VISUAL SERVOING FOR TARGET TRACKING USING A NON-LINEAR 2 DEGREE OF FREEDOM SYSTEM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

BAHADIR BILGIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

AUGUST 2019

Approval of the thesis:

VISUAL SERVOING FOR TARGET TRACKING USING A NON-LINEAR 2 DEGREE OF FREEDOM SYSTEM

submitted by **BAHADIR BILGIN** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _______________

Prof. Dr. M. A. Sahir Arıkan
Head of Department, **Mechanical Engineering** _______________

Assoc. Prof. Dr. Melik Dölen
Supervisor, **Mechanical Engineering, METU** _______________

Assist. Prof. Dr. A. Buğra Koku _______________
Co-supervisor, **Mechanical Engineering, METU**

Examining Committee Members:

Assist. Prof. Dr. Ulaş Yaman
Mechanical Engineering, METU _______________

Assoc. Prof. Dr. Melik Dölen
Mechanical Engineering, METU _______________

Assist. Prof. Dr. A. Buğra Koku
Mechanical Engineering, METU _______________

Assist. Prof. Dr. Kıvanç Azgın
Mechanical Engineering, METU _______________

Assist. Prof. Dr. Kutluk Bilge Arıkan
Mechanical Engineering, TED University _______________

Date:

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

Name, Surname: Bahadır Bilgin

Signature :

ABSTRACT

VISUAL SERVOING FOR TARGET TRACKING USING A NON-LINEAR 2 DEGREE OF FREEDOM SYSTEM

Bilgin, Bahadır
M.S., Department of Mechanical Engineering
Supervisor: Assoc. Prof. Dr. Melik Dölen
Co-Supervisor: Assist. Prof. Dr. A. Buğra Koku

August 2019, 121 pages

Target tracking and pointing systems are utilized in a broad range of applications such as automated surveillance camera applications and satellite tracking applications. In most of these applications, conventional 2-axis gimbal systems are preffered. However, there are some chronic problems, such as their singularity and cabling problems, of the gimbal design that are usually overlooked and traded off. In this thesis, a 2-degree of freedom parallel mechanism that do not suffer from any of the chronic problems of gimbals is designed and prototyped. This 2 degree of freedom prototype (parallel tracking platform) has two output angles which are a non-linear and coupled function of the input actuator angles. A camera is attached to the camera platform of parallel tracking platform for visual servoing purposes. A visual servo controller is developed to acquire an image of a randomly moving target. Aim of the visual servo controller is to fix the target at the center of the field of view. The visual servo controller captures the target with a vision system that can track it, then sends command signal to a motion controller that drives the actuators. The custom motion controller is developed on a microcontroller and employs cascaded prorptional-integral-derivative

(PID) loops as its control architecture. The motion contoller sends position feedback to visual servo controller which is used in kinematic look-up tables to determine the orientation of the camera platform. Position information is used together with image information to determine the command signal in visual servo controller. An experimental setup is set to asses the performance of the parallel tracking platform and visual servo algorithms. The visual servo controller, the motion controller and parallel tracking platform compose a sufficient test setup for target tracking and future visual servo control studies.

Keywords: Visual Servo Control, Target Tracking, Parallel Mechanism, Motion Control

ÖZ

LİNEER OLMAYAN 2 SERBESTLİK DERECELİ SİSTEM İLE HEDEF TAKİBİ İÇİN GÖRÜNTÜLÜ SERVOLAMA

Bilgin, Bahadır
Yüksek Lisans, Makina Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Melik Dölen
Ortak Tez Yöneticisi: Dr. Öğr. Üyesi. A. Buğra Koku

Ağustos 2019 , 121 sayfa

Görüntülü servo algoritmaları hedef takibi uygulamaları, otomatik gözetleme kamerası uygulamaları ve uydu takibi uygulamaları gibi pek çok alanda kullanılır. Bu uygulamaların çoğunda gimbal mekanizmaları kullanılır fakat gimbal mekanizmalarının göz ardı edilen tekillik ve kablaj gibi pek çok problemi vardır. Bu tezde gimbal mekanizmalarında olan problemleri yaşamayan bir paralel mekanizma tasarlanmış ve üretilmiştir. Bu 2 serbestlik dereceli mekanizmanın girdisi ve çıktısı arasında lineer olmayan akuple bir ilişki vardır. Görüntülü servo uygulaması için mekanizmaya bir kamera takılmıştır. Görüntülü servo algoritmasının amacı hedefi bulup ortalamaktır. Görüntülü servo kontrolcüsü hareket kontrolcüsüne komutlar göndererek eyleyicileri sürer. Hareket kontrolcüsünde kademeli bir kontrolcü kullanılmıştır. Hareket kontrolcüsünden gelen pozisyon bilgisi kinematik tablolar ile anlamlandırılarak görüntülü servo kontrolünde kullanılır. Bu tezde paralel mekanizmanın ve görüntülü servo algoritmalarının performansı test edilmiştir. Görüntülü servo kontrolcüsü, haraket kontrolcüsü ve paralel takip mekanizması bir araya gelerek hedef takibi için uygun bir test platformu oluşturmuştur.

Anahtar Kelimeler: Görsel Servo kontrol, Hedef Takibi, Paralel Mekanizma, Hareket Kontrolü

To my family

ACKNOWLEDGMENTS

First and formost, I would like to express my deepest appreciation to my supervisor Dr. Melik Dölen for his patience that cannot be underestimated. His guidance, support and constructive criticism made this thesis possible. His willingness to give his time so generously has been very much appreciated. I'm extremely grateful to him for sacrificing his sleep in order to help me when I most needed it. I had great pleasure of discussing every detail with him.

I cannot begin to express my thanks to my co-advisor Dr. Buğra Koku for his valuable advice and guidance not only during my thesis but my entire academic life. His courses on mechatronics inspired me to study futher in this area. I would like to thank him for providing the brushed DC motors that are used in the parallel tracking platform and 3-D printing the first prototype for me.

I'm extremely grateful to Dr. Ulaş Yaman for offering me a place in his laboratory for the last year. Furthermore, I very much appreciate his kindness which have always encouraged me.

I am also grateful to Dr. Kıvanç Azgın for guiding me during my undergraduate years. I appriciate his willingness to listen attetively and his funny comments which always put a smile on my face.

Special thanks to Dr. Kutluk Bilge Arıkan for attending my thesis defence and his constructive feedback.

I would also like to extend my deepest gratitude to Dr. Murat Durmaz for his enthusiastic encouragement and insightful advice and his profound belief in my work.

I gratefully acknowledge Onur Yarkınoğlu for supporting my self-improvement. I owe a great deal of my technical knowledge and experience to him.

Many thanks to Kaan Erkoç. He extended a great amount of assistance in software

development when I needed it.

I'd like to recognize the patience of İsmail Özçil. He always picked up the phone.

My thanks extend to Smart Manufacturing Lab members for their support and belief in me.

I'm indebted to all my friends for their tolerance to my busy schedule. Special thanks should also go my workstation and 3-D printer; without them to the completion of my dissertation would not have been possible.

Last but not least, I would like to extend my sincere thanks to family. I'm extremely grateful to my father and mother for their unparalleled support.

I could not find a phrase that can express my gratidute to Zeynep Ünal.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

2D	2 Dimensional
3D	3 Dimensional
DOF	Degree of Freedom
USB	Universal Serial Bus
GPIO	General Purpose Input Output
FPU	Floating Point Unit
QEI	Quadrature encoder Interface
PWM	Pulse Width Modulation
ADC	Analog to Digitac Converter
FOV	Field of View
ROI	Region of Interest
ZOH	Zero Order Hold
FPS	Frame Per Second
RMS	Root Mean Square
LUT	Look Up Table
PTP	Parallel Tracking Platform

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

Target tracking and pointing systems are utilized in a broad range of applications. These systems are used to point satellite dish for communication between ground to ground / ground to air / air to air systems. Furthermore, they are employed for radar applications, to point a camera to gather information for unmanned aerial vehicles and surveillance systems. Target tracking and pointing system can also be used to accelerate/move/hold a device for sensor calibration systems and to point and track targets for military applications [6].

In most of these applications, conventional 2-axis gimbal systems are preffered. In order to control the tracking and pointing of gimbal systems with high precision, sophisticated controllers are developed. Gimbals are simple in design: device platform is attached to a rotary axis which is attached on top of another rotary axis. As a result, actuation on rotary axes directly rotate the system around their corresponding axes. Hence, the axes are coupled. Since there are minimal amounts of joints between the actuators and the device platform, it can be said that errors caused on the device platform are minimal. Control of these gimbals are well documented and straightforward. [14–18] . However, there are some chronic problems of the gimbal design that are usually overlooked and traded off.

On conventional 2-axis gimbals, outer axis actuator has to carry the weight of the inner axes which negatively effect its dynamic response characteristics. This fact is usually offset by using bigger actuators on the outer axis. However, it can be said that system characteristics on azimuth and elevation axes are different and non-symmetric.

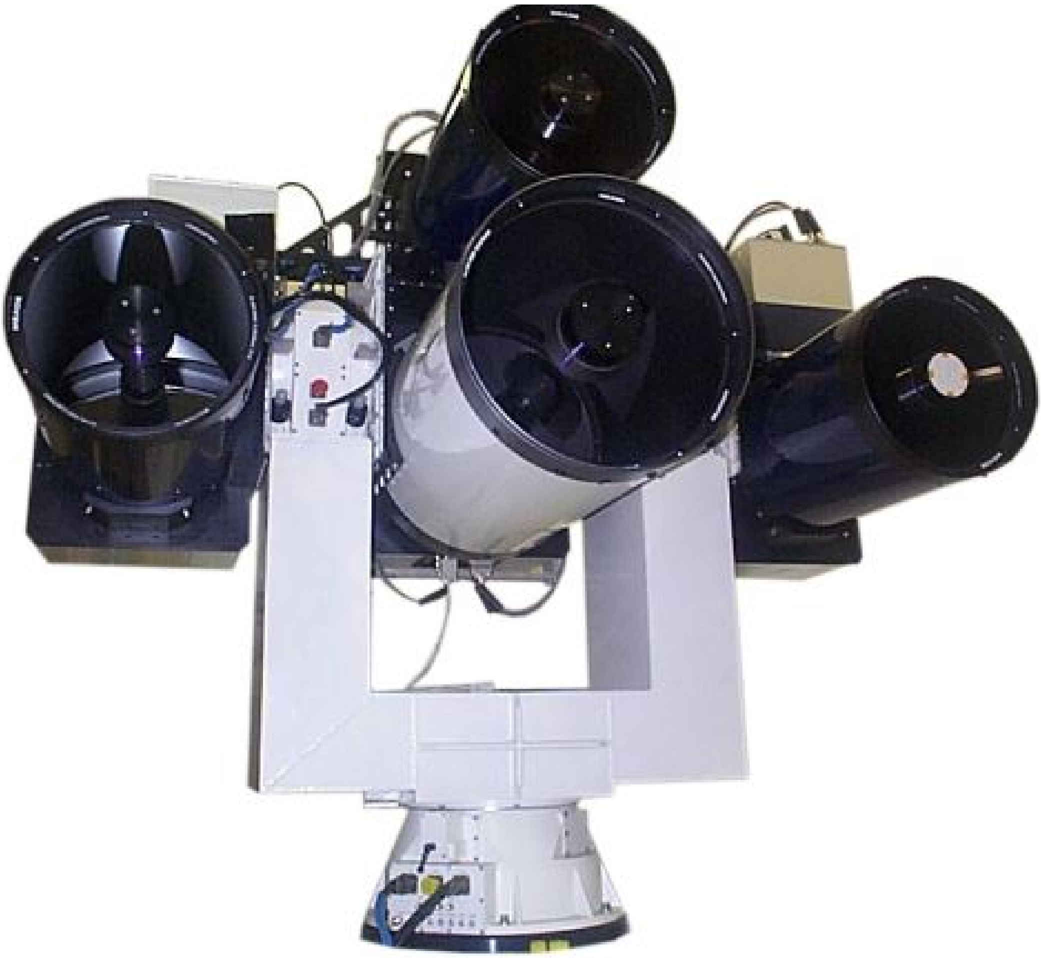


Figure 1.1: Military Application by COBHAM [6].

Conventional 2-axis gimbals employ a singularity in their hemisphere. In order to get out of the singularity (whenever it occurs), special control methods, addition of new equipment/axes are required. Furthermore, cabling of the gimbal and its device platform usually impose restrictions on the gimbal. Gimbals cannot rotate around their hemisphere freely due to this restriction without the usage of equipment called slip rings or rotary unions. These equipment are attached to an axis to allow electricity, fluid or other mediums to be transferred from a stationary part to the rotary part while allowing full turn continuous motion. There are different slip-rings for low frequency signals, high voltage signals, high frequency signals, shielded signals, fiber optics and fluids. Especially if using multiple of these mediums is required in a system; the design, size, and cost of the system as well as interchangeably are greatly affected [19]. Even if a system is made to be fully rotating with slip-rings in reality, changes in the device platform of the system are not easy and results in a lot of backtracking or surrendering the ability to fully rotate.

Different mechanisms are proposed in an attempt to solve the chronic problems of conventional gimbals [7,11,20–25] . These proposed mechanisms are not that simple. As a result, they have axes with coupled motion and lesser precision due to additive errors on joints. However they have their own benefits, they are singularity free, can easily access more than a hemisphere, fully symmetrical motion, easily reachable and changeable components/actuators/sensors. Due to their design the distance between base and device platform does not change. As a consequence, they require no slip rings. Device platform configuration allows for complex designs with no restriction on cabling. In addition, device platform configuration can be changed with no drawback on design throughout the use of the system. Due to their highly coupled and nonlinear nature, these mechanisms require a delicate design of controllers and control architectures. They offer many benefits if the control system is built carefully. These mechanisms also offer a good environment for practicing control algorithms for coupled systems, pointing systems and system with visual feedback. Canfield mechanisms are also used for similar pointing purposes [8,22,24,25].

1.2 Contributions

It is apparent that non-conventional 2-DOF pointing mechanisms are beneficial in some special cases over conventional gimbal designs. The performance of these pointing or tracking mechanisms are related to their control methods. The aim is to test different control methods on a pointing mechanism and compare the results. Since there are no mechanisms at hand, one has to be created from scratch. This mechanism is used in this thesis and in the future it can be used to test and compare tracking and visual servoing control algorithms. Mechanism is designed and manufactured to build this test platform/environment. In order to control the pointing mechanism, electronical design and implementation of control electronics are conducted. Furthermore, motion control software is implemented. A visual servo controller is built and tested.

1.3 Organization of Thesis

This thesis is organized as following: In Chapter 3, a non-conventional 2-DOF mechanism is designed and important remarks about the mechanical aspects of the design are made. In Chapter 4 , kinematic model of the system is obtained. Forward and inverse kinematics of the system are mapped to be used in control architecture for decoupled control. In Chapter 5, dynamic model of the system is constructed and several software experiments are conducted to better understand the characteristics of the parallel tracking platform. In Chapter 6, electronical design and software architecture are briefly explained. Then, motion control architectures built on the system are explained in depth. In Chapter 7, visual servo control architectures are explained. In Chapter 8, experimental results and their implications are discussed. Thesis is concluded in Chapter 9.

CHAPTER 2

REVIEW OF THE STATE OF THE ART

2.1 Introduction

A mechanism can be defined as a serial mechanism, parallel mechanism or hybrid mechanism depending on the arrangement of its connections and joints [26]. Serial mechanisms are the ones with sequential connections of joints and links. While naming these joints, joint types are named from base to the end point of mechanism which can be shortened as (R) for revolute, (P) for prismatic, (U) for universal, (S) for spherical. Conventional gimbals are RR (Revolute-Revolute) type serial mechanisms. Parallel mechanisms are mechanisms where there are multiple serial chains/paths/arms that connect the base of the system to a common end point. Hybrid mechanisms are a chain of one or more serial and parallel mechanisms. In this thesis, the main area of interest is parallel mechanisms used for pointing and tracking applications. These pointing mechanisms utilize electrically powered actuators to orient and control the end effector (tip point) of the system. Hence, the mechanisms of interest are 2-DOF active parallel mechanisms. Different active parallel 2-DOF mechanisms are proposed for their benefits over conventional gimbal systems [20, 21, 27–36]. These systems can be actuated using different electronics that use different control architectures. In this chapter, utilization of different mechanisms are surveyed to create a basis for the design process.

2.2 Parallel Mechanisms Concepts

Firstly, different parallel mechanisms from literature should be investigated before elaborating the design process. Some of the good candidates from the state of art are used in conceptual design for mechanical design of the system in Section 3.2.

2.2.1 Omni-Wrist Family

Multiple mechanisms are designed and patented by the company Rose-Hime Designs. Parallel mechanisms manufactured by the company created under the family name of "Omni-Wrist". Omni-Wrist III, Omni-Wrist V and Omni-Wrist VI are described in the website of the company [37]. Detailed technical information regarding these devices are given in Refs. [10, 20, 35, 38].

2.2.1.1 Omni-Wrist III

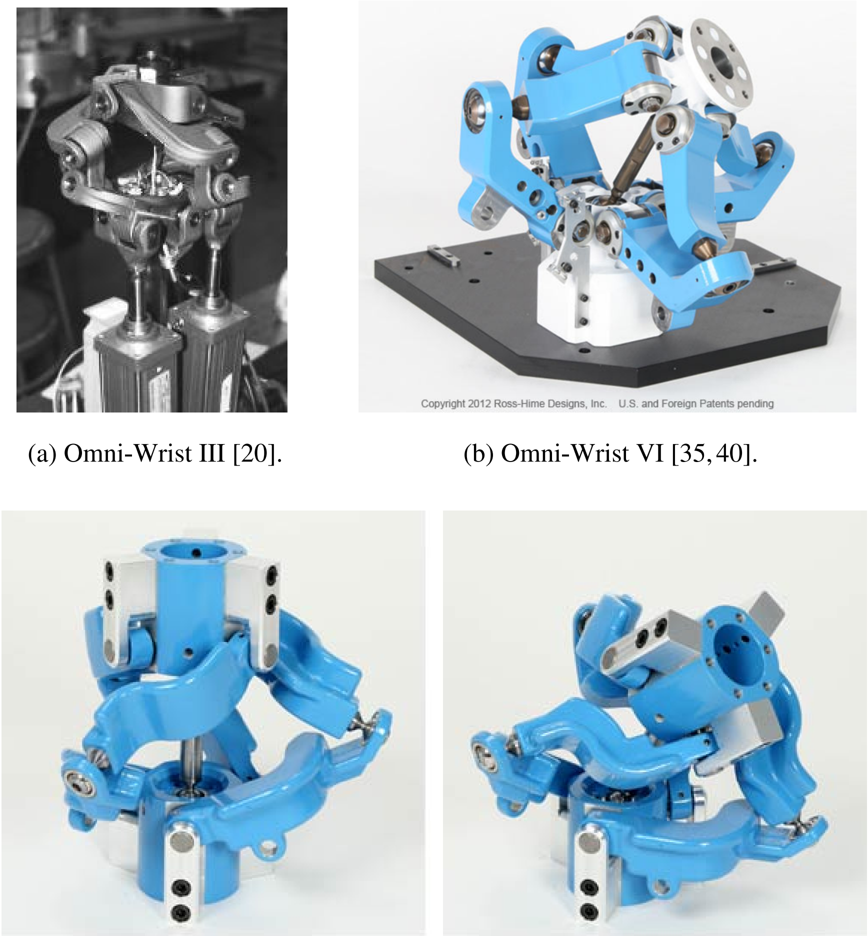
Omni-Wrist III [20] is a $4R^4$ mechanism which emulates the kinematics of a human wrist as shown in Fig. 2.1a. It is primarily designed for optical applications. Computer controlled mechanism is driven by two linear motors. Omni-Wrist III is capable of moving on a full hemisphere through pitch/yaw motion. Linear actuators are employed for mimicking the muscles in human body. Mechanism is singularity free. Omni-Wrist III utilizes translational motion for actuation instead of rotational motion in order to obtain maneuverability without backlash that would be caused by gearbox.

2.2.1.2 Omni-Wrist V

Omni-Wrist V [38, 39] is a SS,3RSR mechanism which can be seen in Fig. 2.1c. The patent for the mechanism [38] claims that it can be controlled by either two rotary actuators or three linear actuators. Increased degree of freedom is achieved by modifying the device platform and removing the central column, essentially making the mechanism 3RSR. Mechanism can move on a full hemisphere through pitch/yaw motion with a repeatability of 0.05 degrees [35].

2.2.1.3 Omni-Wrist VI

Omni-Wrist VI [35,40] is a SS, 4RSR mechanism that is given in Fig. 2.1b. SS serial chain constrains end point of the pointing mechanism to a virtual sphere. System has two degrees of freedom and therefore can be actuated by actuating two of the revolute joints on its arms. System offers benefits of parallel mechanisms. Due to the SS serial chain in the middle carrying most of the weight on its central position and portion of the weight in other positions, the system has better load carrying properties if compared to other parallel mechanisms [26].



Copyright 2012 Ross-Hime Designs, Inc. U.S. and Foreign Patents pending

(a) Omni-Wrist III [20].

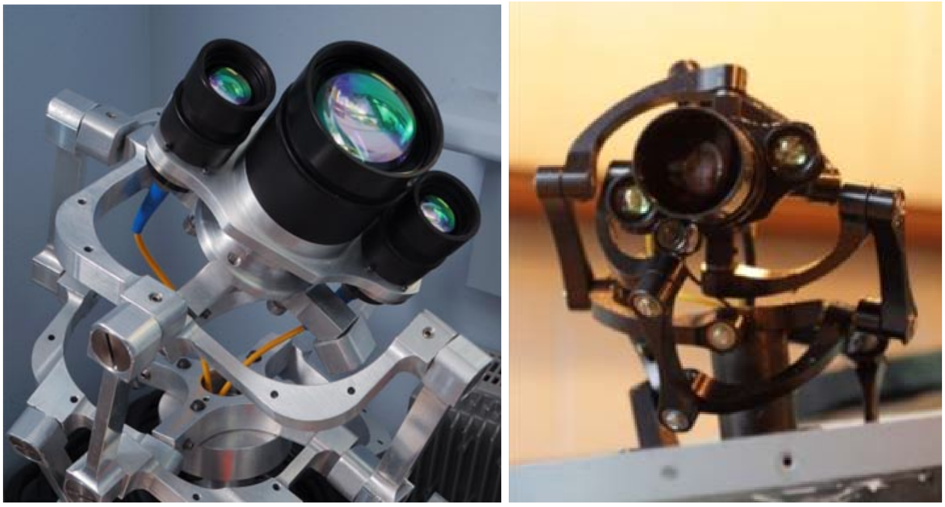
(b) Omni-Wrist VI [35,40].

(c) Omni-Wrist V [38,39].

Figure 2.1: Wide Angle Gimbal Concepts.

2.2.2 Wide Angle Gimbal

Wide Angle Gimbal is a 4R[4] mechanism used for optical applications by Sofka et al. [11], Nikulin et al. [21]. The mechanism is in the form of a parallel robotic linkage that consists four arms, each comprised of three links and four joints that connect the stationary base to device platform as in Fig. 2.2a. Two rotary actuators drive two of the four links attached to the stationary platform to achieve a full hemisphere motion.



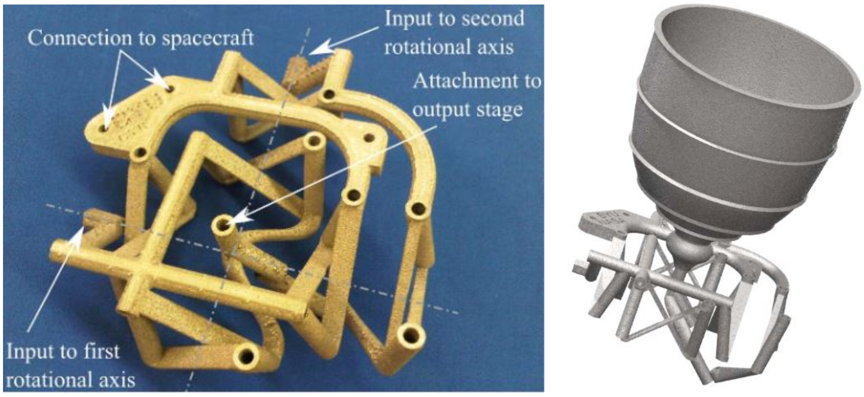
(a) Parallel manipulator by Sofka et al. [11].

(b) Parallel manipulator by Nikulin et al. [21].

Figure 2.2: Wide Angle Gimbal Concepts.

2.2.3 Monolithic Mechanism

Merriam et al. [7] use a monolithic structure as a pointing device for thrusters for space applications which can be seen in Fig. 2.3. This 3D-printed structure is made of titanium. As a result of being a monolithic structure, the system moves only with bending motion. Manipulator contains no bearings and actuators bend/elastically deform the material. This structure is proposed as a candidate for space applications. In zero pressure environment, it is challenging to use bearings since lubrication tend to disperse. Bearingless design eliminates many friction, backlash, lubrication and wear problems that occur in low environmental pressure.


Connection to spacecraft
Input to second rotational axis
Attachment to output stage
Input to first rotational axis

(a) 3-D printed titanium monolithic mechanism [7].

(b) Space-centered application demonstration [7].

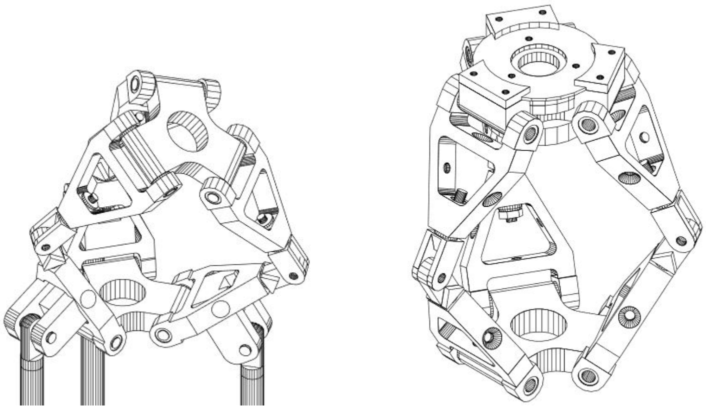
Figure 2.3: Monolithic Mechanism [7].

2.2.4 Canfield Joint

Canfield joint (or Canfield Carpal Wrist) is a 3RSR mechanism developed for space applications by SL Canfield [8] which can be seen in Fig. 2.4. System offers many of the benefits offered by other mechanisms such as symmetry, singularity-free workspace, remotely located actuators, and improved dynamic characteristics. However, this mechanism is intrinsically a 3-DOF system that requires actuation on all three lower links. Thanks to this feature, the mechanism can move back and forwards while being locked to a target.

2.2.5 High Angle Active Link

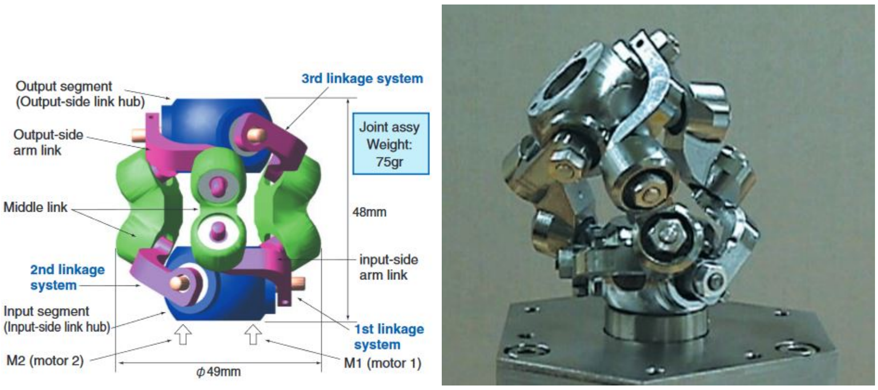
High Angle Active Link [36] is a $3R^4$ mechanism developed for optical applications in Figs. 2.5a2.5b. Just like other mechanisms described in this section, this mechanism also offers singularity-free workspace and remote actuators. Note that in order to achieve the desired DOF, lower six revolute joint axes and upper six revolute joint axes have a common point of intersection.



(a) Canfield Carpal Wrist Concept A (Linear Actuation).

(b) Canfield Carpal Wrist Concept B (Rotating Actuation).

Figure 2.4: Canfield carpal wrist concepts [8].



(a) High Active Link [36].

(b) High Active Link Prototype [41].

Figure 2.5: Parallel mechanism concepts.

2.3 Motion Control of Parallel Mechanisms

2.3.1 Control of Omni-Wrist III

Before starting the design process of controller for Omni-Wrist III, a common design procedure in robotics is followed [9]. Firstly, kinematic- and dynamic performance objectives are specified. Actuators and drive electronics are selected to satisfy those performance objectives and they are used to obtain a mathematical motor model [20]. A reference model, which is used with the mathematical model of the system to obtain the fully decoupling controller for the system is developed [10]. Mechanical system is designed with the considerations of DOF and range of motion. Kinematic operation of the mechanism is evaluated and improved iteratively. After a satisfactory solution is found for the kinematic- and mechanical design by iteration, the inverse kinematics of the system is solved. Finally, this solution is used in controller design. This design procedure can be seen in the Fig. 2.6.

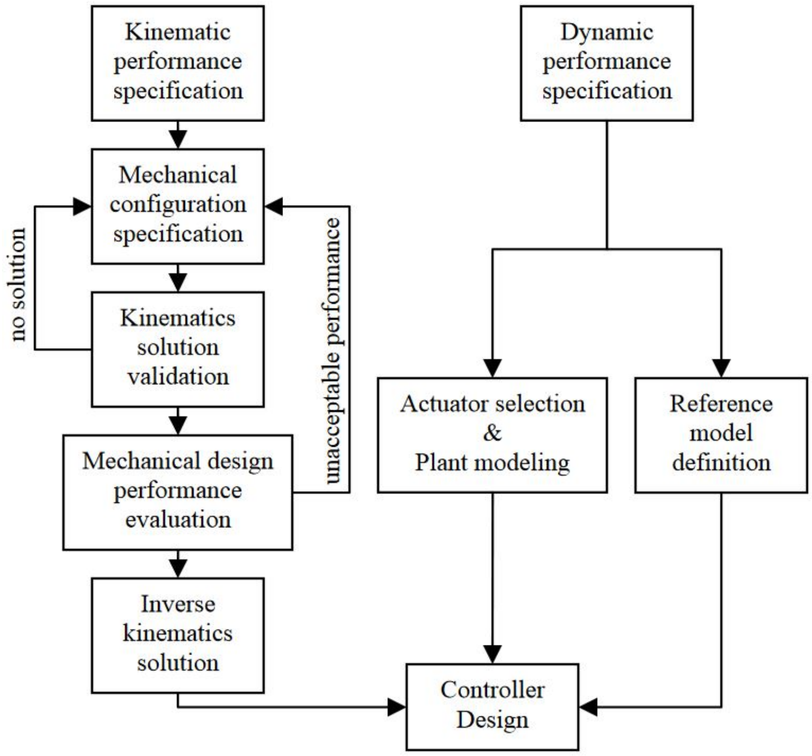
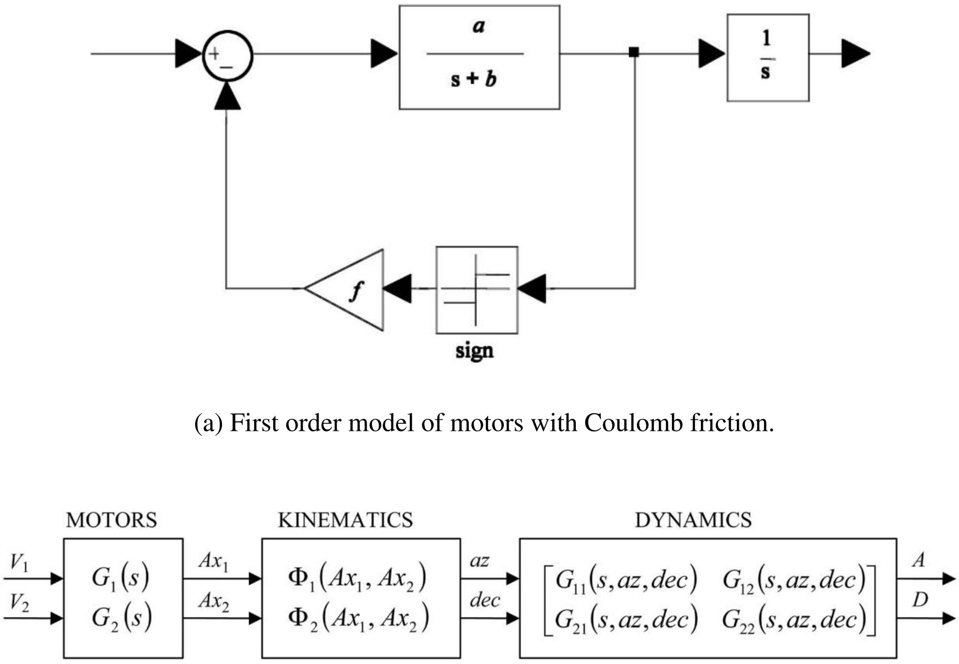

Kinematic performance specification
Dynamic performance specification
Mechanical configuration specification
no solution
unacceptable performance
Kinematics solution validation
Actuator selection & Plant modeling
Reference model definition
Mechanical design performance evaluation
Inverse kinematics solution
Controller Design

Figure 2.6: Integrated design procedure [9].

In order to control Omni-Wrist III, linear motor encoder signals are associated pitch- and yaw coordinates of the sensor mount by kinematic equations using Denavit-Hartenberg method. Hence, solutions for the inverse- and forward kinematics are found. Motors are defined such that, the relationship between manipulated inputs (input voltages) of actuators and displacement output is linear. Motor model also includes a Coulomb friction. Additionally, system response characteristics are mapped for controller by collecting responses of the system on different ranges of hemisphere to have a dynamics estimation of the system. Step references are sent to actuators and a velocity response of a first order system is collected. Collected data is used to fit a and b values in the Eqn. (2.1) by least squares estimation [20]. Control model of the motor along with that of the system can be seen in Fig. 2.7.

$$G(s) = \frac{a}{s+b} \tag{2.1}$$



(a) First order model of motors with Coulomb friction.

(b) Initial control model of Omni-Wrist III.

Figure 2.7: Control models used on Omni-Wrist III [10].

Following the acceptable performance of the system, three new controllers are implemented to improve the performance of previous controller [10]. First, a state-space controller, which decouples the states in steady state, is constructed. Later, a linear model following controller is built that decouples the system only on steady state. Consequently, a linear model based tracking controller that achieves full decoupling is built. State equations Eqns.(2.2)-(2.4) are used for the purpose of the first controller:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{2.2}$$

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} \tag{2.3}$$

$$\boldsymbol{u} = \boldsymbol{W}\boldsymbol{r} - \boldsymbol{F}\boldsymbol{x} \tag{2.4}$$

where $\boldsymbol{x}$ is input state vector including the velocity and position of linear actuator, $\boldsymbol{y}$ is the output (position of the linear actuator) and $\boldsymbol{u}$ is manipulated input to motors. Controller matrix $\boldsymbol{F}$ is introduced so that eigenvalues of closed loop matrix $(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{F})$ can be placed on desired locations. Matrix $\boldsymbol{W}$ is selected for steady state error elimination whenever there is a step reference. This control model can be seen in Fig. 2.8. This control model has unsatisfactory performance. In order to have better performance, control model is improved by adding a following mechanism. This newly added component represents the desired behavior of closed loop motor control system as shown in Fig. 2.9.

