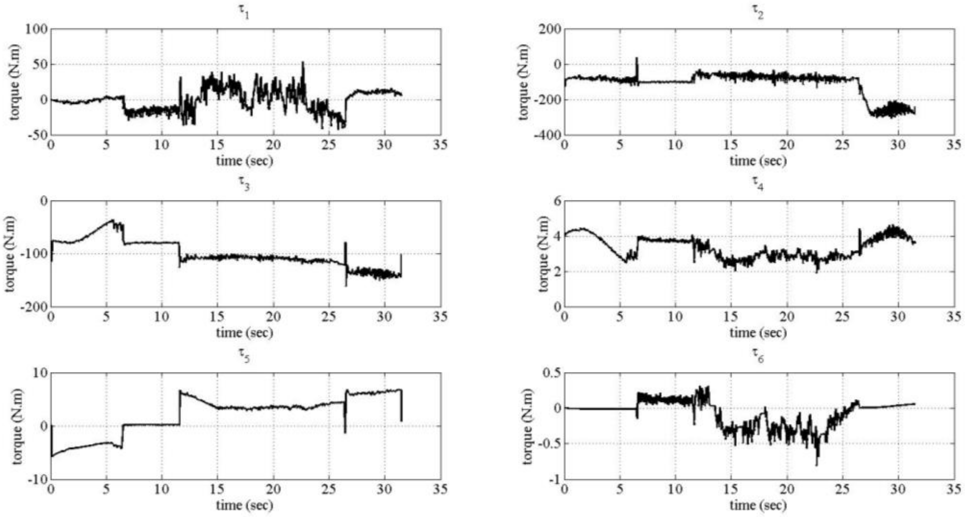


Figure A.21 : Applied joint torques.

Experiment 4:

- Draw a 120 mm long line in x-direction.
- Motion time is 15 seconds in impedance control phase.
- $M_d = diag(2, 2, 12.5, 1.5, 1.5, 1.5)$
- $K_d = diag(2000, 2000, 2500, 1000, 1000, 1000)$
- $B_d = diag(150, 150, 250, 100, 100, 100)$

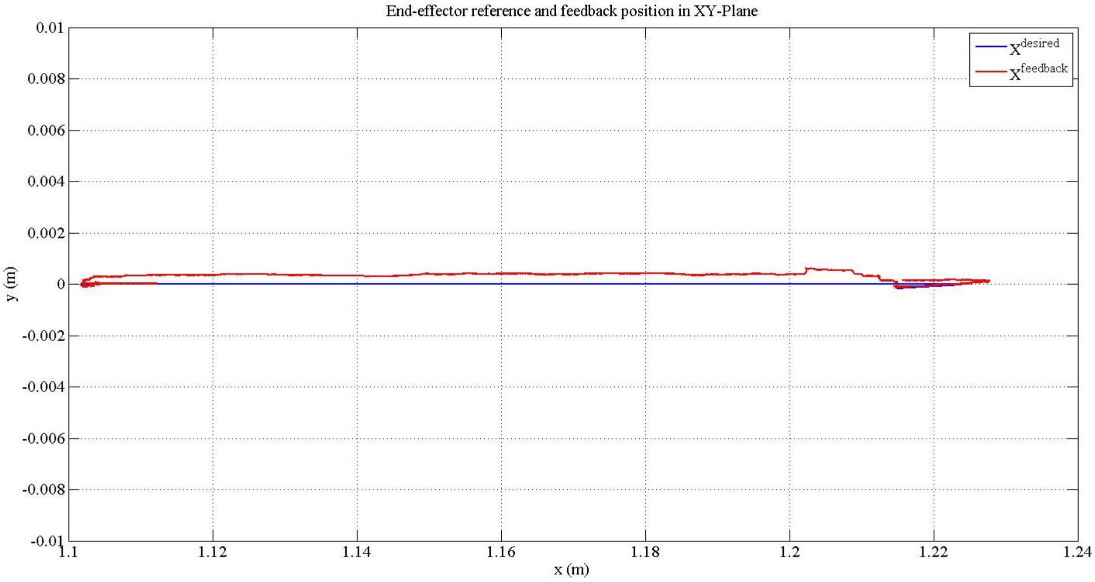


Figure A.22 : Desired and feedback path.

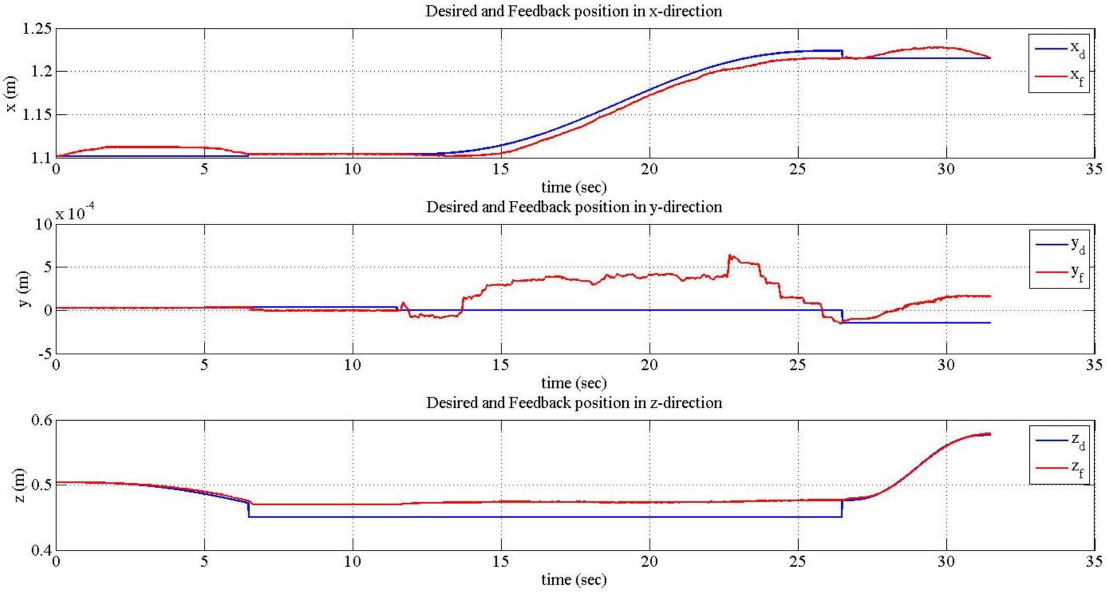


Figure A.23 : Desired and feedback positions.

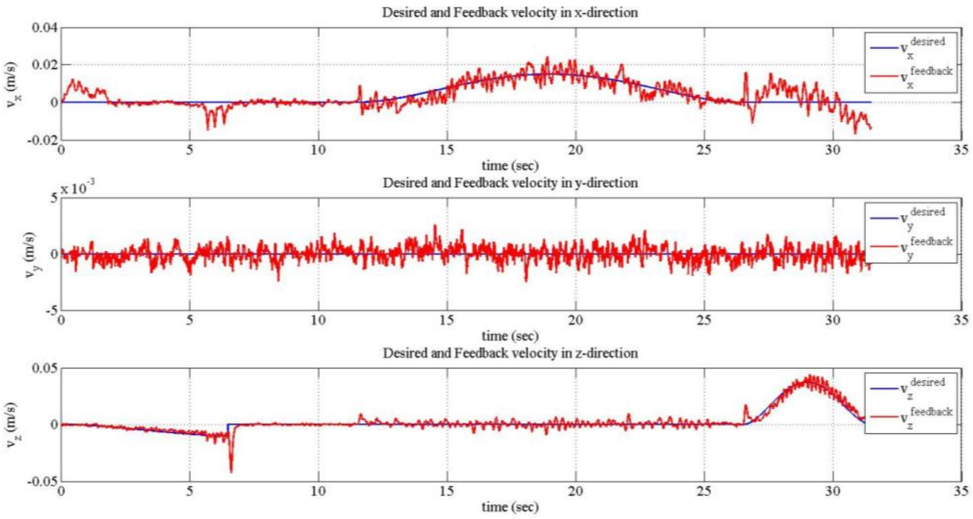


Figure A.24 : Desired and feedback velocities.

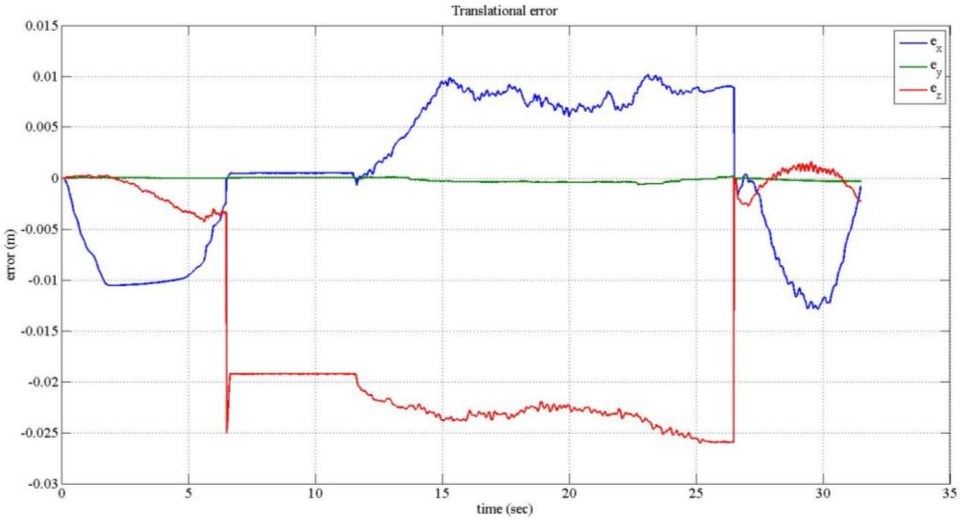


Figure A.25 : Translational errors.

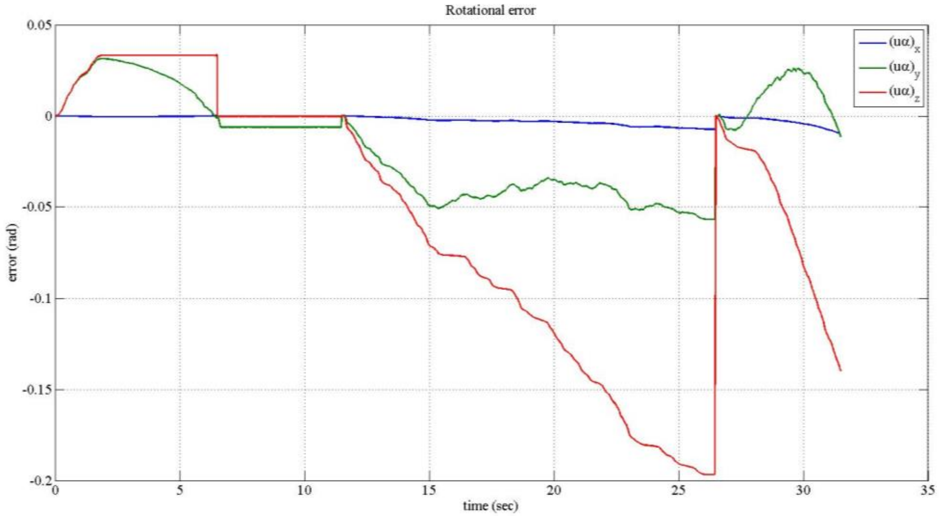


Figure A.26 : Rotational errors.

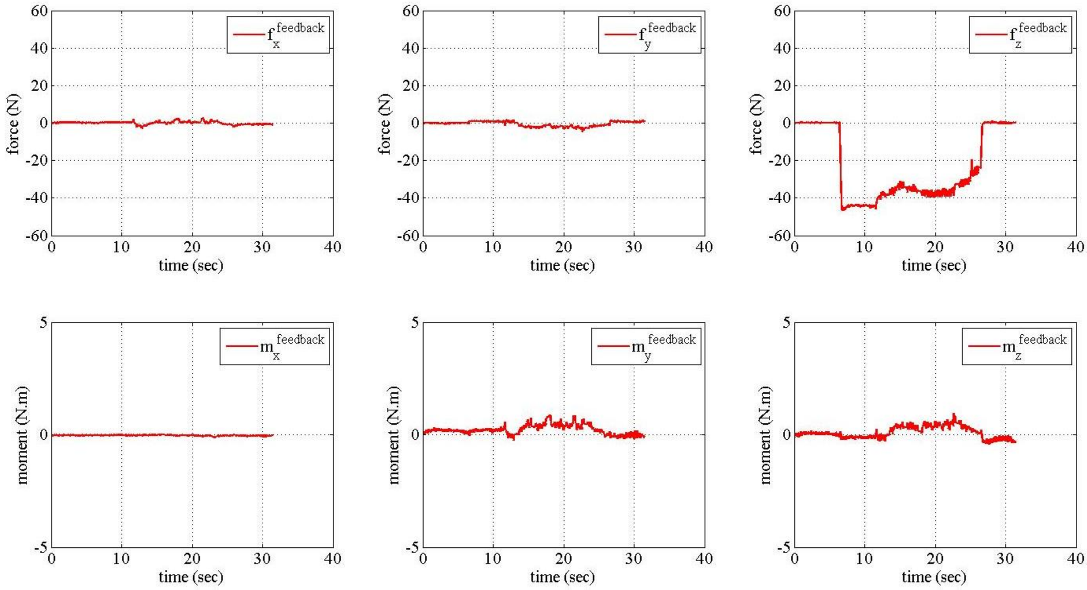


Figure A.27 : Feedback forces.

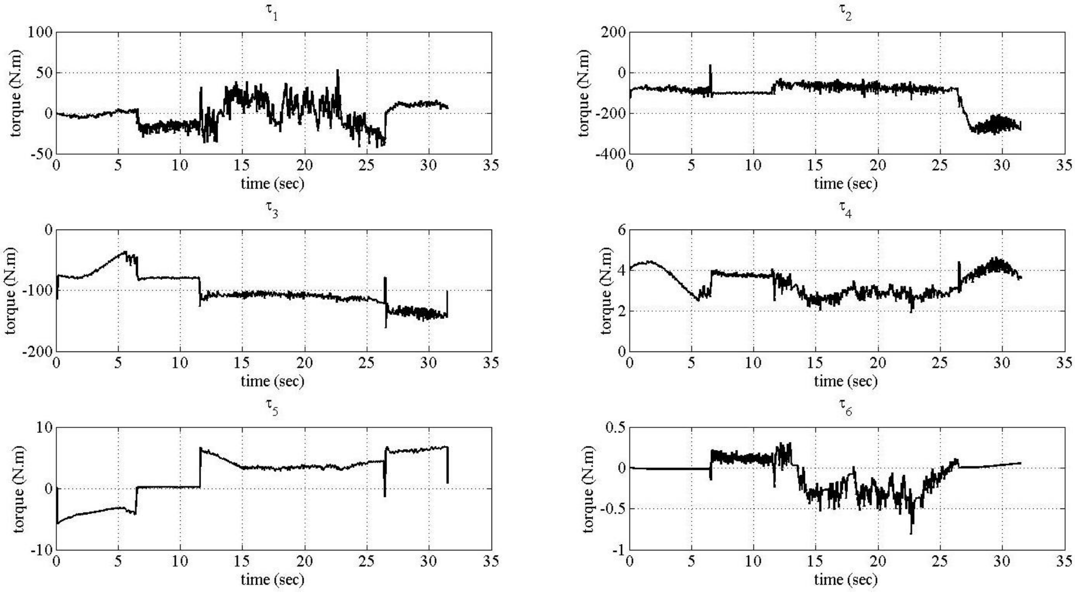


Figure A.28 : Applied joint torques.

Experiment 5:

- Draw a 120 mm long line in x-direction.
- Motion time is 15 seconds in impedance control phase.
- $M_d = diag(2, 2, 10, 1.5, 1.5, 1.5)$
- $K_d = diag(2000, 2000, 2500, 1000, 1000, 1000)$
- $B_d = diag(150, 150, 250, 100, 100, 100)$

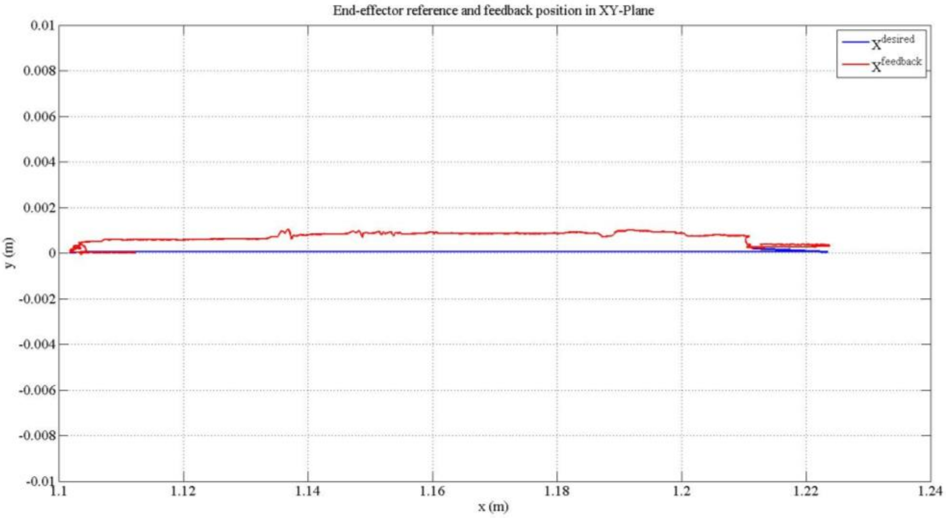


Figure A.29 : Desired and feedback path.

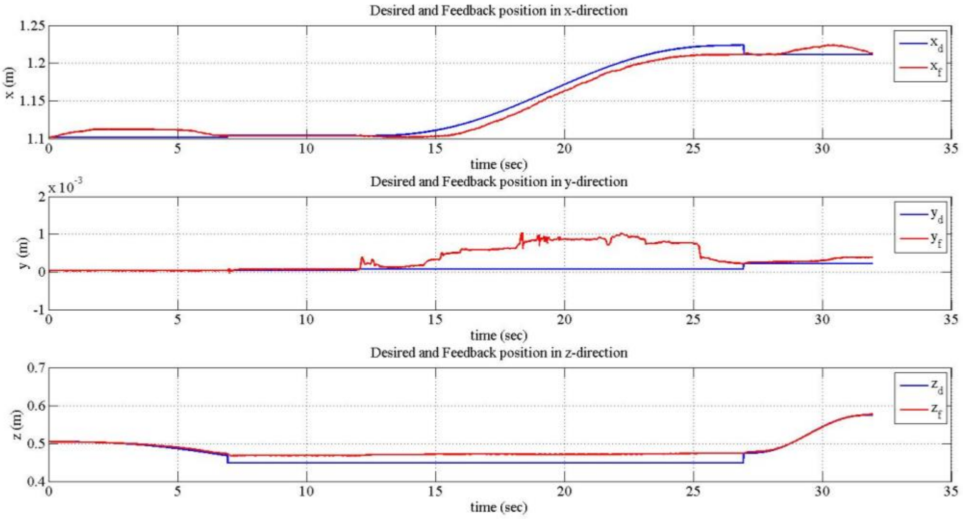


Figure A.30 : Desired and feedback positions.

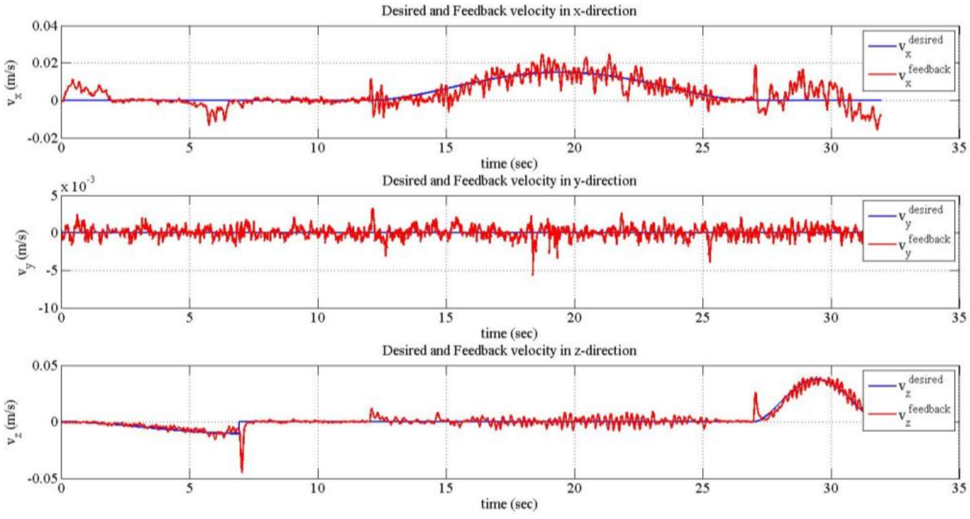


Figure A.31 : Desired and feedback velocities.

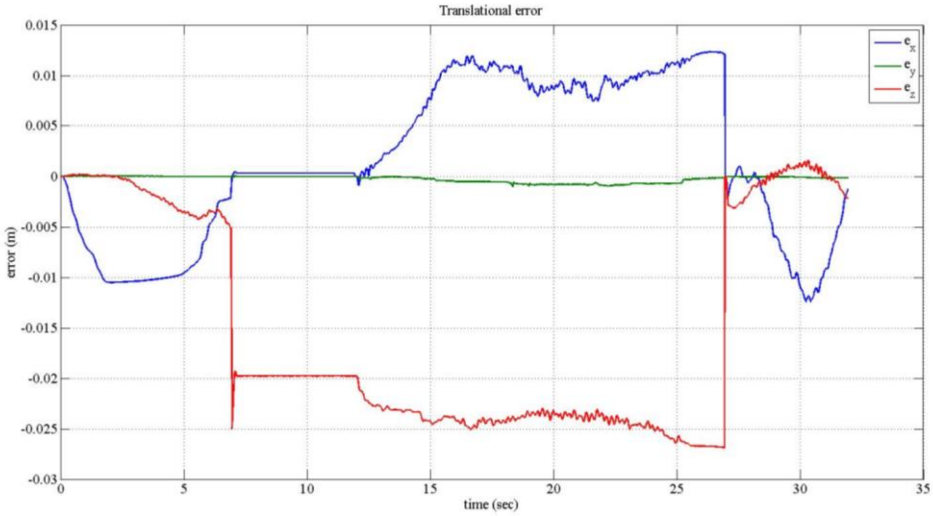


Figure A.32 : Translational errors.

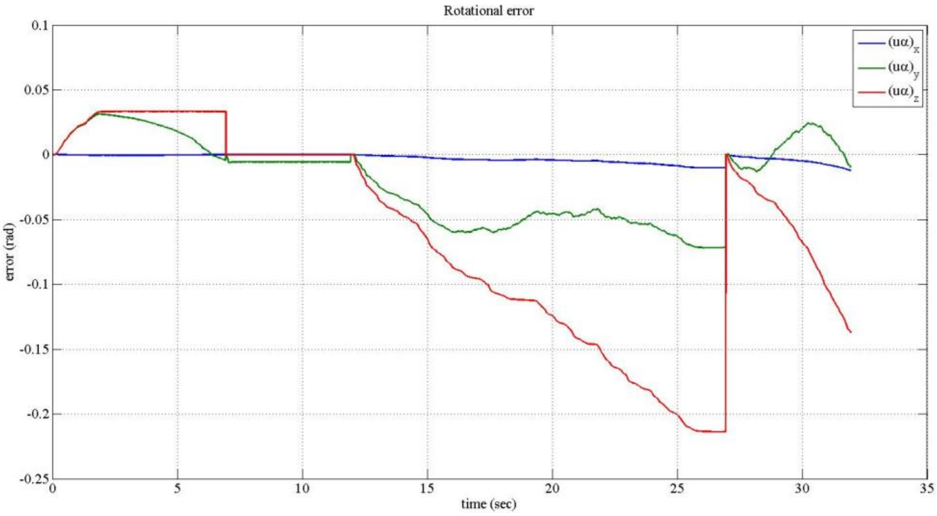


Figure A.33 : Rotational errors.

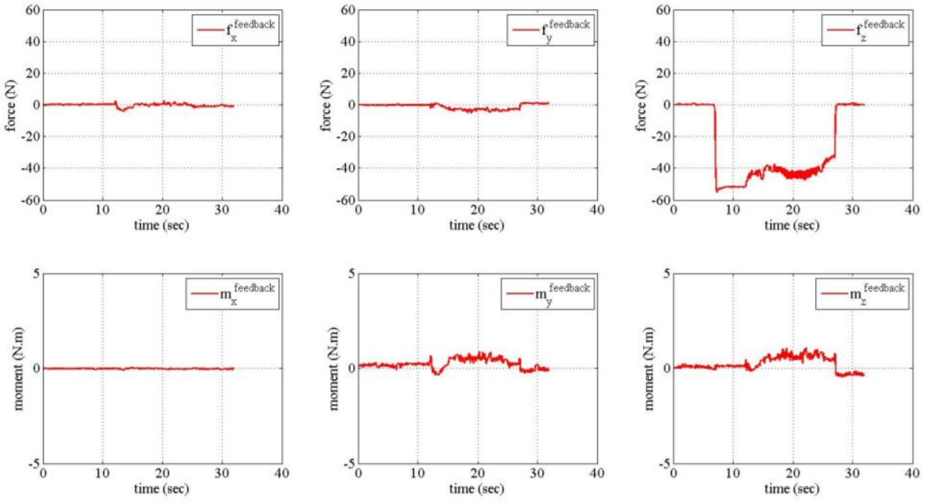


Figure A.34 : Feedback forces.

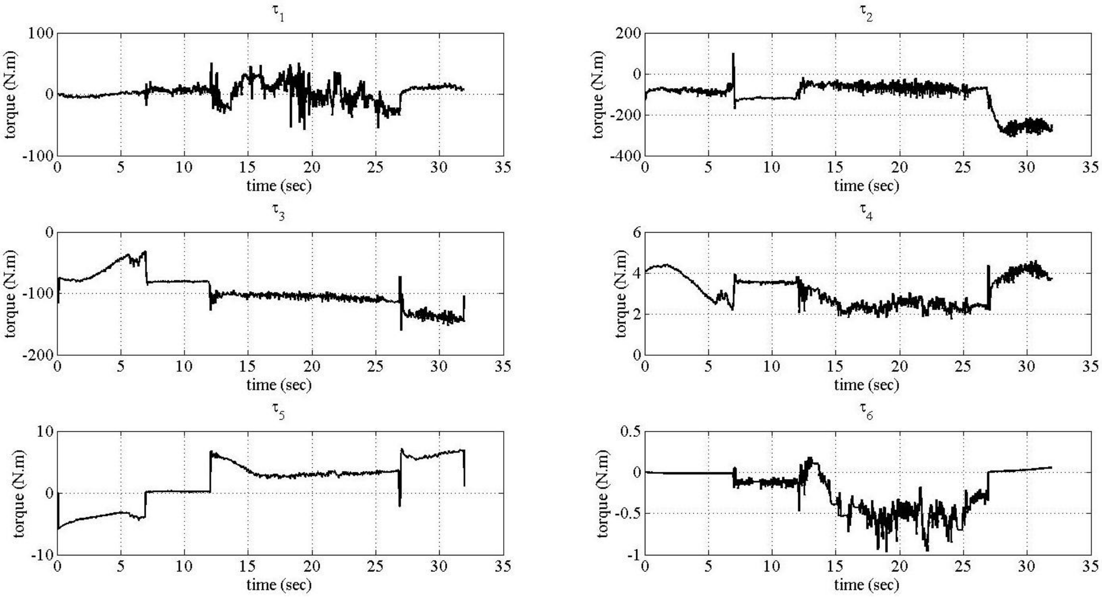


Figure A.35 : Applied joint torques.

APPENDIX B

In this section, further experimental results for hybrid force/position control are given.

Experiment 1:

- Circle in XY-plane.
- Radius of the circle is 75 mm.
- Constant -30 N in z-direction.
- Motion time is 20 seconds in compliant motion phase.
- $K_{fp} = diag(3.5, 3.5, 3.5, 3.5, 3.5, 3.5)$
- $K_{fd} = diag(300, 300, 300, 300, 300, 300)$
- $K_{fI} = diag(2, 2, 2, 2, 2, 2)$

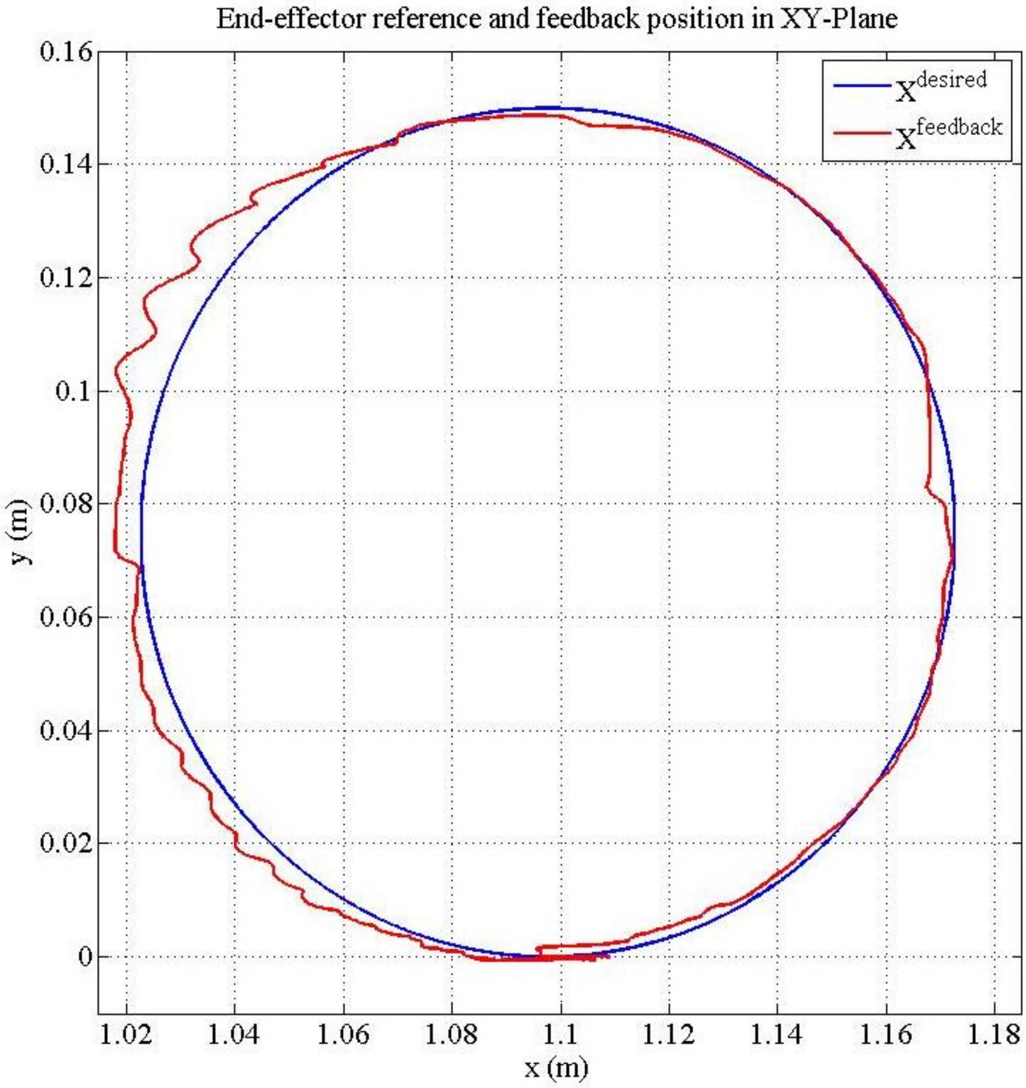


Figure B.1 : Desired and feedback path.

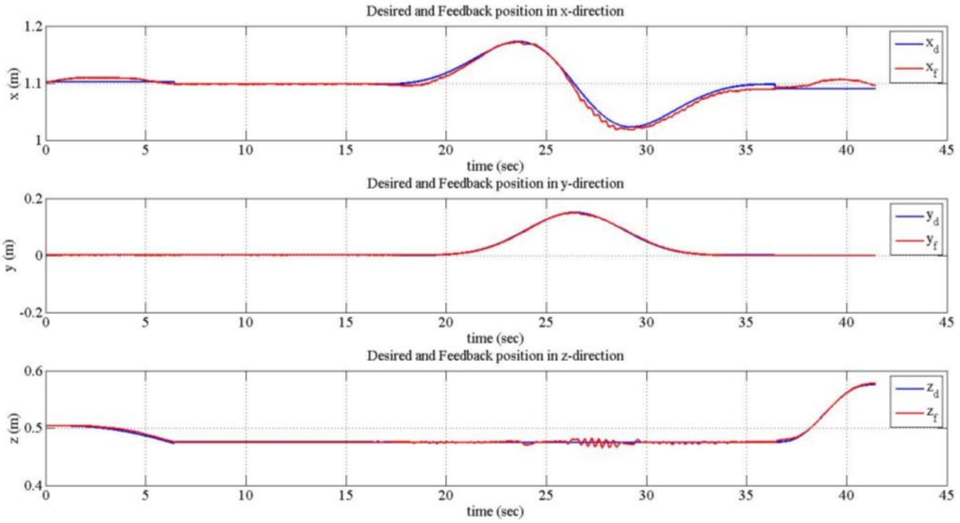


Figure B.2 : Desired and feedback positions.

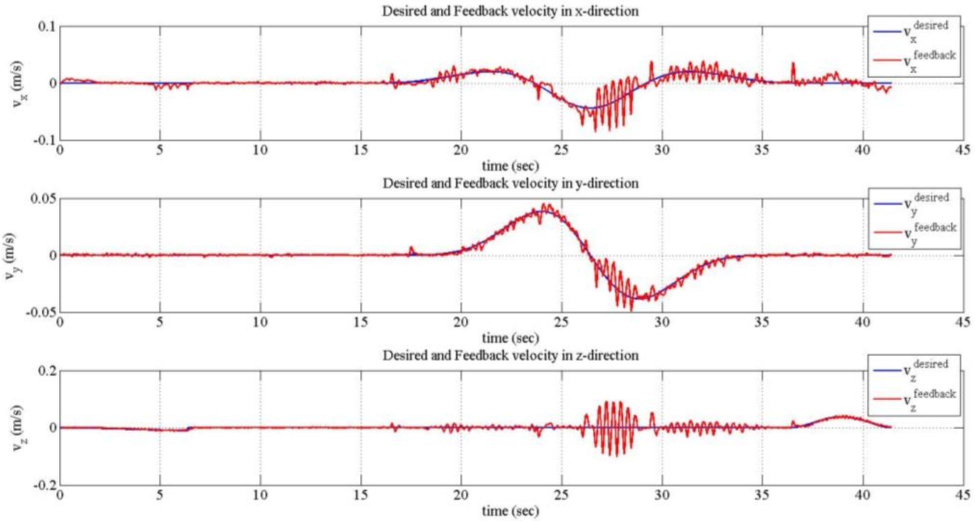


Figure B.3 : Desired and feedback velocities.

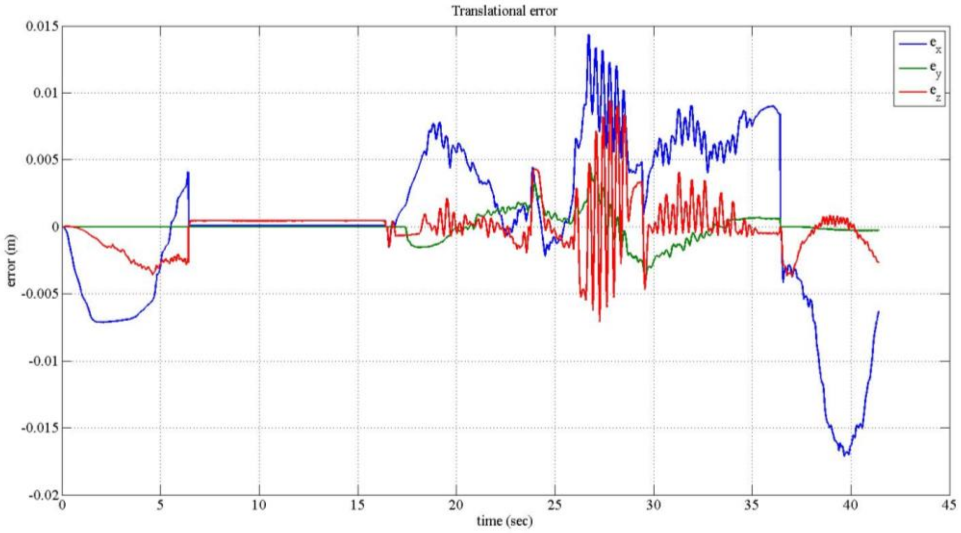


Figure B.4 : Translational errors.

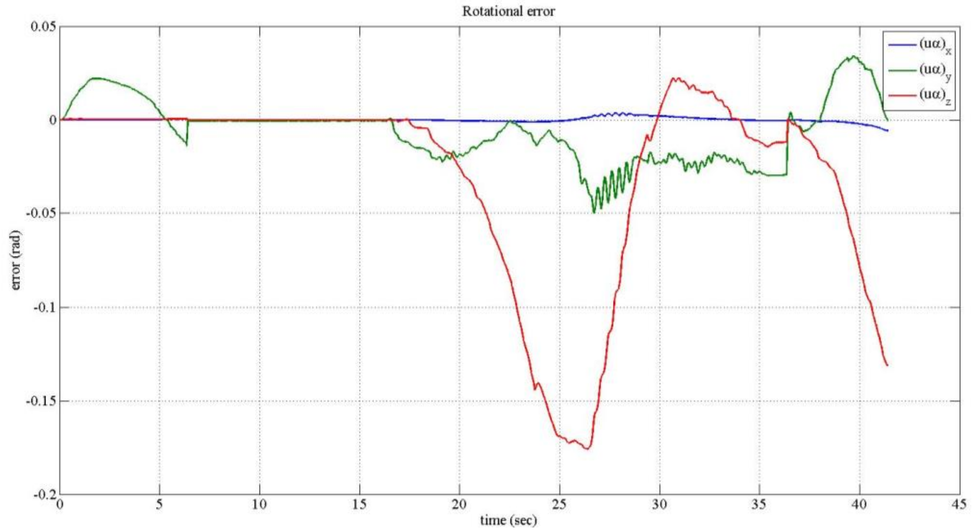


Figure B.5 : Rotational errors.

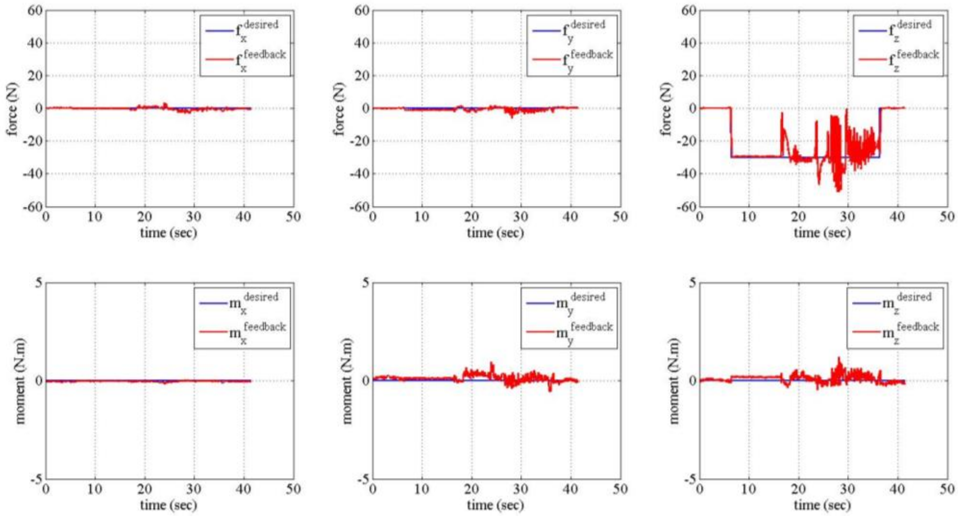


Figure B.6 : Desired and feedback forces.

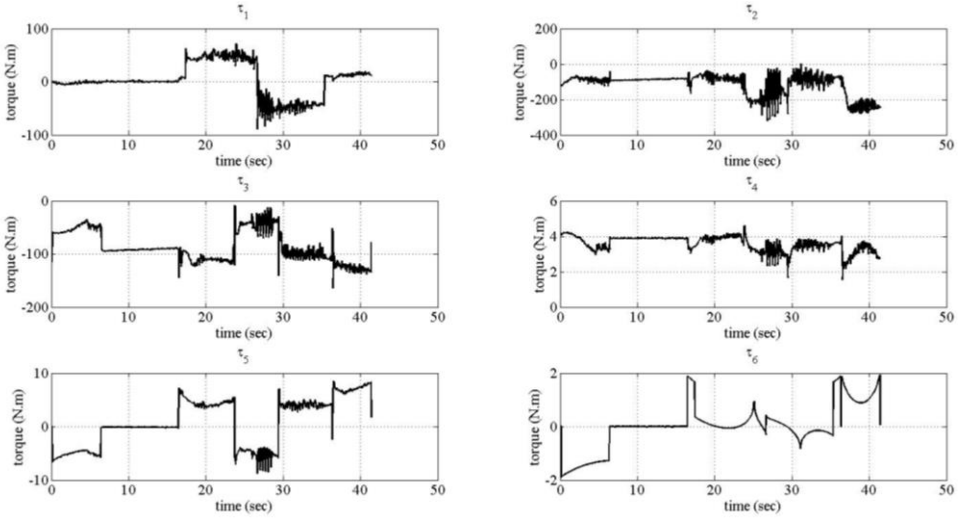


Figure B.7 : Applied joint torques.

Experiment 2:

- Circle in XY-plane
- Radius of the circle is 75 mm.
- Constant -35 N in z-direction.
- Motion time is 20 seconds in compliant motion phase.
- $K_{fp} = diag(3.5, 3.5, 3.5, 3.5, 3.5, 3.5)$
- $K_{fd} = diag(300, 300, 300, 300, 300, 300)$
- $K_{fI} = diag(2, 2, 2, 2, 2, 2)$

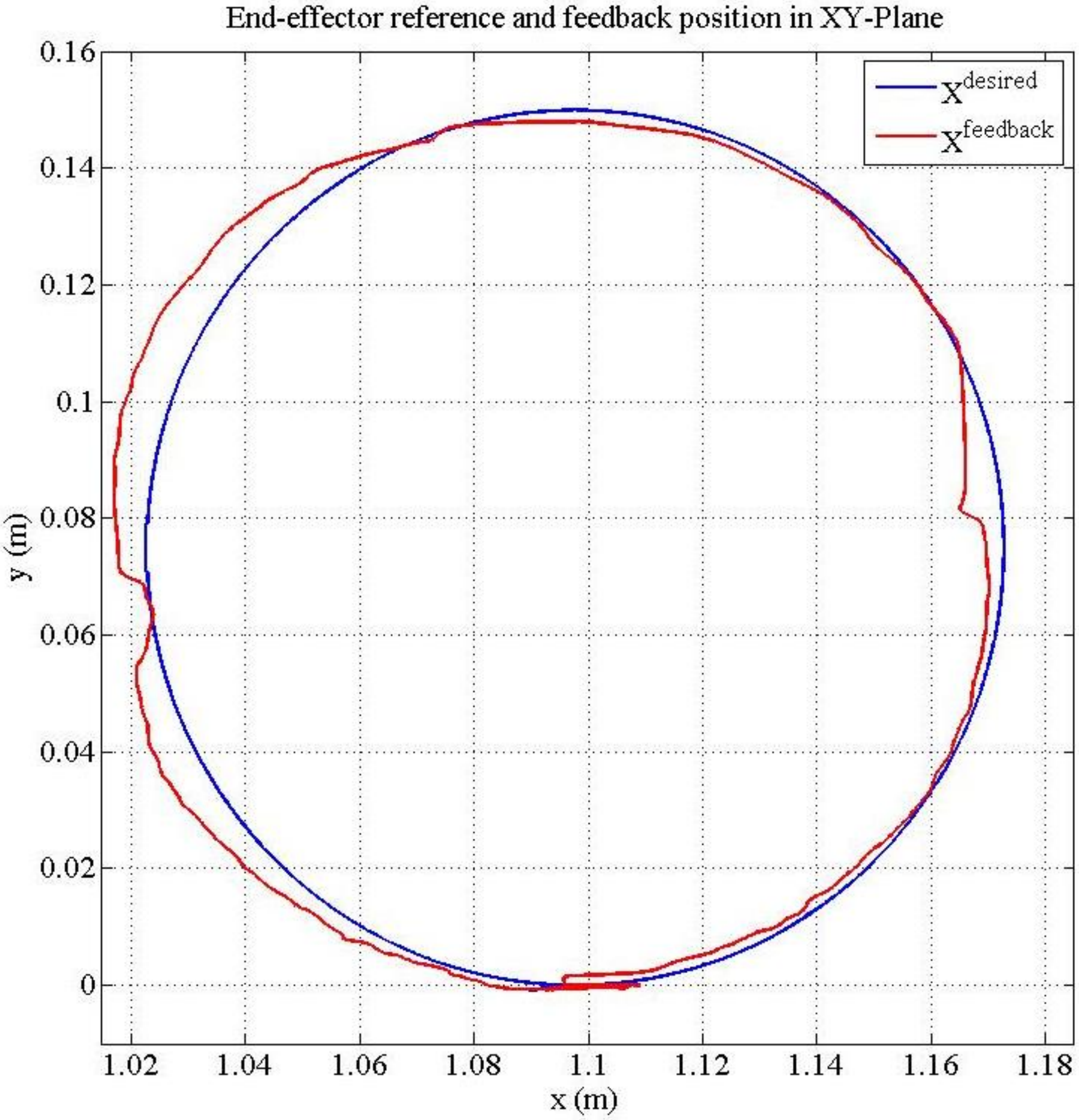


Figure B.8 : Desired and feedback path.

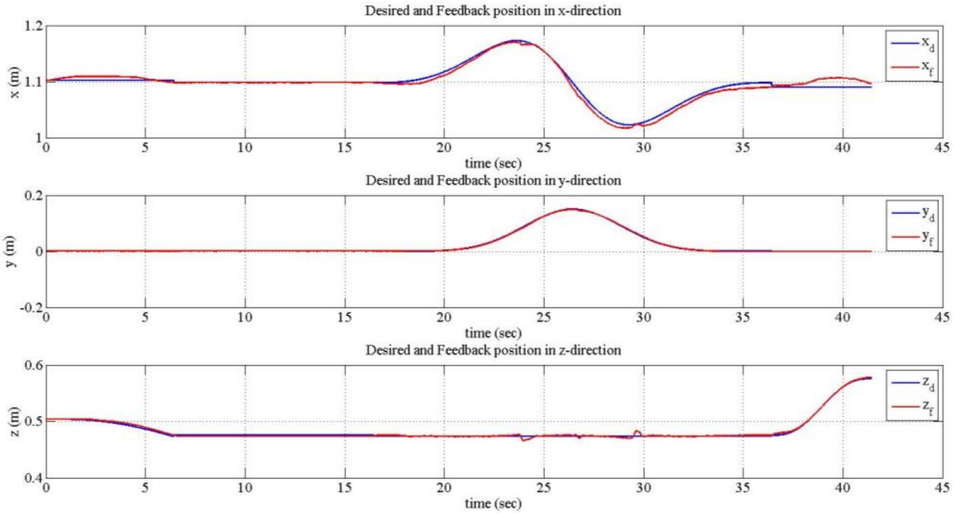


Figure B.9 : Desired and feedback positions.

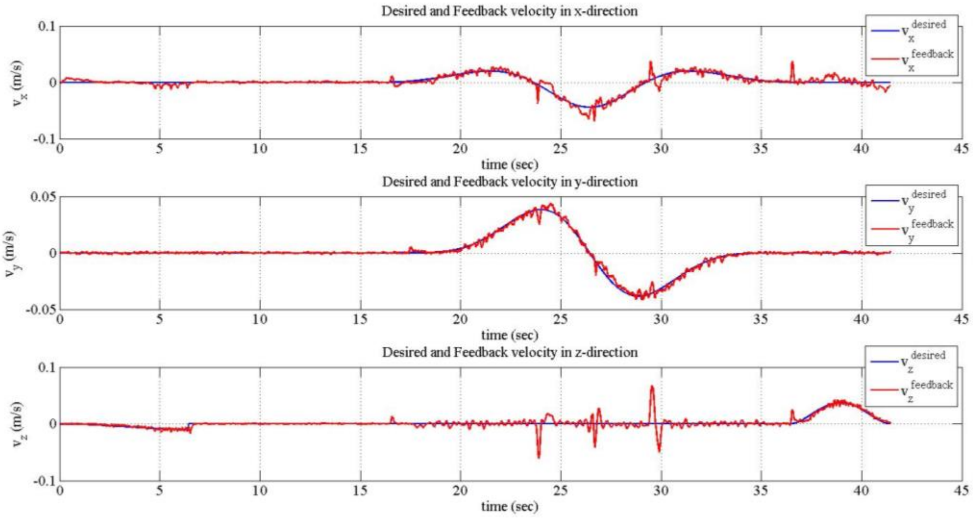


Figure B.10 : Desired and feedback velocities.

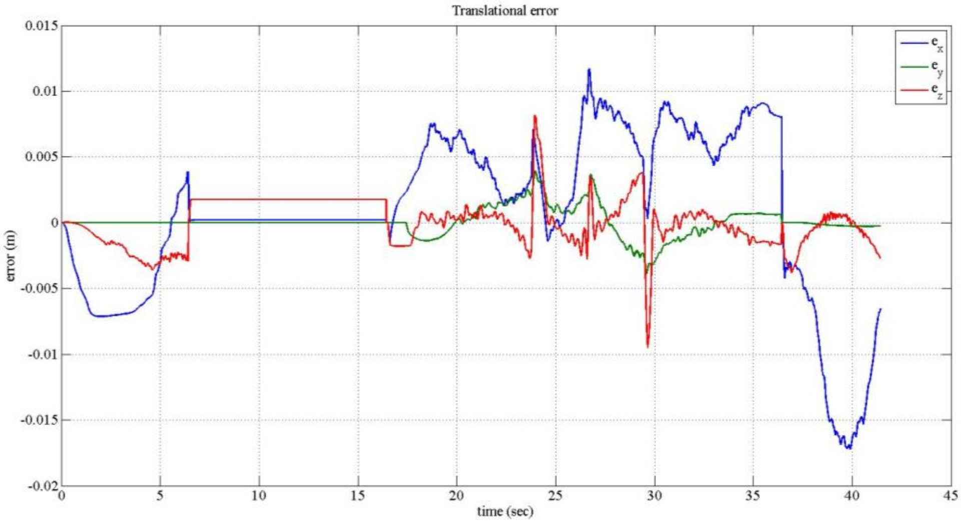


Figure B.11 : Translational errors.