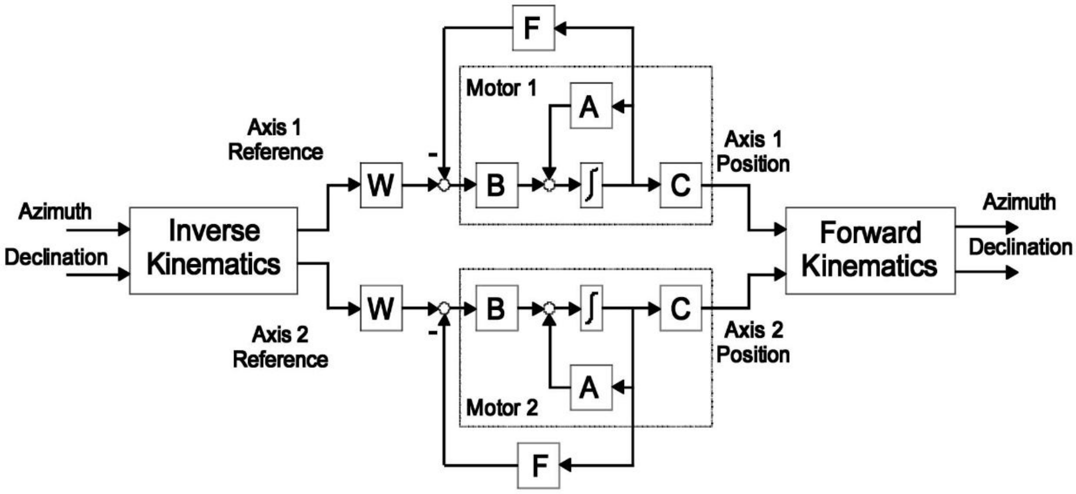


Figure 2.8: State variable controller without full decoupling for Omni-Wrist III [10].

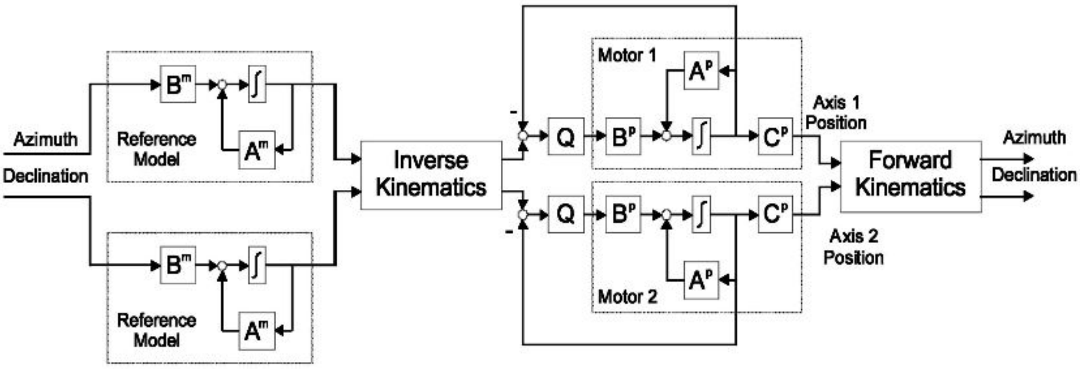


Figure 2.9: Full decoupling controller for Omni-Wrist III [10].

2.3.2 Control of Wide Angle Gimbal

Control of wide angle gimbal is similar to the control of Omni-Wrist III [11] [21]. Kinematic equations governing the motion of the mechanism are obtained. Non-linearities and coupling are eliminated in a fully decoupling controller. At high speed, the non-linearaties cause inconsistent movement depending on the position of the system. For this reason, different gains are mapped to different ranges of angular motion. System switches gains depending on its current angular position for better consistency of movement throughout the full range of motion. In Fig. 2.10, step response of the system in whole range of motion is plotted on left hand side. Hence, step responses are obtained in different directions in no particular order. By using a controller that switches gains depending on the position of system, similar characteristics and step responses can be obtained for different positions of the system as it can be seen in the plot on the right of Fig. 2.10.

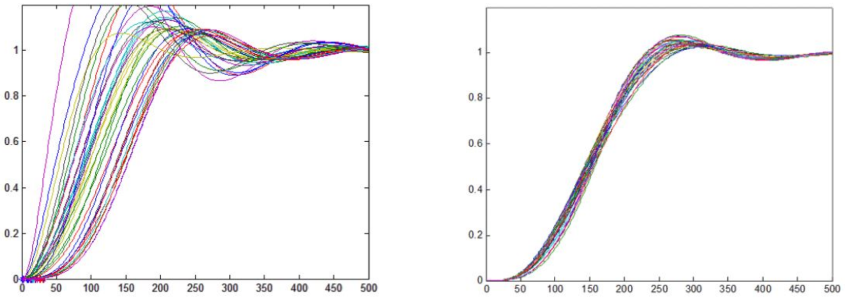


Figure 2.10: Parallel manipulator consistency due to gain switching [11].

2.3.3 Control of High Angle Active Link

High angle active link is controlled using a relation between arm rotation angles to the bend and traverse angles of the tip. Inverse transformation of this relation is obtained to give arm rotation angles for a commanded plant orientation. Positioning accuracy is improved by employing a control method to cancel the backlash caused by the drive mechanism [41].

2.3.4 Control of a 3-PUPU mechanism

Two functions (micro motion and macro motion) are defined for the control of a 3-PUPU mechanism [12]. In order to fulfill the two functions, two controllers are proposed. Two different models are developed for each of the controllers. One of the models are for a 3-UPU system for micro motion control while the macro motion control system uses a 3-PUU system model. For macro motion control, a proportional-integral-derivative (PID) controller is utilized. In order to cancel out the disturbances acting on the platform when it is moving, A H_2 optimal controller is developed against micro vibration control. In order to detect the micro motion of the platform, a triple-axis acceleration sensor is attached to the platform for measurement. Hence, high performance can be obtained using acceleration feedback control strategy to design a robust controller [42]. Analog-to-digital (ADC) signal converters are used to feed back the sensor signals to controllers so that driving outputs can be calculated. These calculation results are sent to actuators to realize vibration isolation.

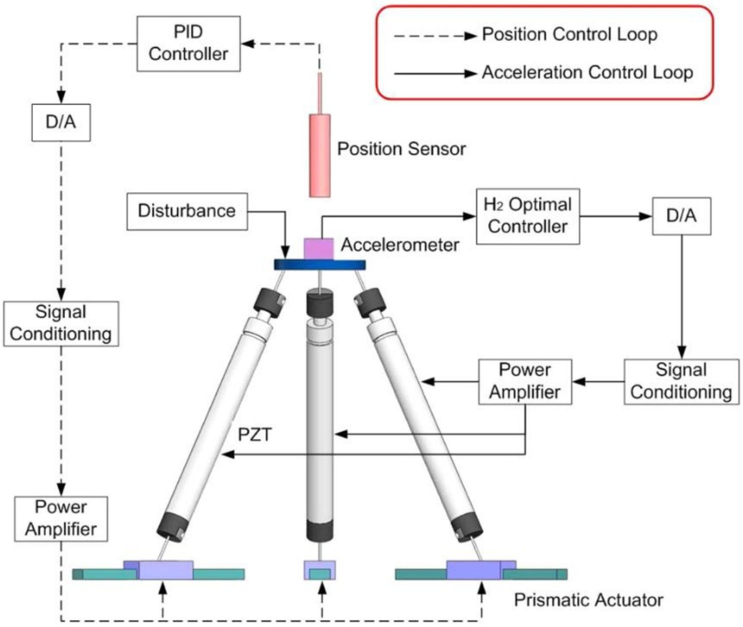


Figure 2.11: Figure of the positioning and vibration control of 3-PUPU system [12]

2.4 Industrial Motion Control

Cascade motion control systems, which make use of successive PID controllers inside a number of nested control loops, are frequently utilized in various motion control and industrial applications [43] including:

- Servo-motor drives and power electronics [44,45],
- CNC machine tools [46],
- Factory automation and robotics [47],
- Aerospace systems and more [48].

Popularity of this classical control scheme in industry is due to the fact that the implementation along with the tuning procedure of a de-facto standard cascade controller are quite simple and well-established in practice. Furthermore, the command tracking- and the disturbance rejection performance of a well-tuned controller are satisfactory (for all intents and purposes) even under the presence of varying physical system parameters such as mass moment of inertia, viscous damping, static/kinematic friction coefficients etc. Despite the fact that the cascade control systems could be conveniently replaced by an equivalent state-space controller, the reign of this classical technique still continues owing to the fact that the topology allows control engineers to be in close contact with the physics of the motion control problem without getting bogged down with abstract mathematical entities. Commercial servomotor drivers incorporate an enhanced version of the cascade motion control method briefly outlined in the previous section. For instance, Figs. 2.12 - 2.14 illustrate the block diagram of the control systems used by Kollmorgen AKD series servomotor drivers [13]. Anti-resonance filters that can be seen on the velocity loop can be any of the digital low pass, notch, lead-lag or biquad filters as it can be seen in Fig. 2.13. Furthermore, feedforward inputs that can be seen on the velocity loop can be any of combination of acceleration feedforward, friction compensation and viscious damping compensation. As an addition to these features, it can be seen from Figs. 2.12 - 2.14, the control loops do include other advanced features such as a state observer for velocity generation (as alternative of taking numerical derivative of position feedback), sophisticated integration- and differentiation algorithms.

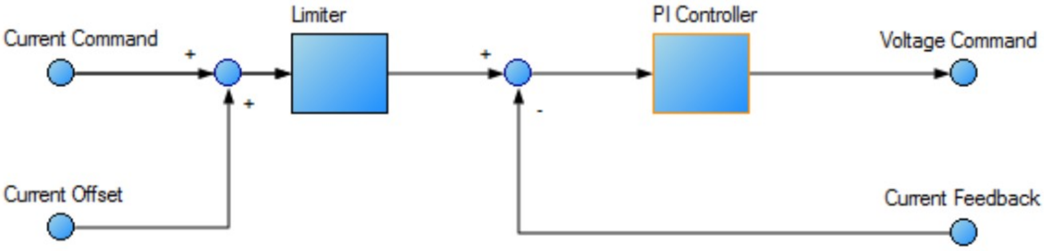


Figure 2.12: Current control loop used in AKD [13].

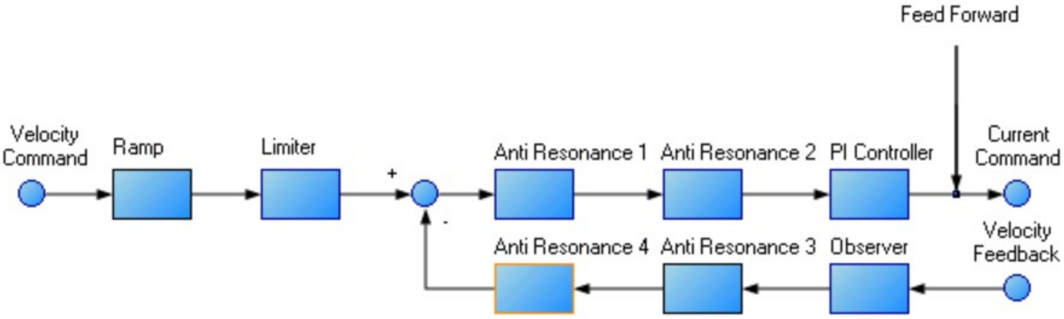


Figure 2.13: Velocity control loop used in AKD [13].

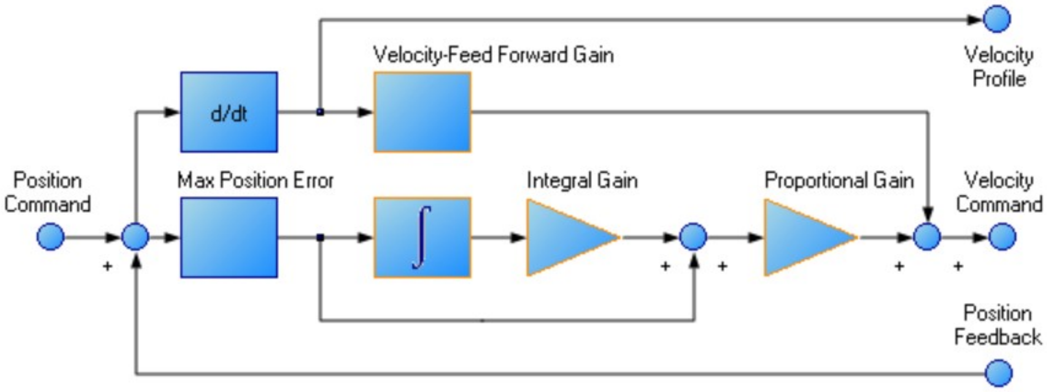


Figure 2.14: Position control loop used in AKD [13].

2.5 Closure

As a result of literature survey, multiple candidates are found for conceptual design process of the system. Additionally, a good understanding of motion control architectures are obtained.

CHAPTER 3

MECHANICAL DESIGN AND IMPLEMENTATION

3.1 Introduction

In this chapter, some of the designs discussed in Chapter 2.2 are used as starting points for the system design. In Section 3.2, different bar mechanisms are compared for target tracking and feasibility. In Section 3.3, selection of components for the system are explained and first preliminary designs are shown. In Section 3.4, the final design of the system is obtained and important details of the design are explained.

3.2 Conceptual Design

Before starting preliminary design, simple bar mechanisms are tested on a CAD software environment. Aim is;

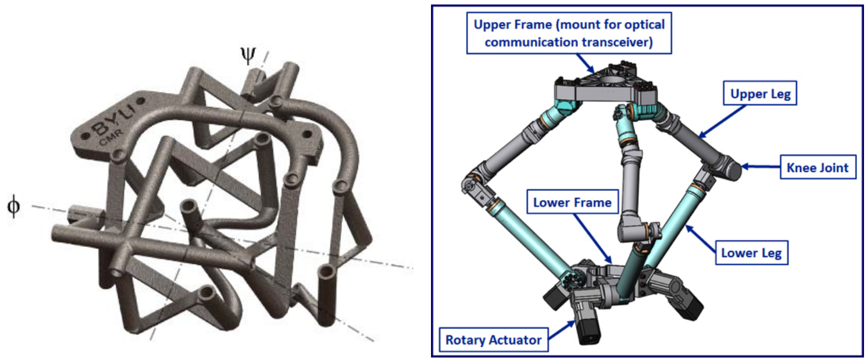
- to have some initial understanding of kinematics of different types of the systems,
- to investigate the range of motion of the system with changes in link dimensions,
- to determine the degree of freedom of the system,
- to check if there are desired number of inputs for desired number of outputs,
- to search for certain conditions which reduce the degree of freedom of the system (like a gimbal-lock).

Manipulator presented in [20] named "Omni-Wrist III" is to be studied. System includes two linear actuators to drive the system. This results in compact system in horizontal axis while occupying considerable space on its vertical axis. These linear actuators are then connected to edge of a link that is attached to a revolute joint. This results in a transformation of linear motion to rotating motion. However, since the system scale is expected to be small, range of motion of output of the system is controlled with slight displacements of the input. This is considered as a drawback since it would reduce the expected/possible resolution of inputs. In addition, cost of linear displacement sensors attached to the linear actuators are expected to be higher than a revolute displacement sensor (encoder).

Parallel manipulator named as "Wide angle gimbal" is conceptially created in CAD software and tested in this study. Design involves four arms with three revolute joints on each arm. There are no other beams that connect the moving platform to the base. System can scan a satisfactory range of motion without gimbal locks and self-collisions. Even though the system only uses revolute joints which can be obtained very cheaply and easily, other mechanical parts and especially links of the system require more complicated manufacturing techniques and higher precision with respect to other mechanism designs. There are respectively more links on each arm which might also create more errors at the tip/platform of the system.

Manipulator named as "Monolithic Mechanism" in Fig. 3.1a is investigated [7]. System moves only with bending motion. In other words, the manipulator contains no bearings and actuators bend/elastically deform the material. Requiring deformation of material results in the system having a narrow range of motion. Furthermore, the material used for the system is required to have special characteristics such as having especially high yield strength. Material is required to be manufactured using a special metal 3-D printing technique which is expected to be expensive.

Another manipulator that was considered is the well known canfield joint as shown in Fig. 3.1b. Design shown at Fig. 3.1b is an example of a canfield joint and made in University of Texas in Austin [22] [23]. Canfield joint involves three arms with two revolute joints and one spherical joint each. System requires three actuators instead of two and has an additional degree of freedom that is translational. Due to this fact causes the system requires more sophisticated control strategies, higher software processing power and overall higher overall cost. Moreover, the distance between the camera platform and the base changes considerably throughout the motion, which causes special care for the cables going to the camera platform from the base against wear and damage after repetative motion.



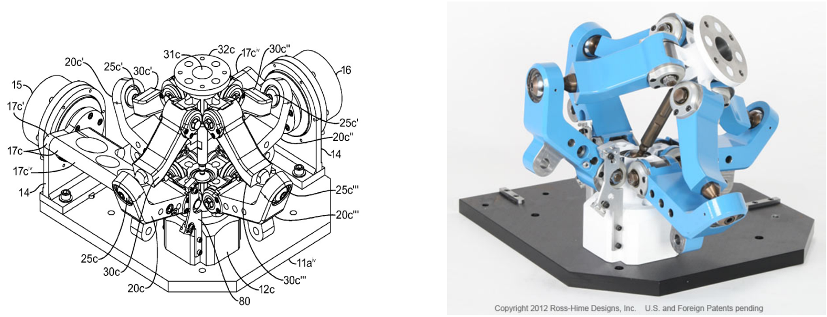
(a) Monolithic Mechanism [7]

(b) Canfield joint example [22] [23]

Figure 3.1: Manipulator Concepts 2.

Manipulator named "Omni-Wrist IV" [40] as patented by Ross-Hime Designs Inc [35] is considered. Patent designs and end-product can be seen in Fig. 3.2 . Design involves five arms. Four of those arms include two revolute joints and a spherical joint each. Fifth joint includes two spherical joints. While number of spherical joints is higher than all other candidates, the system is simpler in other aspects with respect to other candidates. System can be built with simple manufacturing techniques, includes small number of links (two links on each arm), has easy part-changeability and fixed distance between camera platform and base for easier cabling. Due to the low number of links in the system and only two degrees of freedom, the system is expected to be not costly to manufacture.

Manipulator named "Omni-Wrist V" is inspected. Design employs four arms that are similar to the arms of "Omni-Wrist IV" in terms of joints. In other words, each arm include two revolute joints and a spherical joint each. However, those arms are shaped multiple turns and difficult to manifacture. Widely known parallel manipulator "Stewart Platform" is also considered. While the platform has multiple uses and widely used in industry, Stewart platform has six DOF. It would be unnessecarly costly to actuate and control six axes for target tracking which requires two actuators only. Furthermore, it is difficult to create a Stewart platform that makes full hemi-



(a) Figure from patent for Omni-Wrist VI [40]

(b) Omni-Wrist VI [35]

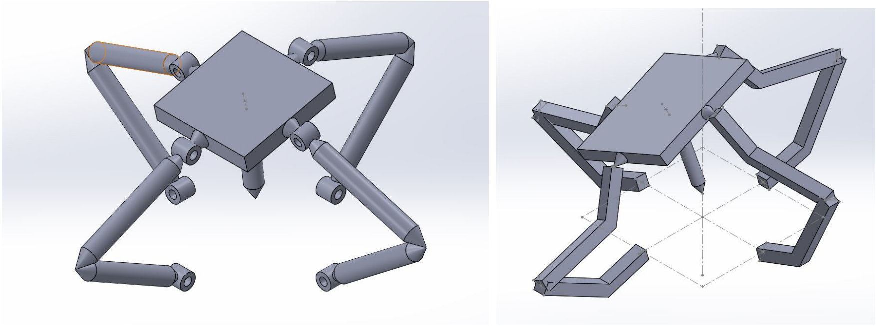
Figure 3.2: Manipulator Concepts 3.

spherical motion. A weighted decision matrix is created to select a mechanism and perform further iterations on that mechanism. Considering all the insights above, it is decided "Omni-Wrist VI" is a good candidate and iterated on.

Table 3.1: Weighted decision matrix for mechanism selection.

Selection Criteria	Weigtht (1-5)	Conventional Gimbal	Omni-Wrist III	Omni-Wrist V	Omni-Wrist VI	Wide Angle Gimbal	Monolithic Mechanism	Canfield Joint	High Angle Active Link	Steward Platform
Singularity Free	5	1	5	5	5	5	5	5	5	5
No restriction on cabling	5	2	5	5	5	5	5	3	5	5
Full Hemisphere Motion	4	5	5	5	5	5	2	4	5	2
Easy to manufacture	3	5	3	2	4	2	1	4	2	2
Complexity	2	5	4	3	4	3	3	3	4	1
Approximate cost	4	5	3	2	4	3	2	3	3	2
Total	**23**	69.57	86.09	78.26	92.17	81.74	65.22	74.78	83.48	64.35

Some conceptual drawings of a mechanism similar to "Omni-Wrist VI" is done in Fig.3.3. Iterations showed that the system has satisfactory range of motion. However, it requires special link designs on arms for not self-colliding. Furthermore, as a very important deduction, the system requires the distance between end points of the arms to be the same length. Without this special condition, the system does not act like a 2-DOF mechanism. This deduction plays a very important role in later stages of design.



(a) Mechanical Concept A

(b) Mechanical Concept B

Figure 3.3: Manipulator Concepts 4.

3.2.1 Design Requirements and Constraints

Several design requirements are given. Designed mechanical system is desired to have a range of motion of 180 degrees on each axis. In other words, the system should be able to turn horizontally any direction. System should be able to identify a circle shaped target of $10 - mm$ diameter from a distance of 5 $[m]$ using the camera attached to its camera platform. Platform of the system should be able to shelter the said camera and its optics, which are expected to be around 30 mm $\times$ 30 mm in height and width. Encoders attached to the system must be on the parts that rotate with the system, instead of being rear-shaft encoders that are attached to the back of motors.

3.3 Preliminary Mechanical Design

Before mechanical design of the system, opto-electro-mechanical components that will be installed to the system must be selected: Motors, encoders, bearings, camera and optics for the camera.

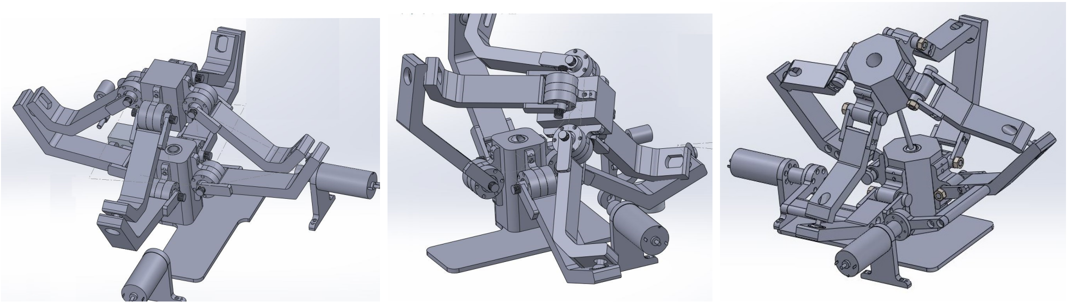
Platform of the system is carried around by arms that are attached to the shaft of the motors. Weight of the camera platform acts on a respectively long lever arm. As a result, motors that will be used in system are required to carry high amounts of torque. Even though stepper motors come to mind as a candidate with high torque capabilities, it should be noted that the system has almost $1 : 1$ ratio between the rotation of motors and and rotation of camera platform. It is predicted that the minimum incremental motion that is caused inherently by a stepper motor will always cause some form of error. Target that is desired to be tracked by the system is a circle with 5mm radius on 5m of distance. A stepper motor can only get under this number with the help of high amounts of micro-stepping and mechanical reduction with a gearbox. In addition, considering the desired high responsive characteristic of the system, it is decided not to use a stepper motor. A BLDC motor would be much more feasible because of its responsiveness and lower minimum incremental motion. Even tough brushless DC (BLDC) motors are used for high precision and high accuracy devices, they require more complex and expensive solutions. It is decided that a brushed DC motor with current control can be used for its motion capabilities, responsiveness and price. However, since the size of the system is small and the required torques are respectively large, a brushed DC motor with gearbox is used. Minimum torque requirement is found in Chapter 5 as 0.4213 $[Nm]$ for utilizing the system while vertical axis parallel tracking platform is looking upwards. With this information in mind, a brushed DC motor and gearbox pair is selected; that is, the DC motor Faulhaber 2342S024CR and its planetary gearhead 30/1S. Selected DC motor has a rated torque of 0.017 $[Nm]$ and a stall torque of 0.085 $[Nm]$. Gearbox of motor has a reduction ratio of $23 : 1$, which means that these torques increase to a rated torque of 0.391 $[Nm]$ and a stall torque of 1.955 $[Nm]$ on gearbox output shaft. Stall torque is significantly higher than the calculated required torque, meaning that the system will not go into stall for prescribed motion. Furthermore, rated torque of the motor-gearbox pair is almost equal to the minimum torque required even though the motor is expected not the be run in rated speeds. Hence, this motor is satisfactory to be used to drive the mechanism. Detailed information of this motor is given in Section 6.2 where the parameters of the motor are used for motion control.

Encoders of the system must be small. As the size gets smaller, it is harder to find inexpensive high resolution encoders due to the decreasing encoder disk size. Also considering the costs, it is decided to use an encoder (AEDC-5550-Z12) with 20000-cpr (4X decoding) having a 6-mm shaft hole. More information of this encoder is given in Section 6.2.

Bearings of the system must also be small in size. They affect the size of everything around them since they must be enclosed. In addition, they also weigh notably since they are made from metal. Stress caused on the bearings are inconsiderable with respect to their load ratings. For revolute joints, flanged ball bearings are preferred for easier assembly.

Camera is desired to have colored pixel format for simpler target recognition, a full resolution frame rate of 60 fps which can be increased by lowering the frame size and/or resolution, global shutter progressive scan CMOS sensor for obtaining images without blurring even in high-speed, USB3.0 5 [Gbit/s] interface for high speed PC connection. Since optics will be attached to the camera, pixel size of the camera does not matter by itself. Additional information for this camera is given in Section 6.2.

A preliminary mechanical design is conducted with the selected components as shown in Fig. 3.4a. Notice that special care must be taken for the mechanism to not self intersect as as shown in Fig. 3.4b. After modifying the design in order to have non-self intersecting system, preliminary configuraton in Fig. 3.4c is obtained. Note that the system is bulkier than desired which must be downsized. This is mostly due to the design of revolute joints. Hence, special care is given to the size of revolute joints and size of all components in the detailed design process.



(a) Preliminary-A Looking Vertical

(b) Preliminary-A Looking Horizontal

(c) Preliminary-B

Figure 3.4: Figures of preleminary designs.

After having some rough estimation of what the system will be like in the preliminary design, it is decided that conventional manufacturing will require 4-axis or 5-axis machining for many of its parts if it is desired to be manufactured from aluminum. In order to have a lightweight system and lower manufacturing costs, it is decided that the prototype should be built by 3-D printing instead. PLA+ is used as the manufacturing material for 3-D printing.

3.4 Detailed Design

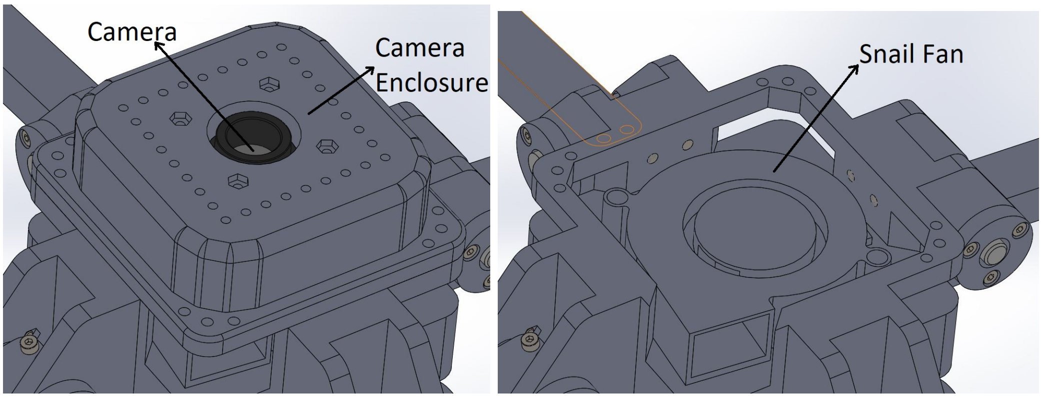
In this section, some details will be explained about the detailed design which might be hidden when looking to the system from outside. Mainly, information about the camera platform of the system, information about the connections of the system and information about sizing of the system is given in this section.

3.4.1 System Camera Platform Design Information

As explained in Section 3.2.1, camera platform of the system have to contain the camera that is selected on preliminary design. This camera must be provided with sufficient heat dissipation to keep the operation temperature under its maximum required operating temperature, which is $75°C$. Usually, this requirement can be satisfied by passive heat dissipation if the camera is enclosed in a metal containment or attached to a metal surface. However, all parts of the system are made with PLA+ material which has low heat conductivity. This means that the camera must be actively cooled. Putting a cooling fan inside the camera platform does not complicate the design thanks to the flexibility of the system. Fins are attached to the electronic components on the camera that generates the heat. Camera platform is designed such that a snail fan is just under the camera as as shown in Fig. 3.5, pulling heat from the electronic components and the fins.

3.4.2 Joint Design Information

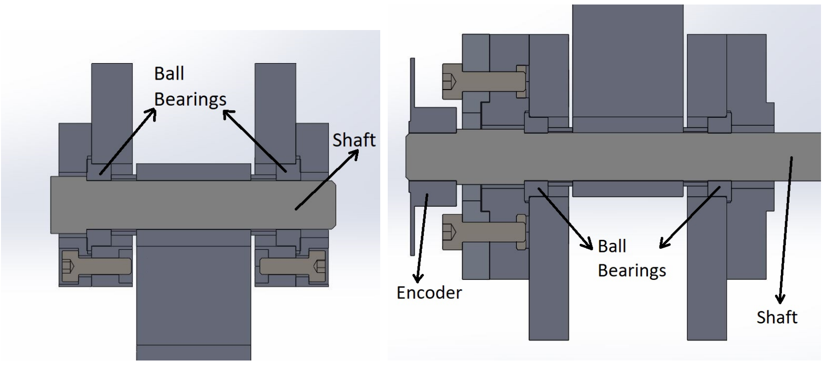
There are two types of joints in the system: Revolute joints and spherical joints. For revolute joints, a miniature stainless steel ball bearing with ISO tolerance of P0 for both inner and outer ring is selected. They are flanged for easier assembly. Bearings are constrained both on inner and outer rings as shown in Fig.3.6. For spherical joints, a miniature sliding spherical bearing is used in which both inner and outter ring are steel that are separated with the radial clearance of 32-68$[\mu\ m]$. They are assembled as shown in Figs. 3.6a and 3.6b.



(a) Camera platform of the system from outside

(b) Snail fan under camera

Figure 3.5: Camera Platform of the system



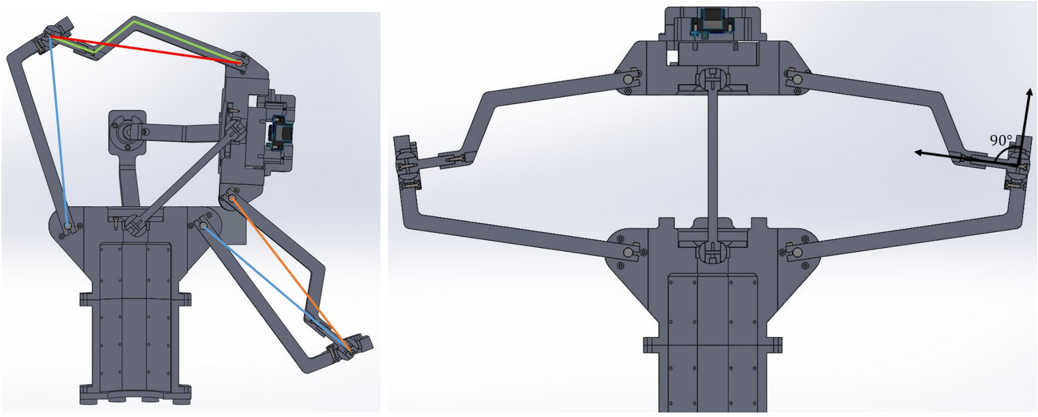
(a) Cross-section of revolute joints

(b) Cross-section of revolute joint attached to motor

Figure 3.6: Ball Bearing Assembly Cross-sections.

3.4.3 Sizing and Shape Considerations

Camera platform of the system is sized according to camera and fan assembly as explained in Section 3.4.1 where the size of that part is considered fixed. Central beam is sized to the smallest possible length such that the camera platform and base of the system will not be contacting when the system rotates to look horizontally. This requirement can be observed better by looking at the Fig. 3.7a or Fig. A.3 in Appendix A. Arms of the system are sized considering the desired range of motion of the system. Longer arms results in higher resolution of motion and higher range of motion with diminishing returns. Drawback is that, higher torques needed from the motor and higher base length. Arms must be sized such that the shafts coming out of the spherical joints (which has a fixed diameter) are not intersecting with its holder. This requirement can be observed better by looking at the broken out sections B and C in the Fig.A.3 in Appendix A. For this reason, thickness of the shafts coming out of the spherical joints are reduced as much as possible. In order to obtain lower thickness, they are specifically built in AISI 1040 steel instead of 3-D printed material.



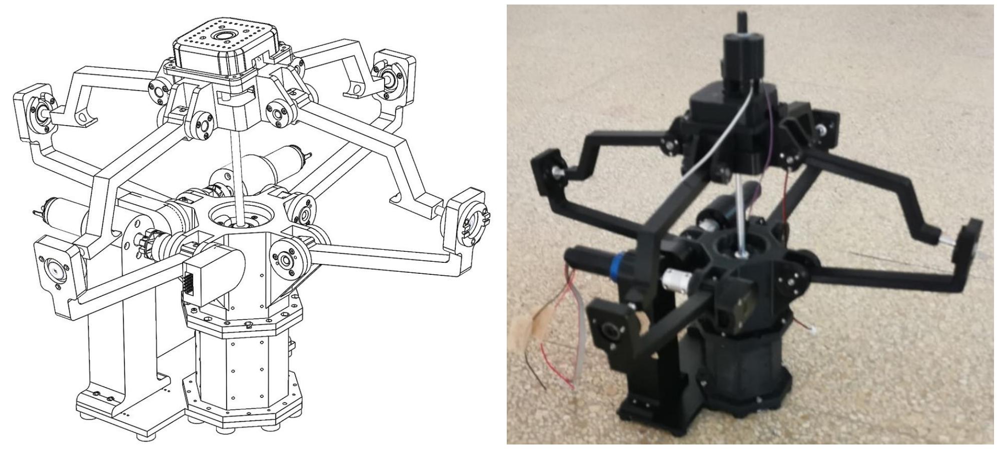
(a) Looking Horizontally

(b) Looking Vertically

Figure 3.7: System Cross-sections.

Shape of lower and upper arms are designed such that; in its vertical position the shafts coming out of the spherical joints are on the central axis of spherical bearing as in Fig.3.7b. In other words, there are 90 degrees between the shafts coming out of the spherical joints and the surface that the spherical joints are inserted in. In its horizontal position, one of the upper links cause a contact with a lower link . This contacting link is the red link in Fig. 3.7a. In order not to have this intersection, the shape of the upper link is modified as much as possible until the link at orange starts intersecting. Arms are sized to the smallest possible length such that the shafts coming out of the spherical joints are at their limits when the system rotates to look horizontally as in Figs. 3.7a and A.3. As a result of the design process, the final desired design of the system is obtained. This design is manufactured using 3-D printing techniques and montaged. Prototype and result of this section is shown in

Fig.3.8. System outter dimensions are $415\ [mm] \times 415\ [mm] \times 315\ [mm]$ while looking vertically and $390\ [mm] \times 310\ [mm] \times 317\ [mm]$ while looking horizontally.



(a) Final Design

(b) Mechanical System Prototype

Figure 3.8: Results of Mechanical Design Process.