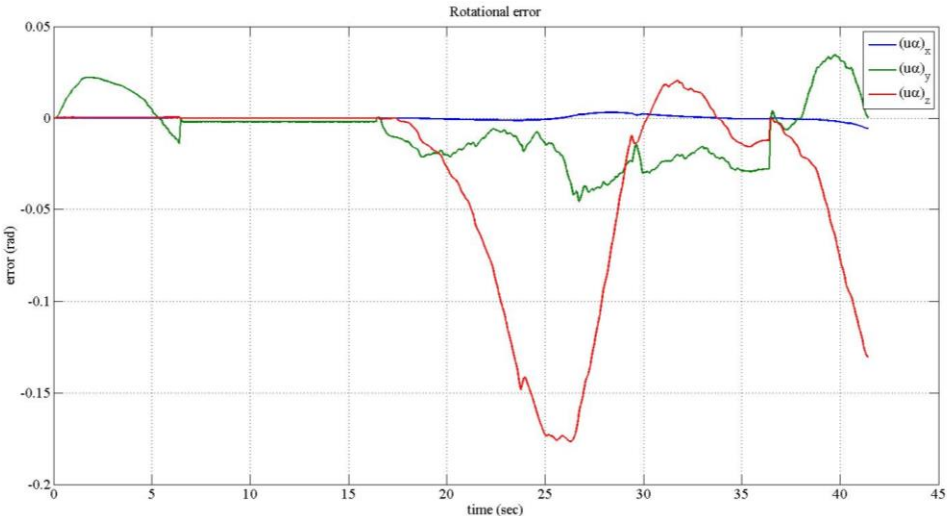


Figure B.12 : Rotational errors.

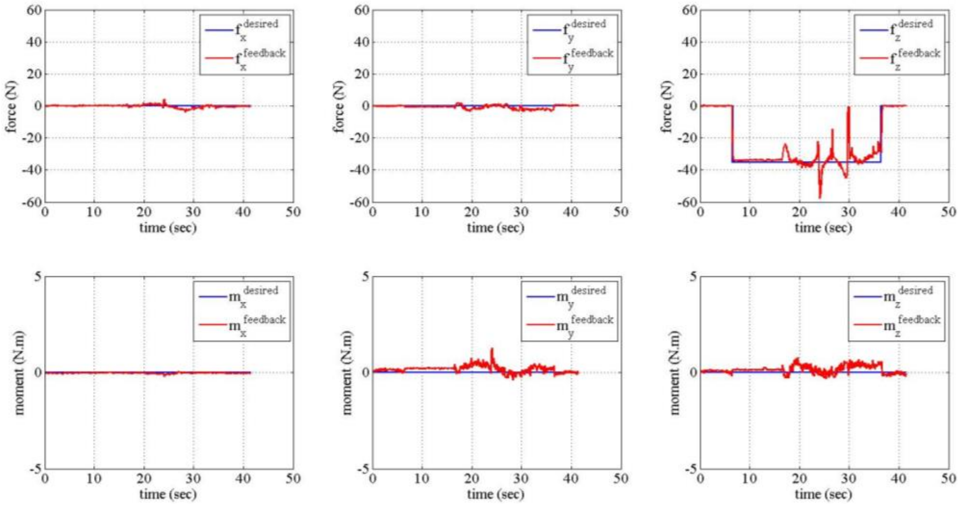


Figure B.13 : Desired and feedback forces.

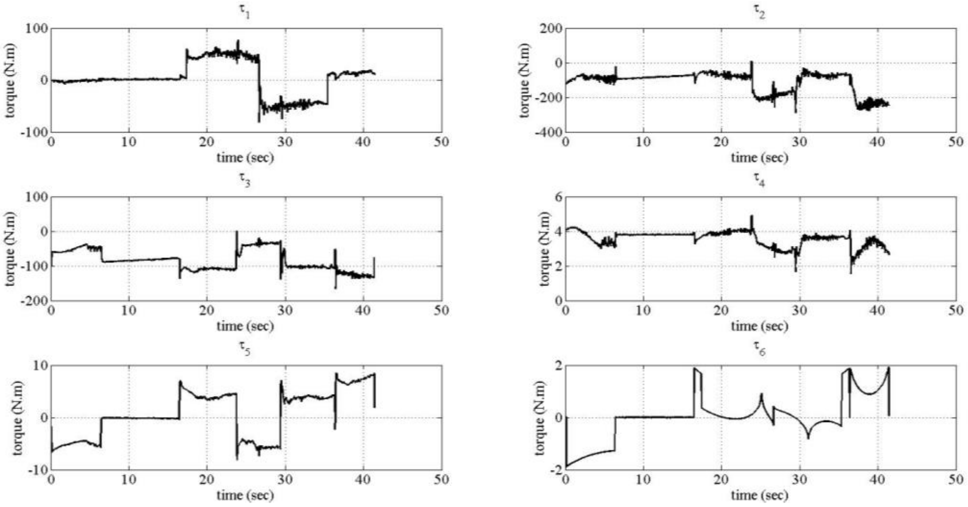


Figure B.14 : Applied joint torques.

Experiment 3:

- Circle in XY-plane
- Radius of the circle is 75 mm.
- Constant -40 N in z-direction.
- Motion time is 20 seconds in compliant motion phase.
- $K_{fp} = diag(3.5, 3.5, 3.5, 3.5, 3.5, 3.5)$
- $K_{fd} = diag(300, 300, 300, 300, 300, 300)$
- $K_{fI} = diag(2, 2, 2, 2, 2, 2)$

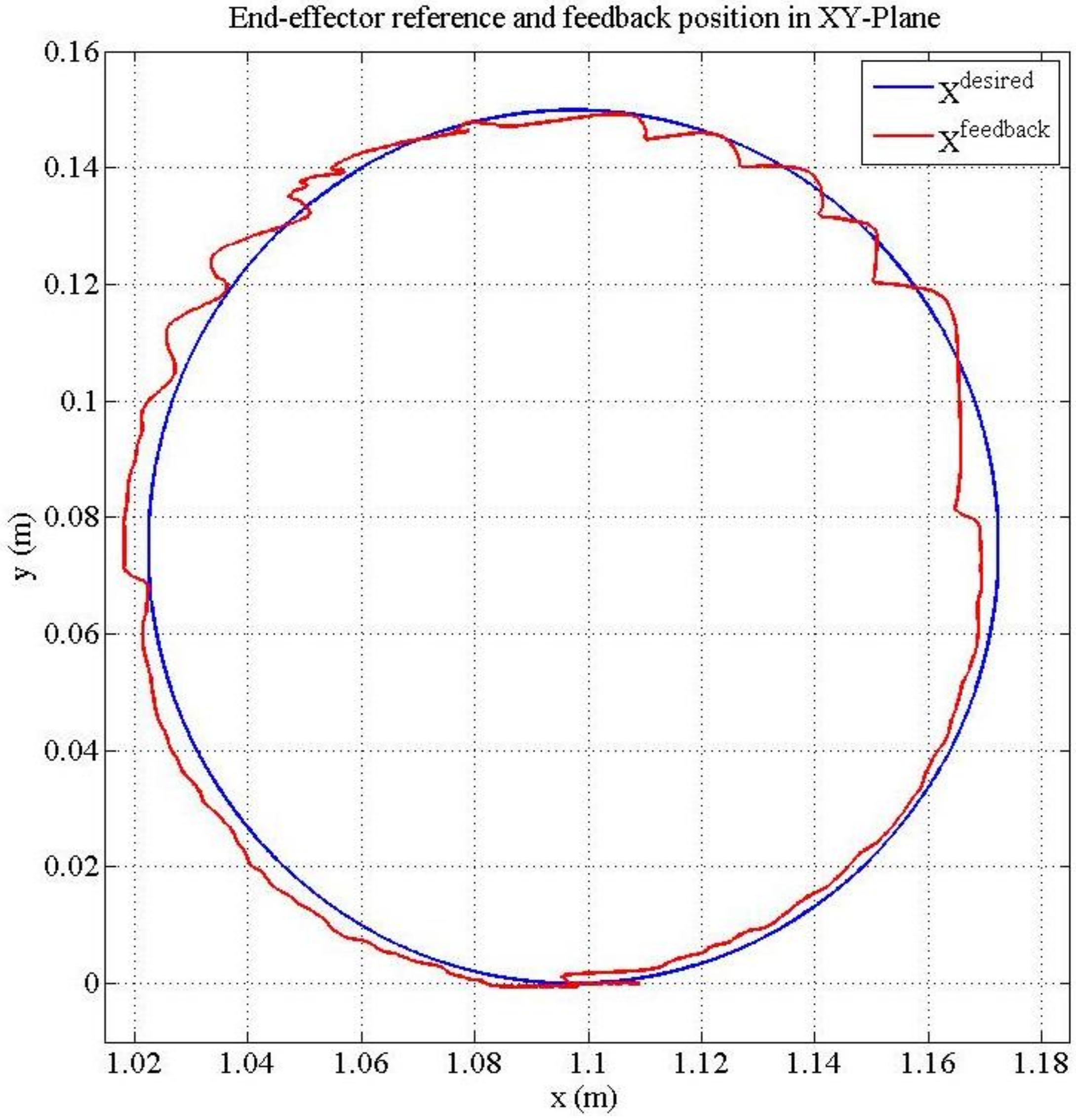


Figure B.15 : Desired and feedback path.

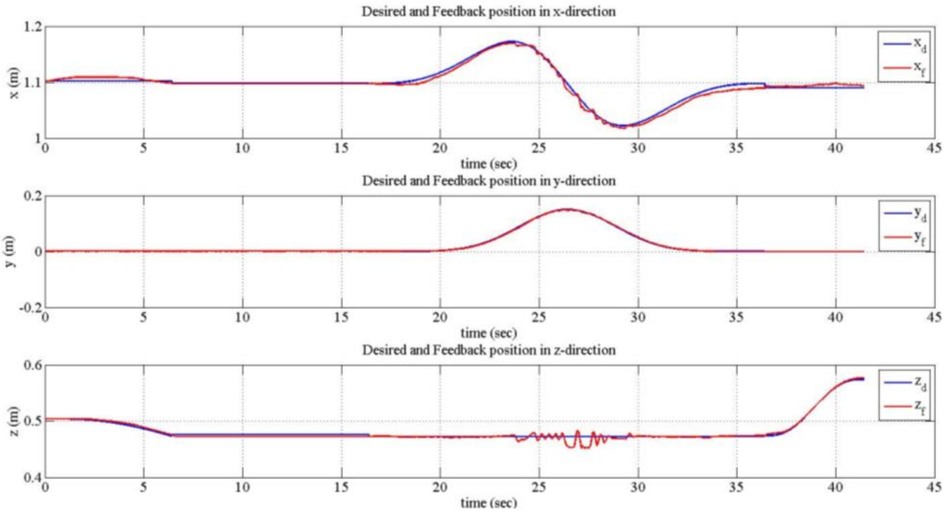


Figure B.16 : Desired and feedback positions.

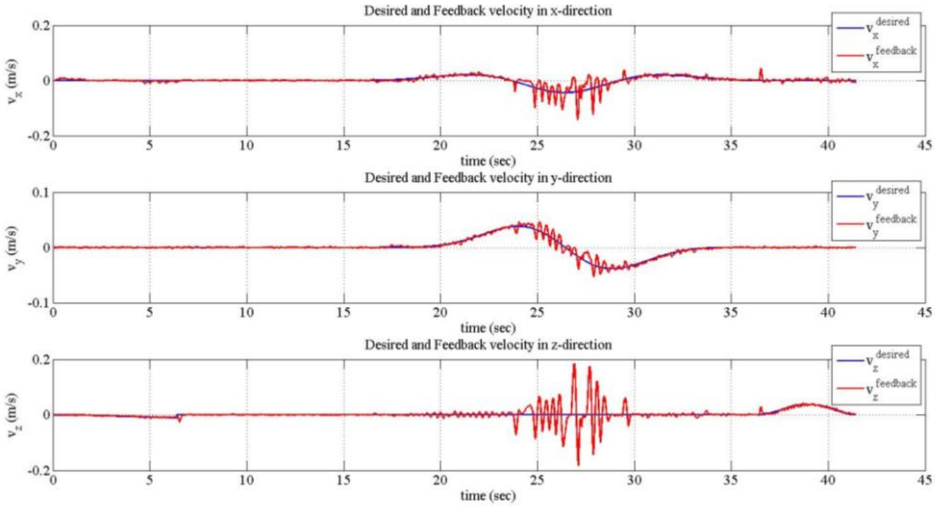


Figure B.17 : Desired and feedback velocities.

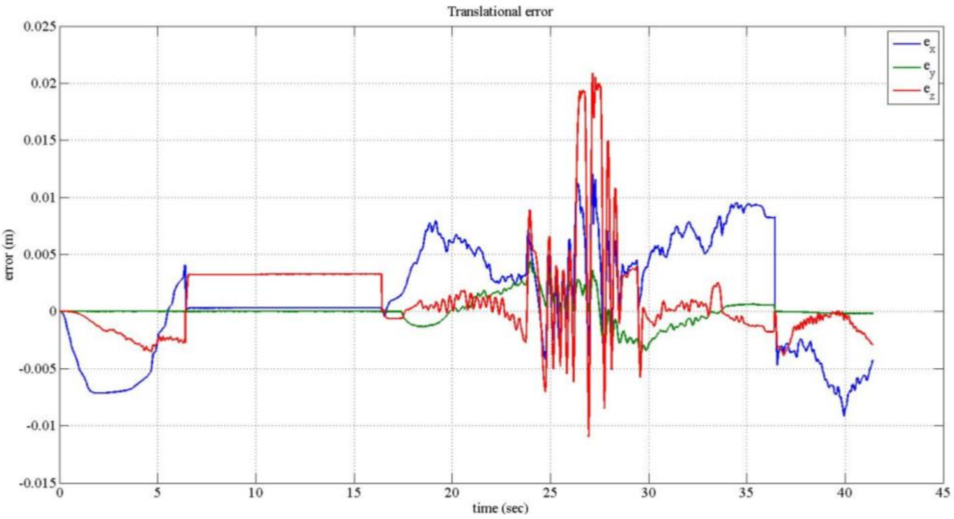


Figure B.18 : Translational errors.

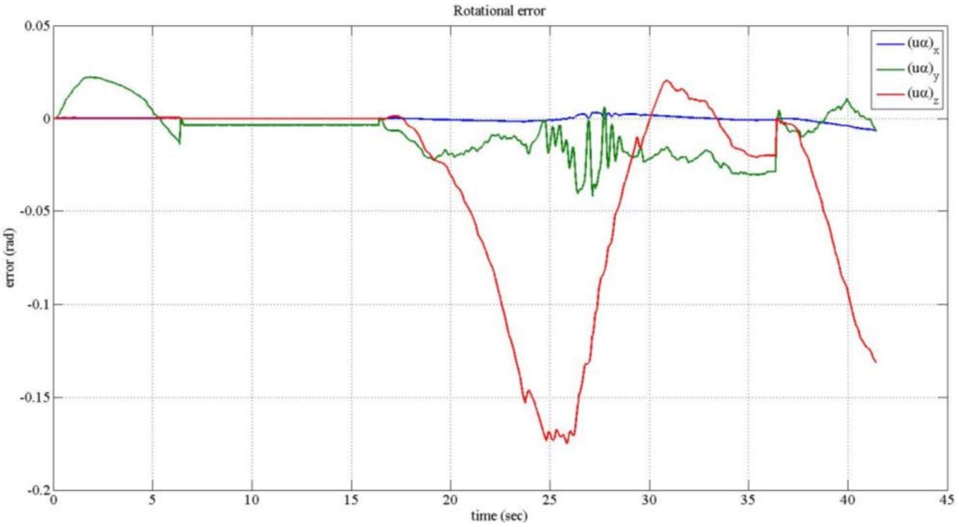


Figure B.19 : Rotational errors.

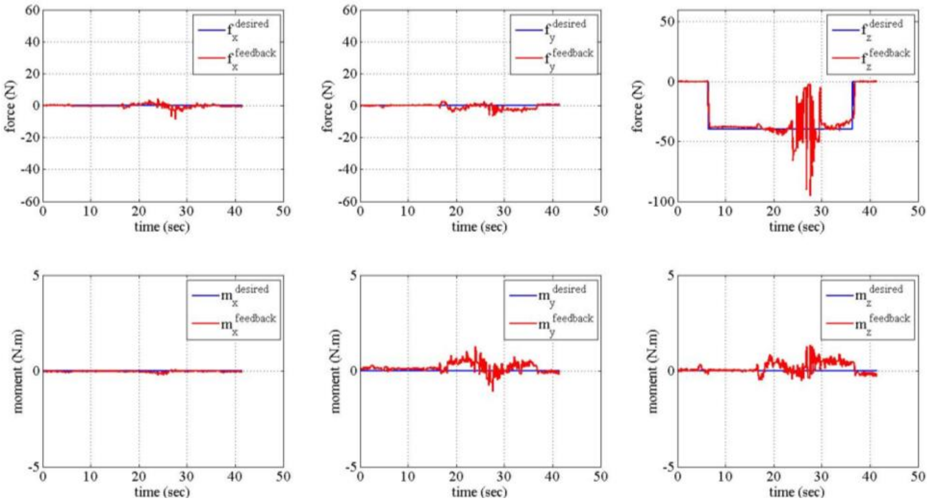


Figure B.20 : Desired and feedback forces.

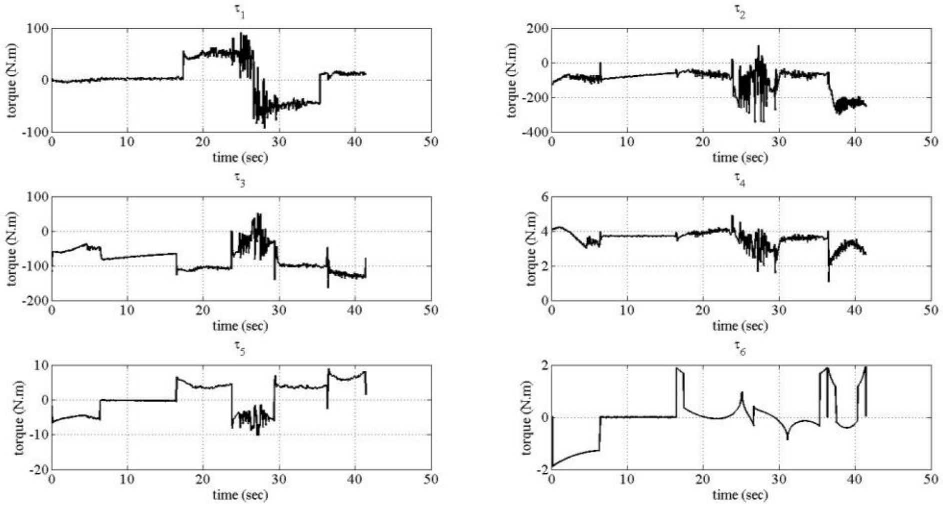


Figure B.21 : Applied joint torques.

Experiment 4:

- Circle in XY-plane
- Radius of the circle is 75 mm.
- Constant -30 N in z-direction.
- Motion time is 20 seconds in compliant motion phase.
- $K_{fp} = diag(3.75, 3.75, 3.75, 3.75, 3.75, 3.75)$
- $K_{fd} = diag(300, 300, 300, 300, 300, 300)$
- $K_{fI} = diag(2, 2, 2, 2, 2, 2)$

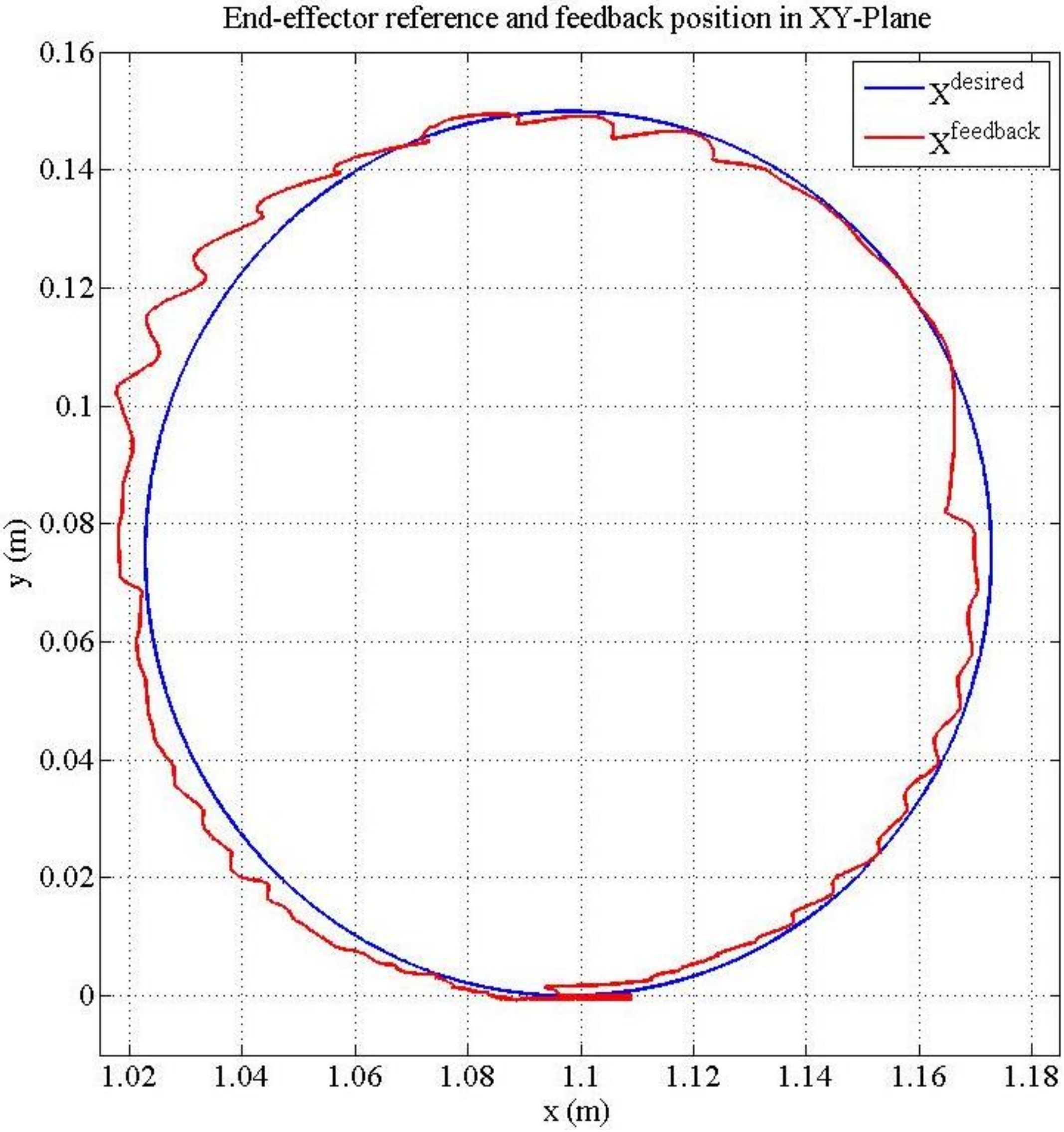


Figure B.22 : Desired and feedback path.

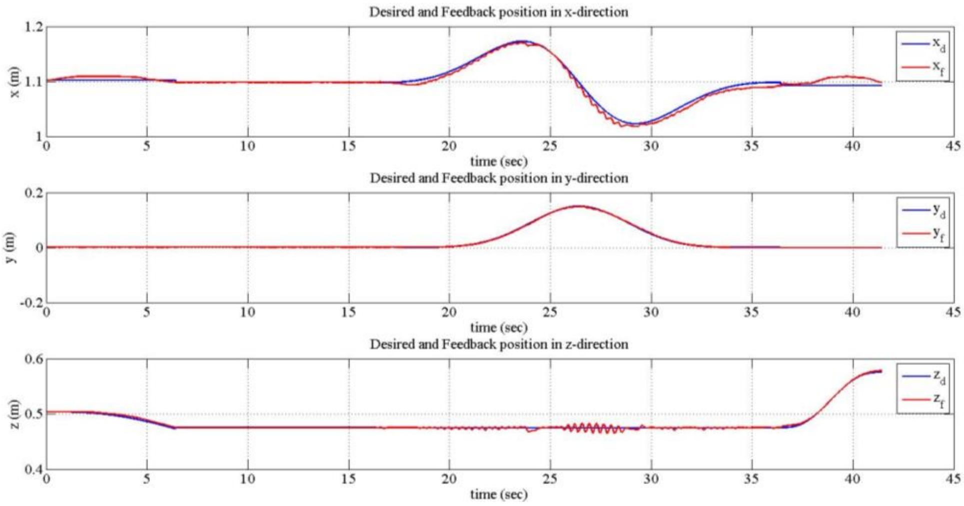


Figure B.23 : Desired and feedback positions.

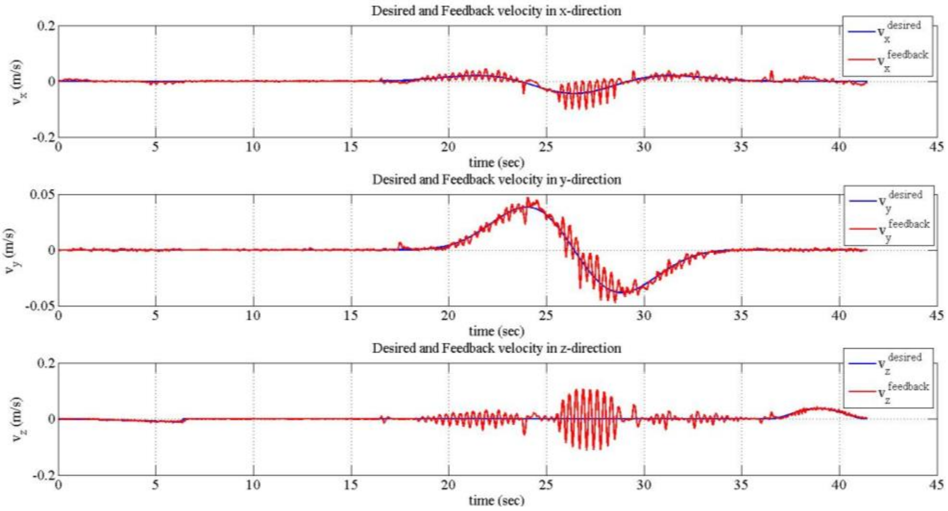


Figure B.24 : Desired and feedback velocities.

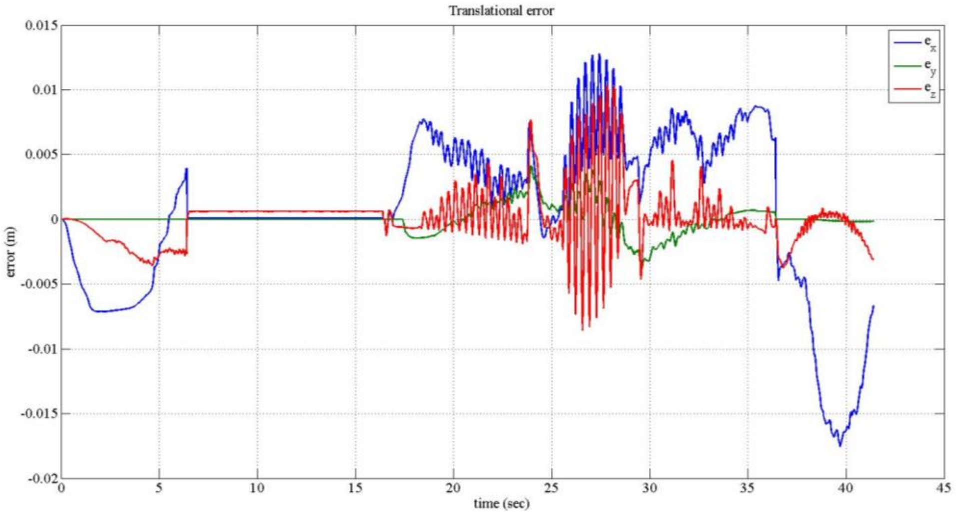


Figure B.25 : Translational errors.

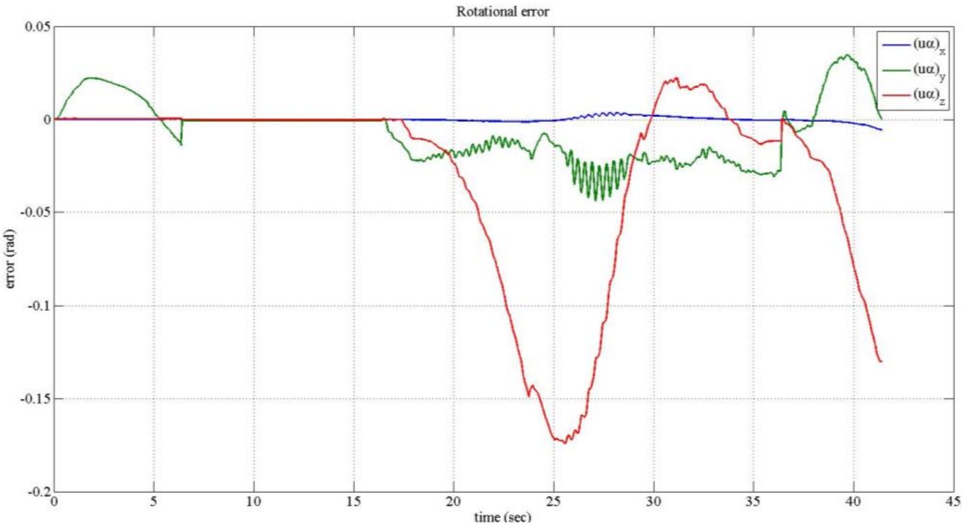


Figure B.26 : Rotational errors.

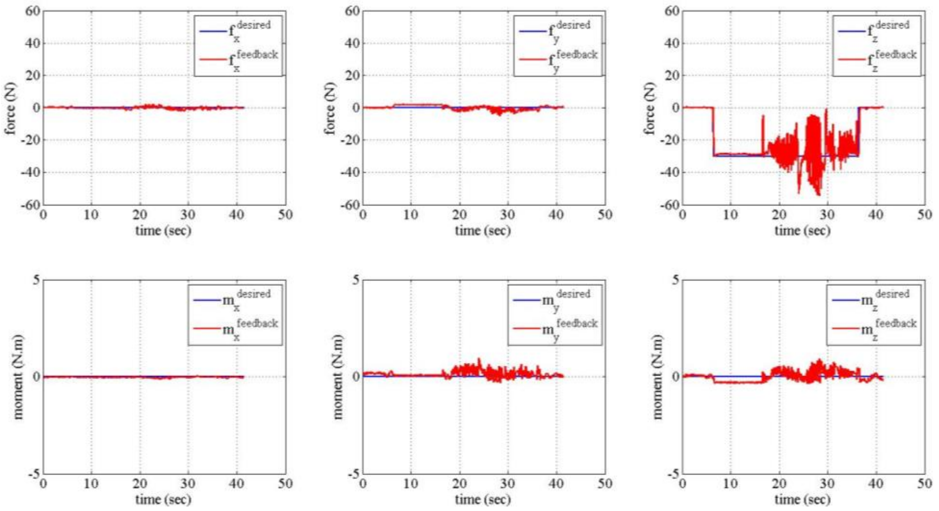


Figure B.27 : Desired and feedback forces.

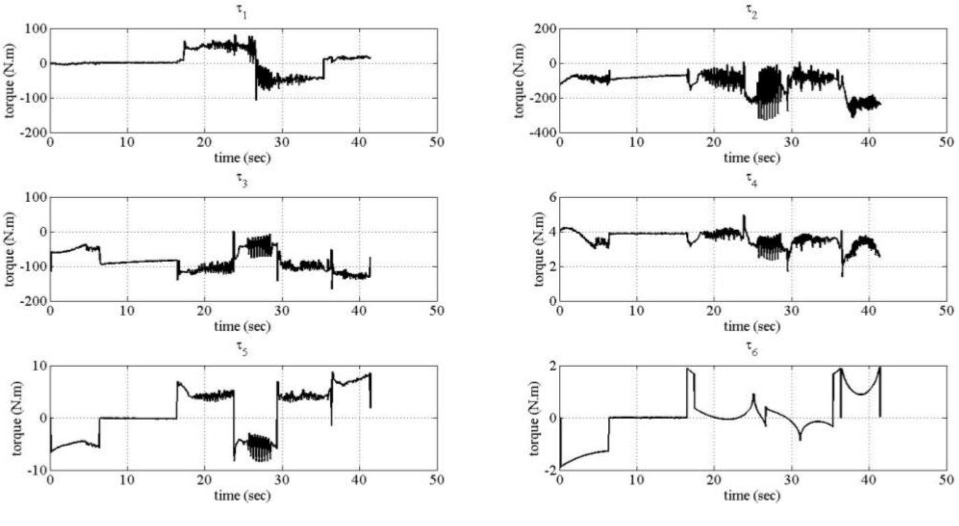


Figure B.28 : Applied joint torques.

Experiment 5:

- Circle in XY-plane
- Radius of the circle is 75 mm.
- Constant -35 N in z-direction.
- Motion time is 20 seconds in compliant motion phase.
- $K_{fp} = diag(3.75, 3.75, 3.75, 3.75, 3.75, 3.75)$
- $K_{fd} = diag(300, 300, 300, 300, 300, 300)$
- $K_{fI} = diag(2, 2, 2, 2, 2, 2)$

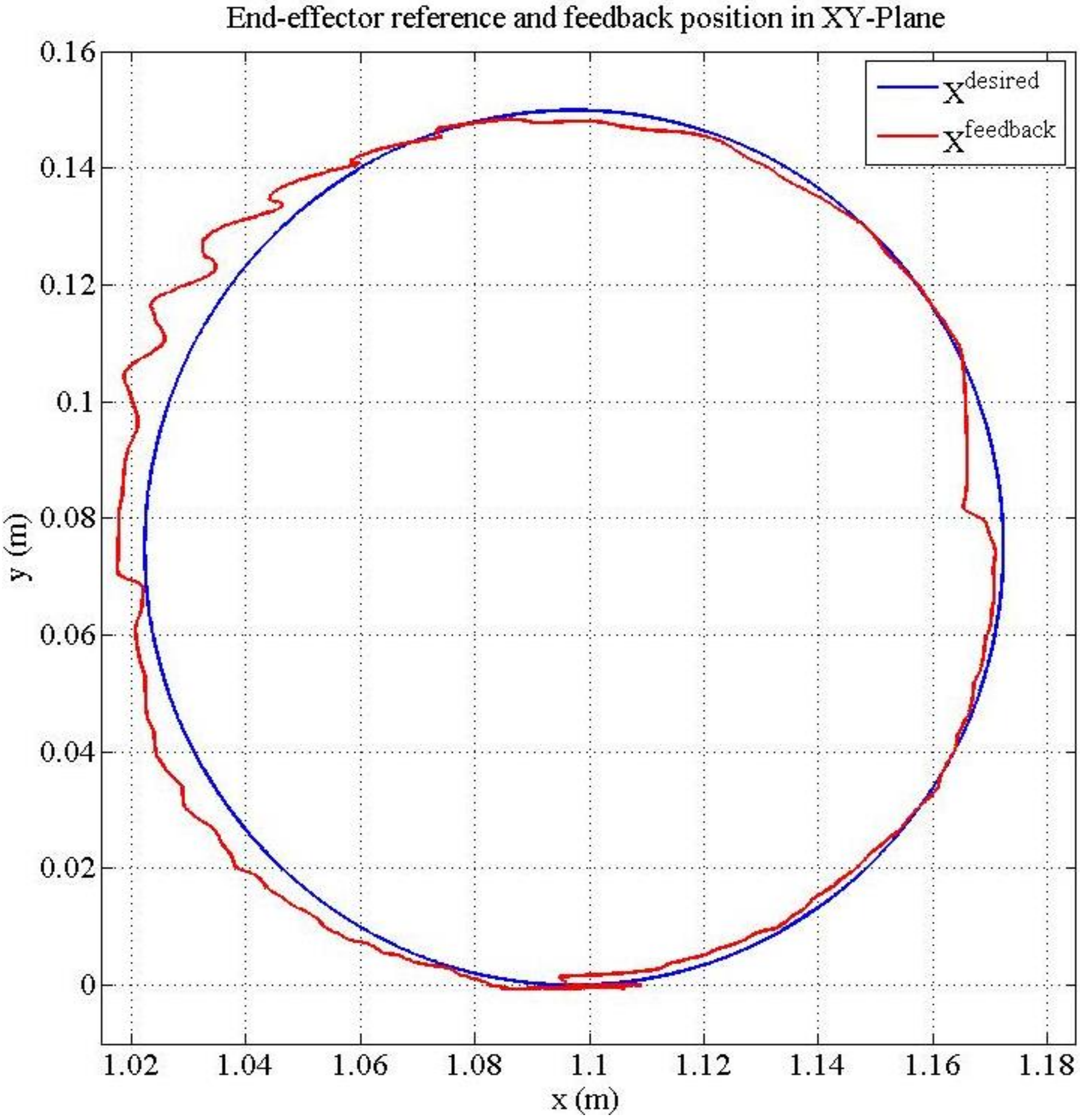


Figure B.29 : Desired and feedback path.

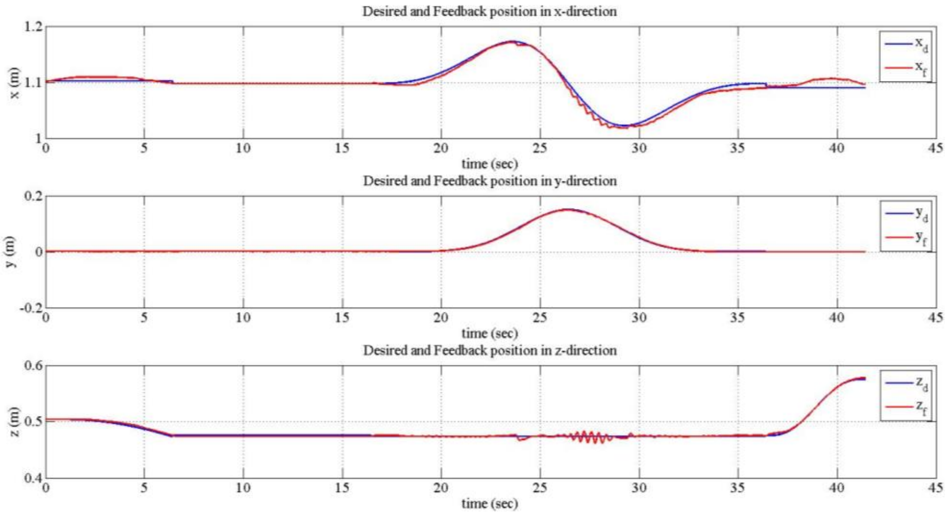


Figure B.30 : Desired and feedback positions.

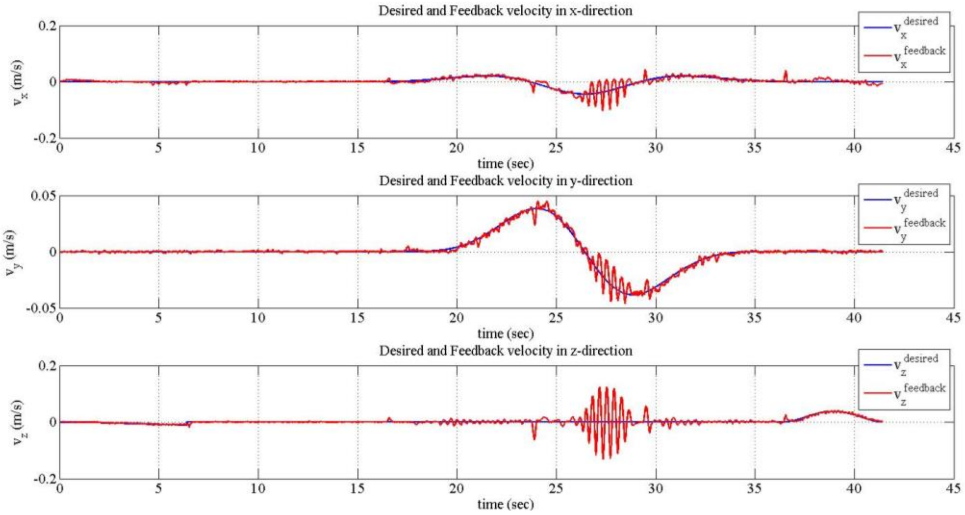


Figure B.31 : Desired and feedback velocities.

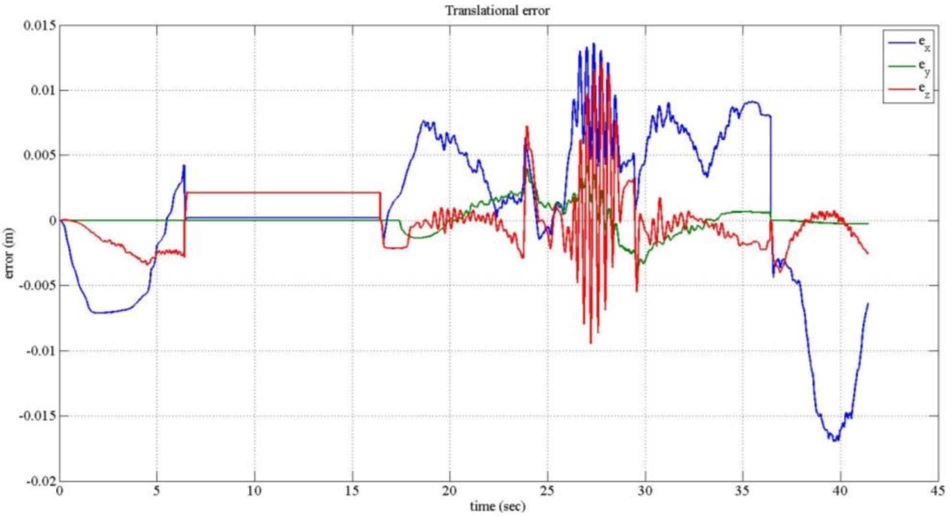


Figure B.32 : Translational errors.

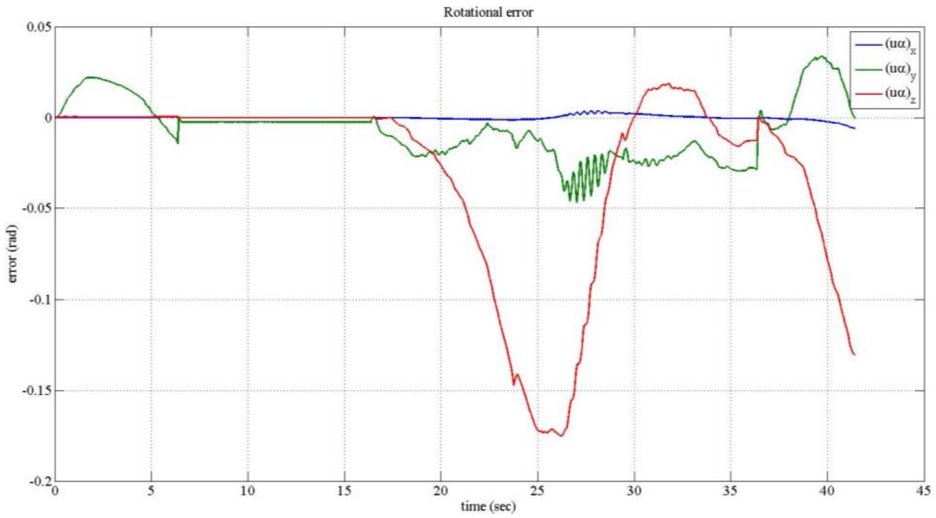


Figure B.33 : Rotational errors.

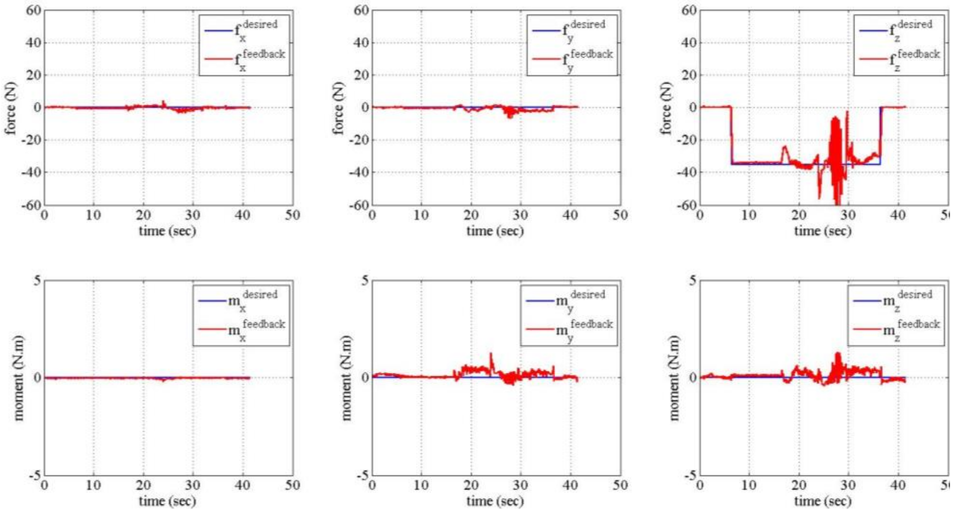


Figure B.34 : Desired and feedback forces.

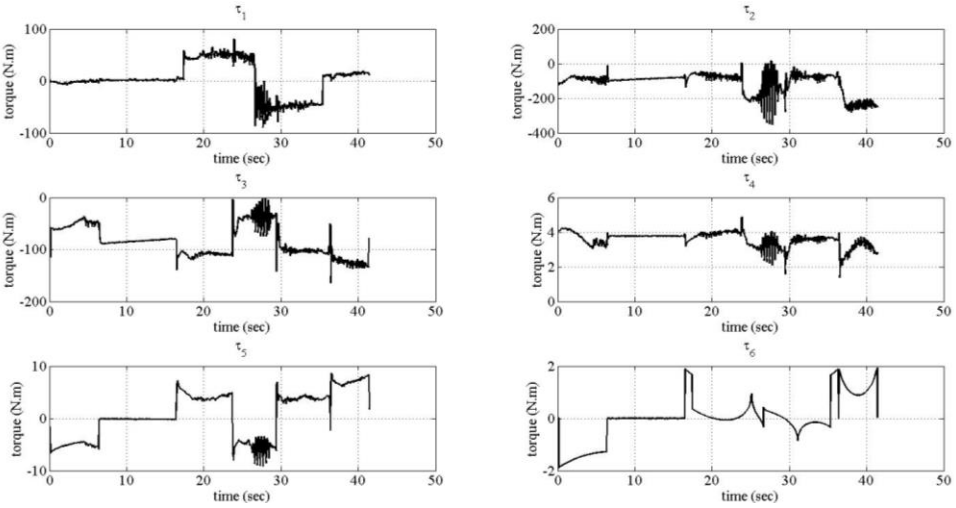


Figure B.35 : Applied joint torques.

Experiment 6:

- Circle in XY-plane
- Radius of the circle is 75 mm.
- Constant -40 N in z-direction.
- Motion time is 20 seconds in compliant motion phase.
- $K_{fp} = diag(3.75, 3.75, 3.75, 3.75, 3.75, 3.75)$
- $K_{fd} = diag(300, 300, 300, 300, 300, 300)$
- $K_{fI} = diag(2, 2, 2, 2, 2, 2)$

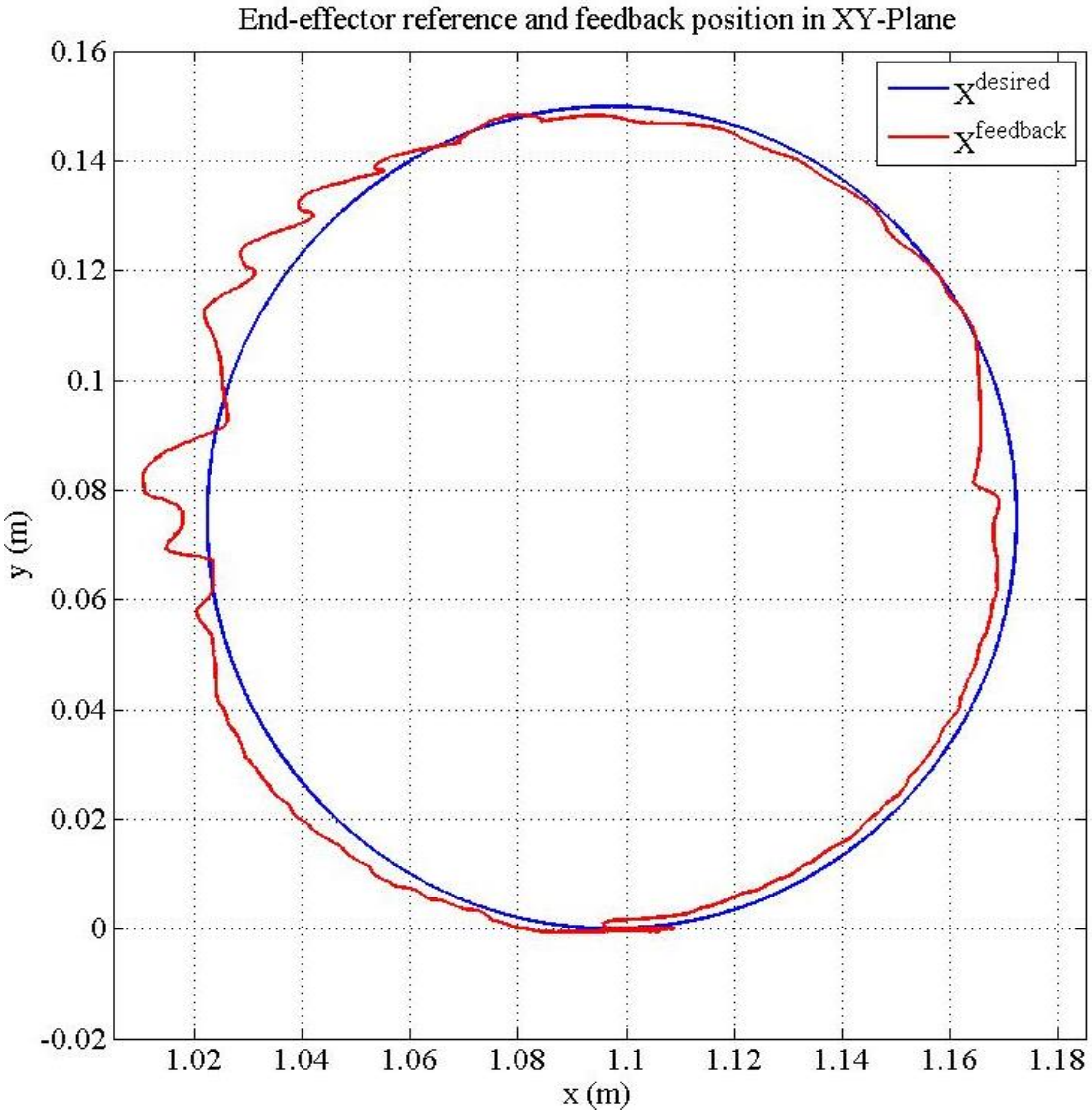


Figure B.36 : Desired and feedback path.