3.5 Closure

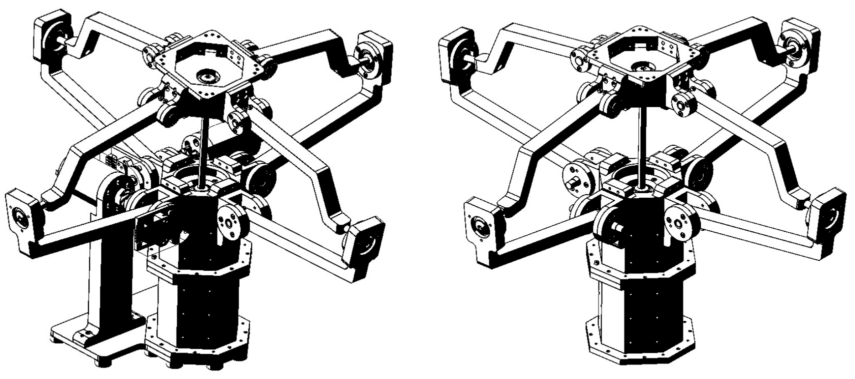
In conclusion, competator parallel mechanisms are compared and Omni-Wrist VI is selected as the best concept. After deciding on the concept, key components such as motors, camera, encoders are selected in preliminery design. In detailed design; camera platform of the system, connections of the system, and sizing of the system are explained in detail. Lastly, designed parallel mechanism is manufactured using 3D-printing techniques and assembled.

CHAPTER 4

KINEMATIC ANALYSIS OF MECHANISM

4.1 Introduction

A unique mechanism (Fig. 4.1a) is designed and used as the tracking platform. Because of its simplicity and presumed easier manufacturability of the system the mechanism described in [35] is used as an inspiration and baseline for design. It is explained in [40] the system has two inputs and two outputs, where both the inputs and outputs are rotational. However, it is required to have the relations between inputs and outputs in order to be able to control and drive the mechanism. A simplified image without motors/encoders etc. (4.1b) is used for the explanation of kinematic analysis of the system throughout this chapter. Even though at this stage of design there is no mechanical design or image of the system, for better explanation of all calculations, images of the finished prototype is used in this part. In reality, all of these



(a) Mechanism.

(b) Simplified Mechanism.

Figure 4.1: Mechanism and Simplified Mechanism.

calculations are made without the a mechanical system as explained in Section 4.2 and Section 4.3. These images are actually obtained at a later stage after kinematic analysis and mechanical design are completed.

4.2 Degree of Freedom Calculation

While it is already expected for the system to have 2 Degree of Freedom (DOF), calculation of number of DOFs is crucial and has very important implications for the design of the system. DOF of the system can be calculated using Gruebler's Equation:

$$DOF = D(N - 1 - j) + \sum_{i=1}^{j} f_i \tag{4.1}$$

where,

D**:** dimentions of the system i.e. 3 for a 2 dimensional system, 6 for a 3 dimensional system,

N**:** total number of links in the mechanism,

j**:** total number of joints in the mechanism,

f_i**:** total number of freedom caused by each joint.

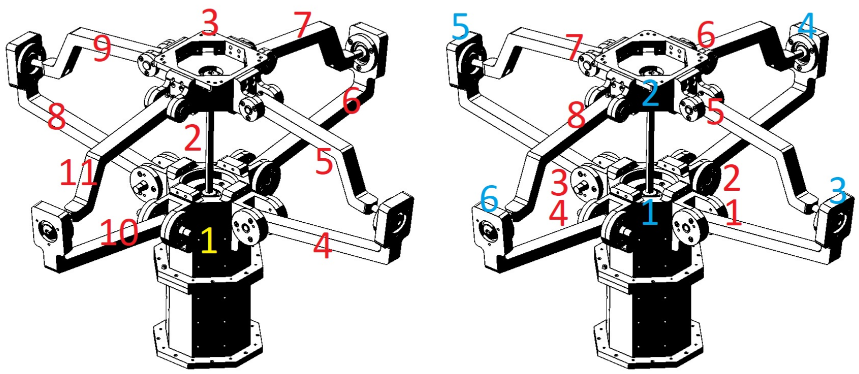
Number of links in the mechanism can be seen in Fig. 4.2a and number of joints in the system can be seen in Fig. 4.2b. Note that a revolute joint allows one degree of freedom while a spherical joints allow three degrees of freedom. Hence, the equation can be rewritten as:

$$DOF = D(N - 1 - j) + n_r + 3n_s \tag{4.2}$$

where the variables are defined as;

n_r**:** number of revolute joints,

n_s**:** number of spherical joints.



(a) Number of links in the system.

(b) Number of revolute joints (red) and spherical joints (blue) in the system.

Figure 4.2: Link and joint nuberings.

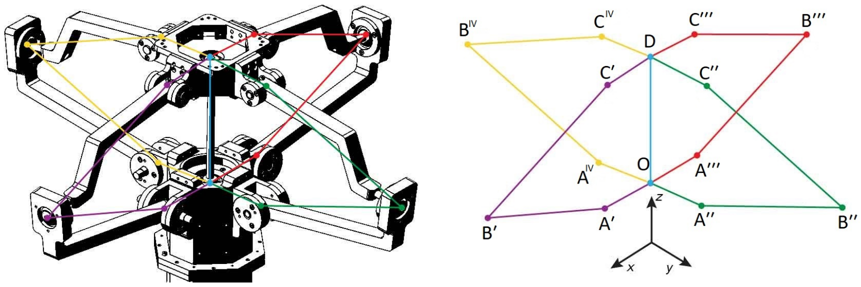
Inserting the values inside the formula yields:

$$DOF = 6(11 - 1 - 14) + 8 + 3 \times 6 = 2 \tag{4.3}$$

Note that there are $8 + 3 \times 6 = 26$ freedoms caused by joints. Note that, the link numbered (2) can move around its own axis, causing the system to have a redundant freedom. This means that the system should have only one remaining useful degree of freedom. Mechanism can only rotate in X axis if there are absolutely no rotation on Y axis and vice versa. However, the system can move in 2-DOF if a certain criteria exists: If links 4-5-6-7-8-9-10-11 have the same length from joint to joint, it creates an exception to the Grübler Criterion (as it is known some mechanisms can have). Exceptions to Grübler Criterion can be caused by various unique geometry properties of mechanisms. In this case, this exception is found by creating simple concept models of the system on a CAD software and running/testing them inside 3-D Simulations (Solidworks).

4.3 Kinematic Modeling

In order to be able to drive and control the system, relations between the inputs and outputs of the system are required to be known. Furthermore, these equations will be used to make sure the system is going to have desired range of motion. The system output is decided to be the state of link (3) in Fig. 4.2a. These relations are found by solving the kinematic equations that define the system. For this purpose, links are drawn on the Fig. 4.1b and Fig. 4.3a. Namings for the joints are made for the mechanism for easier explanation of calculations and they can be seen in Fig. 4.3b. There are four arms that connect O to D which are all identical. They consist of a spherical joint on B and revolute joints on A and C. There is also another connection from O to D that is through link (2) in Fig. 4.2a. These five connections between O and D are shown with different colorings in Fig. 4.3a. This colored link drawings are separated from the mechanical system to have a simpler view of it as in Fig. 4.3b. A single arm of the system is a RSR manipulator. Lengths for the green colored connection in Fig. 4.3b can be shown as in Eqn. (4.4).



(a) Connections between O and D shown on mechanism.

(b) Simplified connections between O and D.

Figure 4.3: Connections between O and D.

$$a' = [OA'], b' = [A'B'], c' = [B'C'], d' = [C'D], e = [OE] \tag{4.4}$$

Translation vector equations from the base to the moving platform on the green colored connection are given in Eqn. (4.5) through Eqn. (4.8), where $\boldsymbol{r} \in \Re^{3\times1}$ are position vectors, $\boldsymbol{u} \in \Re^{3\times1}$ are unit vectors and $\boldsymbol{C} \in \Re^{3\times3}$ are the rotation matrices.

$$\boldsymbol{r}_{(D/O)'} = \boldsymbol{r}_{A'/O} + \boldsymbol{r}_{B'/A'(O)} + \boldsymbol{r}_{C'/B'(O)} + \boldsymbol{r}_{D/C'(O)} \tag{4.5}$$

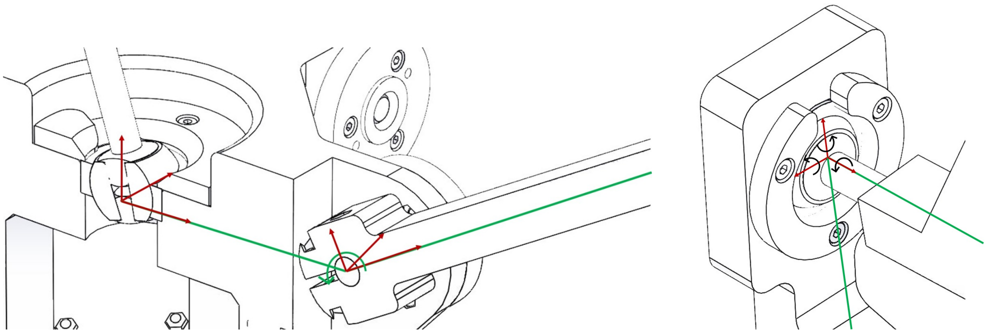
$$\boldsymbol{r}_{(D/O)'} = a' \cdot \boldsymbol{u}_1^{O/O} + b' \cdot \boldsymbol{u}_1^{A'/O} + c' \cdot \boldsymbol{u}_1^{B'/O} + d' \cdot \boldsymbol{u}_1^{C'/O} \tag{4.6}$$

$$\begin{aligned} \boldsymbol{r}_{(D/O)'} = \quad & a' \cdot \boldsymbol{u}_1^{O/O} + b' \cdot \boldsymbol{C}^{O/A'} \cdot \boldsymbol{u}_1^{A'/A'} \\ & + c' \cdot \boldsymbol{C}^{O/B'} \cdot \boldsymbol{u}_1^{B'/B'} + d' \cdot \boldsymbol{C}^{O/C'} \cdot \boldsymbol{u}_1^{C'/C'} \end{aligned} \tag{4.7}$$

$$
\begin{aligned}
\boldsymbol{r}_{(D/O)'} = \quad & a' \cdot \boldsymbol{u}_1 + b' \cdot \boldsymbol{C}^{O,A'} \cdot \boldsymbol{u}_1 \\
& + c' \cdot \boldsymbol{C}^{O,A'} \cdot \boldsymbol{C}^{A',B'} \cdot \boldsymbol{u}_1 + d' \cdot \boldsymbol{C}^{O,A'} \cdot \boldsymbol{C}^{A',B'} \cdot \boldsymbol{C}^{B',C'} \cdot \boldsymbol{u}_1
\end{aligned} \tag{4.8}
$$

where the unit vectors are defined as in Eqn. (4.9).

$$
\boldsymbol{u}_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T, \qquad \boldsymbol{u}_2 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T, \qquad \boldsymbol{u}_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tag{4.9}
$$



(a) $\boldsymbol{C}^{O,A'}$ visualization.

(b) $\boldsymbol{C}^{A',B'}$ visualization.

Figure 4.4: Rotation matrix visualizations.

$\boldsymbol{C}^{O,A'}$ is the rotation matrix that is caused by the rotation of revolute joint at A'. This joint creates a rotation of $\beta_{A'}$ in the direction of $\boldsymbol{u}_2^{A'/A'}$. This rotation matrix can be shown as in Eqn. (4.10).

$$
\boldsymbol{C}^{O,A'} = \begin{bmatrix} cos(\beta_{A'}) & 0 & sin(\beta_{A'}) \\ 0 & 1 & 0 \\ -sin(\beta_{A'}) & 0 & cos(\beta_{A'}) \end{bmatrix} \tag{4.10}
$$

$\boldsymbol{C}^{A',B'}$ is the rotation matrix that is caused by the rotation of spherical joint at B'. This joint creates a rotation of $\alpha_{B'}$ in the direction of $\boldsymbol{u}_1^{B'/B'}$, a rotation of $\beta_{B'}$ in the direction of $\boldsymbol{u}_2^{B'/B'}$ and a rotation of $\gamma_{B'}$ in the direction of $\boldsymbol{u}_3^{B'/B'}$. These rotation matrices can be shown in Eqn. (4.11).

$$
\boldsymbol{C}^{A',B'} = \boldsymbol{C}_1^{A',B'} \cdot \boldsymbol{C}_2^{A',B'} \cdot \boldsymbol{C}_3^{A',B'} \tag{4.11}
$$

$$\boldsymbol{C}_1^{A',B'} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha_{B'}) & -sin(\alpha_{B'}) \\ 0 & sin(\alpha_{B'}) & cos(\alpha_{B'}) \end{bmatrix} \tag{4.12a}$$

$$\boldsymbol{C}_2^{A',B'} = \begin{bmatrix} cos(\beta_{B'}) & 0 & sin(\beta_{B'}) \\ 0 & 1 & 0 \\ -sin(\beta_{B'}) & 0 & cos(\beta_{B'}) \end{bmatrix} \tag{4.12b}$$

$$\boldsymbol{C}_3^{A',B'} = \begin{bmatrix} cos(\gamma_{B'}) & -sin(\gamma_{B'}) & 0 \\ sin(\gamma_{B'}) & cos(\gamma_{B'}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.12c}$$

$\boldsymbol{C}^{B',C'}$ is the rotation matrix that is caused by the rotation of the revolute joint at C'. This joint creates a rotation of $\beta_{C'}$ in the direction of $\boldsymbol{u}_2^{C'/C'}$. This rotation matrix can be shown in Eqn. (4.13).

$$\boldsymbol{C}^{B',C'} = \begin{bmatrix} cos(\beta_{C'}) & 0 & sin(\beta_{C'}) \\ 0 & 1 & 0 \\ -sin(\beta_{C'}) & 0 & cos(\beta_{C'}) \end{bmatrix} \tag{4.13}$$

$$\boldsymbol{C}^{(O/D)'} = \boldsymbol{C}^{O,A'} \cdot \boldsymbol{C}^{A',B'} \cdot \boldsymbol{C}^{B',C'} \cdot \boldsymbol{C}^{C',D'} \tag{4.14}$$

Similar equations can be obtained from other connections between O and D:

- For the red colored connection in Fig. 4.3b, translation vector $\boldsymbol{r}_{(D/O)''}$ and rotation matrix $\boldsymbol{C}^{(O/D)''}$ can be obtained as functions of $\beta_{A''}$, $\alpha_{B''}$, $\beta_{B''}$, $\gamma_{B''}$, $\beta_{C''}$.

- For the yellow colored connection in Fig. 4.3b, translation vector $\boldsymbol{r}_{(D/O)'''}$ and rotation matrix $\boldsymbol{C}^{(O/D)'''}$ can be obtained as functions of $\beta_{A'''}$, $\alpha_{B'''}$, $\beta_{B'''}$, $\gamma_{B'''}$, $\beta_{C'''}$.

- For the purple colored connection in Fig. 4.3b, translation vector $\boldsymbol{r}_{D/(O)^{IV}}$ and rotation matrix $\boldsymbol{C}^{(O/D)^{IV}}$ can be obtained as functions of $\beta_{A^{IV}}$, $\alpha_{B^{IV}}$, $\beta_{B^{IV}}$, $\gamma_{B^{IV}}$, $\beta_{C^{IV}}$.

Translation vector for the blue colored connection in Fig. 4.3b can also be shown as in Eqn. (4.16).

$$\boldsymbol{r}_{(D/O)^V} = e \cdot \boldsymbol{u}_3^{O'/O} \tag{4.15}$$

$$\boldsymbol{r}_{(D/O)^V} = e \cdot \boldsymbol{C}^{O',O} \cdot \boldsymbol{u}_3^{O'/O'} \tag{4.16}$$

$\boldsymbol{C}^{O',O}$ is the rotation matrix that is caused by the rotation of spherical joint at O' and O. This joint creates a rotation of $\alpha_{O'}$ in the direction of $\boldsymbol{u}_1^{O'/O'}$, a rotation of $\beta_{O'}$ in the direction of $\boldsymbol{u}_2^{O'/O'}$ and a rotation of $\gamma_{O'}$ in the direction of $\boldsymbol{u}_3^{O'/O'}$. Rotation matrix of the blue colored connection is not used since it causes same number of angles (unknowns) and equations to appear. These three angles do not appear anywhere else (not even in the translational vector of blue connection) They can be solved only using the equations introduced by the rotation matrix of blue connection. Hence, rotation matrix of blue connection does not contribute to the solution of the system and it is redundant.

In the end, the translational vectors and rotational matrices are obtained that connect O to$D$ in Eqn. (4.18).

$$\boldsymbol{r}_{(D/O)'} = \boldsymbol{r}_{(D/O)''} = \boldsymbol{r}_{(D/O)'''} = \boldsymbol{r}_{(D/O)^{IV}} = \boldsymbol{r}_{(D/O)^V} \tag{4.17}$$

$$\boldsymbol{C}^{(O/D)'} = \boldsymbol{C}^{(O/D)''} = \boldsymbol{C}^{(O/D)'''} = \boldsymbol{C}^{(O/D)^{IV}} \tag{4.18}$$

Eqns. (4.17) and (4.18) can be shown as in Eqns. (4.19b-4.19g) so it is easier to see the number of total equations.

$$\boldsymbol{r}_{(D/O)'} = \boldsymbol{r}_{(D/O)''} \tag{4.19a}$$

$$\boldsymbol{r}_{(D/O)''} = \boldsymbol{r}_{(D/O)'''} \tag{4.19b}$$

$$\boldsymbol{r}_{(D/O)'''} = \boldsymbol{r}_{(D/O)^{IV}} \tag{4.19c}$$

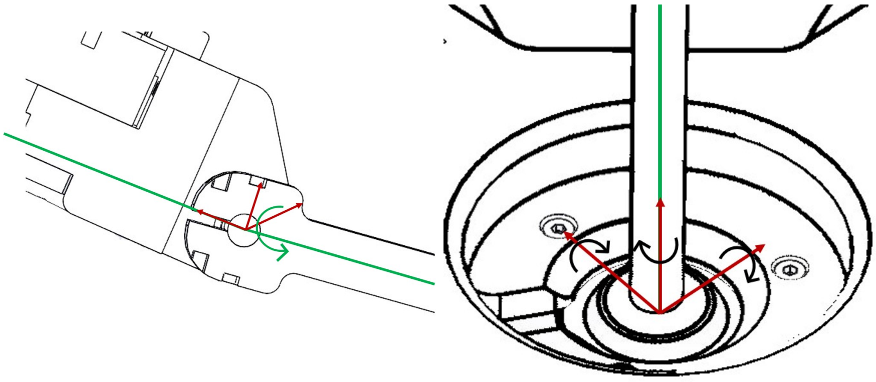
$$\boldsymbol{r}_{(D/O)^{IV}} = \boldsymbol{r}_{(D/O)^V} \tag{4.19d}$$

$$\boldsymbol{C}^{(O/D)'} = \boldsymbol{C}^{(O/D)''} \tag{4.19e}$$

$$\boldsymbol{C}^{(O/D)''} = \boldsymbol{C}^{(O/D)'''} \tag{4.19f}$$

$$\boldsymbol{C}^{(O/D)'''} = \boldsymbol{C}^{(O/D)^{IV}} \tag{4.19g}$$

Each equation in 4.19 gives 3 unique equations. In total 21 equations are obtained. Furthermore, 23 unknown rotations are included those 21 equations. Namely, $\beta_{A'}$,



(a) $\boldsymbol{C}^{B',C'}$ visualization.

(b) $\boldsymbol{C}^{O',O}$ visualization.

Figure 4.5: Rotation matrix visualizations.

$\alpha_{B'}$, $\beta_{B'}$, $\gamma_{B'}$, $\beta_{C'}$, $\beta_{A''}$, $\alpha_{B''}$, $\beta_{B''}$, $\gamma_{B''}$, $\beta_{C''}$, $\beta_{A'''}$, $\alpha_{B'''}$, $\beta_{B'''}$, $\gamma_{B'''}$, $\beta_{C'''}$, $\beta_{A^{IV}}$, $\alpha_{B^{IV}}$, $\beta_{B^{IV}}$, $\gamma_{B^{IV}}$, $\beta_{C^{IV}}$, $\alpha_{O'}$, $\beta_{O'}$ and $\gamma_{O'}$. This means that by defining any of those two rotations (actuating those two rotations), all of the 21 equations can be solved (and have complete knowledge about the state of system). However, analytic solution for those 21 equations is compelling even with the use of computer software. Hence, numeric solution for solving the 21 equations are explained in Section 4.4.

4.4 Numeric Solution

In order to obtain input/output relationships, Eqn. (4.19b)-(4.19g) are solved numerically. It should be noted that, all of the unknown variables are angular. Small-angle assumptions will be made throughout the numeric solution of the system. An initial state of the system in a simple symmetrical state (in which the plant is pointing upwards and all arms are in same alignment) is obtained using the mechanical design described in Section 3. Specifically; lengths of $[OA]$, $[AB]$, $[BC]$, $[CD]$, $[OE]$ and corresponding angles for 23 unknown angular variables $\beta_{A'}$, $\alpha_{B'}$, $\beta_{B'}$, $\gamma_{B'}$, $\beta_{C'}$, $\beta_{A''}$, $\alpha_{B''}$, $\beta_{B''}$, $\gamma_{B''}$, $\beta_{C''}$, $\beta_{A'''}$, $\alpha_{B'''}$, $\beta_{B'''}$, $\gamma_{B'''}$, $\beta_{C'''}$, $\beta_{A^{IV}}$, $\alpha_{B^{IV}}$, $\beta_{B^{IV}}$, $\gamma_{B^{IV}}$, $\beta_{C^{IV}}$, $\alpha_{O'}$, $\beta_{O'}$ and $\gamma_{O'}$. These initial values are inserted into Eqn. (4.19b)-(4.19g) to make sure the equations and initial values for angles are correct. Two of these angles, $\beta_{A'}$ and $\beta_{A''}$ are selected as inputs. Hence, they will be used as known values for the numeric solution. Each combination of these two angles will result in 21 known angles (since there are 21 equations). While working on small-angle assumption, small angles δ_1 through δ_{21} will be added to 21 unknown angular variables. Adding these small angles to unknown angular variables causes changes to the rotation matrices. For example Eqn. (4.10) can be modified as in Eqn. (4.20). Note that c and s inside the Eqn. (4.20) are respectively the shorthand notation for cosine and sine.

$$\begin{aligned}
\boldsymbol{C}_{\delta_1}^{O,A'} &= \begin{bmatrix} cos(\beta_{A'}+\delta_1) & 0 & sin(\beta_{A'}+\delta_1) \\ 0 & 1 & 0 \\ -sin(\beta_{A'}+\delta_1) & 0 & cos(\beta_{A'}+\delta_1) \end{bmatrix} \\
&= \begin{bmatrix} c(\beta_{A'})c(\delta_1)-s(\beta_{A'})s(\delta_1) & 0 & c(\beta_{A'})s(\delta_1)+s(\beta_{A'})c(\delta_1) \\ 0 & 1 & 0 \\ -c(\beta_{A'})s(\delta_1)-s(\beta_{A'})c(\delta_1) & 0 & c(\beta_{A'})c(\delta_1)-s(\beta_{A'})s(\delta_1) \end{bmatrix} \\
&= \begin{bmatrix} c(\beta_{A'})-s(\beta_{A'})\delta_1 & 0 & c(\beta_{A'})\delta_1+s(\beta_{A'}) \\ 0 & 1 & 0 \\ -c(\beta_{A'})-s(\beta_{A'}) & 0 & c(\beta_{A'})-s(\beta_{A'})\delta_1 \end{bmatrix}
\end{aligned} \tag{4.20}$$

where for small angles, assumptions are made as in (4.21).

$$\delta_1 \cong 0, \qquad cos(\delta_1) \cong 1, \qquad sin(\delta_1) \cong \delta_1 \tag{4.21}$$

Similar equations for each rotation matrix shown in Eqns. (4.10), (4.11) and (4.13) as well as their similar counterparts on other arms of the system. As a result, 21 new rotation matrices are obtained with small angle components δ_1 through δ_{21} as in Eqn. 4.20. While numerically solving the nonlinear kinematic equations of the system, these 21 small angle components are used as unknowns at each iteration.

These results are obtained by implementing the following pseudocode in Matlab:

1. Define initial values (known) to 23 angular variables and make sure they satisfy the kinematic equations.

2. Change $\beta_{A'}$ or $\beta_{A''}$ marginally.

3. By inserting the remaining 21 known angular variables, construct/update rotation matrices according to small angle assumption which includes 21 unknown small angle components.

4. By multiplying the rotation matrices (obtained by small angle assumption), obtain the equations (nonlinear system) described from Eqn. (4.19b)-(4.19g).

5. Solve equations by nonlinear regression using Levenberg-Marquardt Method and obtain 21 small angles that satisfy the nonlinear system for the marginal change in $\beta_{A'}$ or $\beta_{A''}$.

6. Add 21 small angles to their corresponding 21 angular variables.

7. Record the 23 angular variables as a satisfactory set of variables that satisfy the kinematic equations in a table.

8. Go to Step 2 until a satisfactory region of interest is obtained.

As a result of implementation of this pseudocode, rotation matrices for the plant ($\boldsymbol{C}^{O/D'}$, $\boldsymbol{C}^{O/D''}$, $\boldsymbol{C}^{O/D'''}$ and $\boldsymbol{C}^{(O/D)^{IV}}$) can be obtained with respect to the two inputs; $\beta_{A'}$ and $\beta_{A''}$. Inverse XYZ Euler transform of any of these rotation matrices yields the rotation components of plant as Euler angles Φ , Θ and Ψ . Hence, discrete point maps for the outputs Φ , Θ and Ψ of the system are obtained with respect to inputs $\beta_{A'}$ and $\beta_{A''}$. These point maps are shown in Figs. 4.6, 4.8 and 4.10. Discrete point maps of the outputs Φ , Θ with respect to inputs $\beta_{A'}$ and $\beta_{A''}$ are also represented in 3-D figures for clarity in Figs. 4.7, 4.9.

4.5 Inverse Kinematic Model and De-Coupling of Axes

In order to control the system, it may be required to know the required input angles in order to have certain output state. In order to have this information, discrete data points obtained in Section 4.4 can be inverted to show inputs $\beta_{A'}$ and $\beta_{A''}$ with respect to outputs Φ, Θ and Ψ. These point maps are shown in Figs. 4.11 and 4.12.

Multiple approaches can be made for decoupling of the axes. Firstly, mapping the inverse of the current positions forwards kinematics is tried to obtain inverse kinematics mapping for current motor positions. Discrete point maps for the outputs Φ, Θ and Ψ of the system are obtained with respect to inputs $\beta_{A'}$ and $\beta_{A''}$ with predefined small angle increments of δ, for any data point in the region of interest, the known

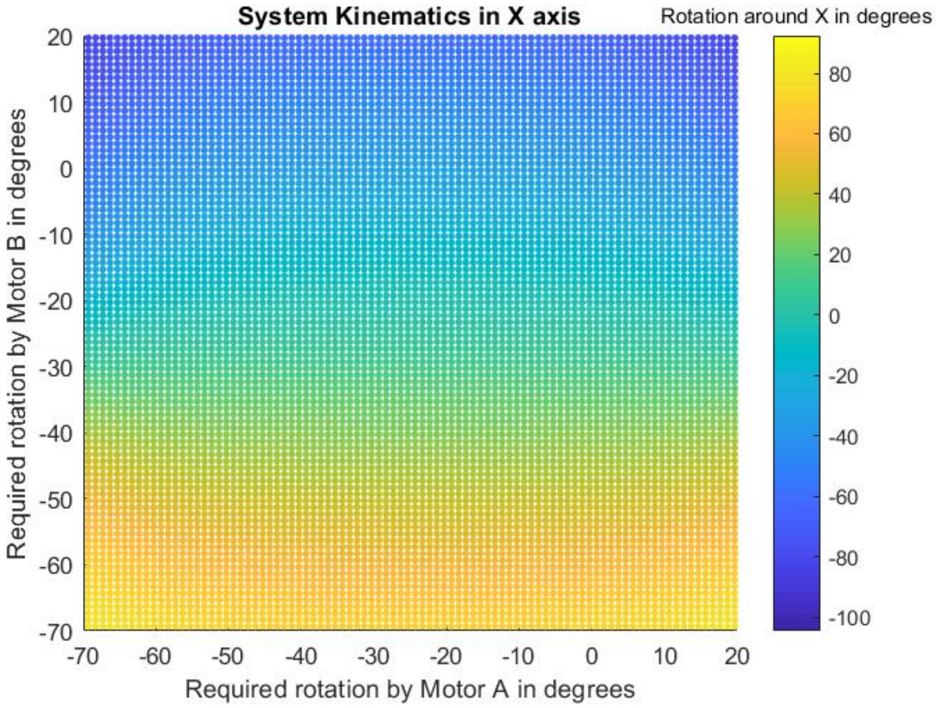


Figure 4.6: System Kinematics in X axis.

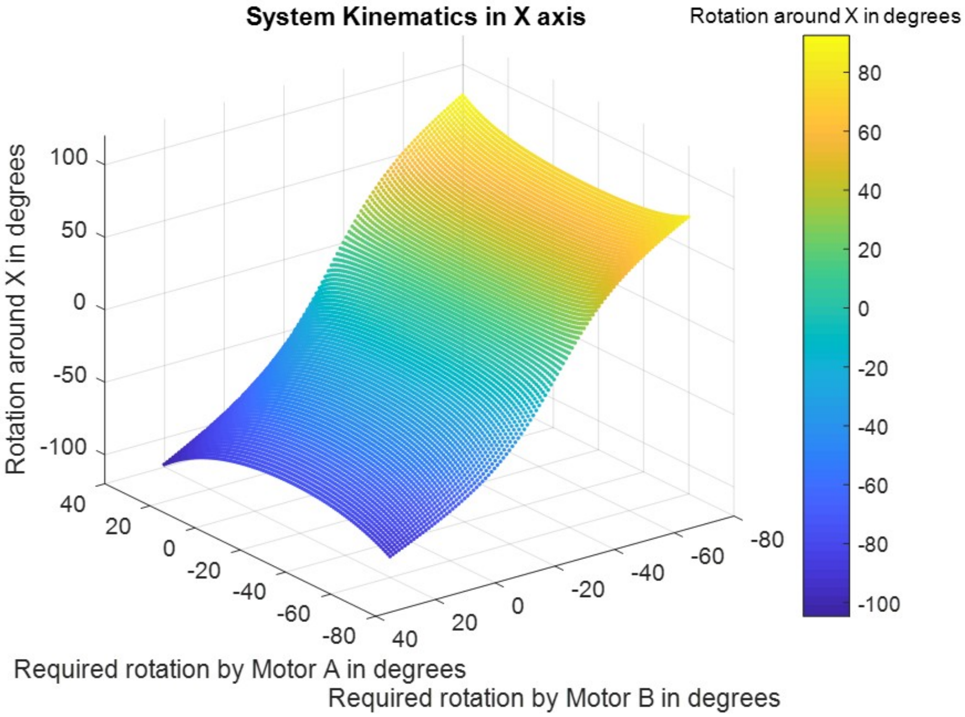


Figure 4.7: System Kinematics in X axis on 3D plot.

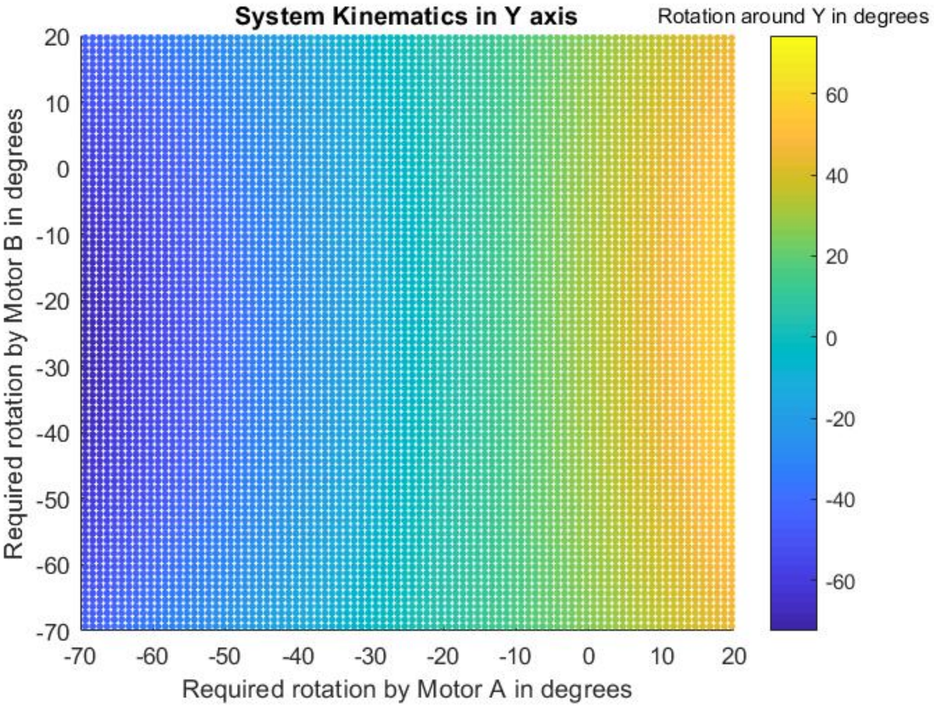


Figure 4.8: System Kinematics in Y axis.

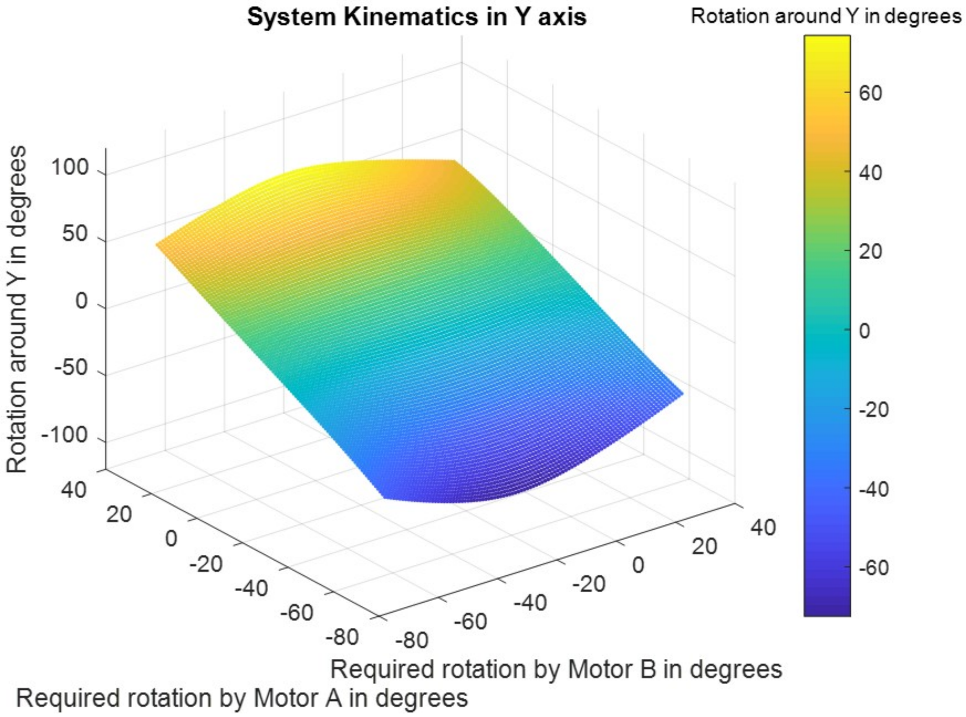


Figure 4.9: System Kinematics in Y axis on 3D plot.