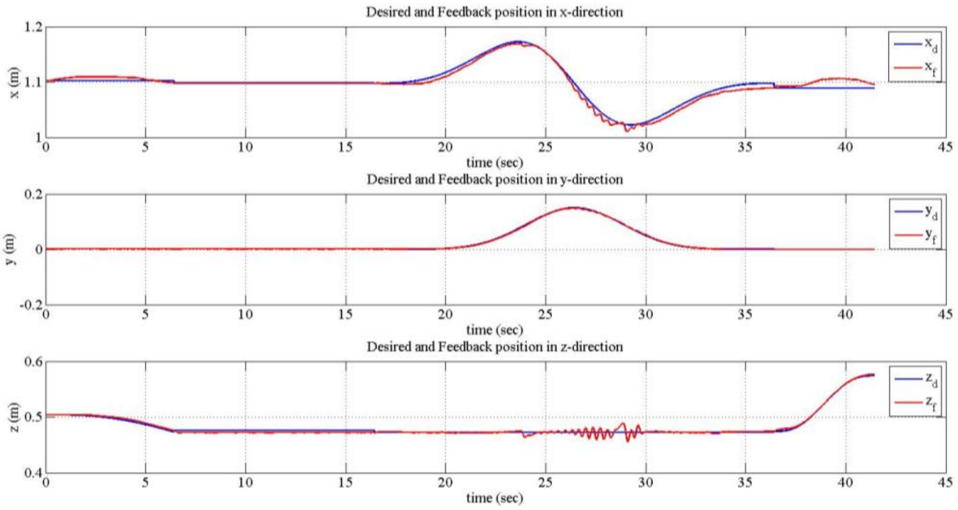


Figure B.37 : Desired and feedback positions.

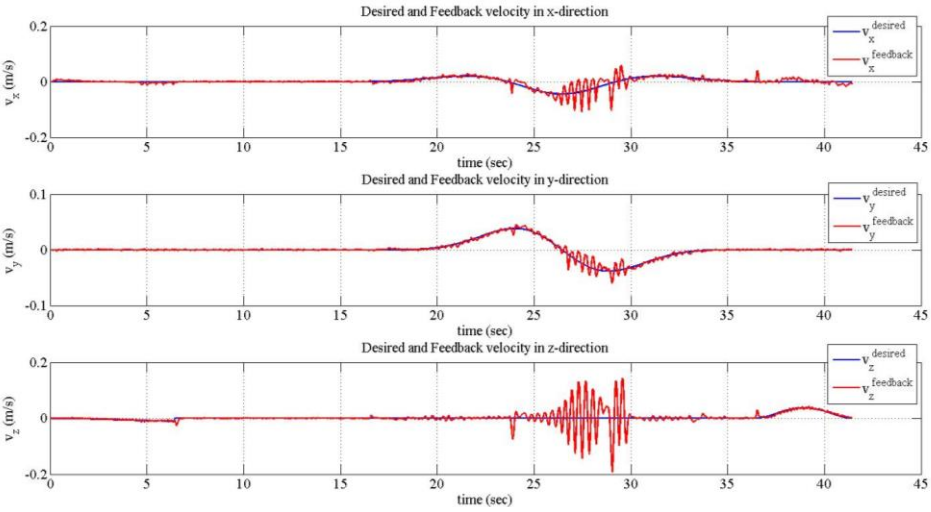


Figure B.38 : Desired and feedback velocities.

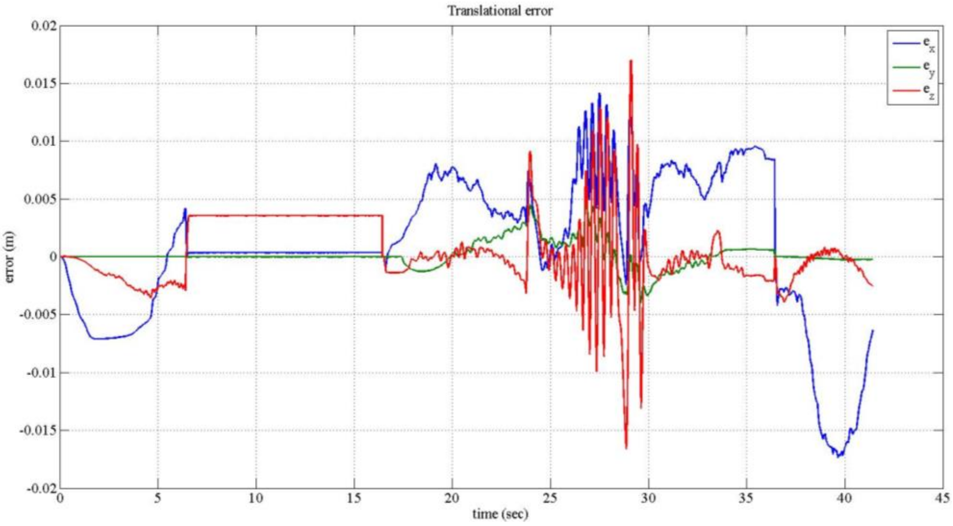


Figure B.39 : Translational errors.

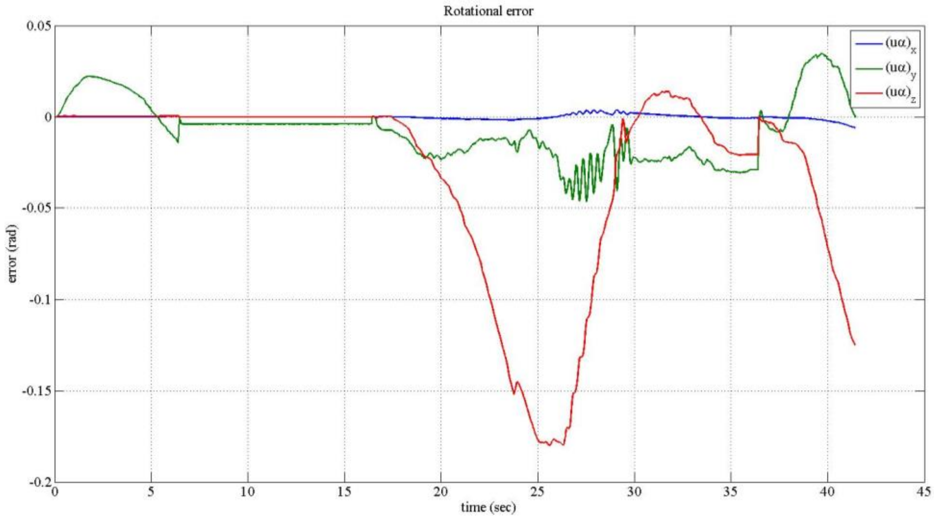


Figure B.40 : Rotational errors.

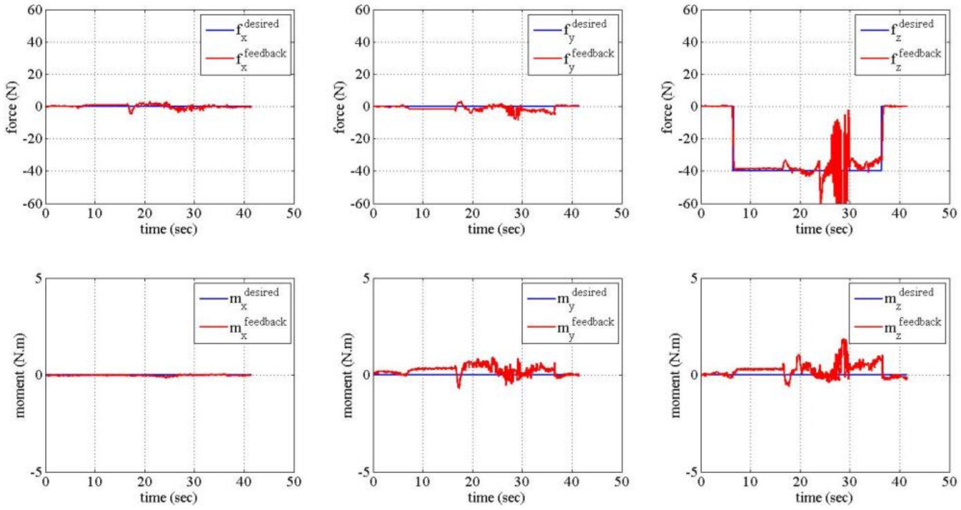


Figure B.41 : Desired and feedback forces.

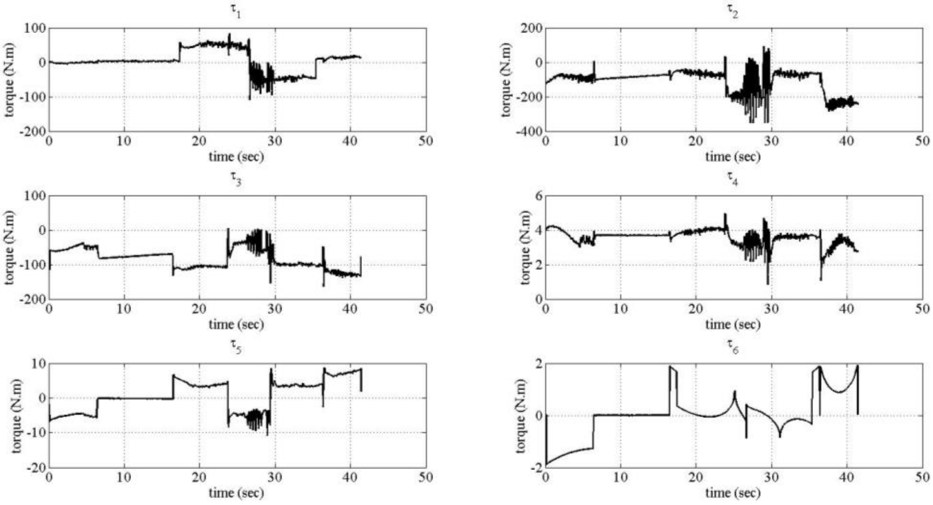


Figure B.42 : Applied joint torques.

Experiment 7:

- Circle in XY-plane
- Radius of the circle is 75 mm.
- Constant -30 N in z-direction.
- Motion time is 20 seconds in compliant motion phase.
- $K_{fp} = diag(3.75, 3.75, 3.75, 3.75, 3.75, 3.75)$
- $K_{fd} = diag(500, 500, 500, 500, 500, 500)$
- $K_{fI} = diag(2, 2, 2, 2, 2, 2)$

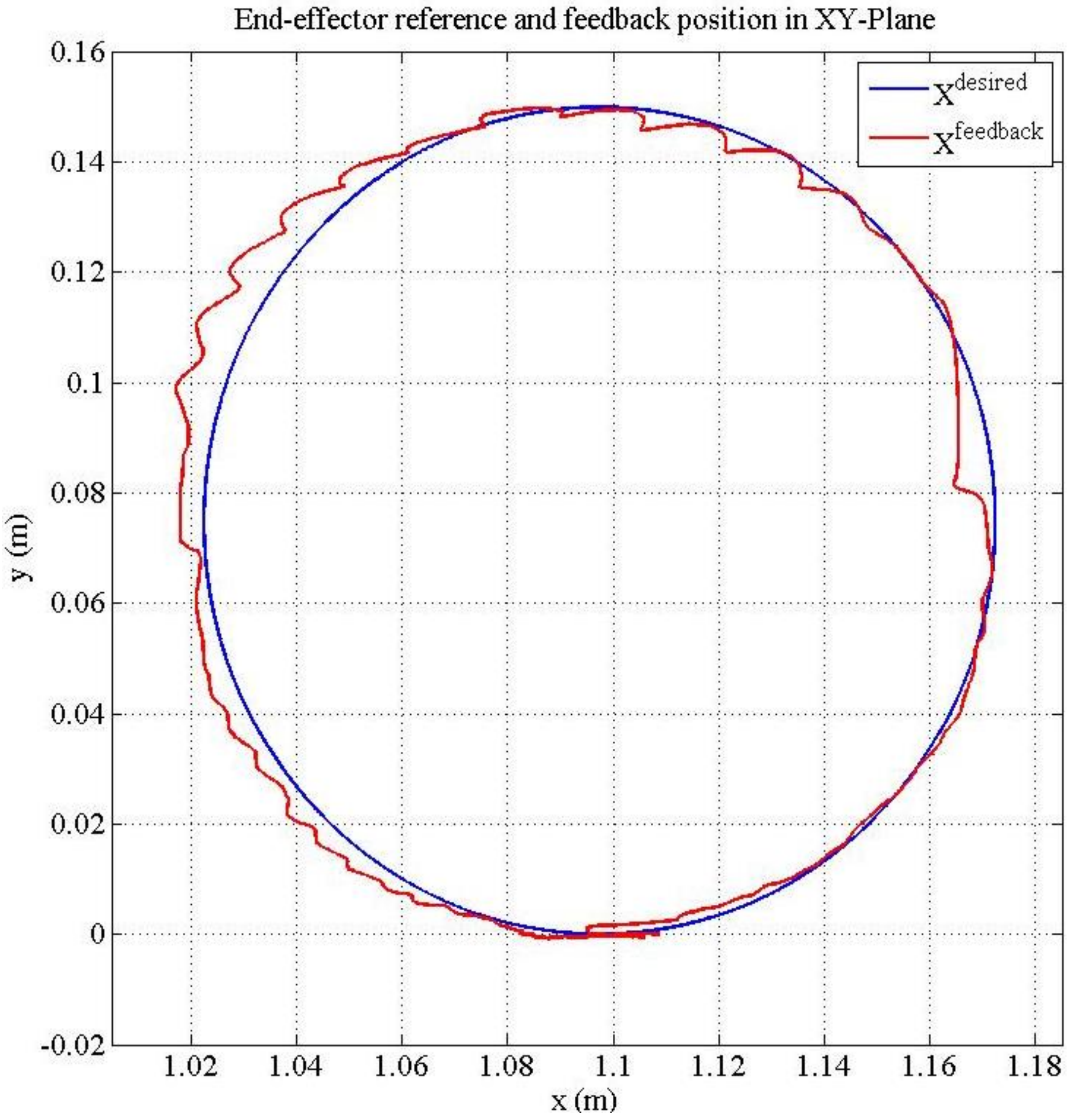


Figure B.43 : Desired and feedback path.

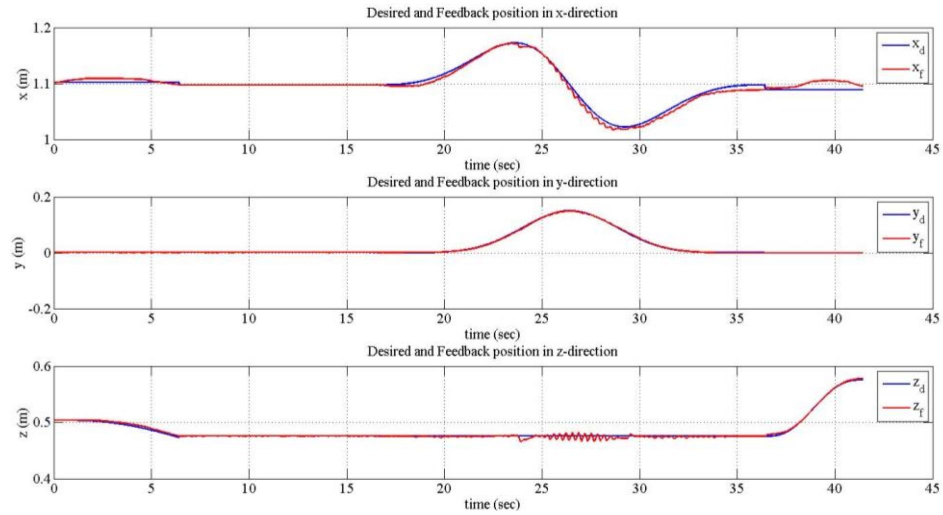


Figure B.44 : Desired and feedback positions.

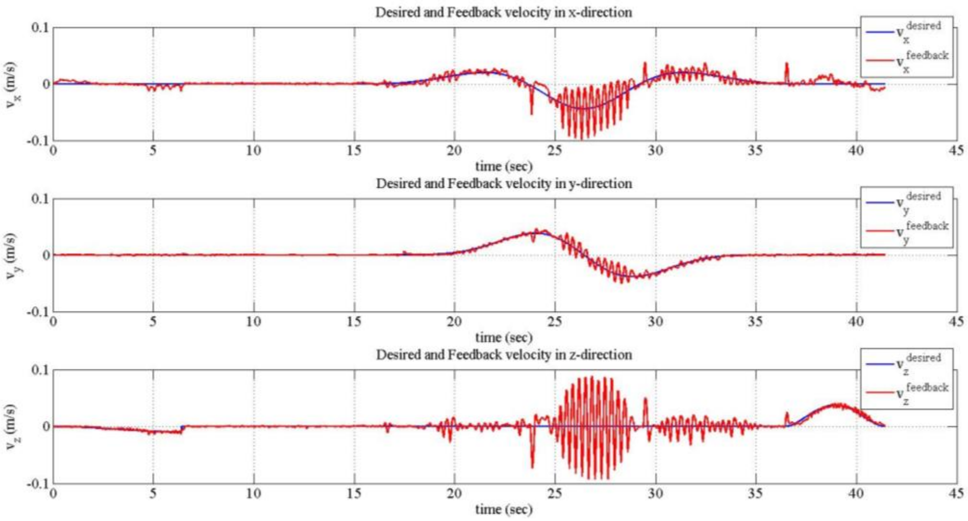


Figure B.45 : Desired and feedback velocities.

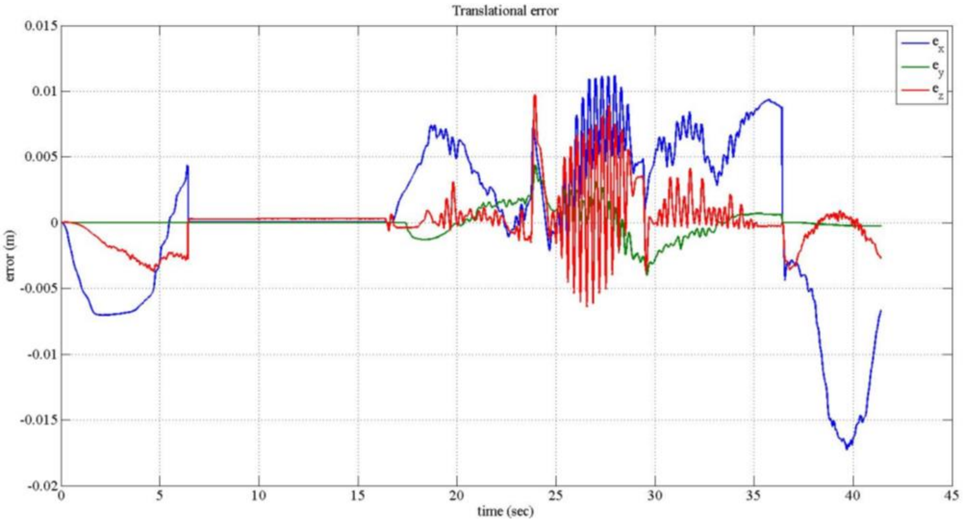


Figure B.46 : Translational errors.

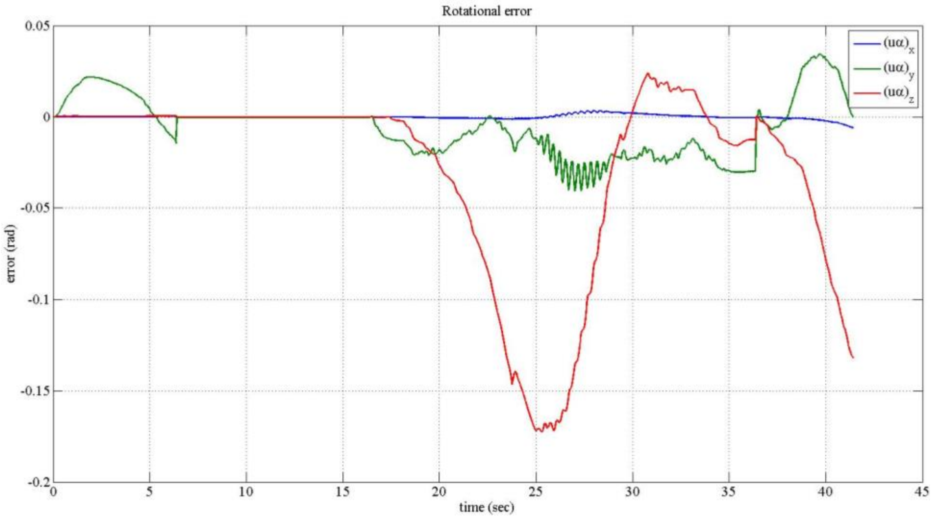


Figure B.47 : Rotational errors.

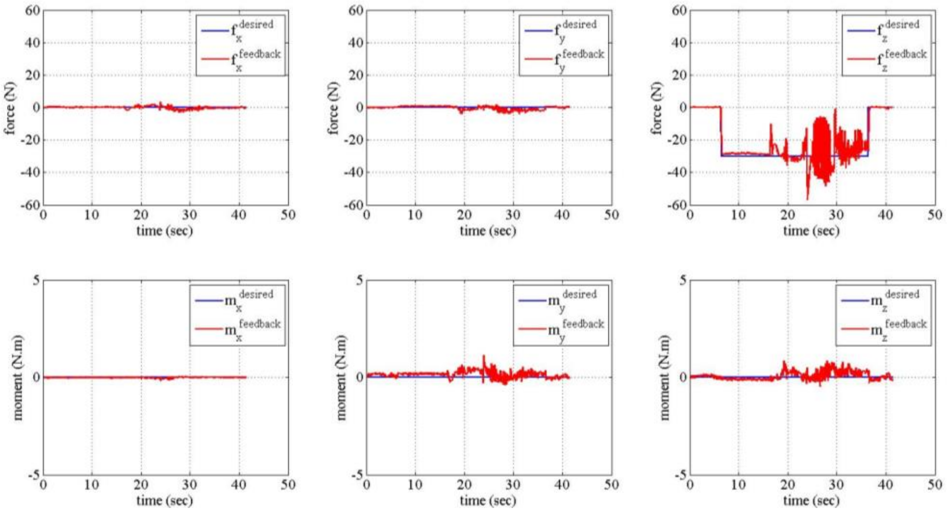


Figure B.48 : Desired and feedback forces.

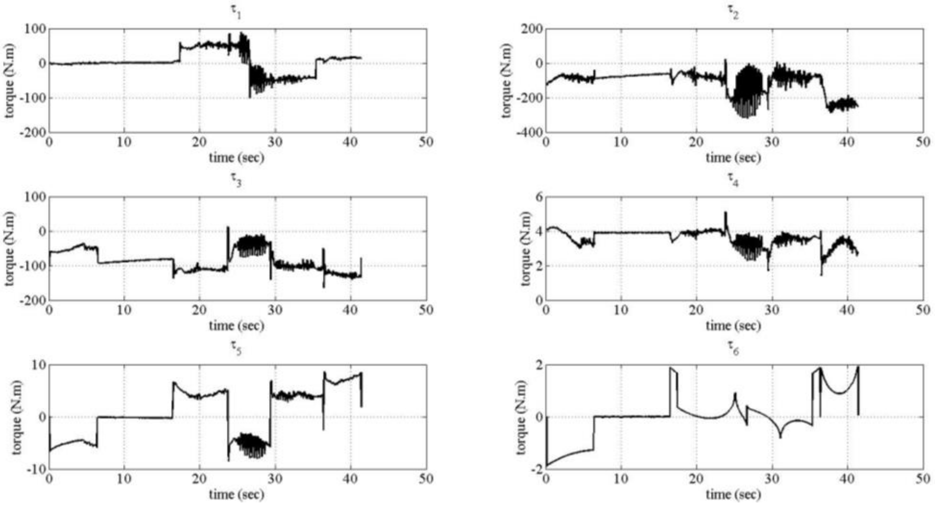


Figure B.49 : Applied joint torques.

Experiment 8:

- Circle in XY-plane
- Radius of the circle is 75 mm.
- Constant -35 N in z-direction.
- Motion time is 20 seconds in compliant motion phase.
- $K_{fp} = diag(3.75, 3.75, 3.75, 3.75, 3.75, 3.75)$
- $K_{fd} = diag(500, 500, 500, 500, 500, 500)$
- $K_{fI} = diag(2, 2, 2, 2, 2, 2)$

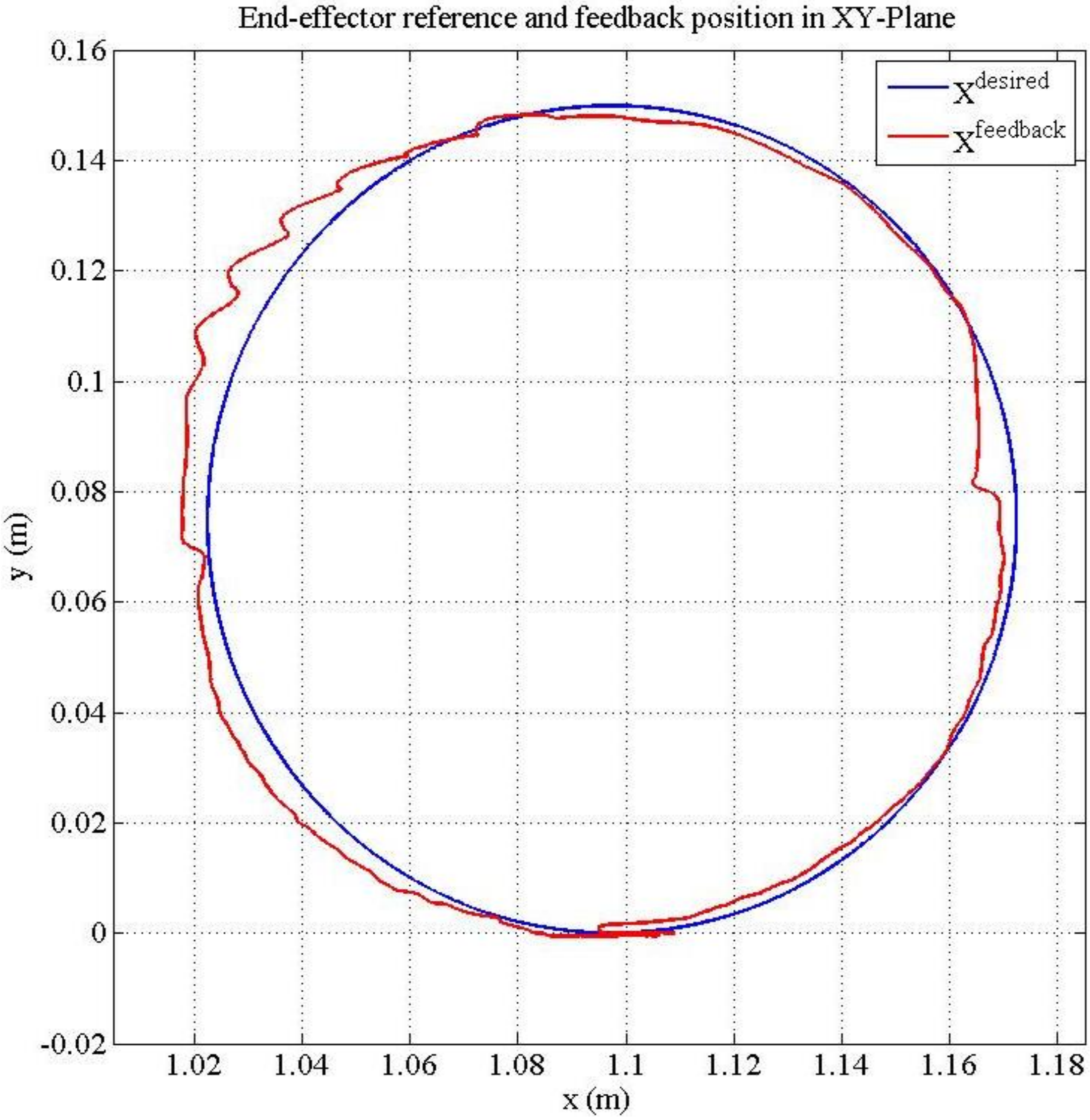


Figure B.50 : Desired and feedback path.

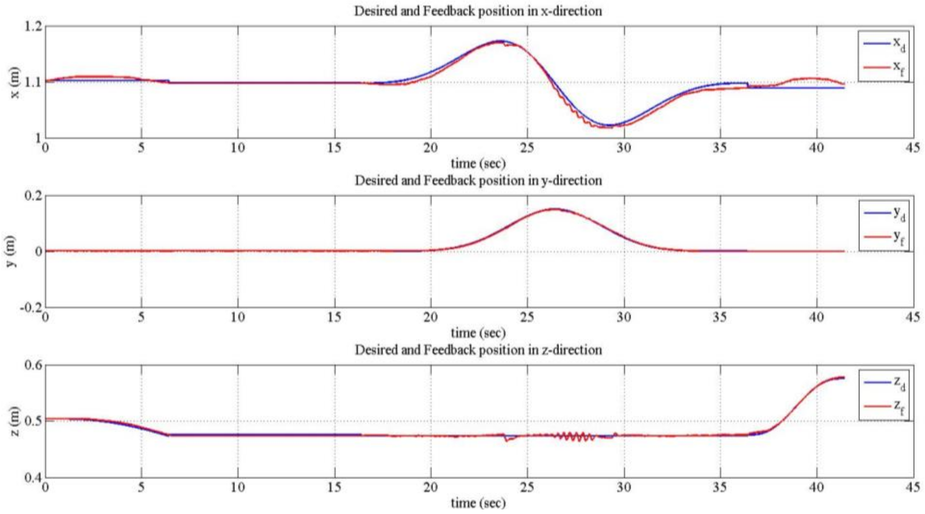


Figure B.51 : Desired and feedback positions.

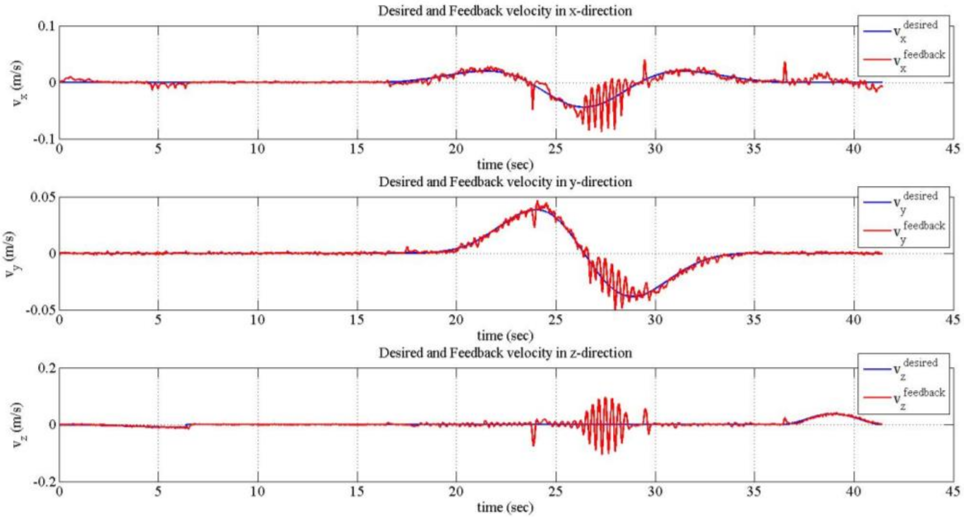


Figure B.52 : Desired and feedback velocities.

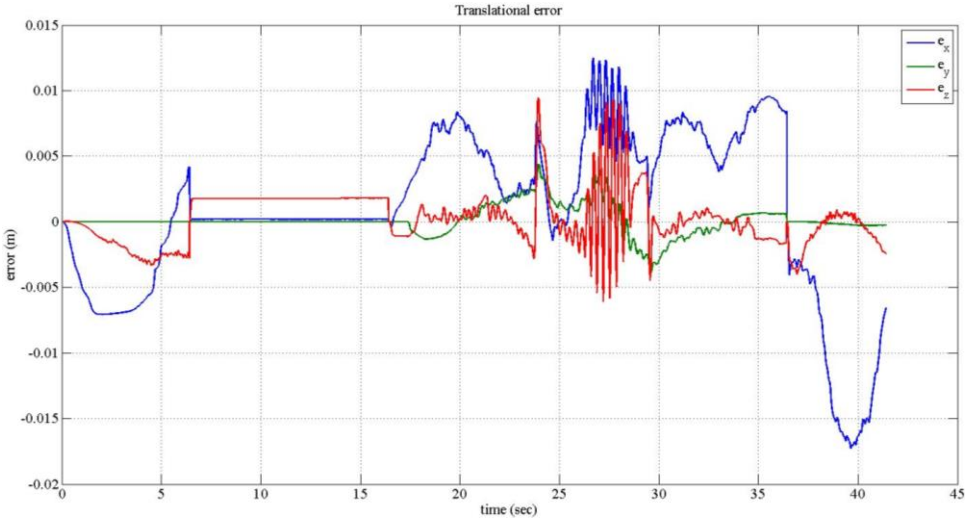


Figure B.53 : Translational errors.

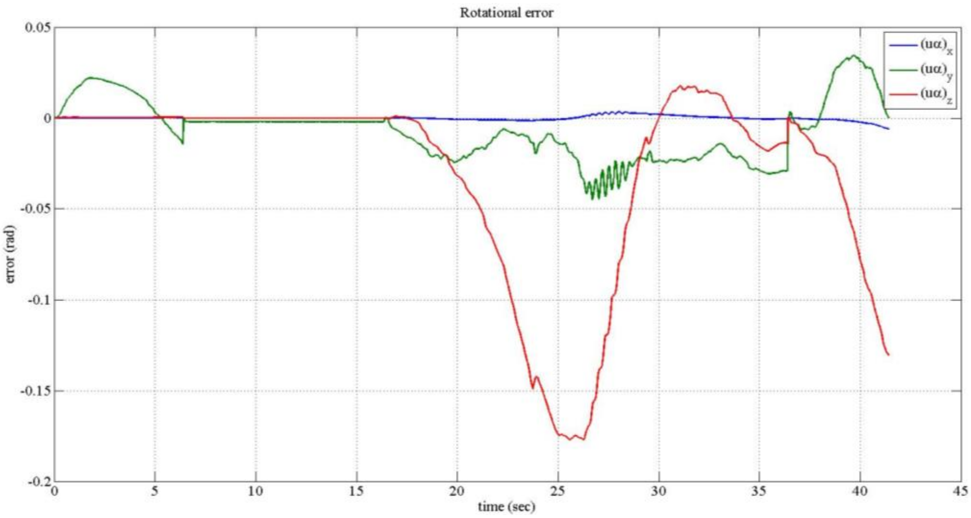


Figure B.54 : Rotational errors.

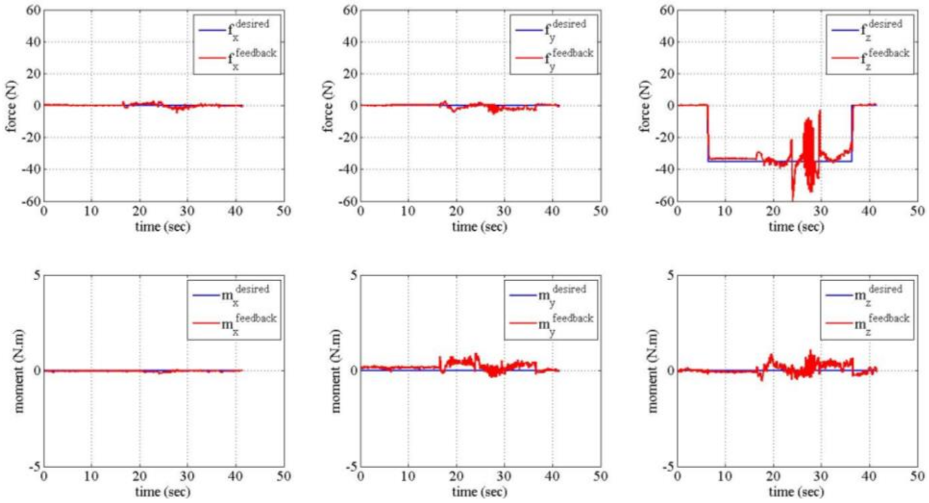


Figure B.55 : Desired and feedback forces.

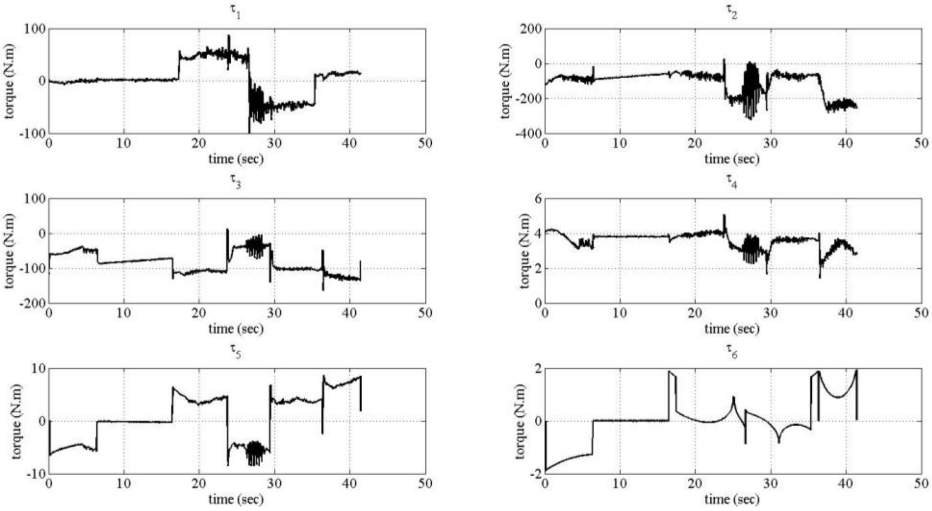


Figure B.56 : Applied joint torques.

Experiment 9:

- Circle in XY-plane
- Radius of the circle is 75 mm.
- Constant -40 N in z-direction.
- Motion time is 20 seconds in compliant motion phase.
- $K_{fp} = diag(3.75, 3.75, 3.75, 3.75, 3.75, 3.75)$
- $K_{fd} = diag(500, 500, 500, 500, 500, 500)$
- $K_{fI} = diag(2, 2, 2, 2, 2, 2)$

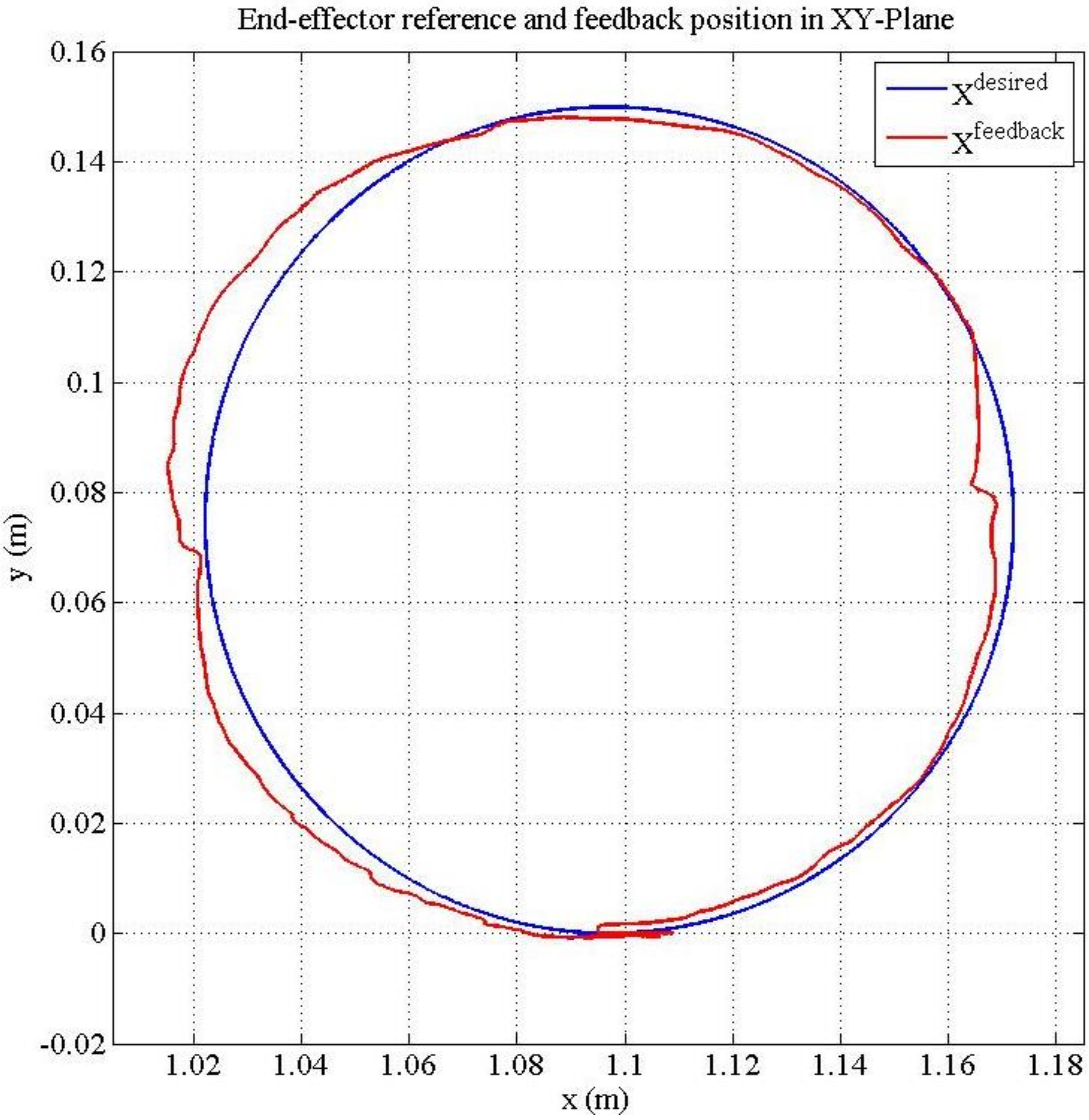


Figure B.57 : Desired and feedback path.

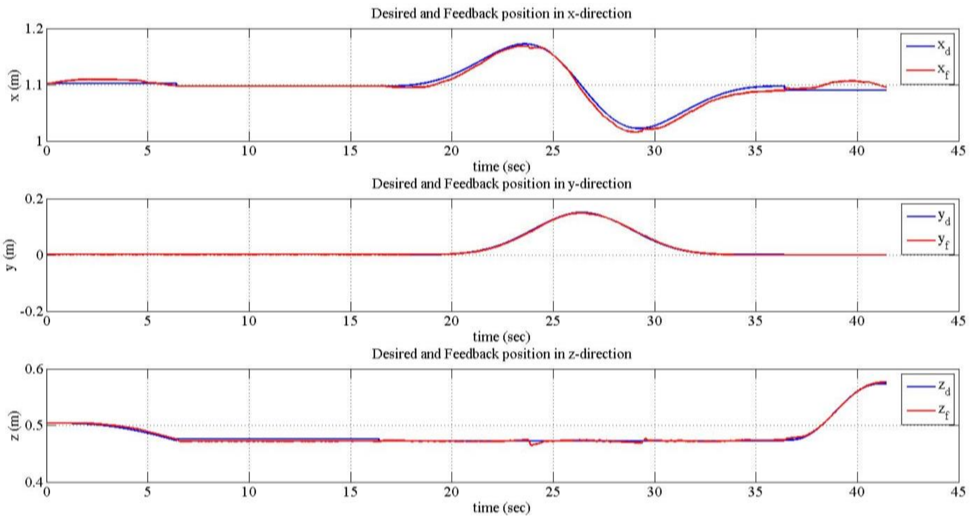


Figure B.58 : Desired and feedback positions.

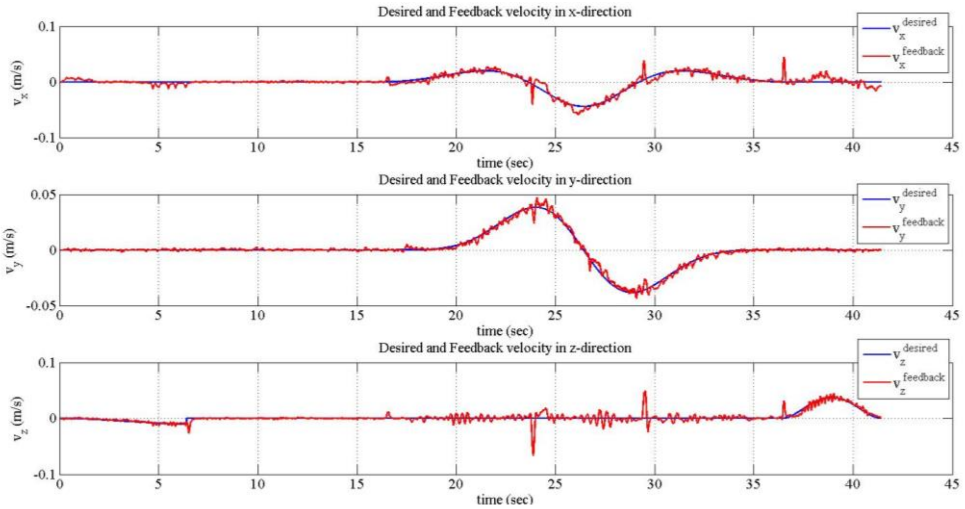


Figure B.59 : Desired and feedback velocities.

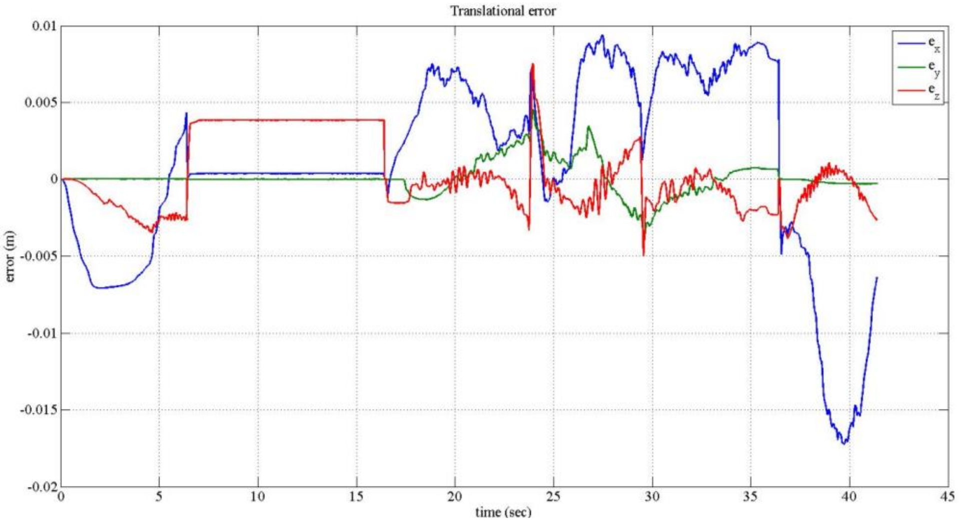


Figure B.60 : Translational errors.

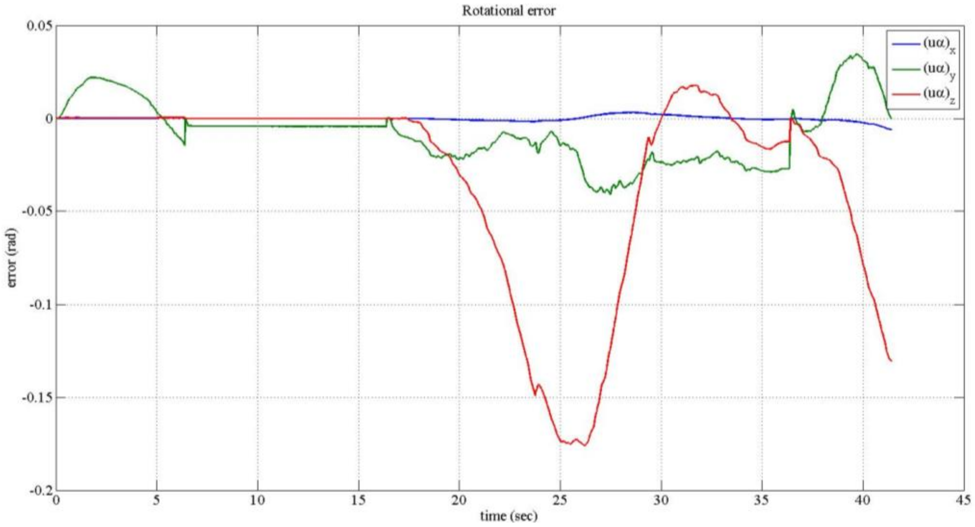


Figure B.61 : Rotational errors.