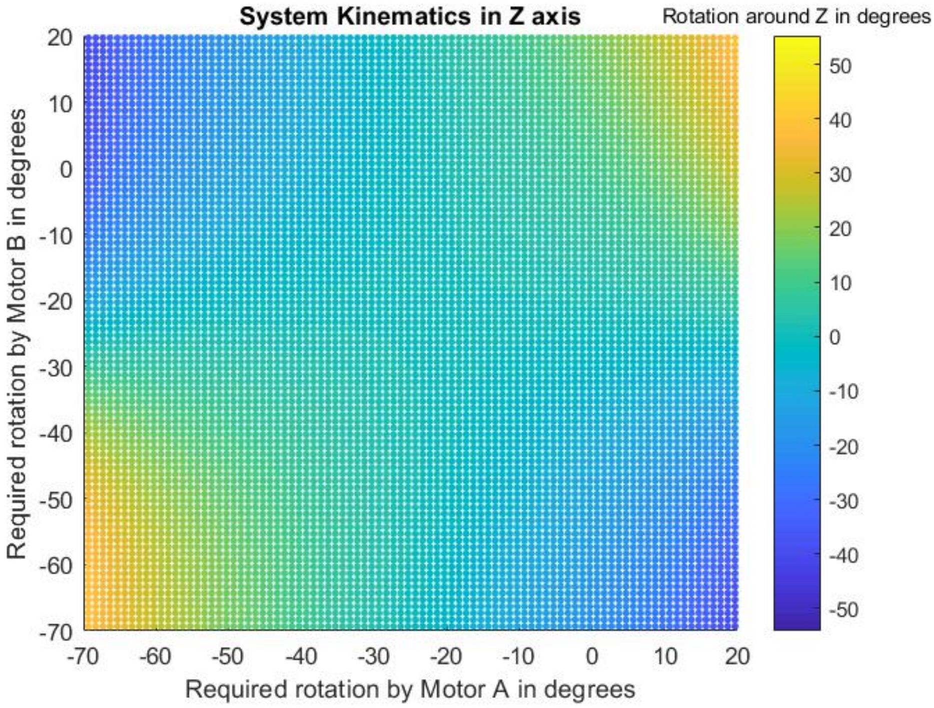


Figure 4.10: System Kinematics in Z axis.

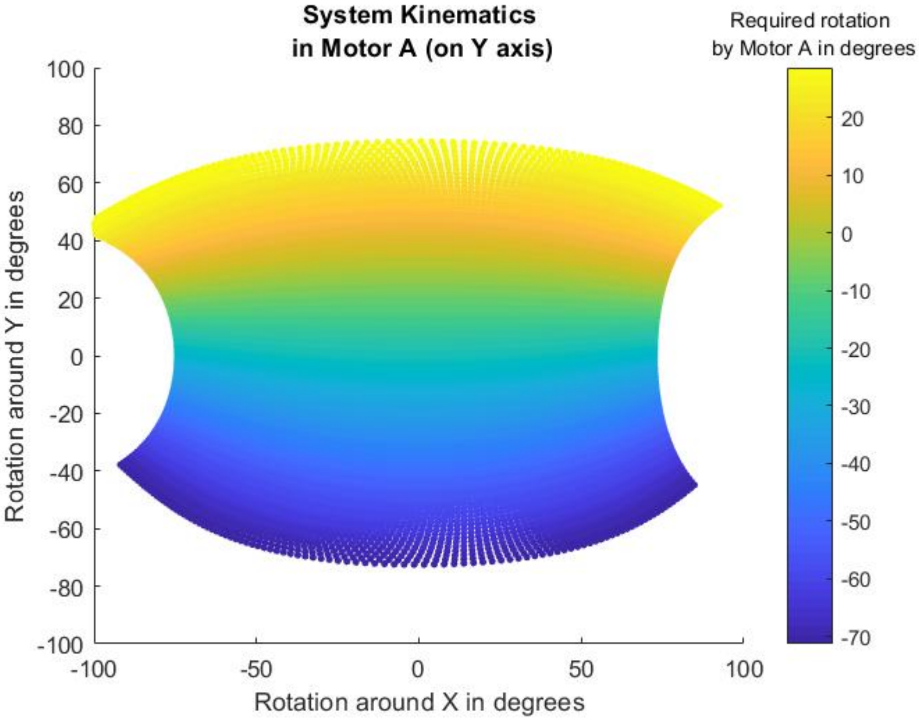


Figure 4.11: System Kinematics in Motor A (on Y axis).

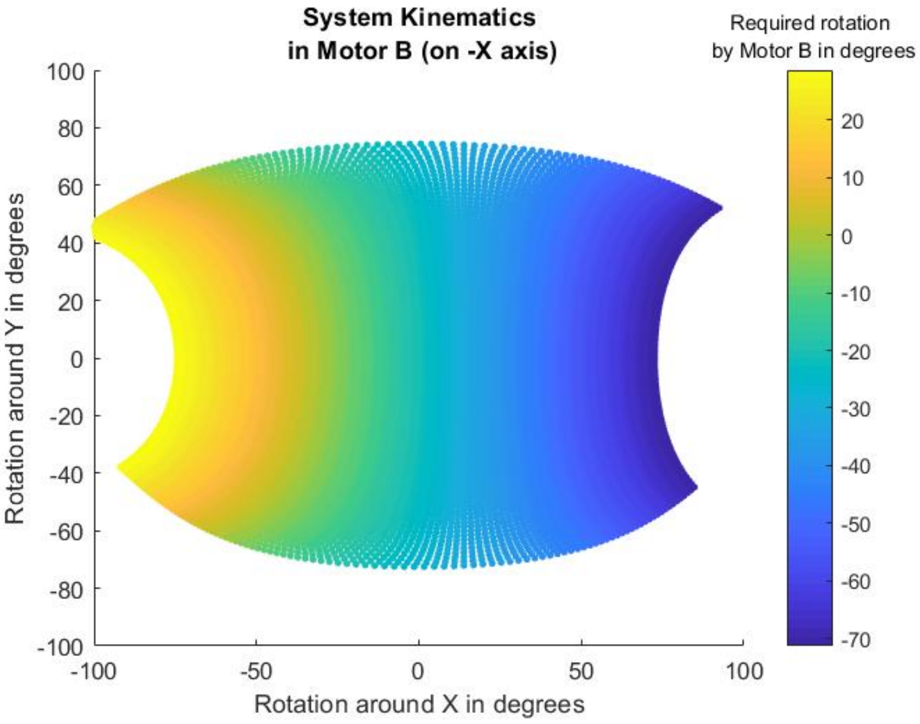


Figure 4.12: System Kinematics in Motor B (on -X axis).

relations can be shown as in Eqns. (4.22a)-(4.22i).

$$\Phi_{\beta_{A'},\beta_{A''}} = F_{\Phi}(\beta_{A'}, \beta_{A''}) \tag{4.22a}$$

$$\Theta_{\beta_{A'},\beta_{A''}} = F_{\Theta}(\beta_{A'}, \beta_{A''}) \tag{4.22b}$$

$$\Psi_{\beta_{A'},\beta_{A''}} = F_{\Psi}(\beta_{A'}, \beta_{A''}) \tag{4.22c}$$

$$\Phi_{\beta_{A'}+\delta,\beta_{A''}} = F_{\Phi}(\beta_{A'}+\delta, \beta_{A''}) \tag{4.22d}$$

$$\Theta_{\beta_{A'}+\delta,\beta_{A''}} = F_{\Theta}(\beta_{A'}+\delta, \beta_{A''}) \tag{4.22e}$$

$$\Psi_{\beta_{A'}+\delta,\beta_{A''}} = F_{\Psi}(\beta_{A'}+\delta, \beta_{A''}) \tag{4.22f}$$

$$\Phi_{\beta_{A'},\beta_{A''}+\delta} = F_{\Phi}(\beta_{A'}, \beta_{A''}+\delta) \tag{4.22g}$$

$$\Theta_{\beta_{A'},\beta_{A''}+\delta} = F_{\Theta}(\beta_{A'}, \beta_{A''}+\delta) \tag{4.22h}$$

$$\Psi_{\beta_{A'},\beta_{A''}+\delta} = F_{\Psi}(\beta_{A'}, \beta_{A''}+\delta) \tag{4.22i}$$

These data points in the mapping allow one to obtain the effect of a unit movement in inputs on the outputs of the system for any data point $\beta_{A'}$, $\beta_{A''}$ in the region of interest. Forward difference method is used to obtain these gains as in Eqns. (4.23a)-(4.23f).

$$\Delta\Phi_{\Delta\beta_{A'}} = \frac{\Phi_{\beta_{A'}+\delta,\beta_{A''}} - \Phi_{\beta_{A'},\beta_{A''}}}{\delta} = K_{\Phi}(\beta_{A'}, \beta_{A''}) \tag{4.23a}$$

$$\Delta\Phi_{\Delta\beta_{A''}} = \frac{\Phi_{\beta_{A'},\beta_{A''}+\delta} - \Phi_{\beta_{A'},\beta_{A''}}}{\delta} = L_{\Phi}(\beta_{A'}, \beta_{A''}) \tag{4.23b}$$

$$\Delta\Theta_{\Delta\beta_{A'}} = \frac{\Theta_{\beta_{A'}+\delta,\beta_{A''}} - \Theta_{\beta_{A'},\beta_{A''}}}{\delta} = M_{\Theta}(\beta_{A'}, \beta_{A''}) \tag{4.23c}$$

$$\Delta\Theta_{\Delta\beta_{A''}} = \frac{\Theta_{\beta_{A'},\beta_{A''}+\delta} - \Theta_{\beta_{A'},\beta_{A''}}}{\delta} = N_{\Theta}(\beta_{A'}, \beta_{A''}) \tag{4.23d}$$

$$\Delta\psi_{\Delta\beta_{A'}} = \frac{\psi_{\beta_{A'}+\delta,\beta_{A''}} - \psi_{\beta_{A'},\beta_{A''}}}{\delta} = P_{\psi}(\beta_{A'}, \beta_{A''}) \tag{4.23e}$$

$$\Delta\psi_{\Delta\beta_{A''}} = \frac{\psi_{\beta_{A'},\beta_{A''}+\delta} - \psi_{\beta_{A'},\beta_{A''}}}{\delta} = R_{\psi}(\beta_{A'}, \beta_{A''}) \tag{4.23f}$$

which allows us to obtain the Eqn. 4.24 for any data point $\beta_{A'}$, $\beta_{A''}$ in the region of interest:

$$\begin{bmatrix} K_{\Phi}(\beta_{A'}, \beta_{A''}) & L_{\Phi}(\beta_{A'}, \beta_{A''}) \\ M_{\Theta}(\beta_{A'}, \beta_{A''}) & N_{\Theta}(\beta_{A'}, \beta_{A''}) \\ P_{\psi}(\beta_{A'}, \beta_{A''}) & R_{\psi}(\beta_{A'}, \beta_{A''}) \end{bmatrix} \cdot \begin{bmatrix} \Delta\beta_{A'} \\ \Delta\beta_{A''} \end{bmatrix} = \begin{bmatrix} \Delta\Phi \\ \Delta\Theta \\ \Delta\Psi \end{bmatrix} \tag{4.24}$$

Note that, rotation around Ψ is not required to be known for our case. Φ and Θ are desired to be used define the position of our target. Hence, equations for Ψ can be omitted.

$$\begin{bmatrix} K_{\Phi}(\beta_{A'}, \beta_{A''}) & L_{\Phi}(\beta_{A'}, \beta_{A''}) \\ M_{\Theta}(\beta_{A'}, \beta_{A''}) & N_{\Theta}(\beta_{A'}, \beta_{A''}) \end{bmatrix} \cdot \begin{bmatrix} \Delta\beta_{A'} \\ \Delta\beta_{A''} \end{bmatrix} = \begin{bmatrix} \Delta\Phi \\ \Delta\Theta \end{bmatrix} \tag{4.25}$$

Hence, as long as the region of the system is known, the effect of a change in any input on the outputs is known. These equations can be reversed as in Eqn. (4.26):

$$\begin{bmatrix} K_{\Phi}(\beta_{A'}, \beta_{A''}) & L_{\Phi}(\beta_{A'}, \beta_{A''}) \\ M_{\Theta}(\beta_{A'}, \beta_{A''}) & N_{\Theta}(\beta_{A'}, \beta_{A''}) \end{bmatrix}^{-1} \cdot \begin{bmatrix} \Delta\Phi \\ \Delta\Theta \end{bmatrix} = \begin{bmatrix} \Delta\beta_{A'} \\ \Delta\beta_{A''} \end{bmatrix} \tag{4.26}$$

Hence, required input changes to have desired changes on outputs are obtained and they are also mapped for marginally small ranges of motion. However, this approach assumes the current region of the system is around the target and moves together with the target. Whenever there is a big displacement between the target and the current position of the system, considerable errors in the direction of movement occurs. It is actually required to directly go to the targets position even with large errors. In other words, it is required to have a mapping that correlates the required absolute angular orientation to motor angles even with high target tracking errors. This mapping can be obtained by different methods. Firstly, an inverse kinematics map can be constructed given only the forward kinematics mapping. For a given angular orientation, the closest points to that orientation can be picked from the outputs of forward kinematics map using a search algorithm. After finding the closest points to desired orientation and consequently their corresponding inputs of forward kinematics map, given angular orientation can then be used to 2-D interpolate between these input points. Repeating this process with equally distanced angular orientations, one can obtain the inverse kinematics map. Secondly, an inverse kinematics map can be obtained similar to the way forward kinematics are obtained. Same method (and same forward kinematics map generating code) can be used to generate the inverse kinematic map just as described in Section 4.3. Only difference from forward kinematics solution is the way inputs (known angles) are defined in the solution. Remembering that there is 23 unknown angles with 21 equations, one can define the angular orientation of plant as the inputs of the system using the following pseudocode which is implemented in Matlab:

1. Define initial values (known) to 23 angular variables and make sure they satisfy the kinematic equations.

2. Change Φ or Θ marginally.

3. By inserting the remaining 21 known angular variables, construct/update rotation matrices according to small angle assumption which includes 21 unknown small angle components.

4. By multiplying the rotation matrices (obtained by small angle assumption), obtain the equations (nonlinear system) described from Eqn. (4.19b)-(4.19g).

5. Solve equations by nonlinear regression using Levenberg-Marquardt Method [49] and obtain 21 small angles that satisfy the nonlinear system for the marginal change in Φ or Θ.

6. Add 21 small angles to their corresponding 21 angular variables.

7. Record the 23 angular variables as a satisfactory set of variables that satisfy the kinematic equations in a table.

8. Go to Step 2 until a satisfactory region of interest is obtained.

This secondly described method is used to obtain inverse kinematic maps in this case. Inverse kinematic maps obtained by this method can be seen in Figs. 4.134.14. Note that these are these inverse kinematic maps have equally distanced inputs unlike the figures that can be seen on Figs. 4.11 4.12.

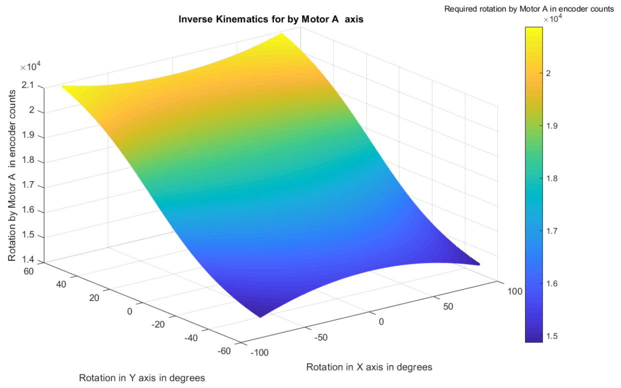


Figure 4.13: System Kinematics in Motor A (on Y axis)

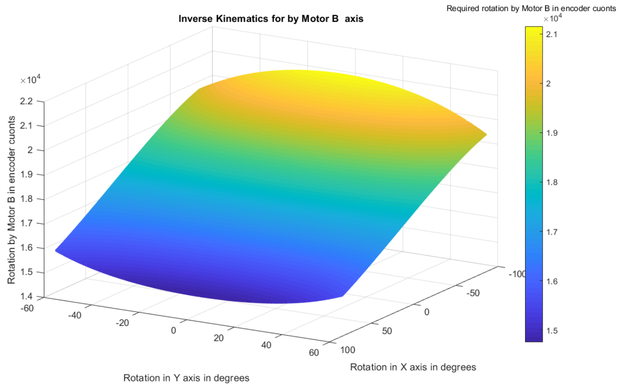


Figure 4.14: System Kinematics in Motor B (on -X axis)

4.6 Closure

As a conclusion, the kinematic model for the system designed in Chapter 3 is obtained. Forward and inverse kinematics are discretely mapped with respect to each other in their range of motion. Required input for any desired outputs are decoupled and mapped in forward and inverse. These discrete maps are saved as matrix forms to be used in later stages of the design, specifically in Section 7.1.1.

CHAPTER 5

DYNAMIC ANALYSIS OF MECHANISM

5.1 Introduction

It is required to know about the dynamical characteristics of the system for multiple reasons. Required torque for desired motions in different orientations are needed for motor selection and deciding on system configuration in Chapter 3. Some system parameters (like inertia) are also needed for controller design calculations for Chapter 6. In order to obtain these dynamical characteristics, a dynamical model is built. Using this model, required parameters are gathered.

5.2 Dynamic Model

Dynamical model is built inside Matlab Simscape environment which can be seen in Fig. 5.1. All bodies are imported from their CAD drawings. Mass, center of mass, moments of inertia and product of inertia of bodies are all calculated based on the predefined densities of parts. 3-D printed parts are defined with a density of 1.25 $[g/cm^3]$ which use PLA+ as printing material. These parts are named Lower Arm 1-4, Upper Arm 1-4, "Base" and "Plant Case" in Fig. 5.2. These parts are grey colored parts in Fig. 5.1. Steel part is defined with a density of 8 $[g/cm^3]$ which is the part named "Central Arm" in Fig. 5.2 and colored blue in Fig. 5.1. Furthermore, the part that symbolizes the camera/fan etc. on the device platform of the system is named "Plant" in Fig. 5.2 and shown in red in Fig. 5.1 which defined with a mass of 100 $[g]$. All bodies are connected with corresponding spherical or revolute joints as shown in Fig. 5.2.

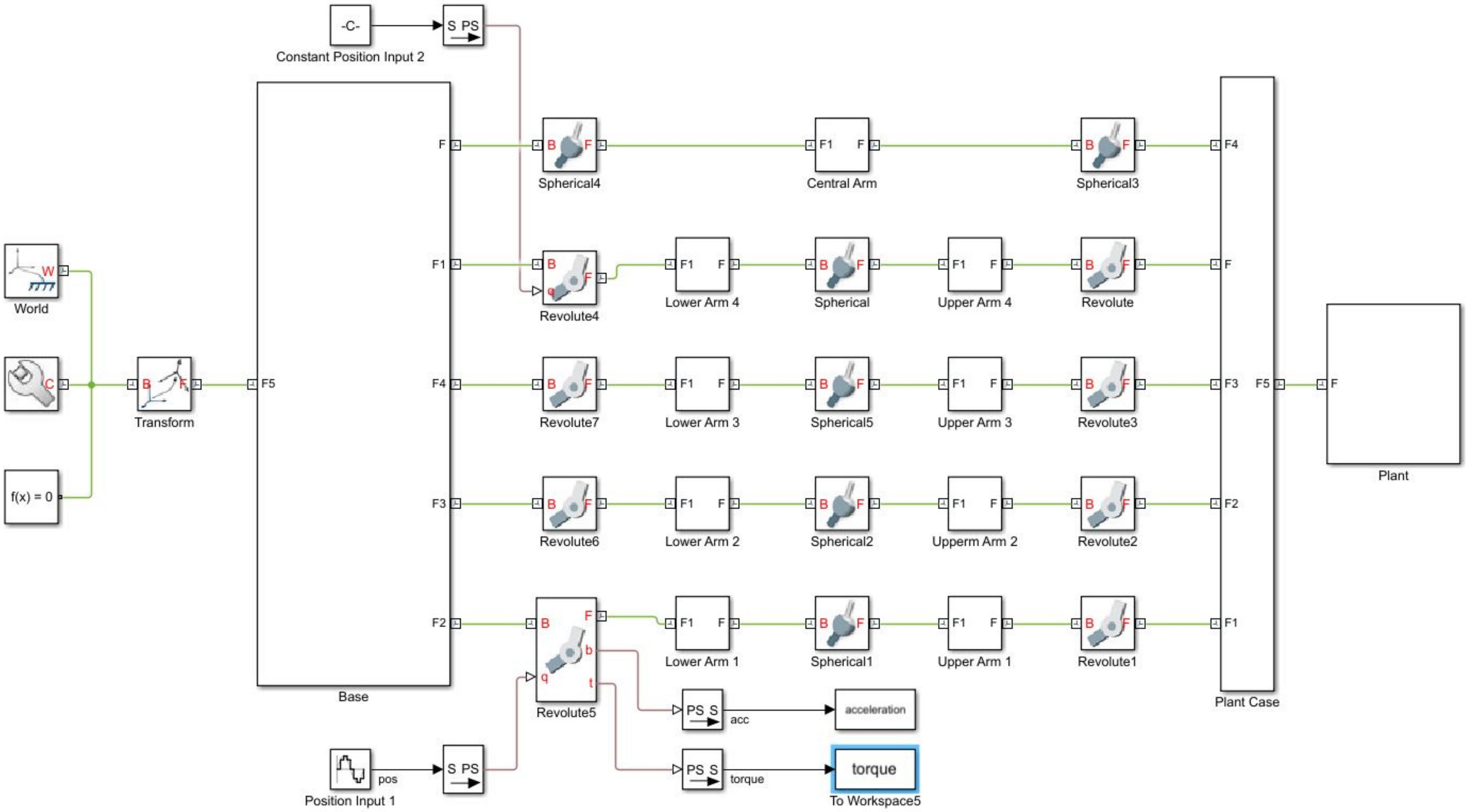


Figure 5.1: Simscape Model.

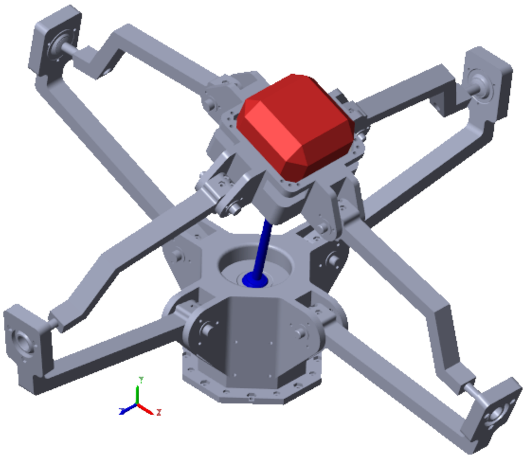


Figure 5.2: Image of Animated Model

5.3 Dynamic Simulations

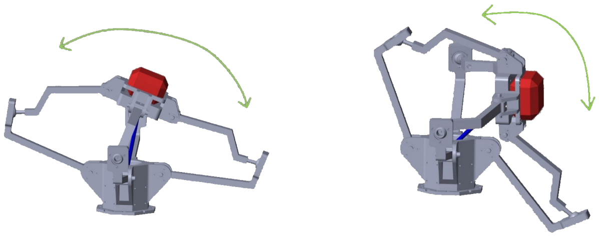
Model created in Section 5.2 is used for simulations. Several cases are simulated which can be divided into two types. First group of simulations are made neglecting gravity. These simulations are made to obtain an approximation for the inertia of the system. In all of the simulations, no viscous friction is defined at joints. Hence, only the inertia affects the torque of the system for the required motion. Torque of the system is automatically calculated by Simscape for the defined motion. Hence, equation of motion can be expressed in its simplest form as shown in Eqn. (5.1).

$$T(t) = J_L(t)\alpha(t) \tag{5.1}$$

In other words, the simulated inertia can be obtained by dividing the simulated torque to the simulated acceleration. Second group of simulations are made with gravitational acceleration in various directions. These simulations are made to obtain- and compare the torque required from the motor in different arrangements.

5.3.1 Dynamic Simulation Configuration I

In first simulation configuration, the inertia around vertical axis is approximated. A desired motion which follows a target at 2 $[m]$ distance doing a sinusoidal movement with 1 $[m]$ amplitude and with 1 $[Hz]$ frequency is used to evaluate equivalent inertia. Gravity is neglected. The system moves around its vertical axis with around ±30°amplitude. Desired motion given to the model can be seen in Fig. 5.4a which can be visualized for as shown in Fig. 5.3a. Resulting accelerations and torques from this motion can be seen in Fig. 5.4b and Fig. 5.4c, respectively. Resulting simulated torque is divided to resulting simulated acceleration which gives the varying inertia with respect to the position which can be seen in Fig. 5.4d. Maximum inertia occurs at ±30 degrees for this motion and it is 5.1210^{-5} $[kgm^2]$ while minimum inertia is at vertical position of mechanism which is 4.9710^{-5} $[kgm^2]$.



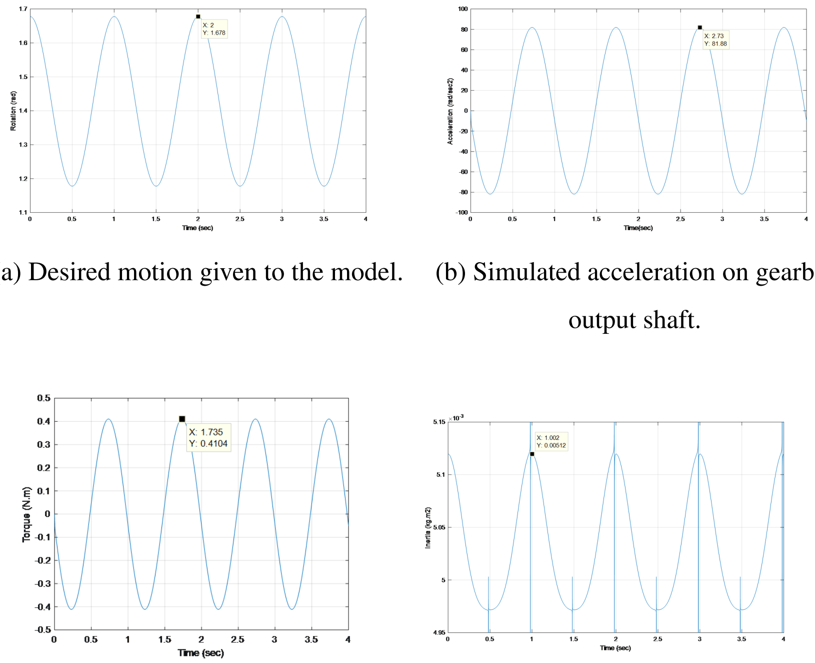
(a) Visualization of motion given to the model for dynamic simulationc configuration I.

(b) Visualization of motion given to the model for dynamic simulation configuration II.

Figure 5.3: Visualization of motion given to the model for dynamic simulation configurations I and II.

5.3.2 Dynamic Simulation Configuration II

In second simulation configuration, the inertia around horizontal axis is approximated. A desired motion which follows a target at 2 $[m]$ distance doing a sinusoidal movement with 1 $[m]$ amplitude and with 1 $[Hz]$ frequency is used. Gravity is neglected. The system moves around its horizontal axis with around ±30°amplitude. Desired motion given to the model can be seen in Fig. 5.5a which is visualized as shown in Fig. 5.3b. Resulting accelerations and torques from this motion can be seen in Fig. 5.5b and Fig. 5.5c, respectively. Simulated torque is divided to resulting simulated acceleration which gives the varying inertia with respect to the position which can be seen in Fig. 5.5d. Maximum inertia occurs at ±30 degrees for this motion and it is 6.3210^{-5} $[kgm^2]$.



(a) Desired motion given to the model.

(b) Simulated acceleration on gearbox output shaft.

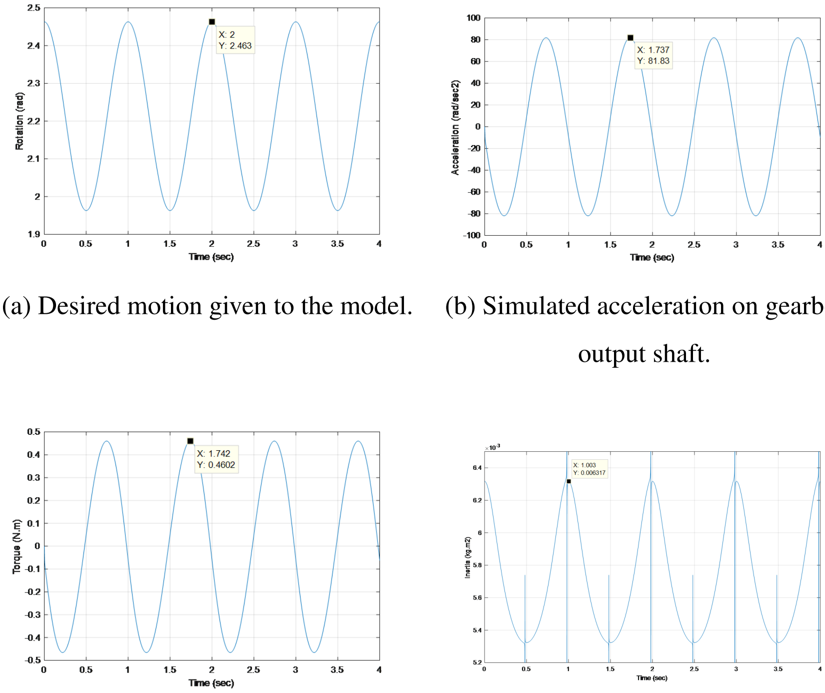
(c) Simulated torque on gearbox output shaft.

(d) Calculated inertia on gearbox output shaft.

Figure 5.4: Simulated Results for dynamic simulation configuration I.

5.3.3 Dynamic Simulation Configuration III

In third simulation configuration, the effects of weight of the system on required motor torque in while the system is in downwards gravitational acceleration environment is investigated. A desired motion which follows a target at 2 $[m]$ distance doing a sinusoidal movement with $1[m]$ amplitude and with 1 $[Hz]$ frequency is used. The system moves around its vertical axis with around ±30 amplitude. There is a gravitational acceleration applied downwards (in -z direction) which can be visualized for reader as shown in Fig. 5.6. Desired motion given to the model can be seen in Fig. 5.7a. Simulated required torque for this simulation can be seen in Fig. 5.7b which peaks at 0.4213 $[Nm]$.



(a) Desired motion given to the model.

(b) Simulated acceleration on gearbox output shaft.

(c) Simulated torque on gearbox output shaft.

(d) Calculated inertia on gearbox output shaft.

Figure 5.5: Simulated Results for dynamic simulation configuration II.

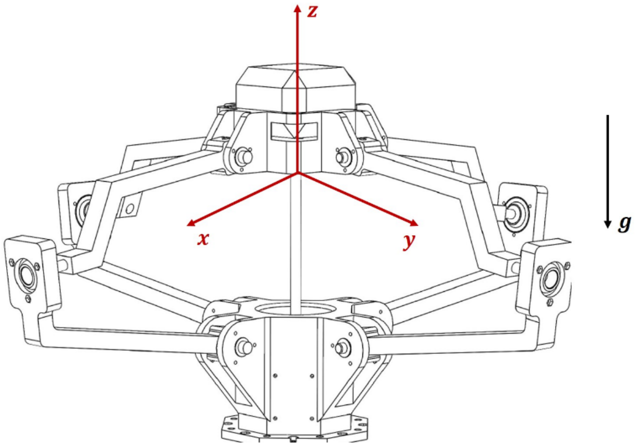
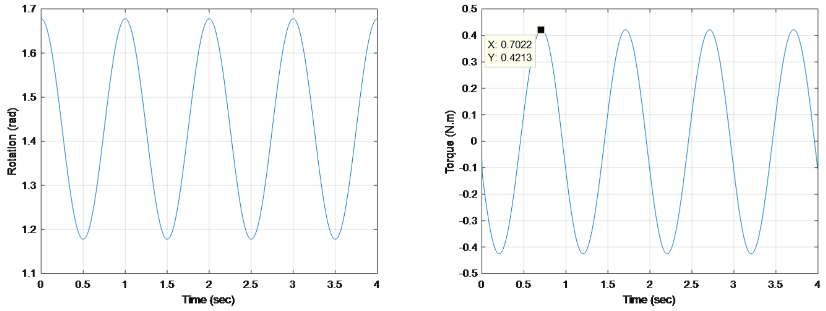


Figure 5.6: Visualization of motion given to the model for dynamic simulation configuration III.



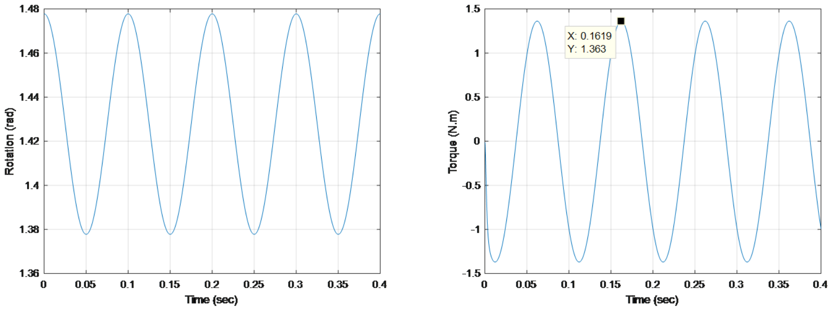
(a) Desired motion given to the model.

(b) Simulated torque on gearbox output shaft.

Figure 5.7: Simulated Results for dynamic simulation configuration III.

5.3.4 Dynamic Simulation Configuration IV

In fourth simulation configuration, the effects of weight of the system on required motor torque in while the system is in downwards gravitational acceleration environment with lesser amplitude and higher frequency is investigated. For this analysis, a target at 2 $[m]$ distance doing a sinusoidal movement with 1 $[m]$ amplitude and with 1 $[Hz]$ is used in an analysis which can be seen in Fig. 5.8a. Simulated required torque for this can be seen in Fig. 5.8b. which peaks at 1.363 $[Nm]$.



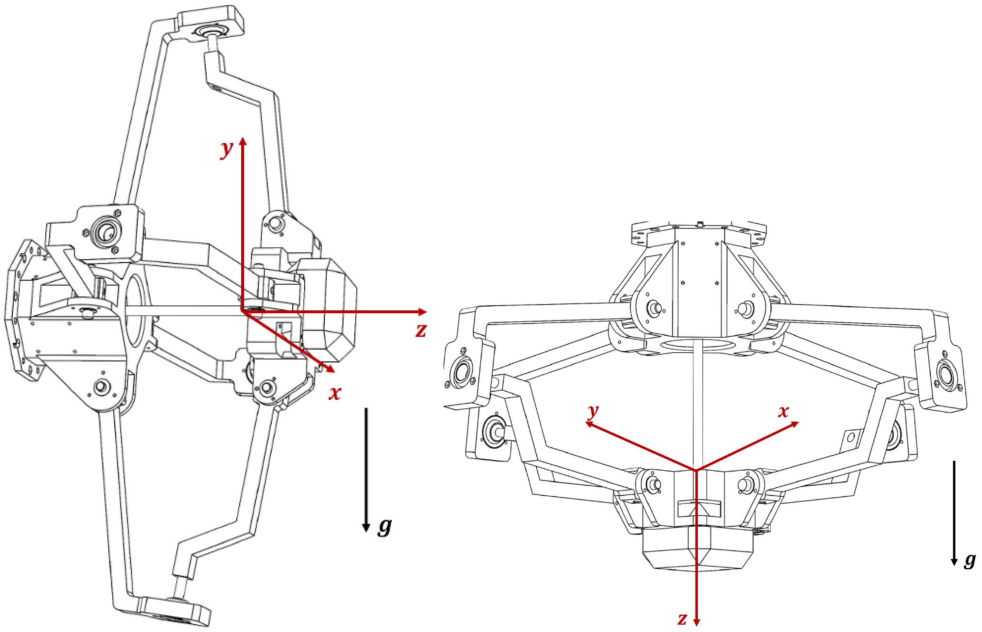
(a) Desired motion given to the model.

(b) Simulated torque on gearbox output shaft.

Figure 5.8: Simulated Results for dynamic simulation configuration IV

5.3.5 Dynamic Simulation Configuration V

In fifth simulation configuration, the effects of weight of the system on required motor torque in while the system is in sideways gravitational acceleration environment is investigated. A desired motion which follows a target at 2 $[m]$ distance doing a sinusoidal movement with 1 $[m]$ amplitude and with 1 $[Hz]$ frequency is used. The system moves around its vertical axis with around ± 30 amplitude. There is a gravitational acceleration applied downwards (in -y direction) which can be visualized for reader as shown in Fig. 5.9a. Desired motion given to the model can be seen in Fig. 5.7a which is the same as motion given to simulation configuration III. Simulated required torque for this simulation can be seen in Fig. 5.10 which peaks at 1.156 $[Nm]$. Note that this is almost three times the torque for doing the same movement in –z direction as in simulation III.



(a) Visualization of direction of gravity for simulation V.

(b) Visualization of direction of gravity for simulation VI.

Figure 5.9: Visualization of direction of gravity for simulation configurations V and VI.

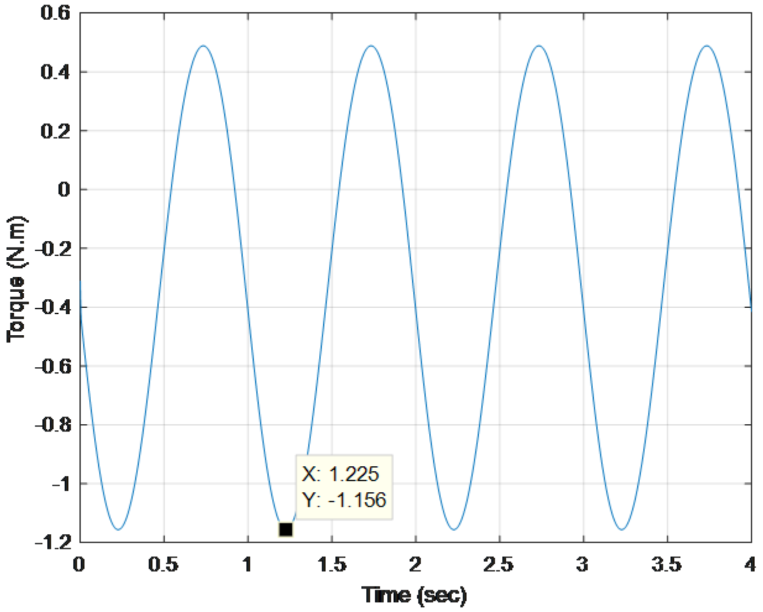


Figure 5.10: Simulated torque on gearbox output shaft for simulation configuration V.

5.3.6 Dynamic Simulation Configuration VI

In sixth simulation, the effects of weight of the system on required motor torque in while the system is in upwards gravitational acceleration environment is investigated. A desired motion which follows a target at 2 $[m]$ distance doing a sinusoidal movement with 1 $[m]$ amplitude and with 1 $[Hz]$ frequency is used. The system moves around its vertical axis with around $\pm30°$amplitude. There is a gravitational acceleration applied downwards (in+ z direction) which can be visualized for reader as shown in Fig. 5.9b. Desired motion given to the model can be seen in Fig. 5.7a which is the same as motion given to simulation configuration III. Simulated required torque for this simulation can be seen in Fig. 5.11 which peaks at 0.8172 $[Nm]$. Note that this is almost two times the torque for doing the same movement in –z direction as in simulation configuration III even though it is less than the torque required for the case in simulation configuration V.

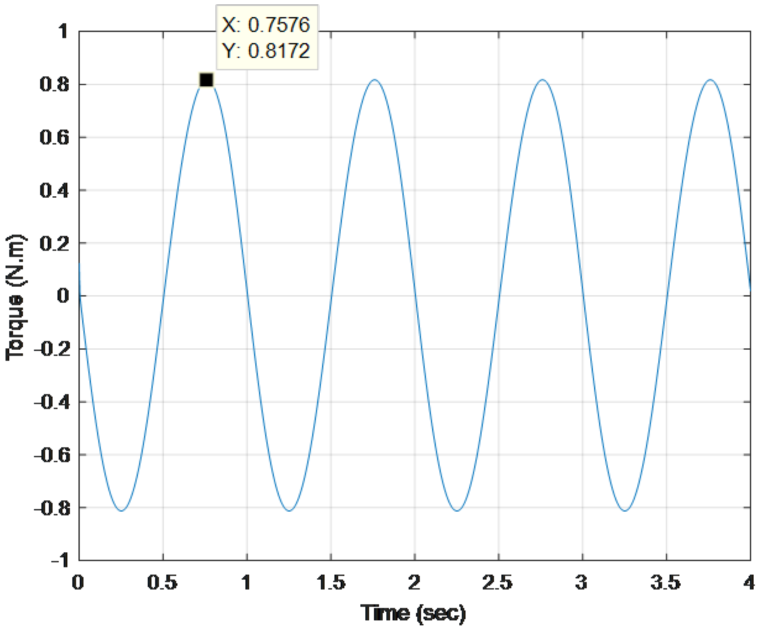


Figure 5.11: Simulated torque on gearbox output shaft for simulation configuration VI

5.4 Closure

Multiple deductions can be made after making these simulations. Firstly, the parallel tracking mechanism is expected to have an inertia between 4.9710^{-5} $[kgm^2]$ and 6.3210^{-5} $[kgm^2]$ in the working range of the system. Furthermore, the parallel targeting mechanism is expected to have a maximum inertia of 5.1210^{-5} $[kgm^2]$ around its vertical (an more linear) region. Secondly, It is estimated that, the minimum torque requirements from the motor to drive such a mechanism is 0.4213 $[Nm]$ for the expected motion of target. This torque increases as the mobility of target increases and a torque requirement of 1.363 $[Nm]$ is given in an example with increased frequency of target. Furthermore, the torque requirements from the motor in different orientations are also found in case the mechanism is assembled and montaged sideways or upside down depending on its utilization. Minimum torque requirements from the motor for sideways case is found to be 1.156 $[Nm]$. Lastly, the minimum torque requirements from the motor for upside down case is found to be 0.8172 $[Nm]$ for the expected motion of target.

CHAPTER 6

MOTION CONTROLLER DESIGN AND IMPLEMENTATION

6.1 Introduction

In this chapter, the motion controller of the parallel tracking platform is designed. In this study, a motion control strategy that makes good use of the paradigm mentioned in Section 2.4 is adopted to control the axes of the parallel tracking platform (PTP). The implementation details follow.

6.2 Parallel Targeting Mechanism Motion Control System

It is explained in Section 2.4 with an example that how a commercial servo-motor driver controller is structured. Unfortunately, due to budgetary restrictions, such commercial systems cannot be directly utilized in this project. Besides, the manufacturers of commercial servo-motor drivers often times do not allow the users to tailor the corresponding algorithms (i.e. the firmware) for the applications with extraordinary features. Also, they do not reveal the realization details of their control algorithms. Hence, in this study, a custom motion controller suitable for the target tracking applications is specially designed and implemented. Fig. 6.1 illustrates the simplified wiring diagram of the overall system.

A generic system in a cascaded motion control application typically includes a voltage source inverter driving an electrical motor. This motor acting as the prime mover (or actuator) could be of different types including brush-type DC motor, synchronous motor, and even induction motor. Consequently, the voltage source inverter is utilized to regulate the phase currents of the motor system such that the desired electromag-

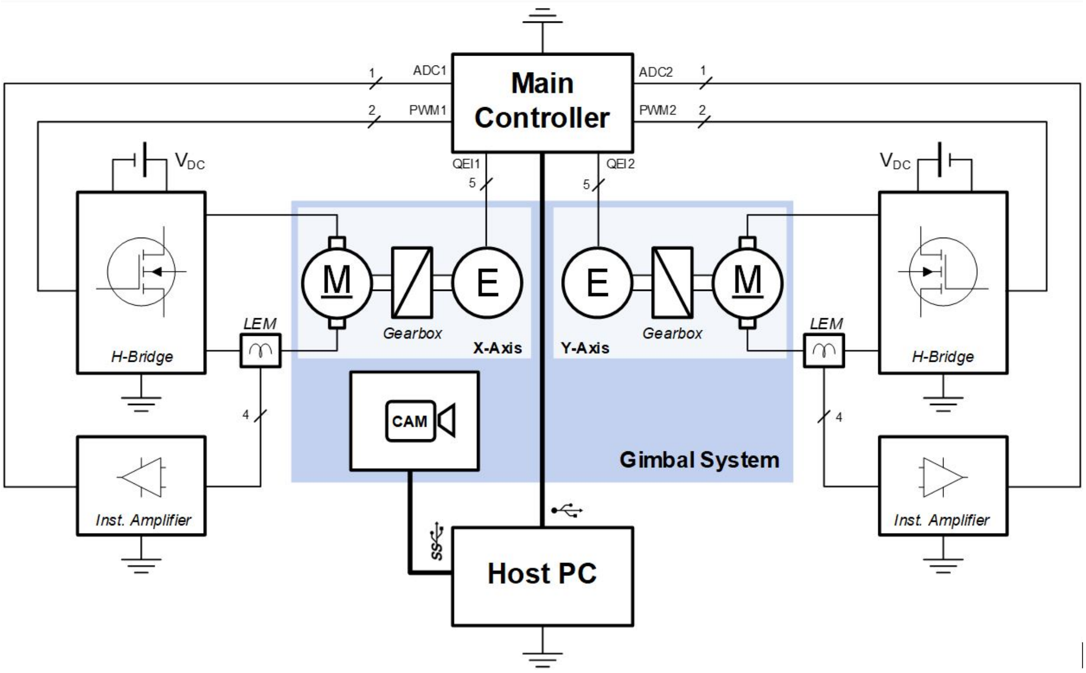


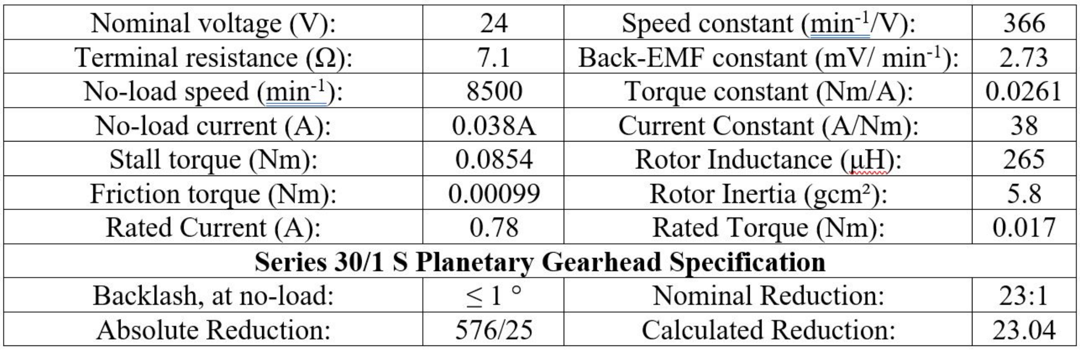
Figure 6.1: Simplified wiring schematics of the motion control system.

netic torque is produced. With the actuation torque closely regulated via the current control loop, one can directly modulate all relevant motion states of the plant under study including, acceleration/deceleration, velocity, and position that are highly correlated with the torque. Hence, the outer-loop generating the current (or torque) reference commands to the innermost/preceding loop regulates the velocity. Finally, the outermost controller feeding velocity commands to the intermittent loop strictly modulates the position.

As can be seen from Fig. 6.1, each axis of the PTP is driven by a brush-type DC motor with a built-in gearbox with a gearing ratio of 23 (see Section3.3). The properties of this motor Faulhaber 2342S024CR and its planetary gear head $30/1S$ 23 : 1 are summarized in Table 6.1 [1, 2].

The angular position of each drive shaft is measured through an optical position encoder AEDC-5550-Z12 using a 5000 count per revolution code wheel attached to the system. The encoder has two channels for reading counts on the code wheel, generating two outputs in quadrature which gives a resolution of 20000 pulses/channel (with 4X decoding) [50]. In this configuration, each motor is driven by an H-bridge. A dual H-Bridge motor driver IC (DRV8848) is selected to drive both motors which

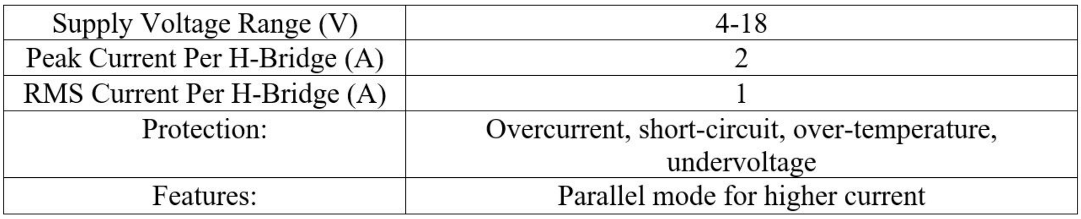
Table 6.1: Spec. of Faulhaber 2342S024CR and its planetary gearhead (30/1S) [1,2].



Nominal voltage (V):	24	Speed constant (min^{-1}/V):	366
Terminal resistance (Ω):	7.1	Back-EMF constant (mV/ min^{-1}):	2.73
No-load speed (min^{-1}):	8500	Torque constant (Nm/A):	0.0261
No-load current (A):	0.038A	Current Constant (A/Nm):	38
Stall torque (Nm):	0.0854	Rotor Inductance (μH):	265
Friction torque (Nm):	0.00099	Rotor Inertia (gcm^2):	5.8
Rated Current (A):	0.78	Rated Torque (Nm):	0.017
Series 30/1 S Planetary Gearhead Specification			
Backlash, at no-load:	≤ 1 °	Nominal Reduction:	23:1
Absolute Reduction:	576/25	Calculated Reduction:	23.04

accepts an independent logic signal for each half-bridge present in the driver IC. An evaluation module of the selected IC (BOOST-DRV8848) is utilized in the system for this purpose whose properties are tabulated in Table 6.2 [3].

Table 6.2: Specification of BOOST-DRV8848 Dual H-Bridge Motor Driver [3].



Supply Voltage Range (V)	4-18
Peak Current Per H-Bridge (A)	2
RMS Current Per H-Bridge (A)	1
Protection:	Overcurrent, short-circuit, over-temperature, undervoltage
Features:	Parallel mode for higher current

To regulate the DC motor currents, a high-performance current transducer (LEM HO 50-S) embodying a current transformer and a Hall-effect sensor is utilized [51]. This type of current transducer have the benefits of having a galvanic separation between the motor current circuit and sensor reading circuit. That is, the effect of the measured current on the motor current is zero. Although the sensitivity of the transducer can be significantly improved by increasing the turns on the primary (winding), the selected current transducer has a reference voltage (zero current voltage) of 2.5 $[V]$. Meanwhile, analog voltage range on the 12-bit ADC of the selected microcontroller is 0-3.3 $[V]$. Hence, increasing the number of turns and consequently the gain of the current transducer would not be satisfactory since its center is at 2.5 $[V]$ and not at the middle of 0-3.3 $[V]$ range of microcontroller. For this reason, a special instrumentation amplifier as shown in Fig. 6.2 is employed to amplify further and offset the output of the transducer. The current transducer LEM HO 50-S has the theoretical sensitivity of 16 $[mV/A]$. With use of the special instrumentation, the zero current voltage is offset to $2[V]$ and sensitivity of current sensing is amplified to 800 $[mV/A]$. Follower Op-amps are employed at entry points of special instrumentation to have high impedance at entry points and to not possibly affect the circuit inside the transducer with resistors. A low pass filter with 10 $[kHz]$ cut-off frequency is added at the end of the circuit. The effect of this circuit is increased sensitivity and shifted output voltage range. This effect can be seen in Fig. 6.2.

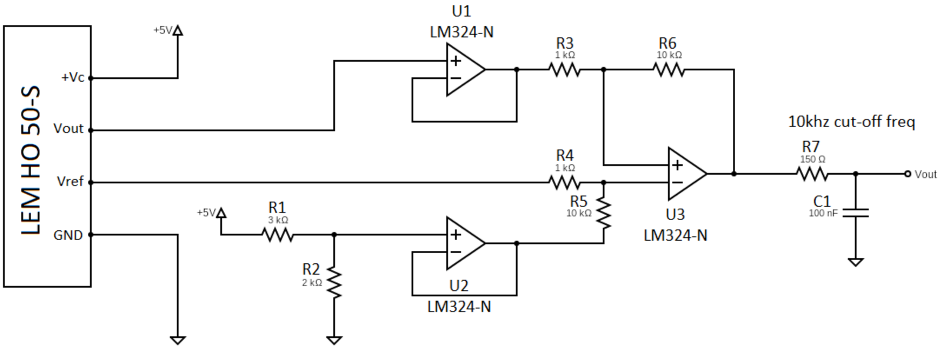


Figure 6.2: Circuit schematics of the current transducer and special instrumental amplifier

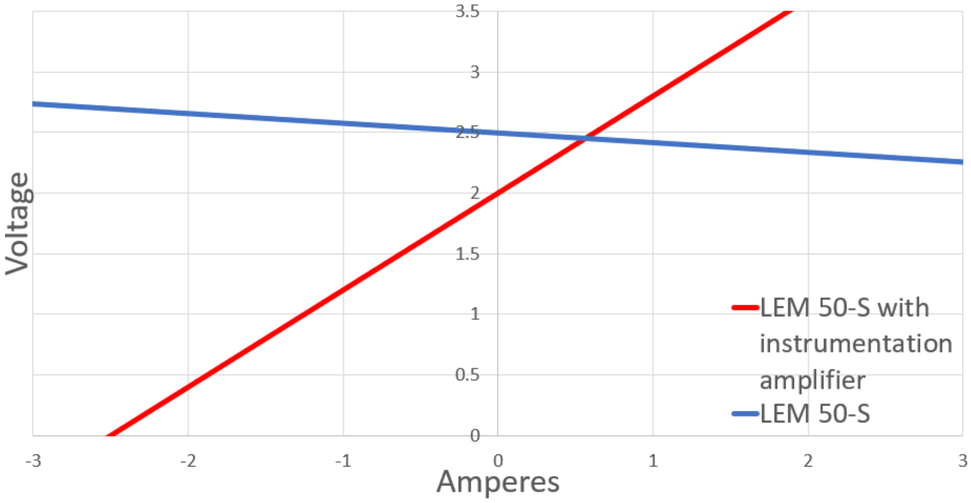


Figure 6.3: Amperes to voltages conversion of current transducer with and without special instrumental amplifier circuit

As for the main controller, Tiva-C Series (TM4C123GH6PM) microcontroller is selected. An evaluation board of selected microcontroller (EK-TM4C123GXL) is employed to be utilized in the system. Table 6.3 highlights some of the important features of this board incorporating a powerful 32-bit ARM Cortex-M4F processor with floating-point arithmetic operation capability [4]. This micro-controller will be employed as a motion controller to process the readings from sensors, send PWM commands to H-bridges, send and receive critical data from visual servo controller while calculating the required PWM outputs using the inputs from sensors and visual servo controller. The selected micro-controller has multiple ADCs for current transducer readings, multiple quadrature encoder interface for encoder readings, multiple USB 2.0 interfaces for communication with main system and in-line debugging of the firmware, PWM modules generating PWM signals and floating point units for faster mathematical fraction calculations. Discrete-time control algorithms discussed in this chapter are all devised in C language using TI's Code Composer Studio (CCS) IDE. Consequently, a high-resolution, global shutter, CMOS camera (Basler Dart daA1600-60uc-S), whose technical attributes are given in Table 6.4, is attached onto the PTP for target-tracking- and visual-servo applications The assembly and implementation of motion control system can be seen in Figs. 6.4, 6.5.

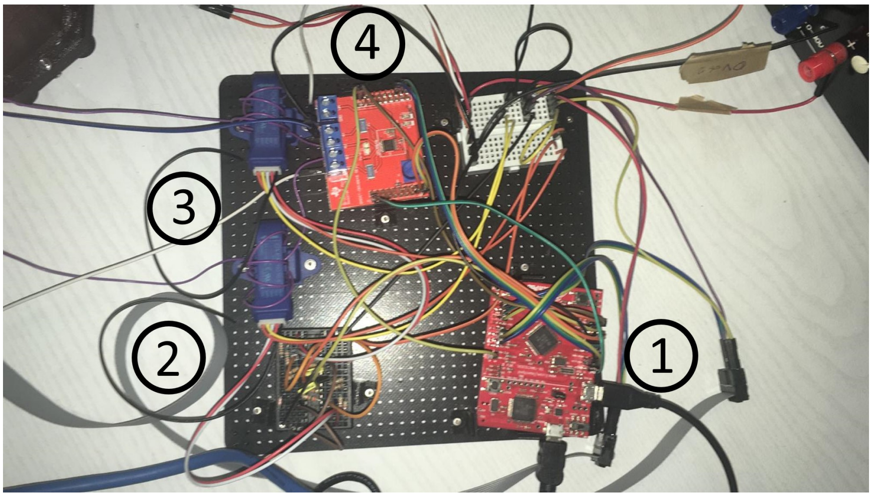


Figure 6.4: Electronics employed in control system. 1) Microcontroller board: EK-TM4C123GXL 2) Instrumental amplifier circuit 3) Current transducers: 2x LEM HO 50-S 4) Dual H-Bridge Motor Driver: BOOST-DRV8848.

Table 6.3: Spec. of TM4C123GH6PM 32-bit ARM® Cortex®-M4F based MCU [4].

Core:	ARM Cortex-M4F processor core
Performance:	80-MHz operation; 100 DMIPS performance
Universal Serial Bus (USB):	USB 2.0 OTG/Host/Device
Pulse Width Modulator (PWM):	Two PWM modules, each with four PWM generator blocks and a control block, for a total of 16 PWM outputs.
Quadrature Encoder Interface (QEI):	Two QEI modules
Analog-to-Digital Converter (ADC):	Two 12-bit ADC modules, each with a maximum sample rate of one million samples/second

Table 6.4: Specification of Basler Dart daA1600-60uc-S [5].

Resolution (H×V pixels):	1600×1200
Sensor Type:	Progressive scan CMOS, Global Shutter
Optical Size:	1/1.8"
Pixel Size (H × V):	4.5µm × 4.5µm
Max Frame Rate (At full resolution):	60

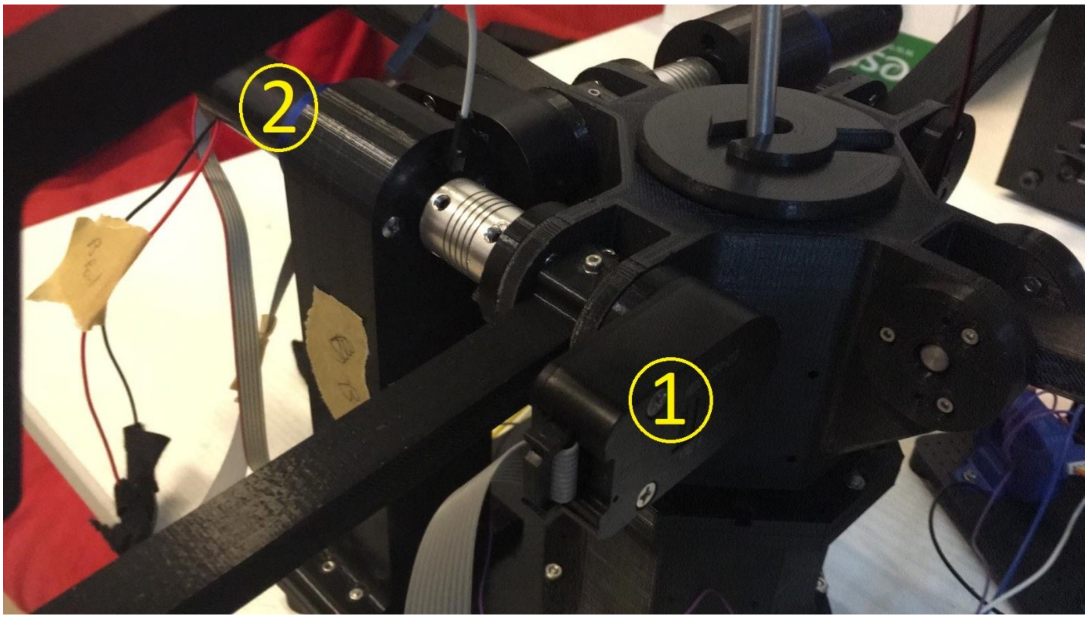


Figure 6.5: Electronics employed in control system. 1) Incremental encoder: AEDC-5550-Z12 2) Brushed DC Motor: Faulhaber 2342S024CR and its planetary gearhead: (30/1S).

Since digital image processing requires a significant amount of computing power, a PC has to be incorporated to the setup so as to acquire the digital images captured the camera via USB 3.0 serial communication interface. The host PC processes a set of images to obtain the relative position of the target (with respect to camera reference frame) and generates a position error signal for each PTP axis in pseudo real-time. Consequently, these reference signals are transferred to the main controller via USB 2.0 serial communication interface to guide the PTP (housing the CMOS camera) to track the target accurately. The details regarding the visual servo algorithms and their implementations will be elaborated in Chapter 7. The next section discusses the model of the plant system.

6.3 Electromechanical Model of the Plant

To control the motion of the PTP accurately, the mathematical models describing the dynamics of the PTP axes must be developed first. For this purpose, model of a DC motor with a built-in gearbox is considered to be driving a PTP axis. The equations for this model can be seen in Eqns. (6.1),(6.2).

$$J_e \frac{d^2\theta}{dt^2} + b_e \frac{d\theta}{dt} = K_t i - \frac{T_L(t)}{N} \tag{6.1}$$

$$L\frac{di}{dt} + Ri = V_a - K_e \omega \tag{6.2}$$

where

θ**:** Motor (i.e. gearbox input shaft) angular position [rad];

ω**:** Motor (i.e. gearbox input shaft) angular velocity [rad/s];

i**:** Motor current [A];

V_a**:** Motor armature voltage [V];

T_L**:** Load torque acting on the gearbox output shaft [Nm];

N**:** Gearing ratio;

K_t**:** DC motor torque constant [Nm/A];

K_e**:** DC motor back EMF constant [V/rad/s];

J_e**:** Equivalent mass moment of inertia [kgm2];

b_e**:** Equivalent viscous damping [Nm/rad/s];

L**:** Winding resistance [Ω].

R**:** Load torque acting on the gearbox output shaft [Nm];

Note that the PTP mechanical system has a direct effect on the important physical parameters of the model as it can be seen in Eqns. (6.3),(6.4).

$$J_e \triangleq J_m + \frac{J_L(t)}{N^2} \tag{6.3}$$

$$b_e \triangleq b_m + \frac{b_L(t)}{N^2} \tag{6.4}$$

Here, J_m is the mass moment of inertia $[kgm^2]$ of the rotor while J_L denotes the mass moment of inertia $[kgm^2]$ of the PTP as reflected onto the gearbox output shaft. Note the quantity J_L depends on the overall position of the PTP in both axes. J_L is obtained using the dynamical model of the system obtained in Chapter 5. Values for J_L are obtained in different points of the system close to the vertical position of the system and a rough estimate for the J_L is taken. Even though there is a gearbox with ratio of N, the effect of J_L on the motor equivalent inertia is significant due to long moment arm attached to the gearbox output shaft. To be on the safe side, a significantly higher J_L value and consequently a marginally higher J_e value is used in calculations. Furthermore, it should be noted that effect of inertia of gearbox on motor shaft is neglected. Likewise, b_m in Eqn. (6.4 is the viscous damping coefficient $[Nm/rad/s]$ of the motor shaft while b_L, which is a function of time, refers to the viscous damping on the gearbox output shaft. Effect of gearbox output shaft viscous damping caused by ball bearings b_L are not significant due to the gearbox ratio with respect to b_m. As an approximation, b_e is taken as equal to bm in calculations. It is critical to note that the load torque T_L in Eqn. (6.1) depends on the payload of the PTP and varies significantly as the PTP axes change their orientations. Expected viscous damping on the gearbox output shaft is assumed to not affect the system.

6.4 Discrete-time Motion Control System

Using the model in given Section 6.3, a motion (axis) control system, which is briefly described in Section 6.2, can be developed. The block diagram of this system is shown in Fig. 6.6. As can be seen, this discrete-time cascade control system consists of three nested loops:

- Innermost PI-based current control loop with back EMF compensation;
- Intermittent velocity control loop with PI-controller;
- PID-based outer position control loop.

In this configuration, since the angular velocity ω is not directly measured, it has to be estimated by taking the first-order (backward) differences of position measurements (θ). Similarly, the back EMF compensation is to be implemented using the velocity estimate. It is critical to note that this scheme is often times considered as a risky trait in case the back EMF constant is overestimated ($\hat{K}_e > K_e$). It will, in turn, introduce positive feedback to the current control loop and may destabilize the system. Hence, as a safer practice, the compensation can be done using the velocity command rather than the measurement itself. Consequently, assuming that the sampling frequency ($1/T$) is relatively high (» $1\ kHz$) and that an effective back EMF compensation (with a reasonable estimate,$\hat{K}_e$) is performed, the simplified PTP axis model (with $T_L = 0$) can be obtained as shown in Fig. 6. Hence, the block diagram of the generic discrete-time control system is illustrated in Fig. 6.8. Here, the transfer functions are defined as shown in Eqns. (6.5a)-(6.5c).

$$\frac{I(z)}{V_a(z)} = (1 - z^{-1})Z\left\{\frac{J_e s + b_e}{s((J_e s + b_e)(L_s + R) + K_t(K_e - \hat{K}_e))}\right\} \tag{6.5a}$$

$$\frac{\omega(z)}{V_a(z)} = (1 - z^{-1})Z\left\{\frac{K_t}{s((J_e s + b_e)(L_s + R) + K_t(K_e - \hat{K}_e))}\right\} \tag{6.5b}$$

$$\frac{\theta(z)}{(V_a(z)} = (1 - z^{-1})Z\left\{\frac{K_t}{s^2((J_e s + b_e)(L_s + R) + K_t(K_e - \hat{K}_e))}\right\} \tag{6.5c}$$

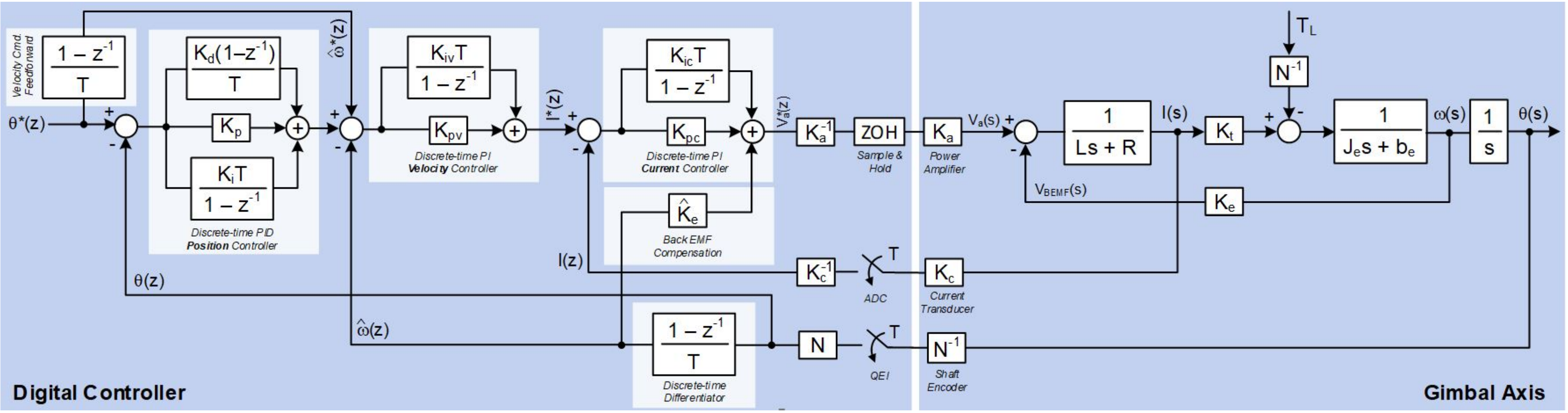


Figure 6.6: Block diagram of the initial concept of motion control system.

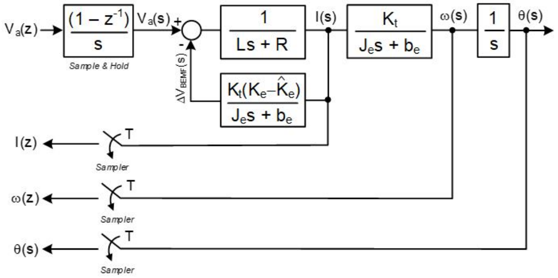


Figure 6.7: Simplified model of the plant.

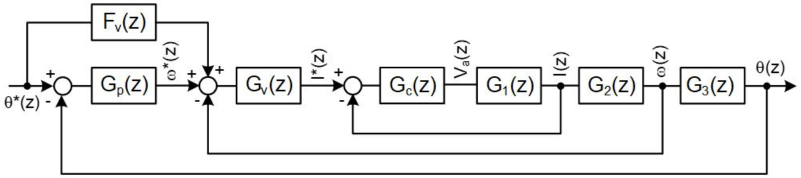


Figure 6.8: Overall discrete-time control system.

where $Z.$ refers to the z-transform of a given complex (Laplace) function. By definition, the transfer functions shown in Fig. 6.8 takes the form as shown in Eqns. (6.6a)-(6.6c).

$$G_1(z) \triangleq \frac{I(z)}{V_a(z)} \tag{6.6a}$$

$$G_2(z) \triangleq \frac{\omega(z)}{I(z)} = \frac{\omega(z)}{V_a(z)}\frac{V_a(z)}{I(z)} = \frac{\omega(z)}{V_a(z)}G_1^{-1}(z) \tag{6.6b}$$

$$G_3(z) \triangleq \frac{\theta(z)}{\omega(z)} = \frac{\theta(z)}{V_a(z)}\frac{V_a(z)}{\omega(z)} = \frac{\theta(z)}{V_a(z)}G_2^{-1}(z) \tag{6.6c}$$

The following section discusses the tuning procedure of the control loops using the discrete-time models devised here.

6.5 Tuning the Digital Controllers

In cascade control, the tuning operation starts out with the innermost loop. Parameters of the controller (i.e. gains) associated with the loop (or the state, to be exact) are selected such that the closed-loop poles (eigenvalues) are placed to the desired locations in the z-plane and that the performance of the controlled state is acceptable within the framework of technical specifications. Once the controller gains are set (and the accompanying closed-loop transfer function is fixed), one proceeds to tune the next controller on the outer loop that inherently incorporates the previously tuned loops.

6.5.1 Current Loop

With the nominal (and identified) parameters of the DC motor system at hand (as discussed in Section 6.3, the discrete-time model of the current loop can be formed according to Eqn. (6.6a).

$$G_1(z) = \frac{0.09822(z - 0.9995)}{(z - 0.0177)(z - 0.9993)} \tag{6.7}$$

The EMF compensation plays a critical role in the dynamics of the loop as it tends to reduce the current/torque generation capability of the driver at high motor velocities.

However, it should be noted that pole at 0.9993 and zero at 0.9995 are introduced by the back-EMF feedback of current loop which can be seen in Fig. 6.7. Hence, pole and zero introduced by back-EMF are already very close to each other. This is a result of the motor of system being considerably small and having a very small back-EMF constant. Assuming that back-EMF constant (i.e. $0.9K_e \leq \hat{K}_e \leq K_e$) is underestimated within 10% error-band, Eqn. (6.6a) takes the form as shown in Eqn. (6.8).

$$G_1(z) = \frac{0.09823(z - 0.9995)}{(z - 0.0177)(z - 0.9995)} \tag{6.8}$$

Root locus of this plant belonging to current loop can be seen in Fig. 6.9. As it can be seen from the figure, the dominant pole of the system diverges to unstable region root locus. However, a PI controller can be added to the system so that the dominant pole will be inside the stable region with desired behavior.