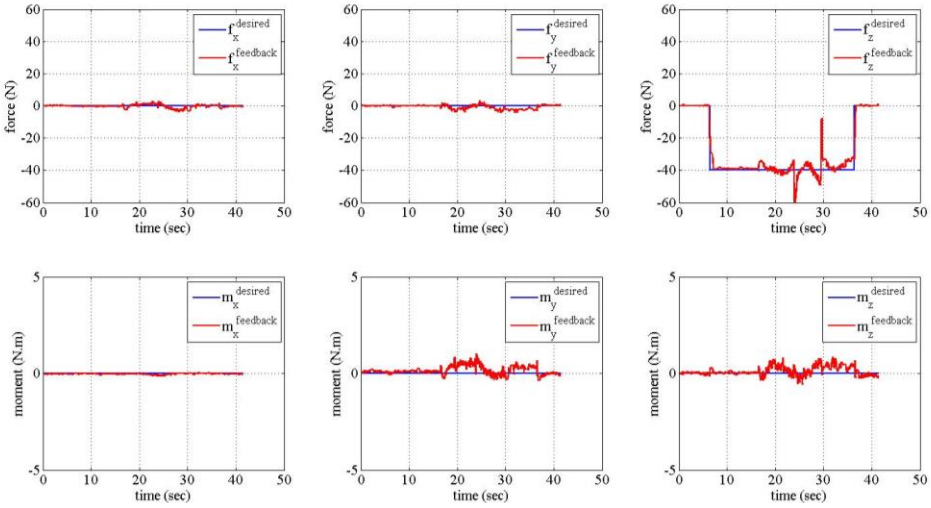


Figure B.62 : Desired and feedback forces.

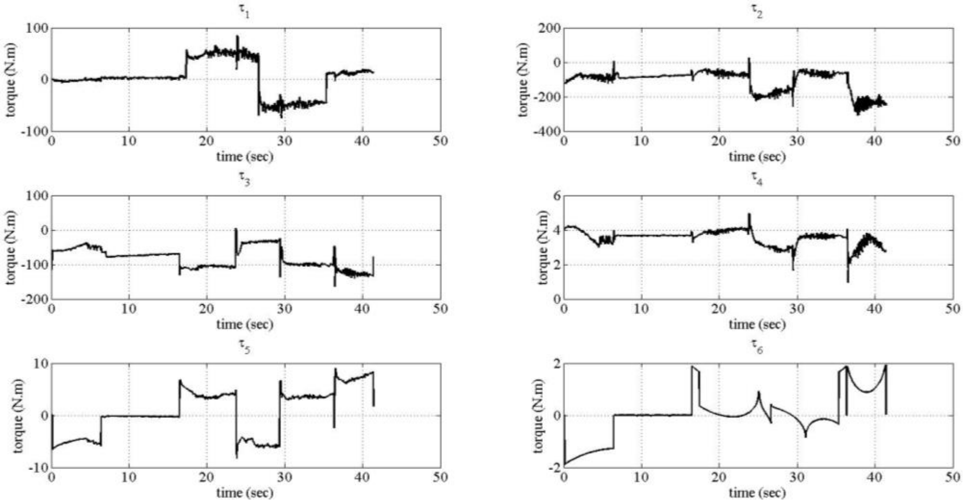


Figure B.63 : Applied joint torques.

Experiment 10:

- Drawing a line in XY-plane
- $\Delta x = 150\ mm$ and $\Delta y = 150\ mm$.
- Constant -30 N in z-direction.
- Motion time is 20 seconds in compliant motion phase.
- $K_{fp} = diag(4.75, 4.75, 4.75, 4.75, 4.75, 4.75)$
- $K_{fd} = diag(350, 350, 350, 350, 350, 350)$
- $K_{fI} = diag(2, 2, 2, 2, 2, 2)$

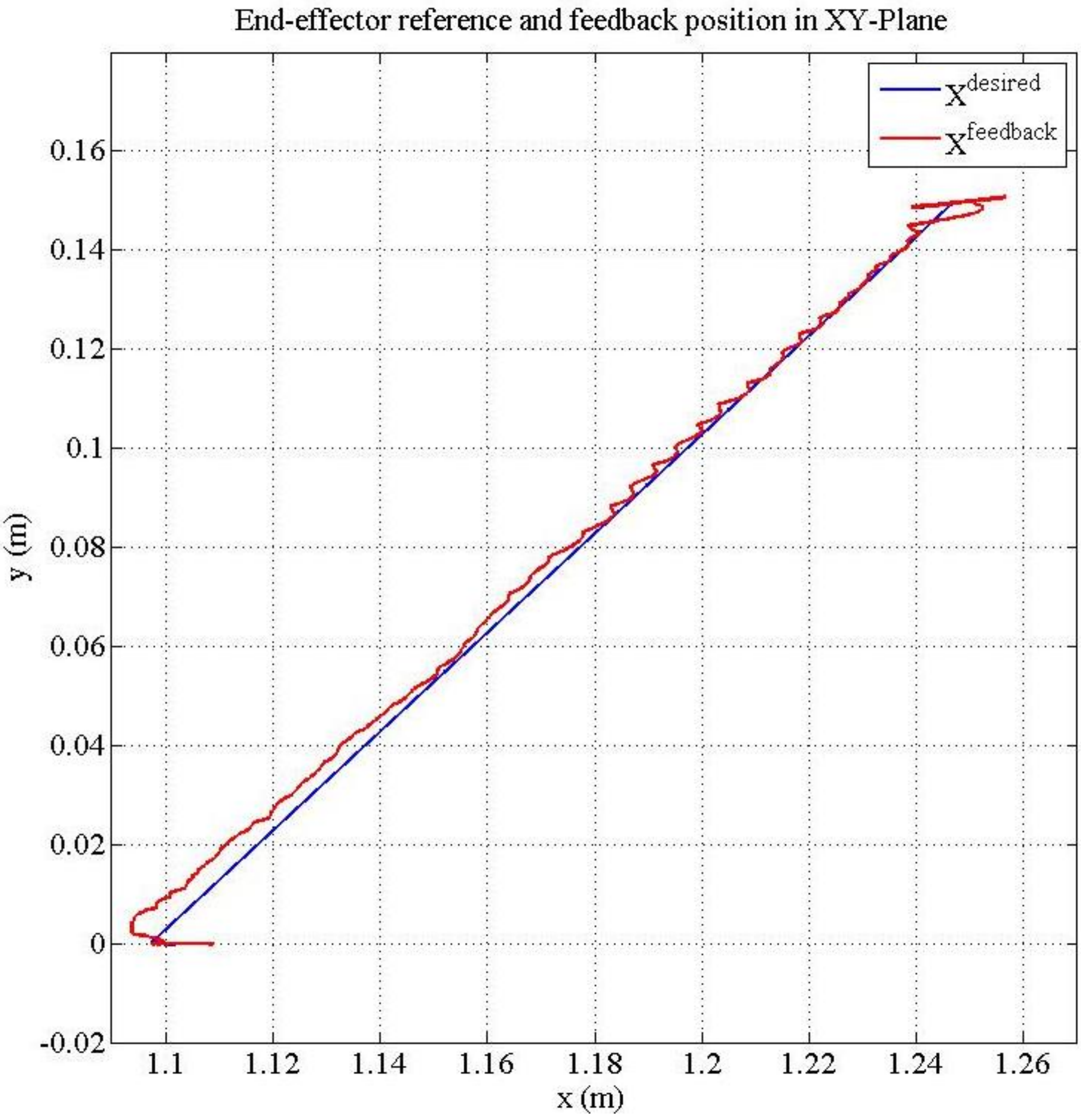
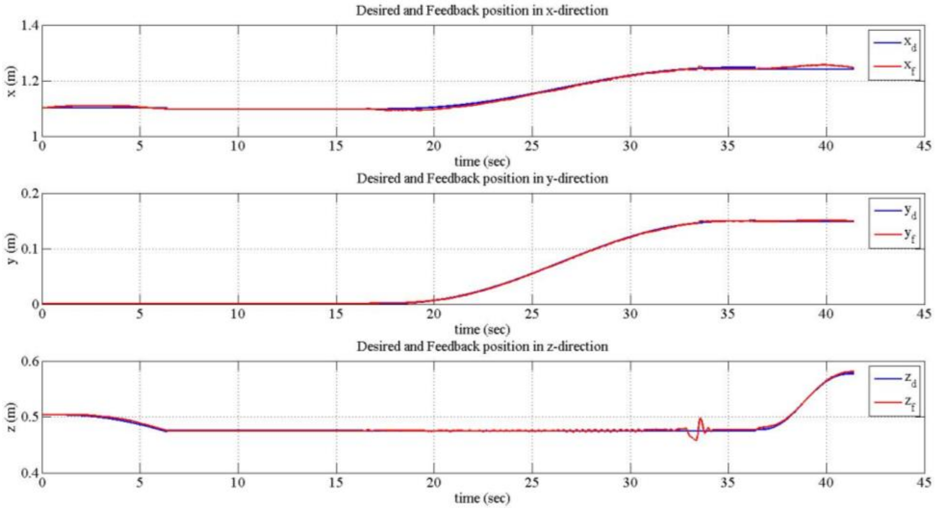


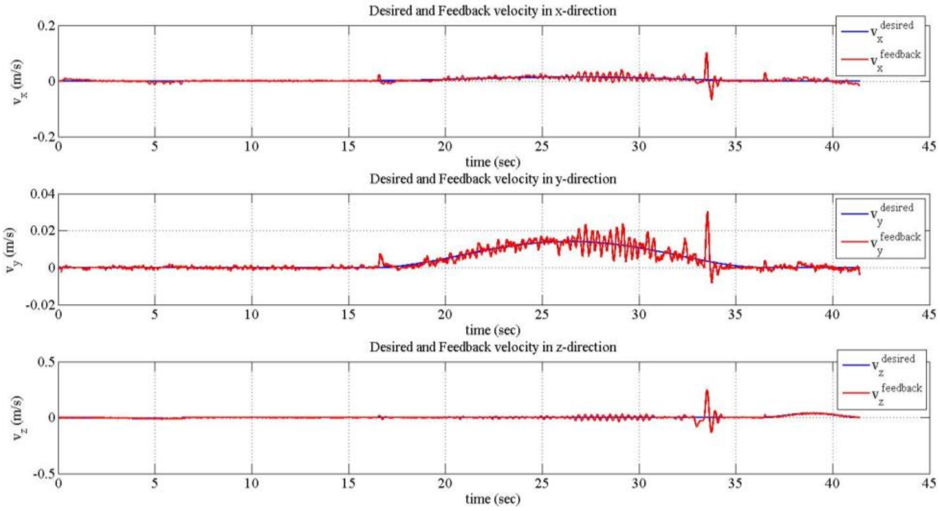
Figure B.64 : Desired and feedback path.


Desired and Feedback position in x-direction

Desired and Feedback position in y-direction

Desired and Feedback position in z-direction

Figure B.65 : Desired and feedback positions.


Desired and Feedback velocity in x-direction

Desired and Feedback velocity in y-direction

Desired and Feedback velocity in z-direction

Figure B.66 : Desired and feedback velocities.

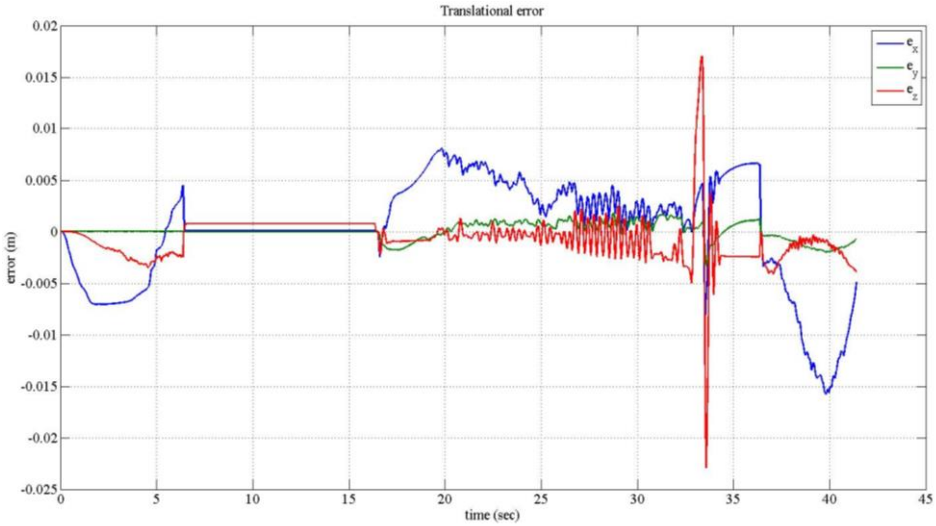

Translational error

Figure B.67 : Translational errors.

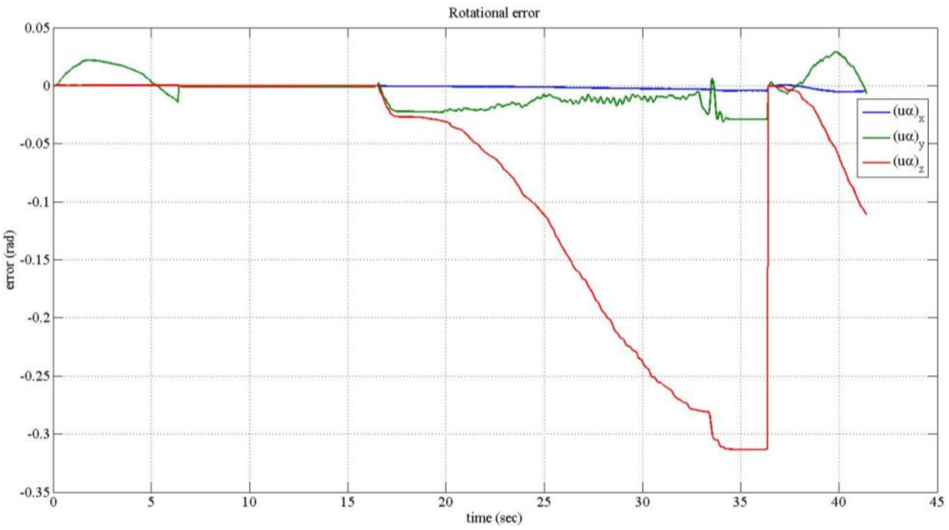


Figure B.68 : Rotational errors.

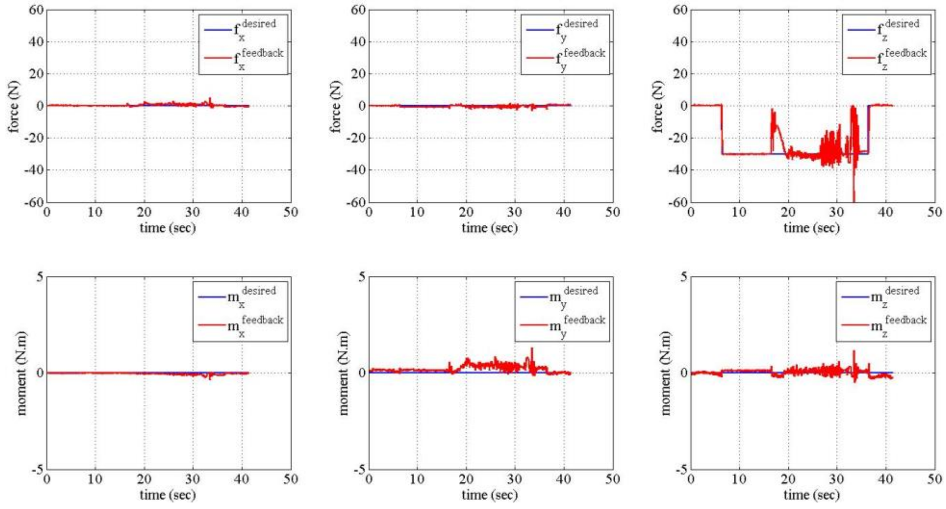


Figure B.69 : Desired and feedback forces.

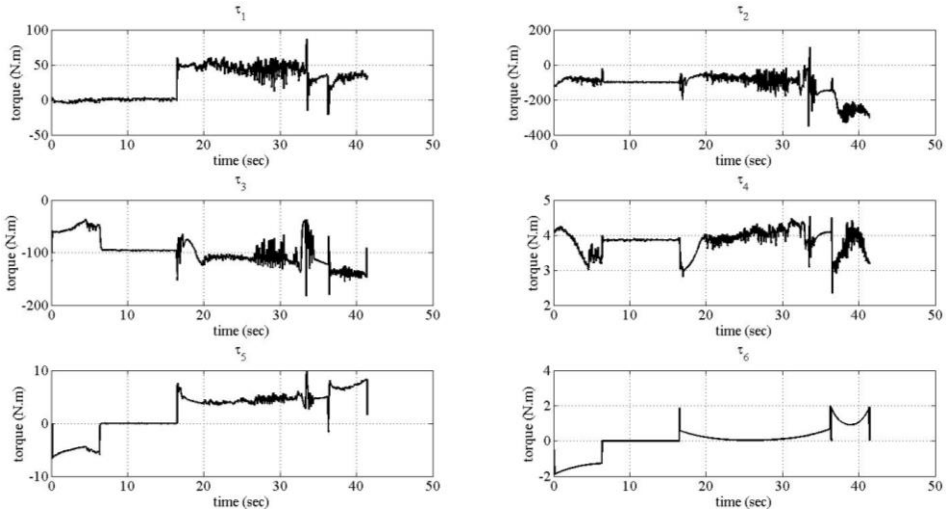


Figure B.70 : Applied joint torques.

Experiment 11:

- Drawing a line in XY-plane
- $\Delta x = 150\ mm$ and $\Delta y = 150\ mm$.
- Constant -35 N in z-direction.
- Motion time is 20 seconds in compliant motion phase.
- $K_{fp} = diag(4.75, 4.75, 4.75, 4.75, 4.75, 4.75)$
- $K_{fd} = diag(350, 350, 350, 350, 350, 350)$
- $K_{fI} = diag(2, 2, 2, 2, 2, 2)$

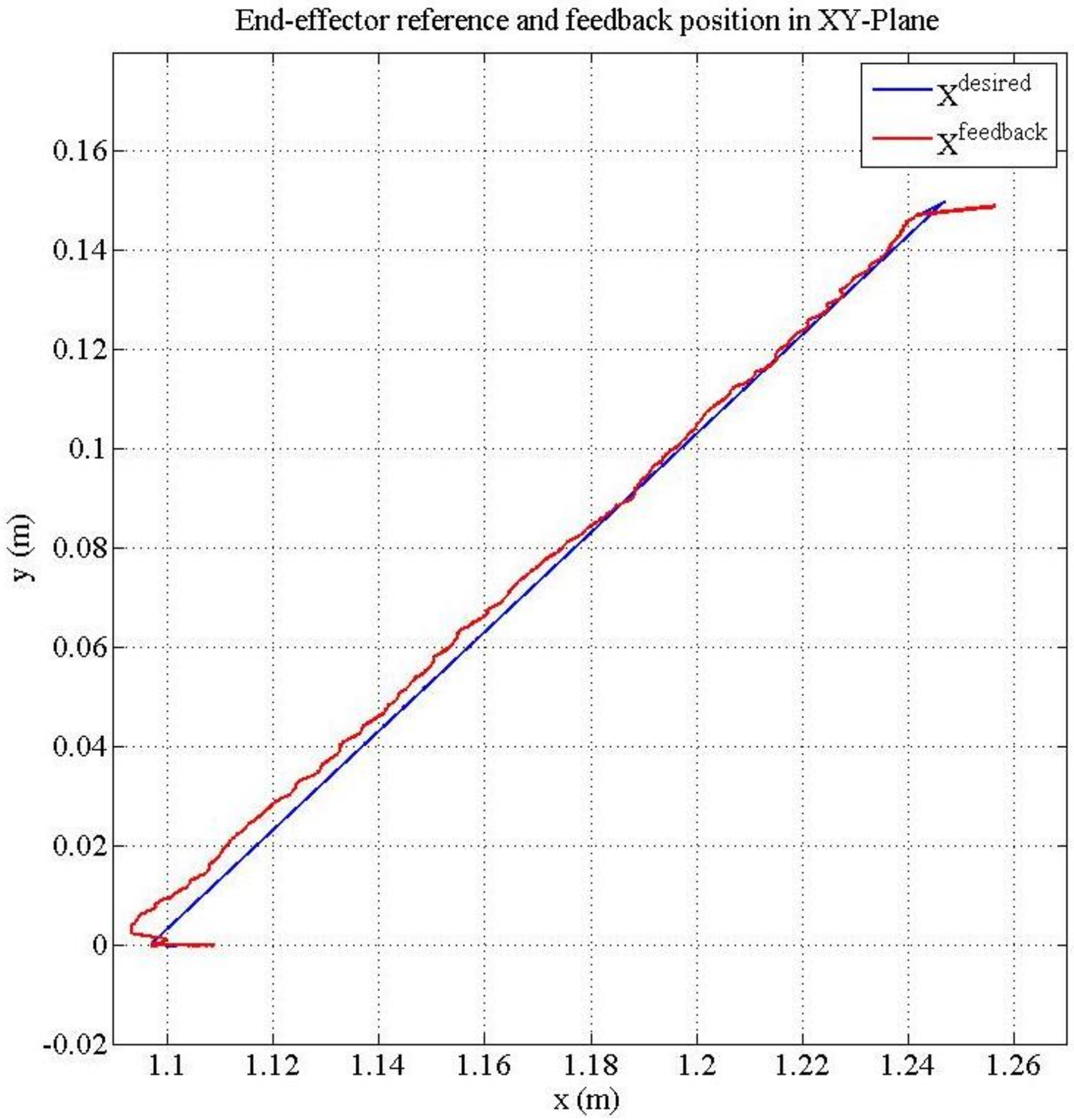


Figure B.71 : Desired and feedback path.

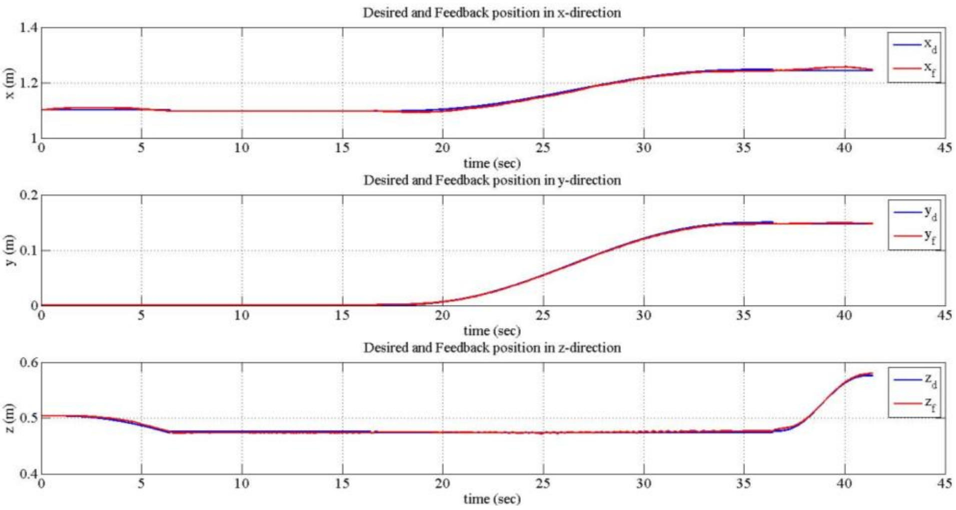


Figure B.72 : Desired and feedback positions.

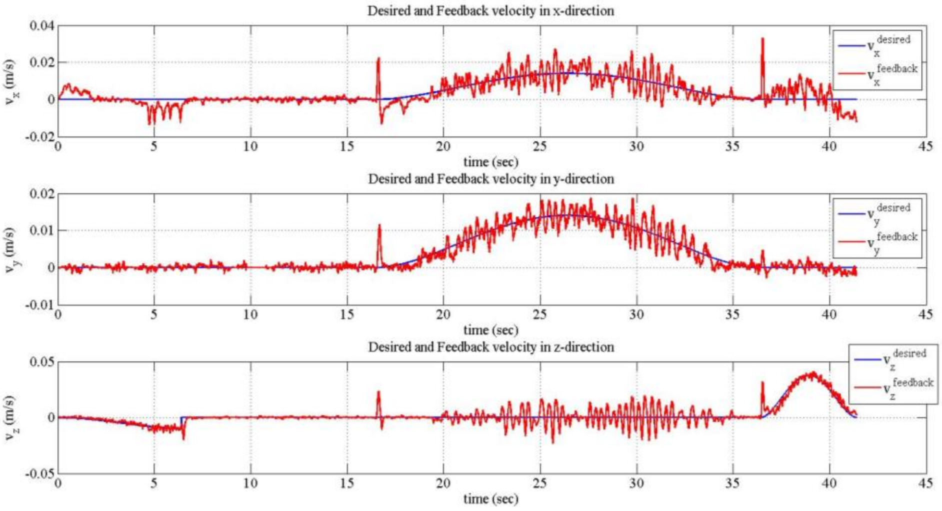


Figure B.73 : Desired and feedback velocities.