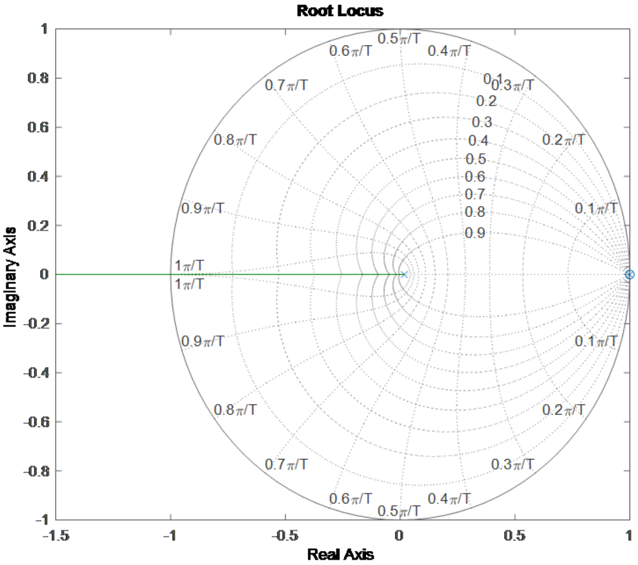


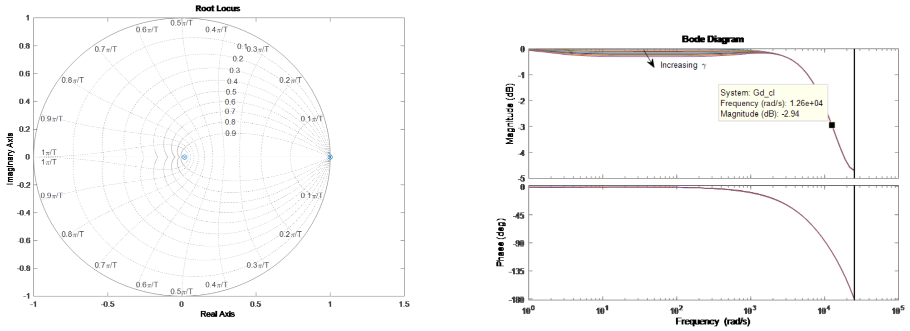
Figure 6.9: Root Locus of the open-loop without control.

Consequently, the discrete-time PI controller with two free parameters can be expressed as shown in Eqn. (6.9).

$$G_c(z) = K\frac{z - b}{z - 1} \tag{6.9}$$

where by definition, $K = K_{ic}T + K_{pc}$ and $b = \frac{K_{pc}}{K_{ic}T + K_{pc}}$. Selecting the zero of the controller to cancel the slowest pole of the plant (i.e. $b = 0.0177$), one can choose the desired dominant pole locations of the closed-loop system using the classical root placement techniques by adjusting the remaining free parameter K. Fig. 6.10 demonstrates the root locus of the closed-loop system along with its Bode plots. The dominant pole is placed such that the cut-off frequency of the tuned-loop is approximately set to $2000\ [Hz]$ (see Fig.6.10b). Note that as a consequence of the Nyquist-Shannon sampling theorem, the maximum achievable bandwidth frequency is half the sampling frequency of this digital servo-loop ($4\ [kHz]$). As can be seen from Fig. 6.10b, the back EMF compensation has a direct effect on control performance where the dimensionless parameter $\gamma \in [0,1]$ is here defined as shown in Eqn. (6.10).

$$\gamma = 1 - \frac{\hat{K}_e}{K_e} \tag{6.10}$$



(a) Root locus

(b) Bode plot

Figure 6.10: Characteristics of the closed-loop current control system.

The (tuned) PI parameters are read off as Kpc = 0.14301; KicT = 7.9211 while the transfer function of the closed-loop current control becomes

$$\frac{I(z)}{I^*(z)} = \frac{0.7921(z-0.0177)(z-0.9995)}{(z-0.2079)(z-0.0177)(z-0.9995)} \tag{6.11}$$

It is critical to notice that in Eqn. (6.11), two of the closed-loop poles appear to be perfectly canceled by the corresponding zeros due to four-digit (after the decimal point) representation of the real numbers in the transfer function. However, the perfect zero-pole cancellation only occurs as $K \to \infty$. For the sake of simplifying the tuning procedure for the outer loops, the closed-loop transfer function of the inner loops are often-times replaced by the lower-order ones assuming that the relevant zero-pole cancellations are to take place [i.e. minimal (state) realization].

6.5.2 Velocity Loop

With the current controller in place, the tuning of the digital PI velocity controller is in order. Using the identified parameters of the system in Section 6.3 , the relevant (discrete-time) transfer functions (as shown in Fig. 6.8) can be obtained. Hence, Eqn. (6.6b) is written as shown in Eqn. (6.12).

$$G_2(z) = \frac{0.05024(z + 0.2985)}{(z - 0.9995)} \tag{6.12}$$

Hence, the transfer function of the plant can be expressed as seen in Eqn. (6.13).

$$\frac{\omega(z)}{I^*(z)} = \frac{0.03979(z + 0.2985)(z - 0.9995)}{(z - 0.2079)(z - 0.9995)(z - 0.9995)} \tag{6.13}$$

Consequently, the discrete-time PI controller (i.e. G_v) in the form of 6.9 is employed. Selecting the zero of the controller to cancel the slowest pole of the plant (i.e. $b = 0.9995$), the dominant poles can be placed to the desired locations in the z-plane using root locus techniques. Fig. 6.11 shows the root locus of the open-loop system. It can be seen that, the affect of a PI controller will be minimal on the system because the slowest pole is very close to 1 (i.e. $b = 0.9995$). However, still a PI controller will be implemented while expecting a small Integral gain in the PI controller.

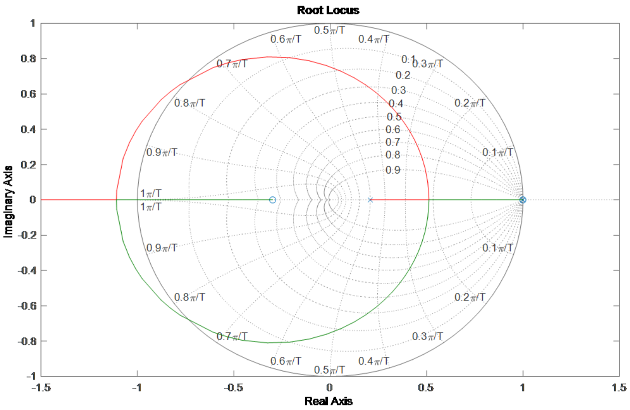


Figure 6.11: Root Locus of the velocity open-loop without control.

After placing the zero on the pole at $b = 0.9995$ the root locus in Fig. 6.12 can be obtained. It can be seen that, if the system and the current loop bandwidth was high enough, closed-loop poles can be placed for up to gains corresponding to $880\ [Hz]$. However, since it is desired to have velocity loop bandwidth to be ten times lower than current loop bandwidth, a gain corresponding to 200 $[Hz]$ is selected. Increasing the gain up to 200 $[Hz]$ does not place the closed-loop poles as complex-conjugates.

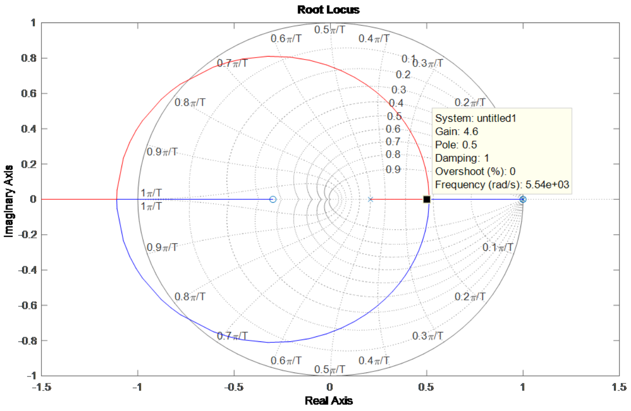
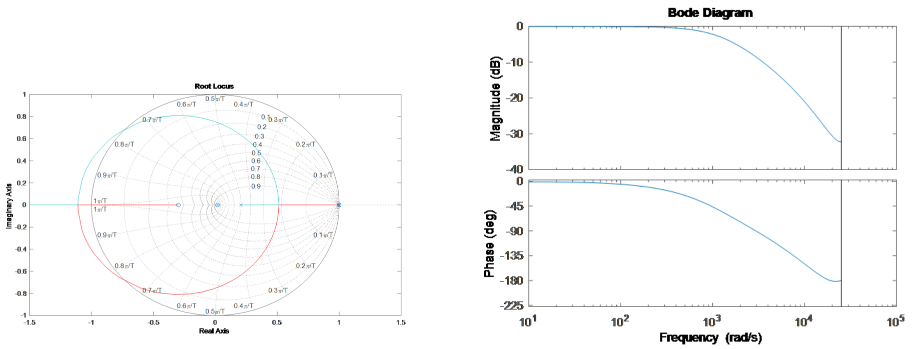


Figure 6.12: Root Locus of the velocity open-loop with I controller added to velocity control.

Fig. 6.13 illustrates the root locus of the closed-loop system along with the Bode plots. In this case, the dominant poles are placed such that the cut-off frequency of the closed-loop is roughly set to 200 $[Hz]$ (Refer to Fig. 6.13).



(a) Root locus

(b) Bode plot

Figure 6.13: Characteristics of the closed-loop velocity control system.

The (tuned) PI parameters are read as $K_{pv} = 2.0481$; $K_{ivT} = 0.0010242$. Transfer function of the closed-loop velocity control system can be shown in Eqn. (6.14).

$$\frac{\omega(z)}{\omega^*(z)} = \frac{0.08153(z+0.2985)(z-0.9995)(z-0.9995)}{(z-0.2079)(z-0.8546)(z-0.9995)(z-0.9995)} \tag{6.14}$$

6.5.3 Position Loop

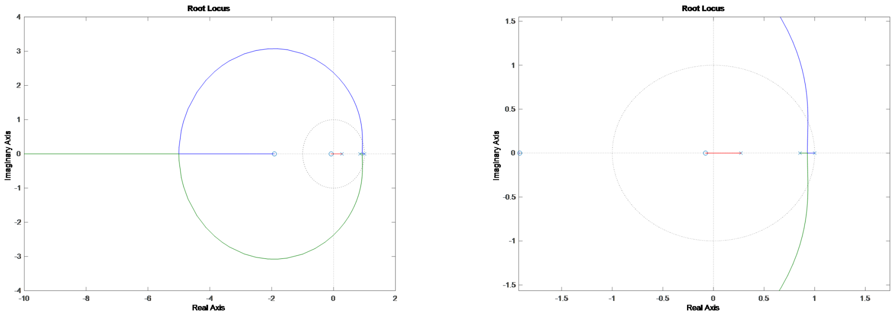
With the velocity controller at hand, the digital P position controller can be tuned. Employing the identified parameters of the system in Section 6.3, the relevant (discrete-time) transfer functions (as shown in Fig. 6.8) can be obtained. Hence, Eqn. (6.6c) is written as shown in Eqn. (6.15).

$$G_3(z) = \frac{5.163 \times 10^{-5}(z+1.9140)(z+0.0788)(z-0.9995)}{(z+0.2985)(z-0.9995)(z-1)} \tag{6.15}$$

Hence, the transfer function of the plant can be expressed as seen in Eqn. (6.16).

$$\frac{\theta(z)}{I^*(z)} = \frac{4.209 \times 10^{-6}(z+1.9140)(z+0.0788)(z-0.9995)}{(z-0.2717)(z-0.8546)(z-0.9995)(z-1)}() \tag{6.16}$$

Consequently, the discrete-time P controller (i.e. G_p) is employed. Again, the dominant poles are placed to the desired locations in the z-plane using root locus techniques as shown in Fig. 6.14. Similarly, Fig. 6.15 illustrates the Bode plots of the closed-loop position control system. In this case, the dominant poles are placed such that the cut-off frequency of the closed-loop is roughly set to 20 $[Hz]$ (Refer to Fig. 6.15). The (tuned) P parameter is found as $K_p = 111.1641$ (with $K_iT = K_d/T = 0$).



(a) Root locus of position open-loop

(b) Root locus of position open-loop

Figure 6.14: Root locus of the closed-loop position control system.

6.5.4 Control Loop Performance Tests

To demonstrate the real system behaves as in the model, sine waves with varying frequency are sent to implemented current loop and the current through the motors

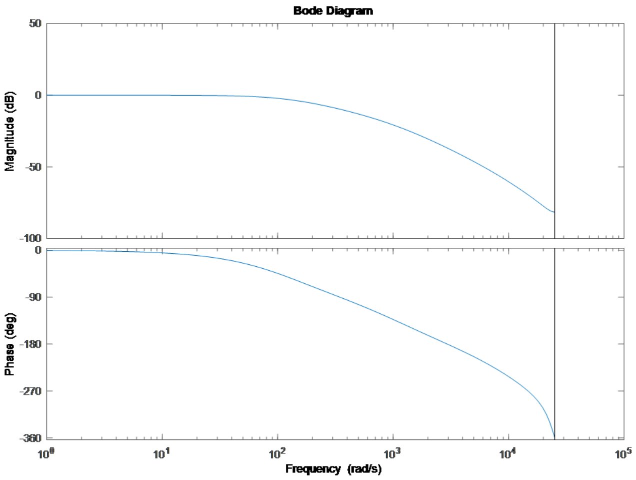
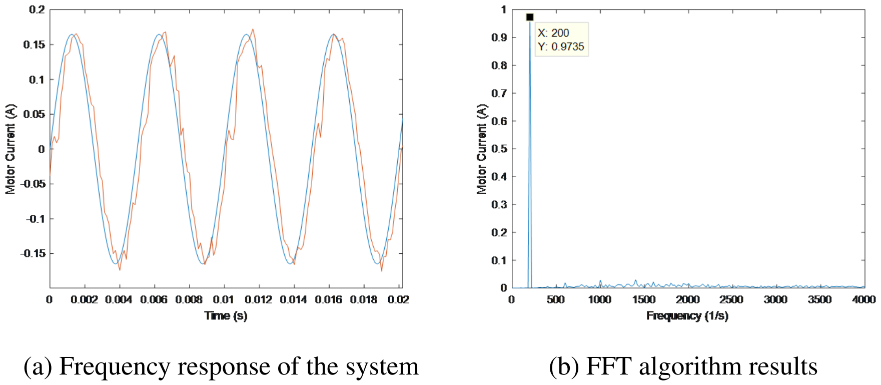


Figure 6.15: Bode plots of the closed-loop position control system.

are measured. These measurements are converted from time domain to frequency domain using fast Fourier transform (FFT) algorithm. Results obtained from frequency tests and their frequencies are given in Figs. 6.16-6.20. Note that FFT results of current measurements are scaled to FFT of commanded current waves so that their corresponding frequency plots are scaled from 0 to 1 to better show the actual gains.

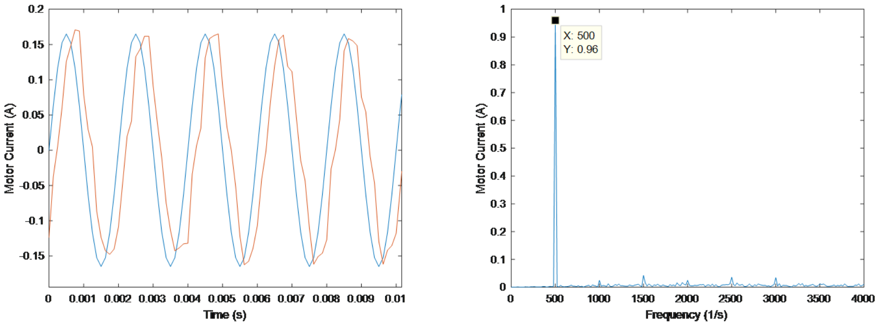


(a) Frequency response of the system

(b) FFT algorithm results

Figure 6.16: Response of the current control system at 200 [Hz].

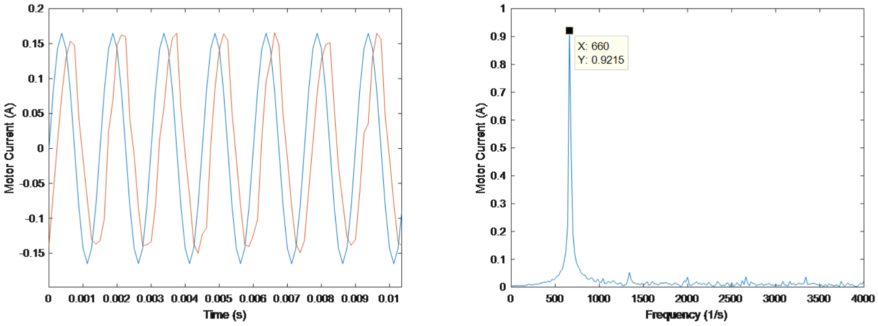
Gains at measured frequencies are compared to bode plots of mathematical model that can be seen in Fig. 6.10b. The comparison can be seen on Table 6.5. The measured gains and the model gains are not very close. Even though cut-off frequency of the current loop is shown to be higher than 1000 $[Hz]$, it is not exactly 2000 $[Hz]$.



(a) Frequency response of the system

(b) FFT algorithm results

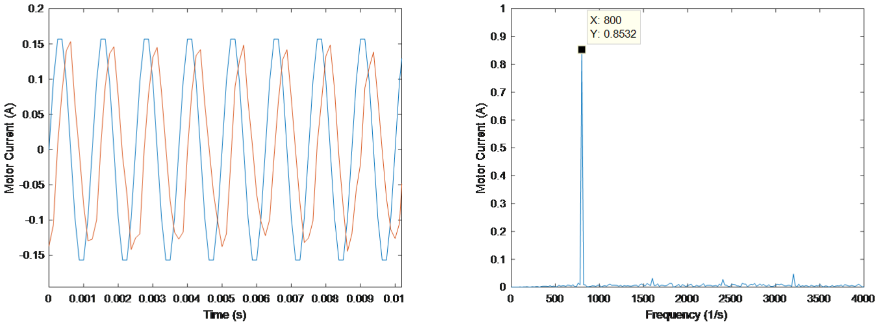
Figure 6.17: Response of the current control system at 500 [Hz].



(a) Frequency response of the system

(b) FFT algorithm results

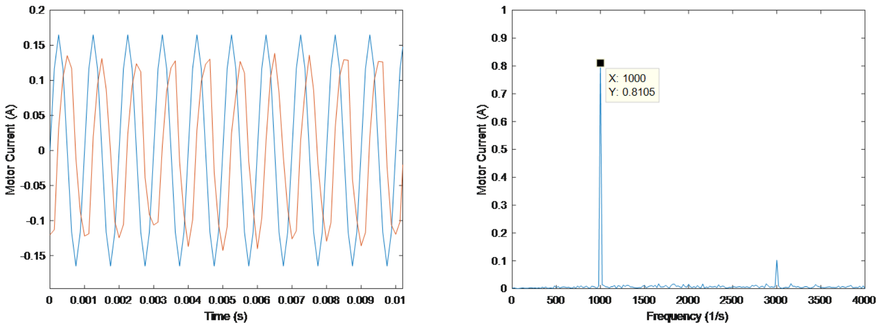
Figure 6.18: Response of the current control system at 660 [Hz].



(a) Frequency response of the system

(b) FFT algorithm results

Figure 6.19: Response of the current control system at 800 [Hz].



(a) Frequency response of the system

(b) FFT algorithm results

Figure 6.20: Response of the current control system at 1000 [Hz].

Table 6.5: Comparison of frequency responses.

Frequency	Gain at mathematical model	Measured gain
200hz	0.9954 (-0.04dB)	0.9735 (-0.23dB)
500hz	0.9757 (-0.21dB)	0.9600 (-0.35dB)
660hz	0.9584 (-0.37dB)	0.9215 (-0.71dB)
800hz	0.9421 (-0.52dB)	0.8532 (-1.37dB)
1000hz	0.9151 (-0.77dB)	0.8105 (-1.82dB)

This is most probably caused by imperfect measurement of the resistor and inductance parameter values used in current loop measurements. Frequency response tests done on outer loops of the system failed to give results as expected. The mechanism oscillated aggressively without a specific frequency. The reason of this behavior is shown by building a full Simulink model of the motion control system with its nonlinearities such as current limits and quantization intervals that is not considered in controller design with root-locus method. The tests on the mathematical model built on Simulink and the real system showed that the nonlinearities caused by significantly small quantization interval of velocity readings obtained from differentiation of encoder position signals cause huge spikes in the current command. The system almost behaves like an on-off controller where velocity loop either demands full scale current in either direction because of the current command limit of $I^*(z)$. The minimum speed of the motor that can be measured can be calculated as:

$$\frac{2\pi N}{T_s c_{enc}} = \frac{2\pi \times 23}{8000^{-1}20000} \cong 57.8[rad/s] \cong 552\ [rpm] \tag{6.17}$$

where

c_{enc}: Number of encoder counts in one revolution (count/rev);

T_s: Sampling time [1/s];

N: Gearbox Ratio.

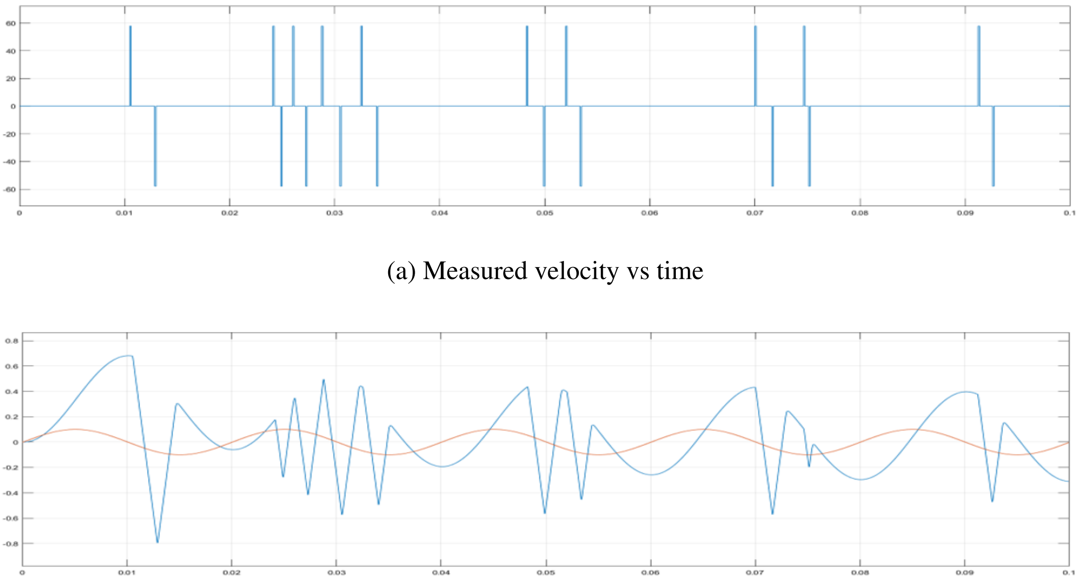
The effect of this huge quantization interval caused by high sampling rate and insufficient number of encoder counts in one revelation can be seen on Simulink model of the control system. Since the gains of PI controller for velocity control is selected with root locus techniques on a continuous system, such a nonlinearity can cause the system to become unstable. A sine wave of amplitude $0.1\ [rad/s]$ and frequency of $50\ [Hz]$ is input to the Simulink model with different encoder quantization intervals. Effects of changing the quantization interval of the encoder can be seen in Figs. 6.21-6.23. In order to reduce this problem caused by quantization interval, a revised and simpler discrete-time control system will be designed.

6.6 Revised Discrete-time Control System

A simpler version of the model explained in Section 6.4 is implemented which does not use velocity loop. Instead, in this control model, position controller directly sends torque commands to the current loop, which is known to operate in desired condition as shown in Section 6.5.4. The block diagram of this system is shown in Fig. 6.25. As can be seen, this discrete-time cascade control system consists of two nested loops:

- Innermost PI-based current control loop;
- PID-based outer position control loop.

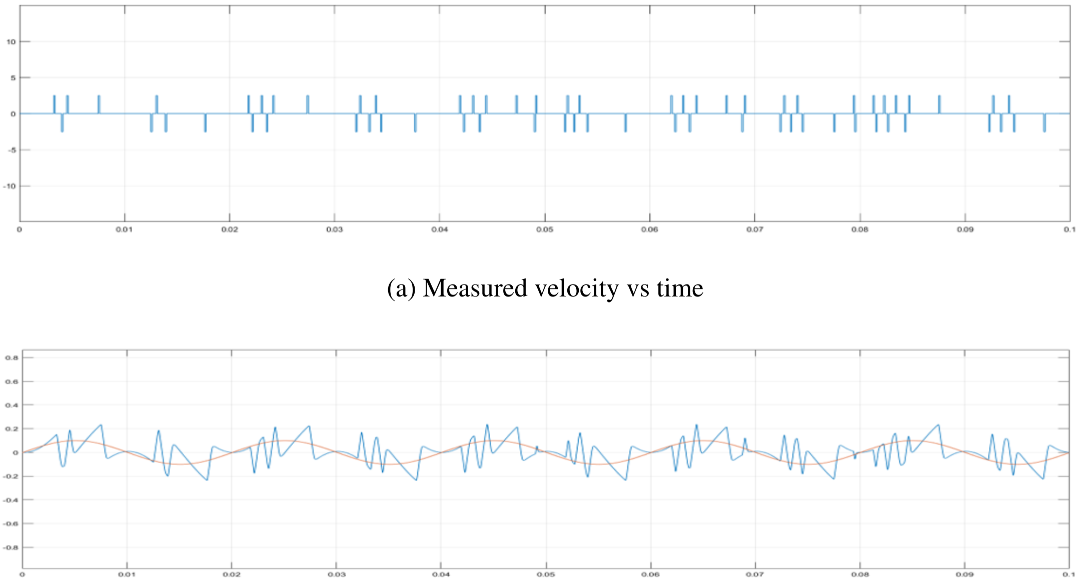
Note that the effect of D gain in position loop and P gain in velocity loop are directly the same as it can be seen in Fig. 6.6. Hence, tuning D to a high value might result in



(a) Measured velocity vs time

(b) Commanded velocity and actual velocity vs time

Figure 6.21: Velocity loop performance when min. speed measurement is 552 [rpm]

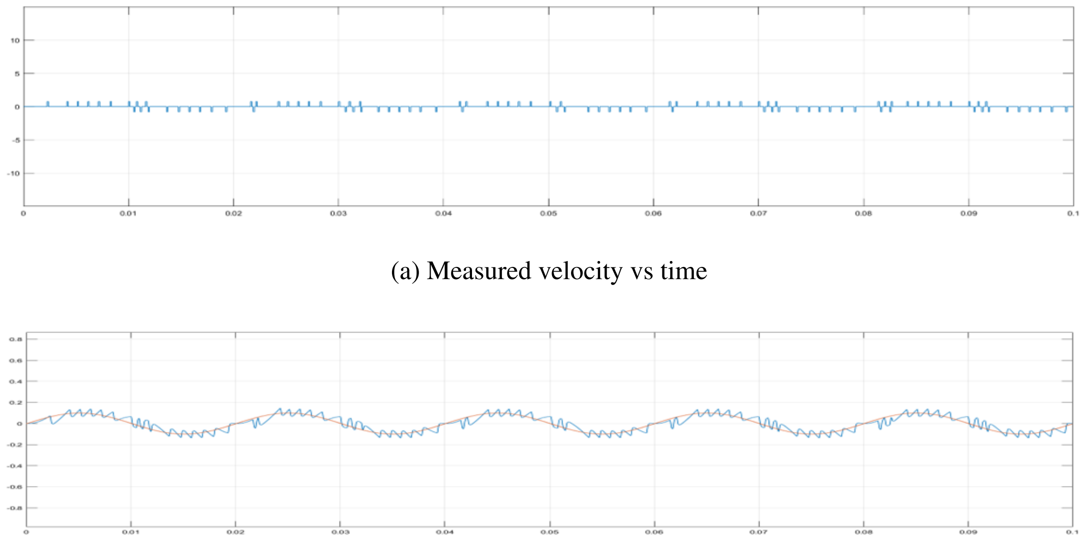


(a) Measured velocity vs time

(b) Commanded velocity and actual velocity vs time

Figure 6.22: Velocity loop performance when min. speed measurement is 48 [rpm]

unsatisfactory performance just as in earlier model. The back EMF compensation is neglected. It was already noted in Subsection 6.5.1 that effect of EMF compensation



(a) Measured velocity vs time

(b) Commanded velocity and actual velocity vs time

Figure 6.23: Velocity loop performance when min. speed measurement is 15 [rpm]

is minimal since the motor used in the system is very small with respect to conventional industrial servo motors. The block diagram of the generic discrete-time control system is illustrated in Fig. 6.24. Hence, the transfer functions are defined as shown in Eqns. (6.18).

$$\frac{I(z)}{V_a(z)} = (1 - z^{-1})Z\left\{\frac{J_e s + b_e}{s((J_e s + b_e)(L_s + R) + K_t(K_e - \hat{K}_e))}\right\} \tag{6.18a}$$

$$\frac{\theta(z)}{V_a(z)} = (1 - z^{-1})Z\left\{\frac{K_t}{s^2((J_e s + b_e)(L_s + R) + K_t(K_e - \hat{K}_e))}\right\} \tag{6.18b}$$

where $Z.$ refers to the z-transform of a given complex (Laplace) function. By definition, the transfer functions shown in Fig. 6.24 takes the form which can be shown as Eqns. (6.19).

$$G_1(z) \triangleq \frac{I(z)}{V_a(z)} \tag{6.19a}$$

$$G_2(z)G_3(z) \triangleq \frac{\theta(z)}{I(z)} = \frac{\theta(z)}{V_a(z)}\frac{V_a(z)}{I(z)} = \frac{\theta(z)}{V_a(z)}G_1^{-1}(z) \tag{6.19b}$$

Noting that current loop is the same as in Subsection 6.5.1, its tuning is not repeated. One can directly move the tuning of position loop.

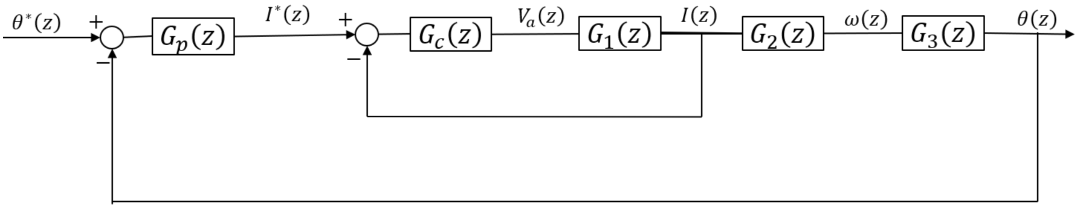


Figure 6.24: Overall discrete-time control system.

6.6.1 Position Loop

The relevant (discrete-time) transfer functions (as shown in Fig. 6.24) can be obtained. Hence, Eqn. (6.19b) is written as Eqns. (6.20).

$$G_2(z)G_3(z) = \frac{1.019 \times 10^{-6}(z + 1.9132)(z + 0.0788)(z - 0.0177)}{(z - 0.0177)(z - 0.9998)(z - 1)} \tag{6.20}$$

Hence, the transfer function of the plant can be expressed as Eqn. (6.21).

$$\frac{\theta(z)}{I^*(z)} = \frac{8.06910^{-7}(z + 1.9132)(z + 0.0788)(z - 0.0177)(z - 0.0177)}{(z - 0.0177)(z - 0.0177)(z - 0.2079)(z - 0.9998)(z - 1)} \tag{6.21}$$

Consequently, the discrete-time PID controller (i.e. G_p) can be employed with three free variables as shown in Eqn. (6.22).

$$G_c(z) = K\frac{(z - p_1)(z - p_2)}{(z - 1)z} \tag{6.22}$$

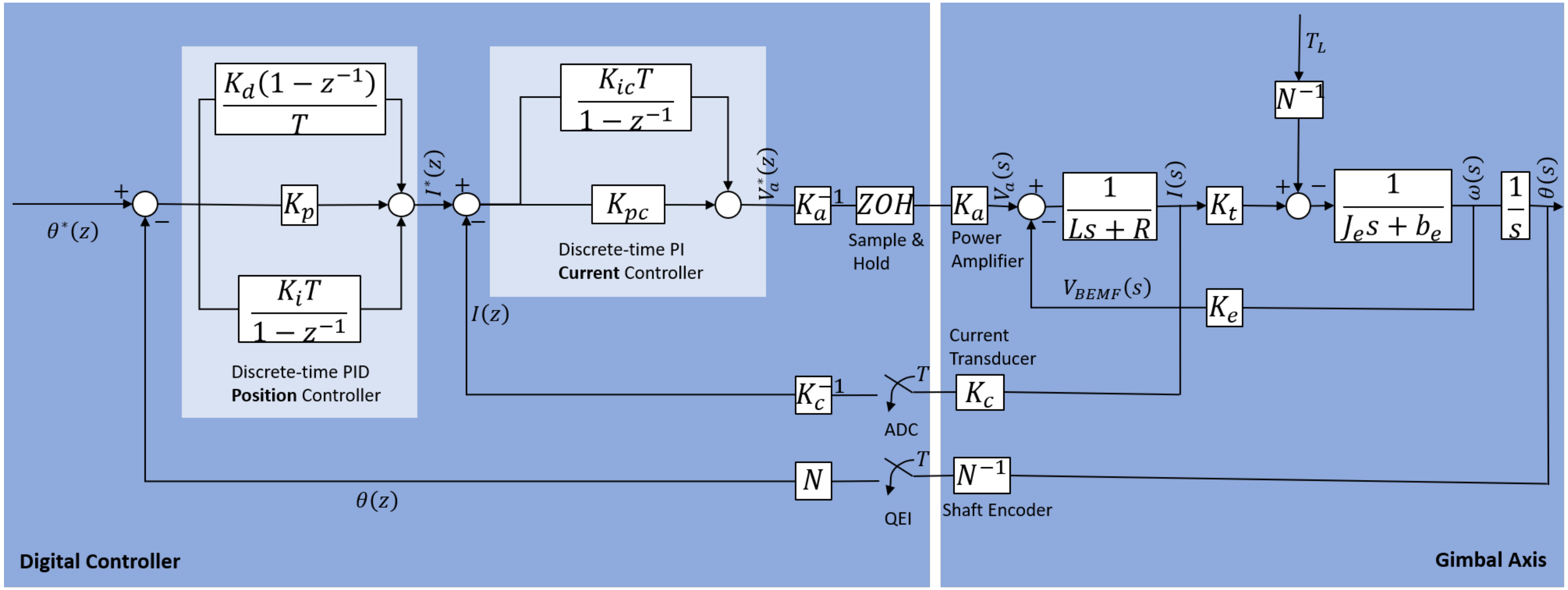


Figure 6.25: Block diagram of the revised concept of motion control system.

Again, the dominant poles are placed to the desired locations ($p_1 = 0.998, p_2 = 1$) in the z-plane using root locus techniques as shown in Fig. 6.26. In this case, the dominant poles are placed such that the cut-off frequency of the closed-loop is roughly set to $5\ [Hz]$. Higher frequencies require higher K_{pd} gains which is not desirable. Fig. 6.27a illustrates the Bode plots and 6.27b illustrates the step response of the closed-loop position control system.