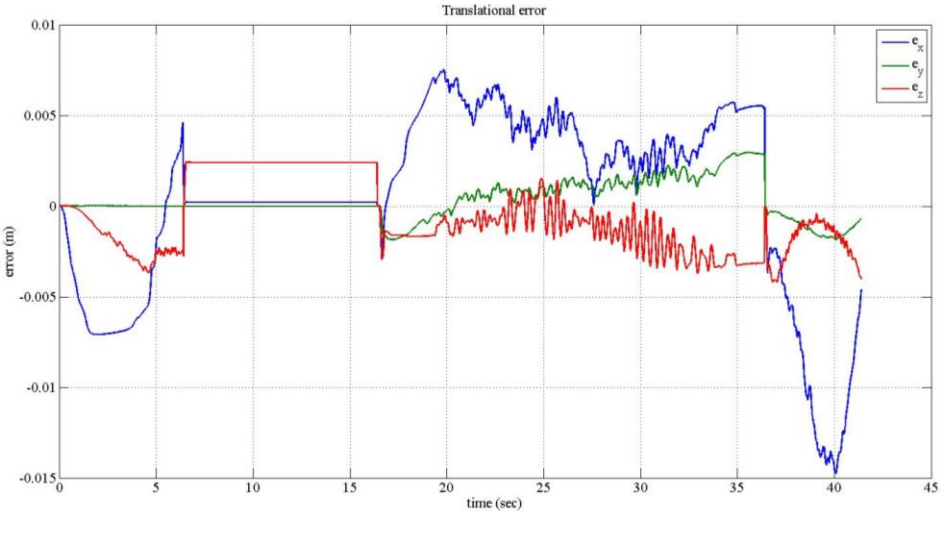


Figure B.74 : Translational errors.

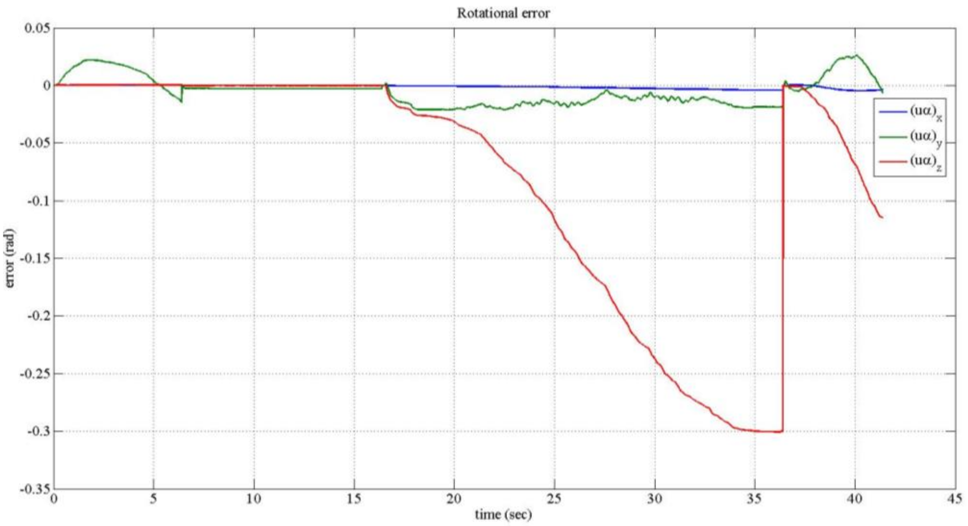


Figure B.75 : Rotational errors.

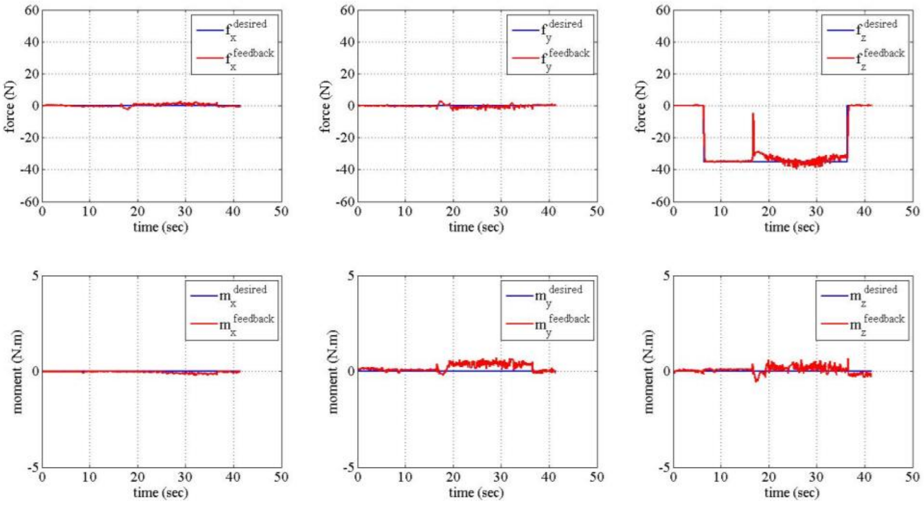


Figure B.76 : Desired and feedback forces.

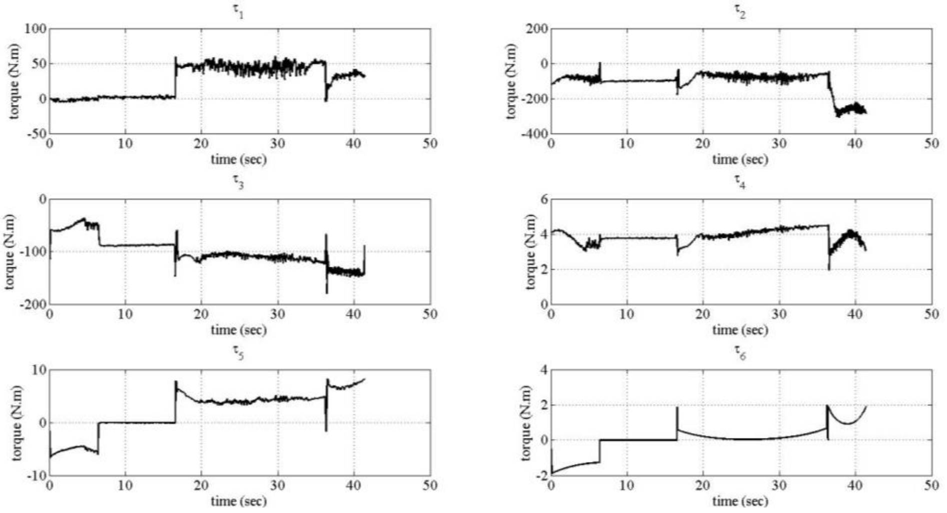


Figure B.77 : Applied joint torques.

Experiment 12:

- Drawing a line in XY-plane
- $\Delta x = 150\ mm$ and $\Delta y = 150\ mm$.
- Constant -40 N in z-direction.
- Motion time is 20 seconds in compliant motion phase.
- $K_{fp} = diag(4.75, 4.75, 4.75, 4.75, 4.75, 4.75)$
- $K_{fd} = diag(350, 350, 350, 350, 350, 350)$
- $K_{fI} = diag(2, 2, 2, 2, 2, 2)$

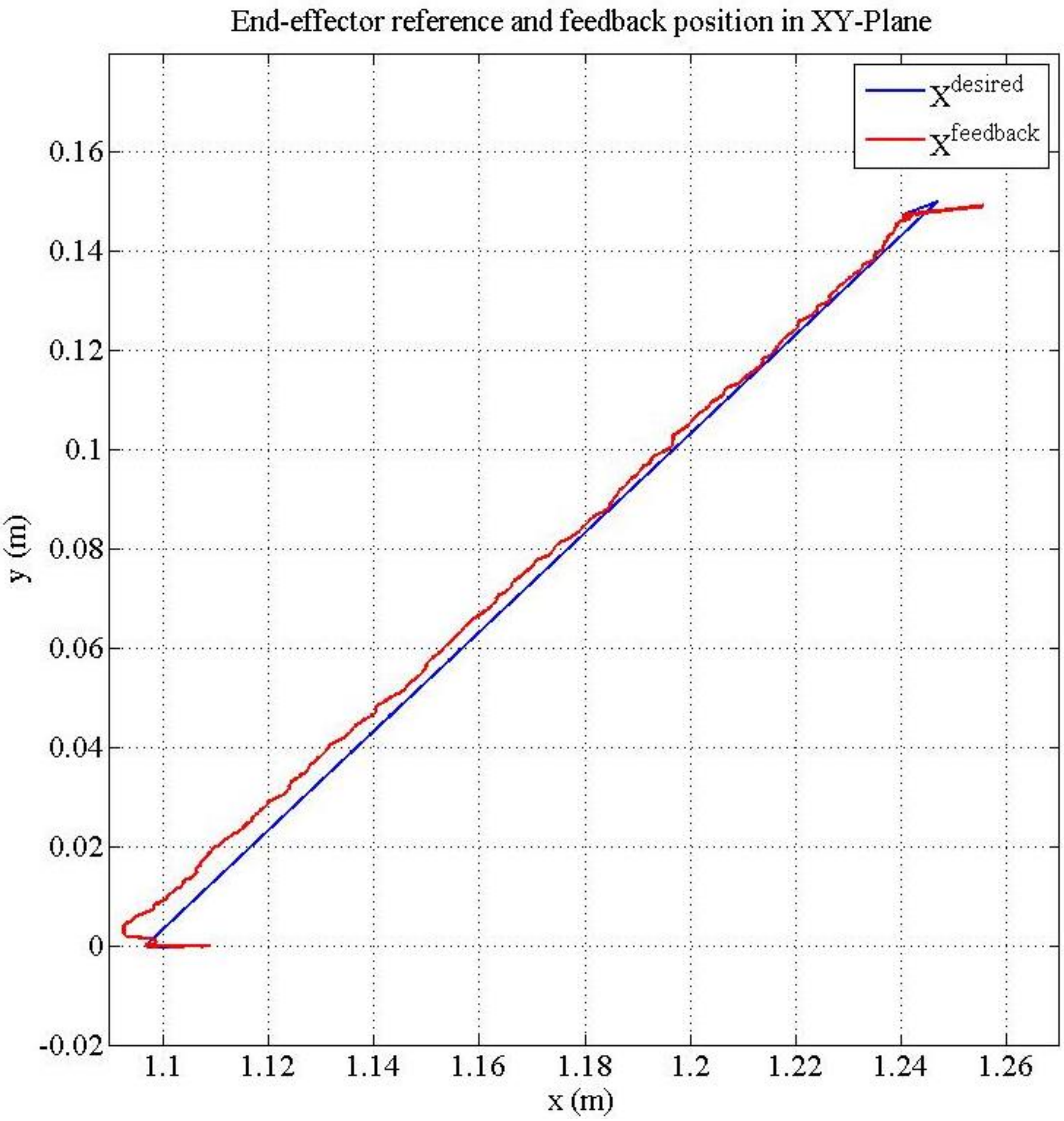


Figure B.78 : Desired and feedback path.

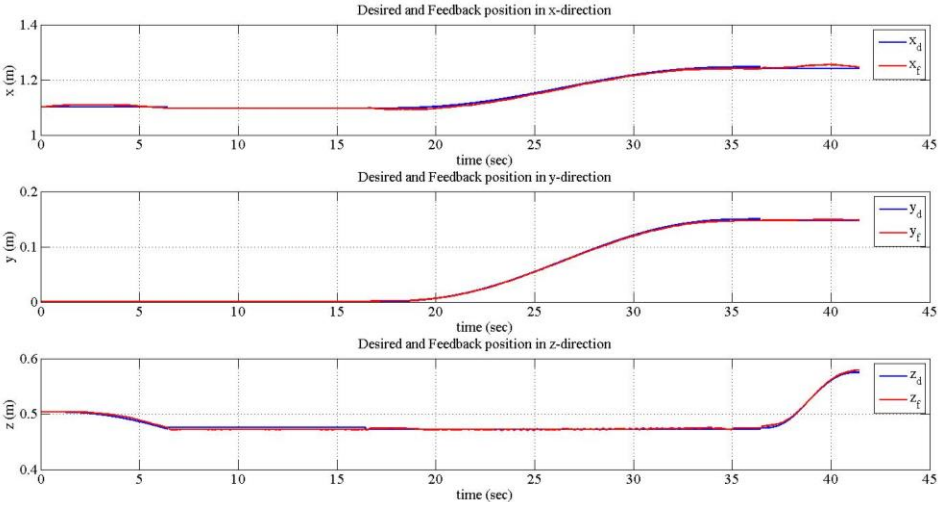


Figure B.79 : Desired and feedback positions.

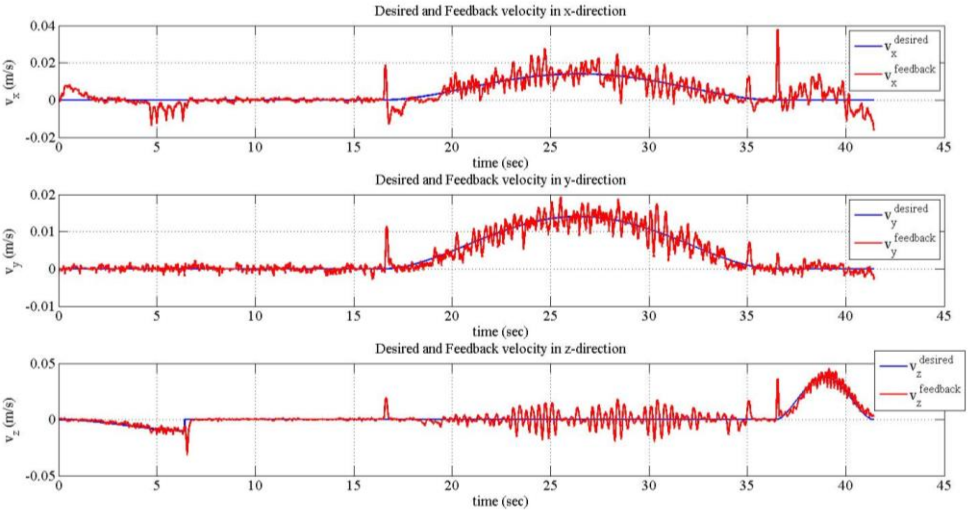


Figure B.80 : Desired and feedback velocities.

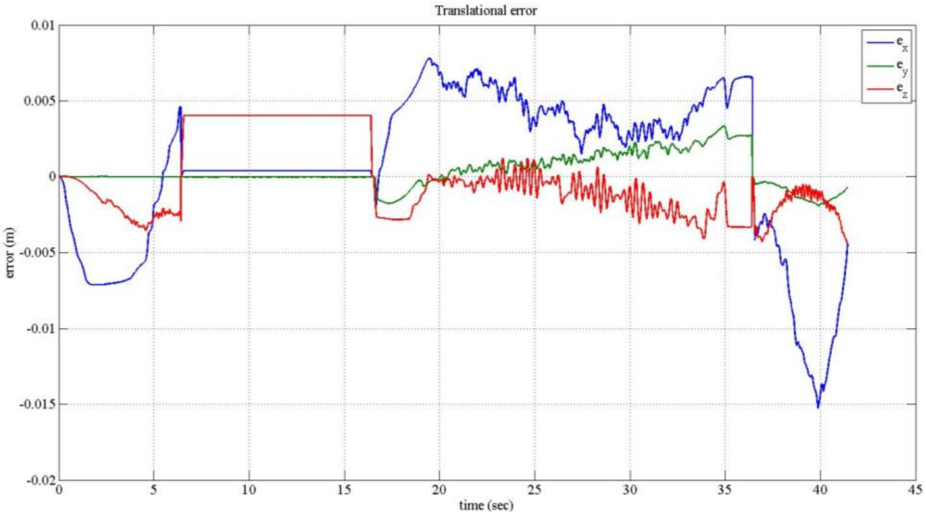


Figure B.81 : Translational errors.

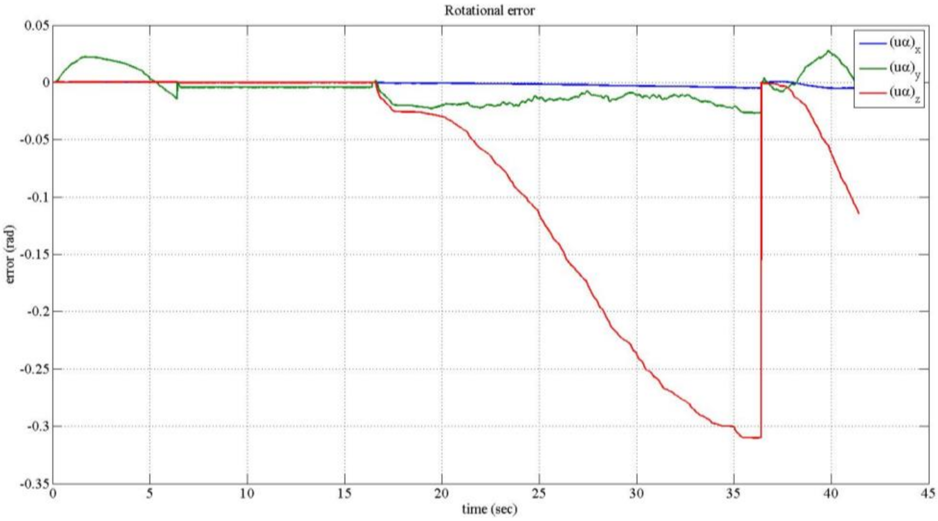


Figure B.82 : Rotational errors.

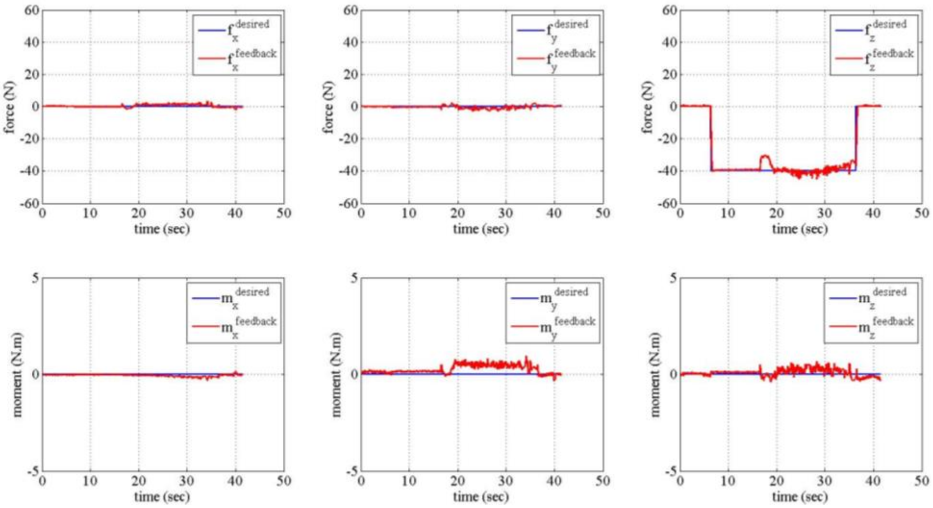


Figure B.83 : Desired and feedback forces.

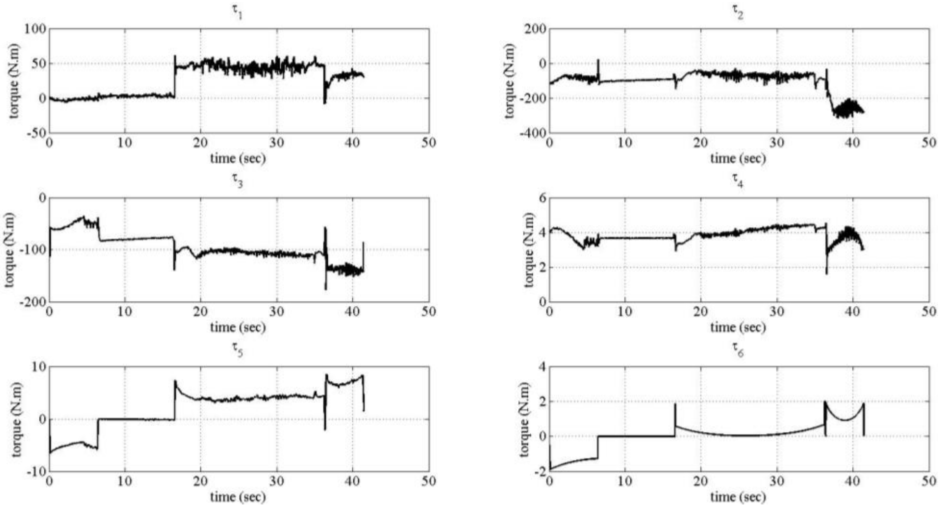


Figure B.84 : Applied joint torques.

APPENDIX C

```
// ----------------------------------------------------------------
// _Filter structure
// ----------------------------------------------------------------
typedef struct __Filter
{
        double fbkVal[MAX_NB_JNT];        // Feedback data to be filtered
        double estVal[MAX_NB_JNT];        // Estimated value

        double q[MAX_NB_JNT];             // Process noise covariance
        double r[MAX_NB_JNT];             // Measurement noise covariance

        double p[MAX_NB_JNT];             // Estimation error covariance
        double k[MAX_NB_JNT];             // Kalman gain

} _Filter;

_Filter frcFilter;

void main()
{
        kalmanFilter(&frcFilter);

}

// ----------------------------------------------------------------
// Kalman Filter
// ----------------------------------------------------------------
void kalmanFilter(_Filter* fltr)
{
        int i;

        for (i = 0; i < 6; i++)
        {
                fltr->p[i] = fltr->p[i] + fltr->q[i];

                fltr->k[i] = fltr->p[i] / (fltr->p[i] + fltr->r[i]);
                fltr->estVal[i] = fltr->estVal[i] + fltr->k[i] * (fltr->fbkVal[i]
- fltr->estVal[i]);
                fltr->p[i] = (1.0 - fltr->k[i])*fltr->p[i];
        }

}
```

APPENDIX D

Here, we will consider matrix inversion by partitioning which can be efficient for large matrices like 6 × 6 manipulator Jacobian matrix.

Let

$$J = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \tag{D.1}$$

$$J^{-1} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \tag{D.2}$$

where the elements of J are chosen to be

$$\begin{aligned} a_{11} &= \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix} \\ a_{12} &= \begin{bmatrix} J_{14} & J_{15} & J_{16} \\ J_{24} & J_{25} & J_{26} \\ J_{34} & J_{35} & J_{36} \end{bmatrix} \\ a_{21} &= \begin{bmatrix} J_{41} & J_{42} & J_{43} \\ J_{51} & J_{52} & J_{53} \\ J_{61} & J_{62} & J_{63} \end{bmatrix} \\ a_{22} &= \begin{bmatrix} J_{44} & J_{45} & J_{46} \\ J_{54} & J_{55} & J_{56} \\ J_{64} & J_{65} & J_{66} \end{bmatrix} \end{aligned} \tag{D.3}$$

and $a_{11}, a_{12}, a_{21}, a_{22}$ have the same sizes as $b_{11}, b_{12}, b_{21}, b_{22}$ respectively. The elements of J^{-1} can be found by either of the following formulas [19]

$$\begin{aligned} b_{11} &= (a_{11} - a_{12}(a_{22})^{-1}a_{21})^{-1} \\ b_{12} &= -(a_{11} - a_{12}(a_{22})^{-1}a_{21})^{-1}a_{12}(a_{22})^{-1} \\ b_{21} &= -(a_{22})^{-1}a_{21}(a_{11} - a_{12}(a_{22})^{-1}a_{21})^{-1} \\ b_{22} &= (a_{22})^{-1} + (a_{22})^{-1}a_{21}(a_{11} - a_{12}(a_{22})^{-1}a_{21})^{-1}a_{12}(a_{22})^{-1} \end{aligned} \tag{D.4}$$

or

$$\begin{aligned} b_{11} &= (a_{11})^{-1} + (a_{11})^{-1}a_{12}(a_{22} - a_{21}(a_{11})^{-1}a_{12})^{-1}a_{21}(a_{11})^{-1} \\ b_{12} &= -(a_{11})^{-1}a_{12}(a_{22} - a_{21}(a_{11})^{-1}a_{12})^{-1} \\ b_{21} &= -(a_{22} - a_{21}(a_{11})^{-1}a_{12})^{-1}a_{21}(a_{11})^{-1} \\ b_{22} &= (a_{22} - a_{21}(a_{11})^{-1}a_{12})^{-1} \end{aligned} \tag{D.5}$$

The choice between equation (D.4) and equation (D.5) can be made by considering relative simplicity of calculations.

Another useful formula is for the determinant of the partitioned matrix calculated with equation (D.6).

$$\begin{aligned} det(J) &= det(a_{22})det(a_{11} - a_{12}(a_{22})^{-1}a_{21}) \\ det(J) &= det(a_{11})det(a_{22} - a_{21}(a_{11})^{-1}a_{12}) \end{aligned} \tag{D.6}$$

If we solve $det(J) = 0$ symbolically, we can easily find the singular configurations of the manipulator.

CURRICULUM VITAE

Name – Surname	**:** Serhat AKBAŞ
Date and Place of Birth	**:** 04.07.1989, Altındağ
E-Mail	**:** serhatakbas89@gmail.com

EDUCATION :

- **B.Sc. :** 2012, Marmara University, Faculty of Engineering, Mechanical Engineering.