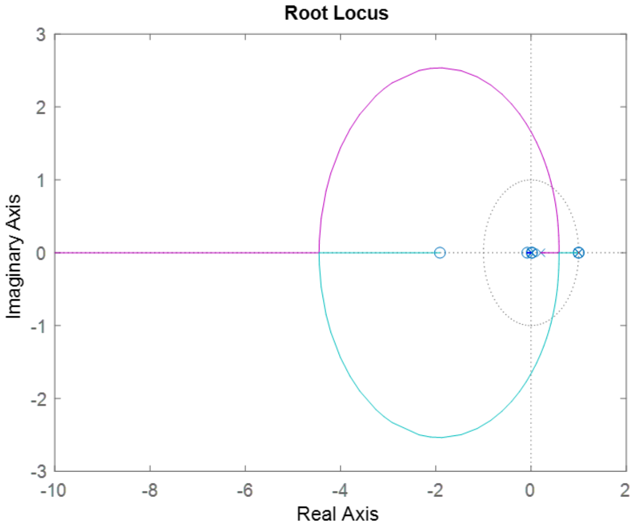
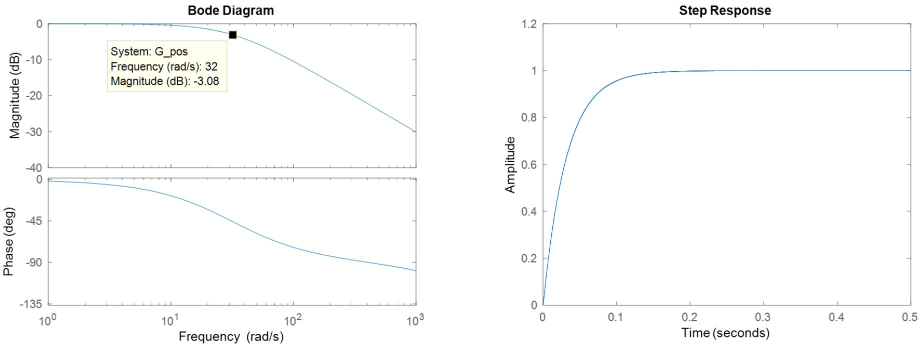


Figure 6.26: Root locus of position open-loop

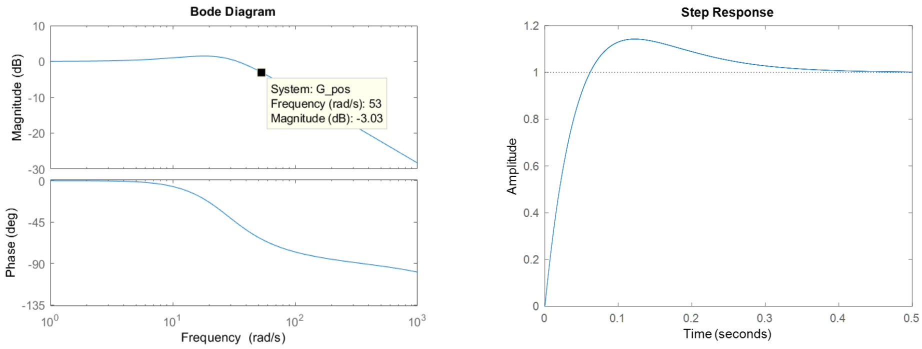


(a) Bode plots of position control loop.

(b) Step Response of position control loop.

Figure 6.27: Root locus of the closed-loop position control system.

The (tuned) P,I and D parameters are found as $K_p = 0.2398, K_i \cong 0, K_d = 0.0150$. In an attempt to have faster response characteristics and higher cut-off frequency, the pole locations are shifted slightly to the positions ($p_1 = 0.997, p_2 = 0.999$). This newly tuned systems P,I and D parameters are $K_p = 0.5696, K_i = 3.6045, K_d = 0.0182$. Note that, K_d value should not be increased much as it would cause undesired behaviour as explained in Section 6.6 . Fig. 6.28a illustrates the Bode plots and 6.28b illustrates the step response of the closed-loop position control system.



(a) Bode plots of position control loop.

(b) Step Response of position control loop.

Figure 6.28: Root locus of the closed-loop position control system.

6.7 Motor Controller Software Implementation

Motion controller software starts with the initializations of the modules it uses: General Purpose Input Output (GPIO) module, Floating Point Unit (FPU) module, Quadrature encoder Interface (QEI) module, Pulse Width Modulation (PWM) Module, Analog to Digital Converter (ADC) module, Universal Serial Bus (USB) module as well as timers are initialized and prepared for use. When initializations end, motion controller enters its control loop. Control loop contains a cascaded loop structure. Detailed information about this control loop and calculations inside are explained in Section 6.5. Motion controller always stays inside the control loop unless USB communication is needed. Control loop cycles with a predefined frequency. It should be noted that, the number of clock cycles (time) that passes while inside receiving or sending data through USB interface is measured to be negligible and does not affect the control loop performance. Motion controller sends encoder information to system controller with a predefined frequency. If a command is received from system controller, the command is checked for its meaning. If it is a command sent for the control loop; the data received is forwarded to the related parameter inside control loop.

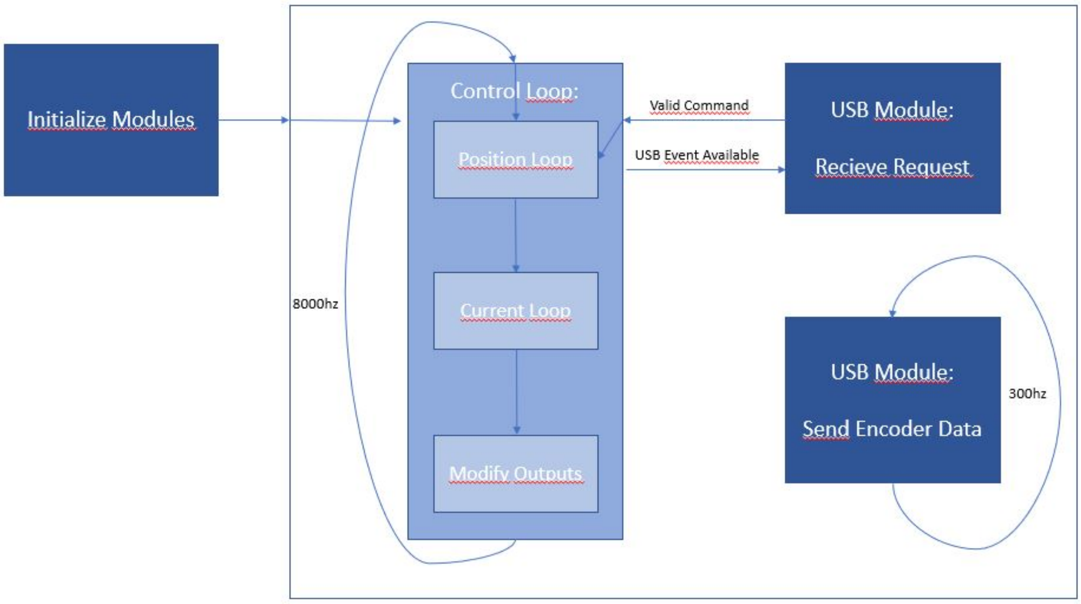


Figure 6.29: Motion controller simplified architecture

The control loop that is designed in Section 6.6 and seen in Fig. 6.25 is implemented inside the code. If one looks at the Fig. 6.25, there are gains at sensor inputs and voltage outputs of the system. These gains can be calculated as shown in Eqns. (6.23)-(6.25).

$$K_a = \frac{T_{fullpwm}}{V_{bus}} = \frac{1000}{12} \cong 83.333 \tag{6.23}$$

where

$T_{fullpwm}$: Number of cycles in a full pwm cycle. [1/80M s];

V_{bus}: Maximum voltage (bus voltage) of power amplifier [V].

Similarly,

$$K_c^{-1} = \frac{V_{adcrange}}{N_{adc}} \times K_{sens} = \frac{3.3}{4096} \times 0.8 \cong 642.1312 \times 10^{-6} \tag{6.24}$$

where

$V_{adcrange}$: Voltage range of the ADC of microprocessor [V];

N_{adc}: Quantization interval (resolution) of ADC of microprocessor;

K_{sens}: Current sensing of special instrumental amplifier (section 6.2) [A/V].

Similarly,

$$N^{-1} = \frac{2 \times \pi}{N_{encrot}} \times N_{gearbox} = \frac{2 \times \pi}{20000} \times 23 \cong 7.226 \times 10^{-3} \tag{6.25}$$

where

N_{encrot}: Number of pulses in one revolution of encoder disc;

$N_{gearbox}$: Gearbox Ratio.

Note that in Subsection 6.5.1 , gains for the controller is found as, $K_{pc} = 0.14301$; $K_{icT} = 7.9211$, which results in following PI controller:

$$G_{cc}(z) = \frac{8.064z - 0.143}{z - 1} \tag{6.26}$$

Similarly, in Subsection 6.6.1 , gains for the controller is found as, $K_p = 0.5696$, $K_i = 3.6045$, $K_d = 0.0182$, which results in following PID controller:

$$G_{cp}(z) = \frac{146.296z^2 - 292.032z + 145.737}{z^2 - z} \tag{6.27}$$

The calculations implemented on software that controls the system according to calculations made in this chapter can be found on Appendix B.

6.8 Closure

In this chapter, the motion controller of the parallel tracking platform is designed. Firstly, architecture of the motion controller is described in depth in Section 6.2. Next, electromechanical model of plant is constructed in Section 6.3. A discrete controller is built to control this model in Section 6.4 and 6.5 . However, tests conducted on the system showed that the results are not as desired from the real system in Section 6.5.4. This unexpected short-coming is caused by non-sufficent quantization interval of encoder readings on motor shaft. Due to this, another discrete controller is built in Section 6.6. Lastly, information is given about the implementation of this discrete controller in Section 6.6.1.

CHAPTER 7

VISUAL SERVO DESIGN AND IMPLEMENTATION

7.1 Introduction

Visual servo control system is explained in detail in this section. In Section 7.1.1, both image processing method and trajectory prediction algorithm are explained. Coupled visual servo controller utilizes image processing and trajectory prediction to track the target in Section 7.2. In order to improove the performance of the visual servo controller, forward and inverse look up tables are employed in Section 7.3.

7.1.1 Parallel Targetting Mechanism Visual Servo Control System

In the Visual Servo Controller; firstly, target position θ^*, ψ^* is obtained by the camera attached to the plant of the parallel tracking platform which is rotated in X axis by θ and Y axis by ψ. Camera then generates an image which is sent to the visual servo controller with a prespecified sampling rate (300 [Hz] in this case) through USB3.0 bus. In addition, encoder signals $\beta_{A'}$ and $\beta_{A''}$ are sent back to the visual servo controller by the motion controller with a sampling rate of 300 $[Hz]$ through an USB2.0 interface. When the desired commands are generated by visual servo control algorithms, they are also sent to the motion controllers with the rate of 300 $[Hz]$. Target position is obtained by firstly, thresholding the image to desired target color. Each pixel is asigned by a one or zero depending on thresholding conditions. If treshold conditions are satisfied a one is asigned to the pixel, otherwise a zero is asigned. Secondly, zeroth and first order moments of this image is taken. Zeroth image moment adds all pixels in thersholded image to get the number of pixels above the threshold. Zeroth and first order moments can be shown as:

$$\mu_{0,0} = \sum_{x=o}^{w}\sum_{x=o}^{w} f(x,y) \tag{7.1}$$

$$\mu_{1,0} = \sum_{x=o}^{w}\sum_{x=o}^{w} xf(x,y) \tag{7.2}$$

$$\mu_{0,1} = \sum_{x=o}^{w}\sum_{x=o}^{w} yf(x,y) \tag{7.3}$$

Hence, center of the target position can be found by:

$$\Delta X = \frac{\mu_{1,0}}{\mu_{0,0}} \tag{7.4}$$

$$\Delta Y = \frac{\mu_{0,1}}{\mu_{0,0}} \tag{7.5}$$

Field of view of the camera with lens is known to be 8 in 1600 pixels. An approximation is made such that there is minimal circular lens distortion on image and one pixel of displacement is around 0.01 radial displacement. ΔX and ΔY values can be turned into rotation errors $\Delta\theta^*$ and $\Delta\psi^*$ by using this ratio. Visual servo controllers that will be explained use this simple image processing method to obtain target positions. Using $\Delta\theta^*, \Delta\psi^*, \beta_{A'} and \beta_{A''}$ two different visual servo controllers are designed in sections 7.2 and 7.3. In both of these visual servo controllers, target trajectory prediction is required. For this purpose, third order polynomals are used to fit last four target positions to the curves shown in 7.6.

$$x = a_1t^3 + b_1t^2 + c_1t + d_1 \tag{7.6a}$$

$$y = a_2t^3 + b_2t^2 + c_2t + d_2 \tag{7.6b}$$

Least square method is used to fit these curves to the four points defined as (x_{k-3}, t_{k-3}), (x_{k-2}, t_{k-2}), (x_{k-1}, t_{k-1}),(x_k, t_k) to obtain satisfactory constants for a_1,b_1 ,c_1 ,d_1 ,a_2, b_2 ,c_2 , d_2. With all variables at hand, one can predict the next position of target by calculating the equations shown in 7.7.

$$x_{k+1} = a_1t_{k+1}^3 + b_1t_{k+1}^2 + c_1t_{k+1} + d_1 \tag{7.7a}$$

$$y_{k+1} = a_2t_{k+1}^3 + b_2t_{k+1}^2 + c_2t_{k+1} + d_2 \tag{7.7b}$$

7.2 Coupled Visual Servo Control System

A controller is built with the assumption that parallel targetting mechanism will turn in linear relationship with motor rotations. Note that while this is a small assumption while tracking a target around vertical position of parallel targetting mechanism, in other positions this is a huge assumption which should affect the performance considerably. In vertical position of parallel targetting mechanism, a rotation around $\beta_{A''}$ affects θ angle while a rotation around $\beta_{A'}$ affects ψ angle. Hence, rotation errors and motor angles are paired as $\Delta\theta^*$, $\beta_{A''}$ and $\Delta\psi^*$, $\beta_{A'}$ and added with eachother to obtain the target position for motors. These target methods are fed to the curve fitting and trajectory prediction algorithm which is explained in section 7.1.1. Output of prediction algorithm is directly sent to the motion controller. Architecture of this control method can be seen in Fig. 7.1.

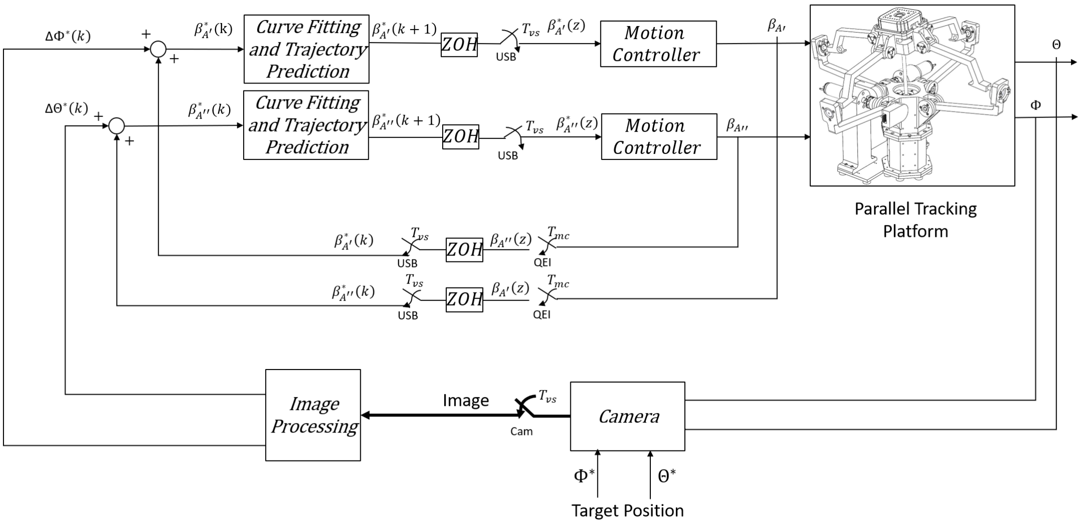


Figure 7.1: Coupled visual servo controller ($T_{mc} = 1/8000, T_{vs} = 1/300$)

7.3 Decoupled Visual Servo Control System

Another controller is built without the assumption that parallel targetting mechanism will turn in linear relationship with motor rotations. For this purpose forward- and inverse kinematic maps are inserted into the system which are used as loop-up tables. Forward kinematic look up table (LUT), maps the range of possible parallel tracking platform motor positions with equal spacing to their corresponding plant (i.e. camera) rotation angles θ and ψ. It is used to obtain the cameras rotation by using motor angles $\beta_{A''}$ and $\beta_{A'}$ as an input to the forward look up table. Not every point is available in look-up table. Hence, bilinear interpolation is used to obtain rotation angles θ and ψ corresponding to motor angles $\beta_{A''}$ and $\beta_{A'}$ using the look-up table at any point. Calculated rotation angles θ and ψ are added to rotation errors obtained from camera $\Delta\theta^*$ and $\Delta\psi^*$ to obtain calculated global position of the target. Global position of the target is used in trajectory prediction to obtain the target position in a future time (i.e. next time step in this case) as explained in section 7.1.1. Similarly to what is done in forward LUT, inverse LUT is used to turn the global position of the target in next time step to the corresponding motor angles $\beta_{A''}$ and $\beta_{A'}$. The architecture of this control method can be seen in Fig. 7.2.

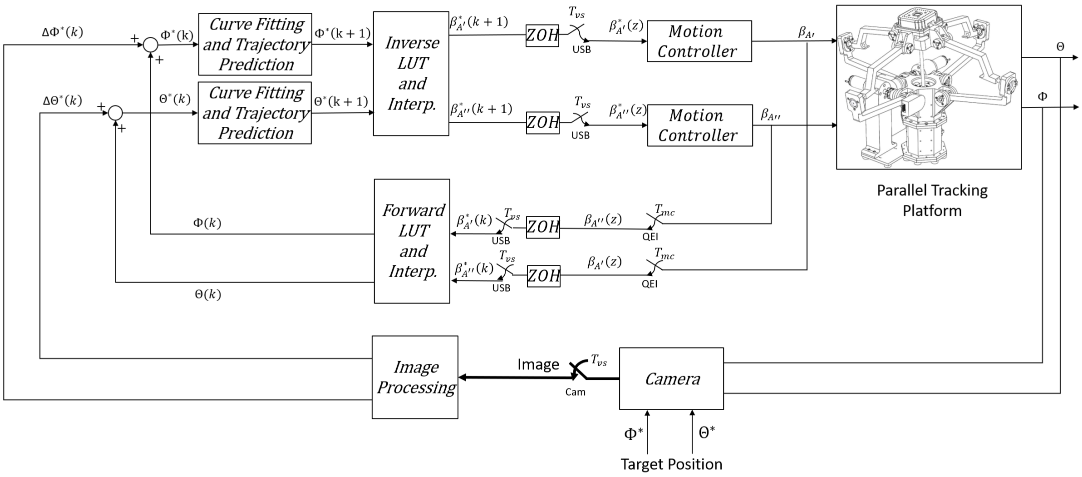


Figure 7.2: Decoupled visual servo controller ($T_{mc} = 1/8000, T_{vs} = 1/300$)

7.4 Visual Servo Control System Implementation

Visual Servo Controller software starts with the initializations of the libraries it uses. All the discrete maps obtained in 4 are imported inside. After initializations are done, three threads are created: One for visual servoing, one for sending data, one for receiving data. Visual servoing thread loop frequency is synced with the camera FPS. In this loop; image matrix is obtained from the camera, target is found in the image, target position is sent to visual servoing loop as explained in section 7.1.1.

Output of visual servo controller are sent to motion controllers. It is desired to not use commands that block and poll for USB communication to not disturb visual servo thread and command frequency. For this reason, separate threads are created for USB communication so that blocking calls and interrupts are absent from visual servo thread. In coding, multiple libraries are used.

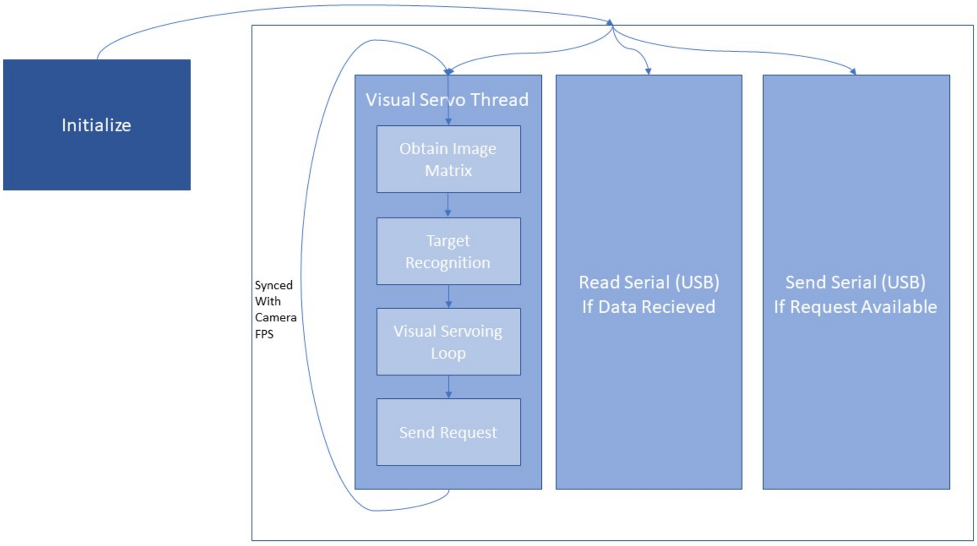


Figure 7.3: Visual Servo controller simplified architecture

- For visualization of the target and images to the user, OpenCV libraries are used.
- For multithreading, Boost.thread library is utilized.
- For camera configuration, Basler pylon SDK is employed.

Simplified architecture of the visual controller software can be seen in Fig. 7.3

7.4.1 Camera Configuration

The camera that is used allows for 60 FPS operation in maximum 1600×1200 (12-bit color) region of interest. This 60 FPS image acquisition operation can be changed both ways depending on the most dominant of the following bottlenecks/characteristics of the camera [5]:

- Maximum USB3.0 data transfer rate that is allowed by the combination of host PC USB3.0 driver hardware and firmware.
- Data packet size (Height of the region of interest of image times Color information available in the image) .
- Predefined exposure rate of the camera sensor.

Using this information, camera frame rate can be improved by reducing the region of the interest to 280×210 and color information quality to 8bits resolution. Exposure rate is used as the bottleneck for the image acquisition and fixed to 3300 $[us]$. This allows the camera to operate at 300 FPS when desired. Note that visual servoing loop is synchronized with respect to the camera FPS. This allows that with the reduced ROI, visual servoing loop can be used with much higher bandwidth which is preferred for the control of the system. Note that the camera configuration can be changed in air. This feature is put to use as explained in Subsection 7.4.2.

7.4.2 Target Search State Machine

It is expected from the PTP to find a target without any outside help after giving the system the command to search for track a target. For this reason, 3 different states are built into the implementation. The software switches between the states depending on some conditions while searching and tracking for a target. These three states can be explained as:

- If no target is found inside the 1600×1200 camera image; send position commands to motion controller that represent an expanding and contracting Archimedes spiral until the target is found.
- If target is found but the target is not at the central area of the image; run visual servo control to track the target with 1600×1200 camera image and 60 $[Hz]$ sampling time until target is lost or target is at the centeral area of the image.
- If target is found and the target is at the central area of the image; run visual servo control to track the target with 280×210 camera image and 300 $[Hz]$ sampling time until target is lost.

7.5 Closure

In this chapter visual servo control system is explained in detail. The visual servo controller recieves data from camera and motion controller to track the target. Image processing and trajectory prediction algorithms are utilized to track the target. In order to improove the performance of tracking, forward and inverse look-up tables are employed. Furthermore, target search state machine implementation is also explained.

CHAPTER 8

EXPERIMENTAL RESULTS

8.1 Introduction

In order to test the performance of the system and the effect of implemented decoupling target tracking experiments are conducted. In this chapter firstly, test setup will be explained in Section 8.2. Secondly, target paths and configurations will be explained in Section 8.3. Using the test setup and target configurations explained in these sections, several tests are conducted and their results will be given in Section 8.4. Lastly, conclusions from these test results will be made.

8.2 Test Setup

It is explained in Chapter 6 that the parallel targeting mechanism can be controlled by connecting encoders and motors to the motion controller. It is also explained in Chapter 7 that the parallel targetting mechanism can be controlled by a PC (i.ei visual servo controller) by connecting to the motion controller. This PC sends appropriate control commands to motion controller using a USB 2.0 interface using a camera attached to the parallel targeting mechanism for visual feedback. The camera attached to the parallel targetting mechanism connects to the visual servo controller PC via a USB3.0 interface. For experiments, a new PC is introduced for target trajectory generating and projecting purposes. This PC is connected to a projector for projecting the target. The projector is directed to ceiling for projecting the target on a flat surface. The ceiling is approximately 3 meters apart from the parallel targeting mechanism. In the tests, parallel targeting mechanism is turned on and it searches around by itself using the search algorithm as explained in Chapter 7 . After finding the target, parallel targetting mechanism locks on the target and tracking errors in terms of pixels and motor positions sent from the motion controller are recorded. These results are given in Section 8.4. A representation of this test setup configuration can be seen in Fig. 8.1 and a photo of actual test setup can be seen in Fig. 8.2.

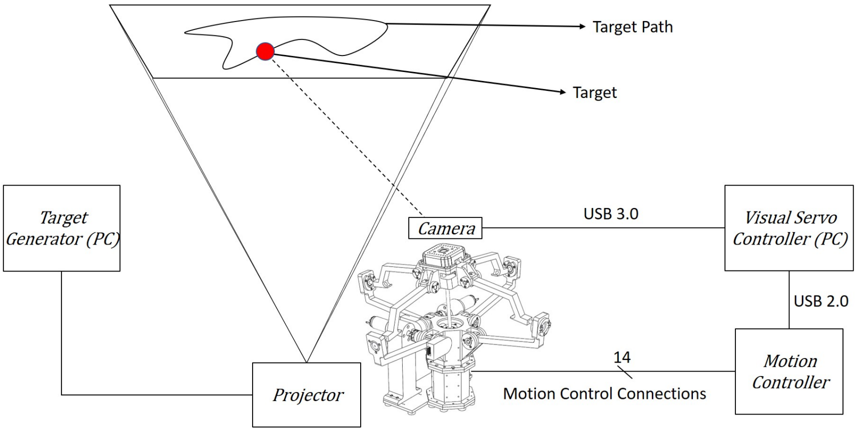


Figure 8.1: Test setup configuration

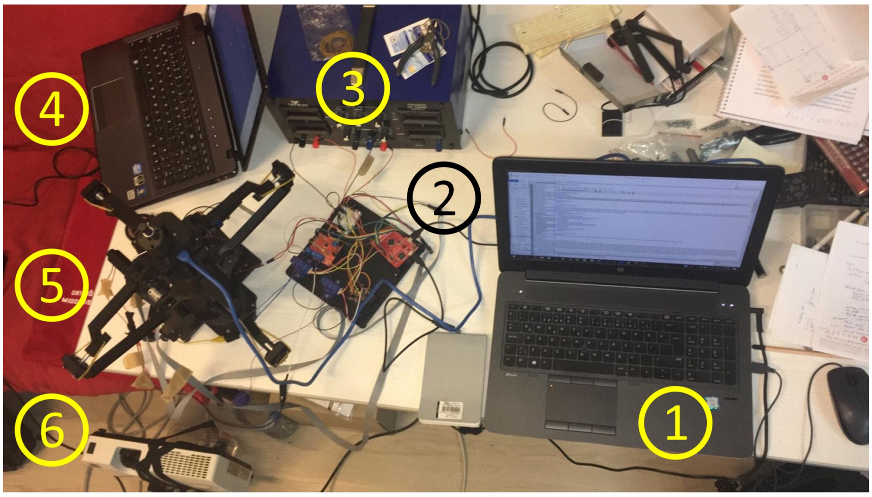
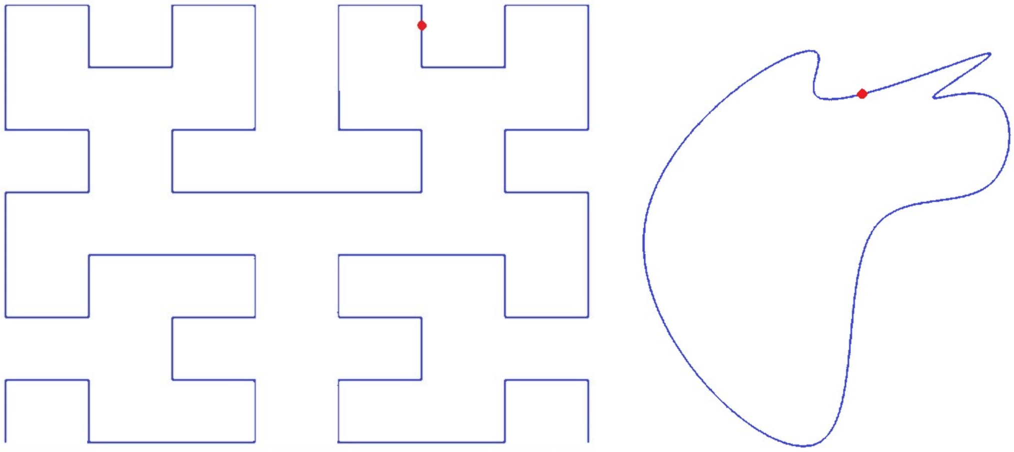


Figure 8.2: Test Setup. 1) Visual Servo Controller (PC) 2) Motion Controller 3) Power supply 4) Target Generator (PC) 5) Parallel Targetting Mechanism 6) Projector

8.3 Target Configuration

A target of 5mm is projected onto a surface 3 meters ahead of the mechanical system. This target is made to move in two different paths to test the implemented algorithms described in chapter 7. The first path includes a portion where the red colored target zig-zags with a moderate speed, and a curved portion where it speeds up. This first target path can be seen in Fig. 8.3b. The second path includes a third degree Hilbert R-tree shape in which the target makes quick turns throughout the motion. This second target path can be seen in Fig. 8.3a. Six different target configuration are used throughout the experiments. First three of the tests (i.e. test case 1-3) are done with the target path in Fig. 8.3a. In these three tests, speeds of the target increase from test one to test three. Similarly, the last three tests (i.e. test case 4-6) are done with the target path in 8.3b and the speeds of the targets increase from test four to six. Speeds of targets are selected so that the fastest cases (case 3 and 6) are almost at the fastest speed the target tracker can track relaibly which is approximately around 1.1m/s at 3m distance (around 0.42 rad/s). This speed increases in straitgh movements and slows down in curves. Similarly, case 2 and case 5 target speeds are approximately around 1m/s at 3m distance. Furthermore,case 1 and case 4 target speeds are approimately around 0.9m/s at 3m distance.



(a) Target paths for experiments

(b) Blob shaped target path with zig-zag

Figure 8.3: Third degree Hibert R-tree shaped target path

8.4 Resuts and Discussion

All of the test cases explained in section 8.3 are conducted for a visual servo control system with- and without decoupling. Results of different tests can be seen in figures 8.4 through 8.15. Inside these figures, four subfigures can be observed. In the top left and top right subfigures, error in x axis and error in y axis measured by the camera can be observed in arcsec. In the left bottom subfigure, absolute error throughout the motion can be seen in arcsec. In the bottom right subfigure, forward transform results of motor positions are plotted to give a representation of target movement in degrees. It should be noted that the results are taken for around 2 minutes for each case. Usually, the target loops throughout the path multiple times in that time span. However, a random single loop is taken to make clean examples in bottom right subfigure. Average, standart deviation, RMS and maximum error values of experimental results are tabulated in Table 8.1. Note that the bottom right figures are represented in terms of rotation around axes. As a result, the paths seen in bottom right figures are skewed with respect to Fig. 8.3.

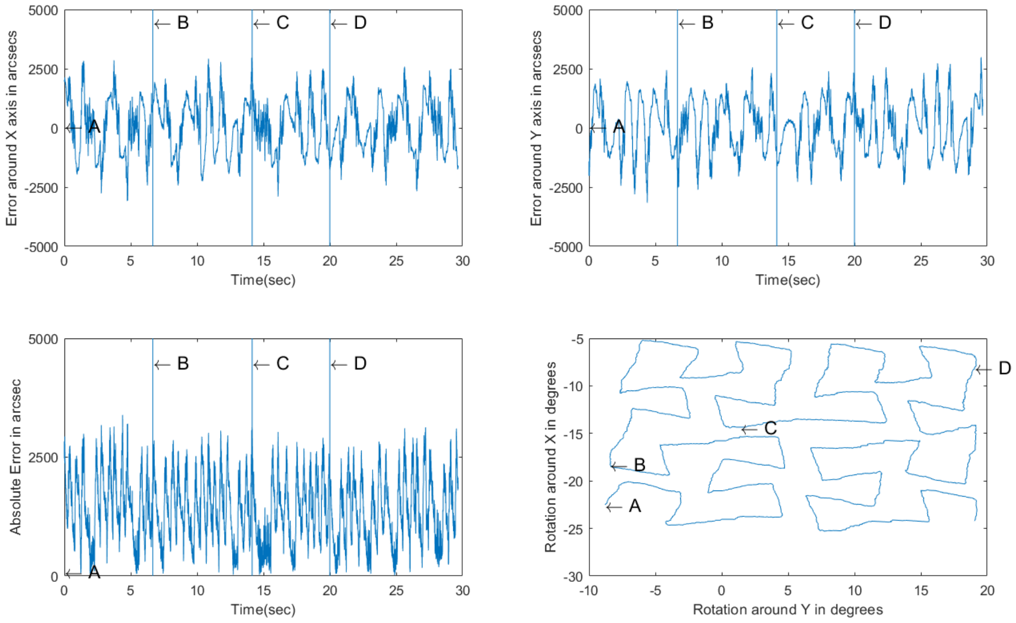


Figure 8.4: Test results coupled visual servo control for case 1

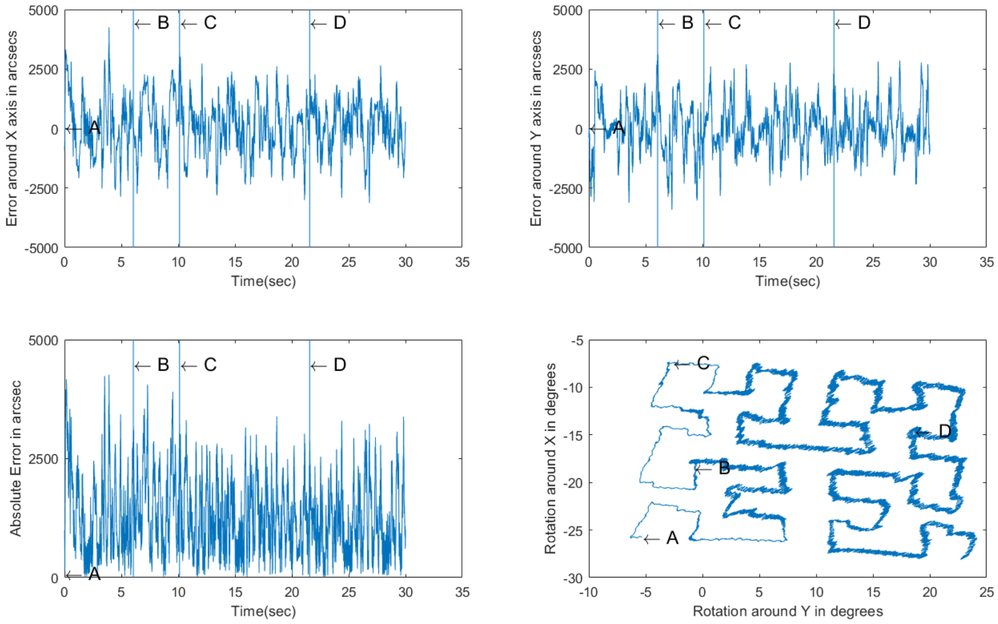


Figure 8.5: Test results decoupled visual servo control for case 1

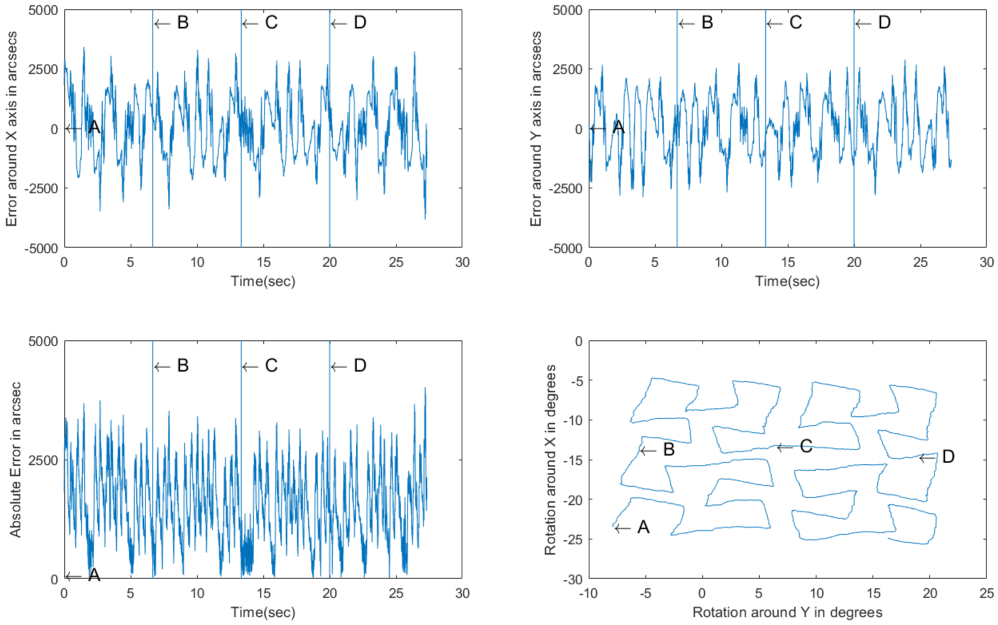


Figure 8.6: Test results coupled visual servo control for case 2

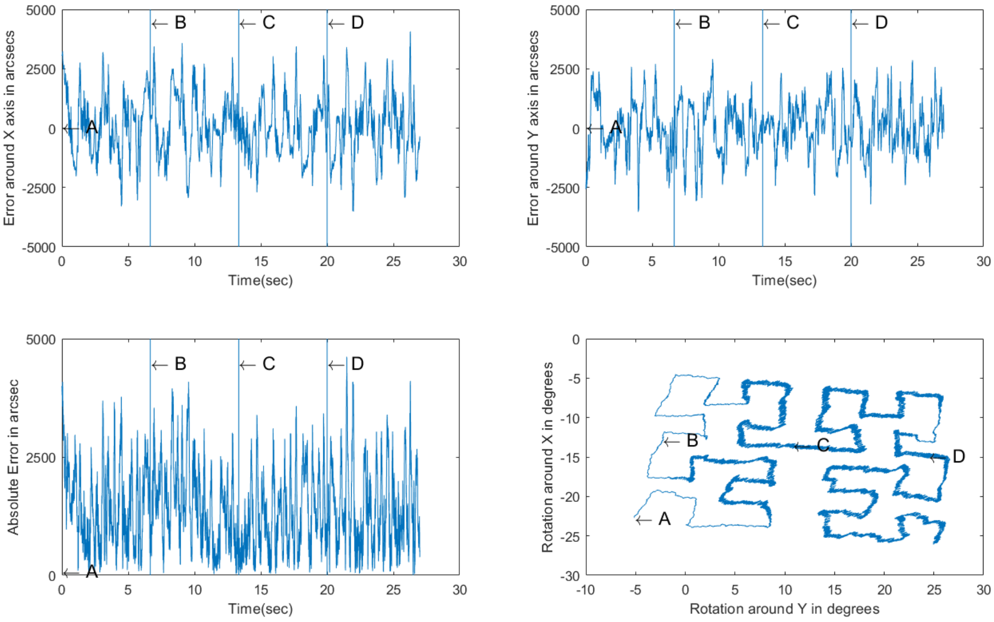


Figure 8.7: Test results decoupled visual servo control for case 2

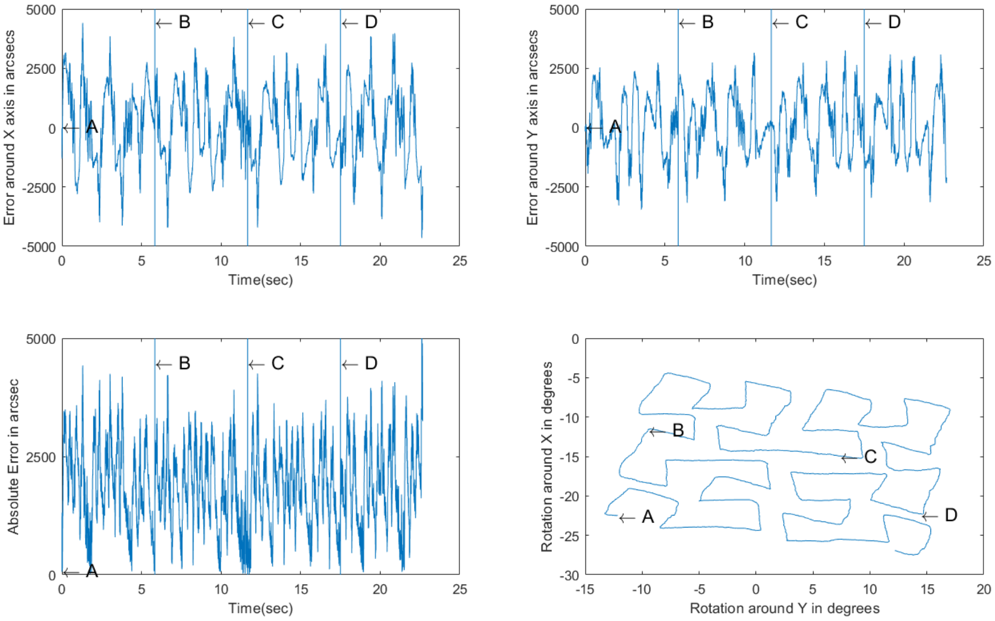


Figure 8.8: Test results coupled visual servo control for case 3

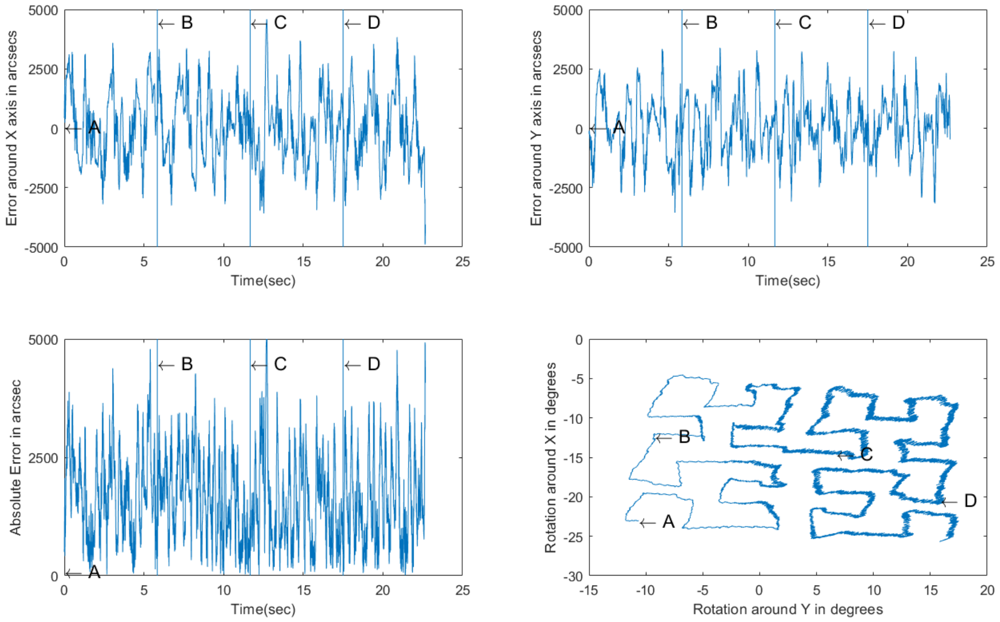


Figure 8.9: Test results decoupled visual servo control for case 3

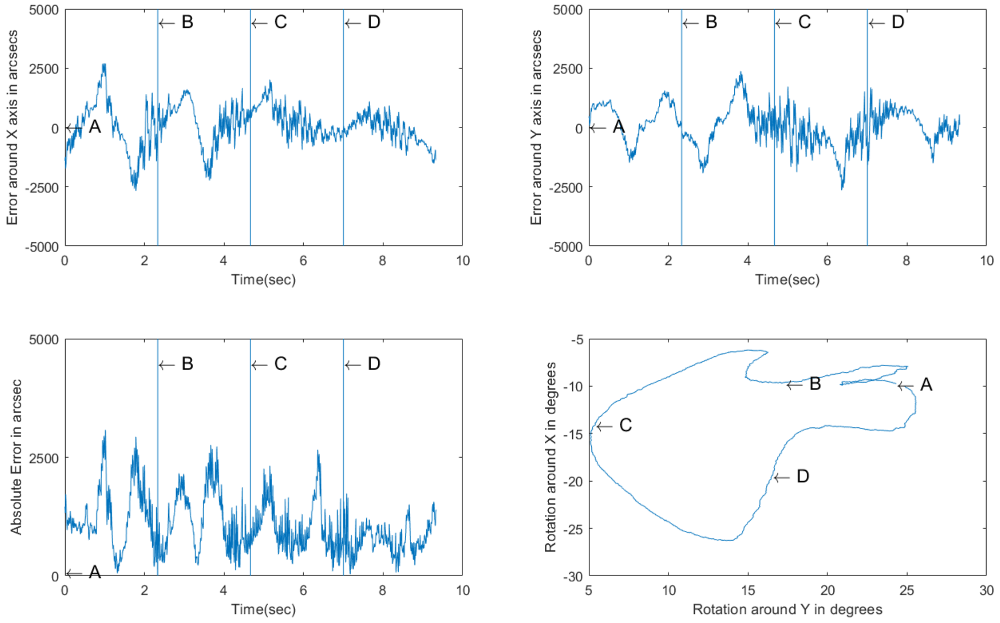


Figure 8.10: Test results coupled visual servo control for case 4

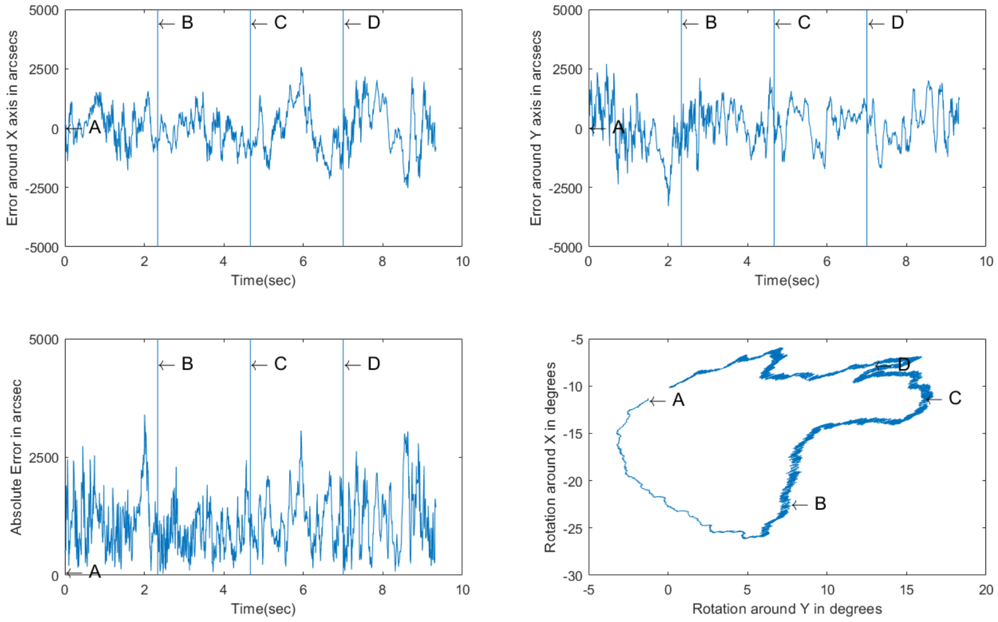


Figure 8.11: Test results decoupled visual servo control for case 4

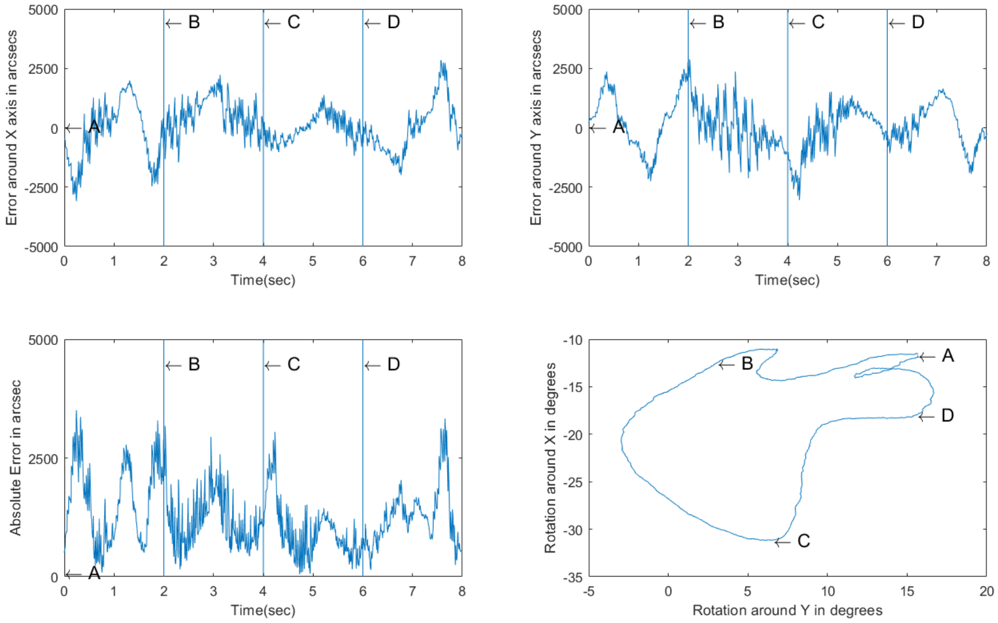


Figure 8.12: Test results coupled visual servo control for case 5

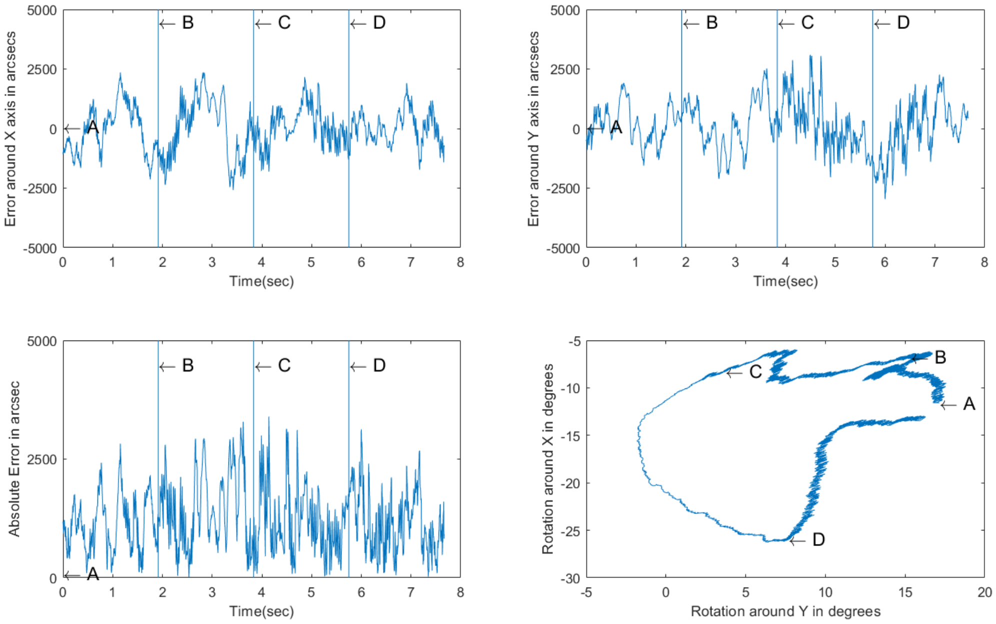


Figure 8.13: Test results decoupled visual servo control for case 5

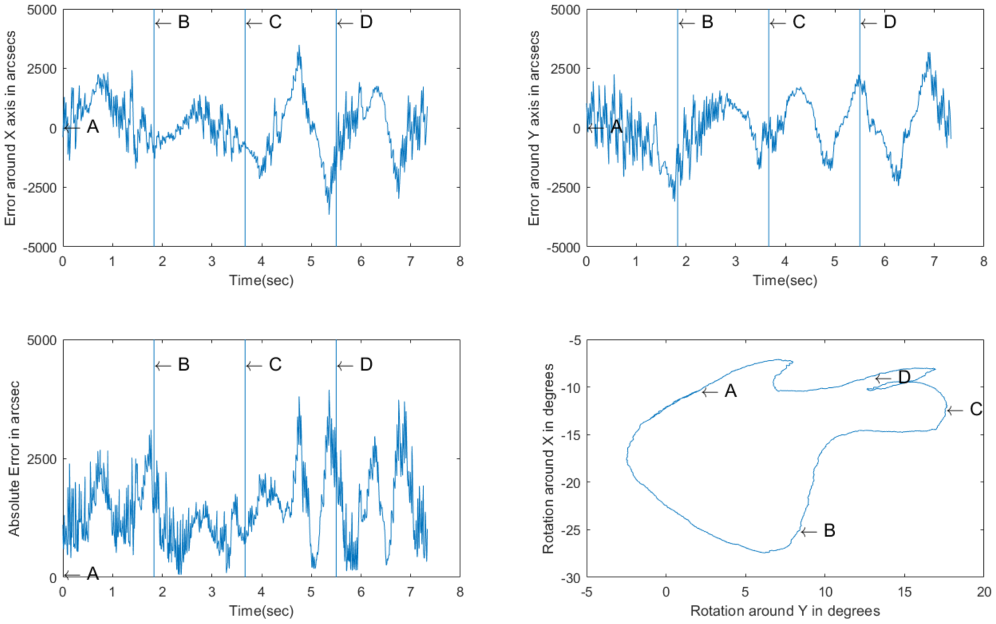


Figure 8.14: Test results coupled visual servo control for case 16

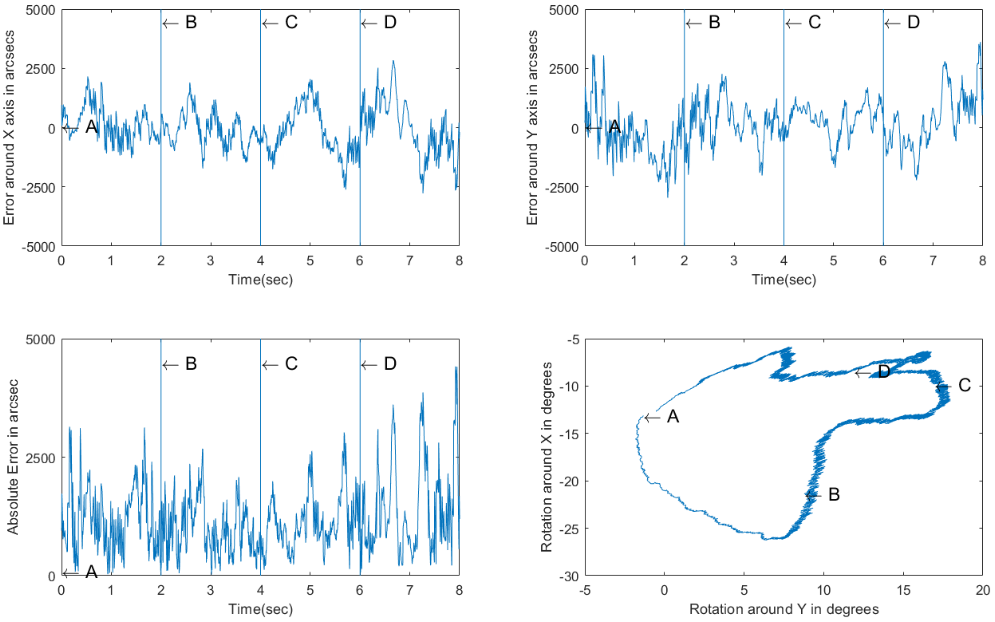
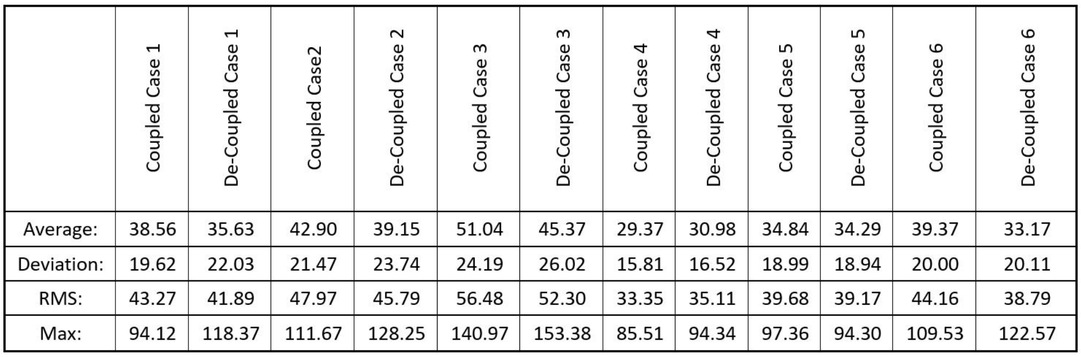


Figure 8.15: Test results decoupled visual servo control for case 1

Table 8.1: Average, standart deviation, RMS and maximum absolute error values of experimental results



	Coupled Case 1	De-Coupled Case 1	Coupled Case2	De-Coupled Case 2	Coupled Case 3	De-Coupled Case 3	Coupled Case 4	De-Coupled Case 4	Coupled Case 5	De-Coupled Case 5	Coupled Case 6	De-Coupled Case 6
Average:	38.56	35.63	42.90	39.15	51.04	45.37	29.37	30.98	34.84	34.29	39.37	33.17
Deviation:	19.62	22.03	21.47	23.74	24.19	26.02	15.81	16.52	18.99	18.94	20.00	20.11
RMS:	43.27	41.89	47.97	45.79	56.48	52.30	33.35	35.11	39.68	39.17	44.16	38.79
Max:	94.12	118.37	111.67	128.25	140.97	153.38	85.51	94.34	97.36	94.30	109.53	122.57

It can be observed that there are oscillations that occur in de-coupled controller test results. They almost always start at sharp turns and edges. While it is expected to have overshoots while target tracking at sharp turns due to third order target trajectory planning, de-coupled controller goes into an oscillatory motion unless the target returns back to a smooth path. This is most probably becouse of the coarse spacing of datapoints in forward- and inverse kinematic maps. While they include 86400 data points each, they are spaced with 0.5 degree intervals at their inputs. This is considerably large for the small movements done while target tracking. Furthermore, bilinear interpolation is used to get values between datapoints in forward- and inverse kinematic maps. However, it is a well-known fact that the behaviour between datapoints is not linear. A higher order interpolation like a cubic interpolation that uses higher number of data points could be used for removing this unsatisfactory performance. Due to these oscillations on decoupled visual controller experiment results, they almost always have higher maximum absolute error with respect to its coupled counterpart. Even though there are oscillations caused by the inaccuracies caused by maps as explained above, de-coupled control experiments almost always show lesser average error and RMS values. Repeatedly, standart deviation of decoupled experiments are worse with respect to their coupled counterparts. This means that de-coupled controller has better tracking performance and center of the oscillations are closer to the center of target with respect to its coupled counterpart. However, oscillations cause the decoupled tracker to overshoot to high error values while coupled controller tracks the target at a farther distance with much higher consistency.

8.5 Closure

In this chapter firstly, test setup are explained in Section 8.2. Secondly, target paths and configurations are explained in Section 8.3. Using the test setup and target configurations explained in these sections, several tests are conducted and their results are given in Section 8.4. As a conclusion, it is seen that while decoupled visual servo controller have better target tracking performance, it tracks with high oscillations which cause a higher standart deviation with respect to a coupled visual servo controller.

CHAPTER 9

CONCLUSIONS AND FUTURE WORK

In this thesis, a unique 2-DOF test platform (parallel tracking platform) is designed and prototyped in which both outputs of the system are a non-linear and coupled function of the input actuators. A camera is attached to the said platform for visual servoing purposes. Aim is to acquire an image of a randomly moving target, capture the target with a vision system that can track it, then give feedback to actuators to control the line of sight and fix the target at the center of the field of view. Multiple steps are followed to achieve this aim. Firstly, kinematic analysis is conducted to obtain numerical relations between inputs and outputs of the parallel tracking platform. Input-output relations are mapped and used in visual servo control algorithms. Dynamical analysis of the system is conducted in Simulink Simscape to better understand the system behaviour. A cascaded discrete motion controller is constructed to control the actuators attached to the test platform. Visual servo control algorithms that decouple the motion of the system are developed. Finally, an experimental setup is set to asses the performance of the parallel tracking platform and visual servo algorithms. In the experiment scenarios, the parallel tracking platform tracks a target that moves on a prespecified path. Following conclusions can be made:

- The selected mechanism is shown to be feasible when it comes to its flexibility on device platform cabling.
- In order to be able to actuate such a mechanism, lower and upper components of the arms should have the exact same length between joints.
- Since sensing device used in tracking is also attached to the device platform, any clearance between actuators and the sensing device may cause undesired behaviour.

- To improve the operation in full hemisphere, forward and inverse kinematic maps need to be utilized. Morover, using a higher resolution map and/or using higher order linear interpolation methods improve tracking performance.

- Motion control performance is highly dependant on quantization intervals of sensing devices/current limits of H-bridges or other non-linearities introduced by components utilized in system.

- In a cascaded control scheme, if one of the control loops is causing undesired response characteristics due to unreliable sensing devices, that control loop can be removed from the cascade. Even though this does not guarantee better performance, it is better than nothing.

- The visual servo controller, the motion controller and parallel tracking platform compose a sufficient test setup for target tracking and future visual servo control studies.

Following studies can be conducted in future:

- A new platform can be built from aluminum alloys for higher precision manufacturing and performance.

- Encoders placed on motor side can be utilized for higher position and velocity resolution for motion control. Encoder on gearbox output shaft can still be used for visual servo control without getting affected by backlash introduced by gearbox.

- Realistic target simulation can be added to experimental setup and target configuration.

- Trajectory prediction allows visual servo controller to obtain multiple predicted positions for target. These positions can be diferantiated to obtain desired acceleration from axes which can directly used as current commands for motion controller.

- Laser pointer attached to the platform can be used the test and improve the capabilities of visual servo algorithm for non-illimunating targets.

- Different visual servo control algorithms can be studied and tested in this platform.

REFERENCES

[1] Faulhaber, *Faulhaber Series 2342…CR DC-Micromotors Datasheet*. Faulhaber.

[2] Faulhaber, *Faulhaber Series 30/1 S Planetary Gearhead Datasheet*. Faulhaber.

[3] T. Instruments, *BOOST-DRV8848 Dual H-Bridge Motor Driver Datasheet*. Texas Instruments.

[4] T. Instruments, *Tiva™TM4C123GH6PM Microcontroller DATA SHEET*. Texas Instruments.

[5] B. AG, *USER'S MANUAL FOR USB 3.0 CAMERAS*. Basler AG.

[6] COBHAM, "multi-axis-gimbal-systems." `https://www.cobham.com/advanced-electronic-solutions/motion-control-and-positioning/stabilised-gimbal-systems/multi-axis-gimbal-systems/`, 2018.

[7] E. G. Merriam, J. E. Jones, S. P. Magleby, and L. L. Howell, "Monolithic 2 dof fully compliant space pointing mechanism," *Mechanical Sciences*, vol. 4, no. 2, pp. 381–390, 2013.

[8] S. L. Canfield, "Development of the carpal wrist; a symmetric, parallel-architecture robotic wrist," 1997.

[9] J. Sofka and V. Skormin, "Integrated approach to electromechanical design of a digitally controlled high precision actuator for aerospace applications," pp. 261 – 265, 11 2006.

[10] J. Sofka, V. Skormin, V. Nikulin, and D. J. Nicholson, "Omni-wrist iii - a new generation of pointing devices. part ii. gimbals systems - control," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, pp. 726–734, April 2006.

[11] J. Sofka, “Wide angle gimbal.” `http://sofka.org/portfolio/gimbal/`, 2017.

[12] Y. Yun and Y. Li, “Modeling and control analysis of a 3-pupu dual compliant parallel manipulator for micro positioning and active vibration isolation,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 134, no. 2, p. 021001, 2012.

[13] Kollmorgen, *AKD User Guide*. Kollmorgen.

[14] T. Öztürk, *Angular Acceleration Assisted Stabilization of A 2-DOF Gimbal Platform*. PhD thesis, Master’s Thesis, Middle East Technical University, Ankara, 2010.

[15] R. J. Bauer, “Kinematics and dynamics of a double-gimbaled control moment gyroscope,” *Mechanism and machine theory*, vol. 37, no. 12, pp. 1513–1529, 2002.

[16] J. Hilkert, “Inertially stabilized platform technology concepts and principles,” *IEEE control systems magazine*, vol. 28, no. 1, pp. 26–46, 2008.

[17] M. Abdo, A. Toloei, A. R. Vali, and M. R. Arvan, “Cascade control system for two axes gimbal system with mass unbalance,” *International Journal of Scientific & Engineering Research*, vol. 4, no. 9, pp. 903–913, 2013.

[18] B. Ekstrand, “Equations of motion for a two-axes gimbal system,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, pp. 1083–1091, July 2001.

[19] Moflon, “Rotary solutions.” `https://www.moflon.com/products.html`.

[20] J. Sofka, V. Skormin, V. Nikulin, and D. J. Nicholson, “Omni-wrist iii - a new generation of pointing devices. part i. laser beam steering devices - mathematical modeling,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 2, pp. 718–725, 2006.

[21] V. V. Nikulin and R. M. Khandekar, “Demonstration of a mobile tracking system

for laser communications," in *2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel*, pp. 000781–000785, Nov 2010.

[22] T. U. of Texas at Austin, "Canfield joint." `https://www.cem.utexas.edu/content/canfield-joint`.

[23] T. U. of Texas at Austin, "Bt-cem high precision pointing system." `https://www.youtube.com/watch?v=_EOHgflAEHE`.

[24] S. Yimin, Q. Yang, D. Gang, and S. Tao, "Type synthesis of 2-dof rotational parallelmechanisms actuating the inter-satellite linkantenna," *Chinese Journal of Aeronautics*, vol. 29, no. 6, pp. 1795–1805, 2016.

[25] X. Huo, P. Wang, and W. Li, "Kinematic optimal design of a 2-dof parallel positioning mechanism employing geometric algebra," *Advances in Applied Clifford Algebras*, vol. 28, p. 20, Feb 2018.

[26] N. Bajaj, A. Spiers, and A. M. Dollar, "State of the art in artificial wrists: A review of prosthetic and robotic wrist design," *IEEE Transactions on Robotics*, vol. 35, pp. 261–277, 02 2019.

[27] X. Duan, Y. Yang, and B. Cheng, "Modeling and analysis of a 2-dof spherical parallel manipulator," *Sensors*, vol. 16, p. 1485, 09 2016.

[28] A. Cammarata, "Optimized design of a large-workspace 2-dof parallel robot for solar tracking systems," *Mechanism and machine theory*, vol. 83, pp. 175–186, 2015.

[29] D. Sanchez, "Surgical instrument with a universal wrist," Oct. 17 2006. US Patent 7,121,781.

[30] X. Duan, Y. Yang, and B. Cheng, "Modeling and analysis of a 2-dof spherical parallel manipulator," *Sensors*, vol. 16, no. 9, p. 1485, 2016.

[31] B. M. Schena, "Center robotic arm with five-bar spherical linkage for endoscopic camera," June 25 2013. US Patent 8,469,945.

[32] J. M. Wiitala *et al.*, "Design of an overconstrained and dextrous spherical wrist," *Journal of Mechanical Design*, vol. 122, no. 3, pp. 347–353, 2000.

[33] S. Hernandez, S. Bai, and J. Angeles, "The design of a chain of spherical stephenson mechanisms for a gearless robotic pitch-roll wrist," *Journal of Mechanical Design*, vol. 128, no. 2, pp. 422–429, 2006.

[34] L. Barker, M. Neber, and H. Lee, "Design of a low-profile two-axis solar tracker," *Solar Energy*, vol. 97, pp. 569–576, 2013.

[35] ROSS-HIME DESIGNS Inc , "Robotic manipulator with spherical joints," U.S. Patent No. US9630326B2, Apr. 2017.

[36] K. Sone, H. Isobe, and K. Yamada, "High angle active link," *Special Issue Special Supplement to Industrial Machines*, 2004.

[37] R.-H. D. Inc, "Omni-wrist vi." https://www.anthrobot.com/tech/, 2012.

[38] ROSS-HIME DESIGNS Inc , "Robotic manipulator with spherical joints," U.S. Patent No. US20150114163A1, Apr. 2015.

[39] R.-H. D. Inc, "Omni-wrist v." `http://www.anthrobot.com/omni-wrist-v/`, 2012.

[40] R.-H. D. Inc, "Omni-wrist vi." `https://www.anthrobot.com/omni-wrist-vi/`, 2012.

[41] H. Isobe and Y. Nishio, "Parallel link high speed angle control equipment (phace)," *NTN TECHNICAL REVIEW*, no. 80, pp. 42–47, 2012.

[42] S. Dyke, B. Spencer Jr, P. Quast, M. Sain, D. Kaspari Jr, and T. Soong, "Acceleration feedback control of mdof structures," *Journal of Engineering Mechanics*, vol. 122, no. 9, pp. 907–918, 1996.

[43] Yun Li, Kiam Heong Ang, and G. C. Y. Chong, "Patents, software, and hardware for pid control: an overview and analysis of the current art," *IEEE Control Systems Magazine*, vol. 26, pp. 42–54, Feb 2006.

[44] J. Liu, P. Zhang, and F. Wang, "Real-time dc servo motor position control by pid controller using labview," in *2009 International Conference on Intelligent Human-Machine Systems and Cybernetics*, vol. 1, pp. 206–209, IEEE, 2009.

[45] H. Zhou, "Dc servo motor pid control in mobile robots with embedded dsp," in *2008 International Conference on Intelligent Computation Technology and Automation (ICICTA)*, vol. 1, pp. 332–336, Oct 2008.

[46] Y. Koren, "Control of Machine Tools," *Journal of Manufacturing Science and Engineering*, vol. 119, pp. 749–755, 11 1997.

[47] P. Rocco, "Stability of pid control for industrial robot arms," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 606–614, Aug 1996.

[48] S. Bouabdallah, A. Noth, and R. Siegwart, "Pid vs lq control techniques applied to an indoor micro quadrotor," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, pp. 2451–2456 vol.3, Sep. 2004.

[49] B. P. Flannery, W. H. Press, S. A. Teukolsky, and W. Vetterling, "Numerical recipes in c," *Press Syndicate of the University of Cambridge, New York*, vol. 24, p. 78, 1992.

[50] A. Technologies, *AEDC-55xx / AEDC-56xx High Resolution Two or Three Channel Encoders Datasheet.* Avago Technologies.

[51] Lem, *LEM 50-S Current Transducer Datasheet.* Lem.

Appendix A

DIMENSIONS OF MECHANISM

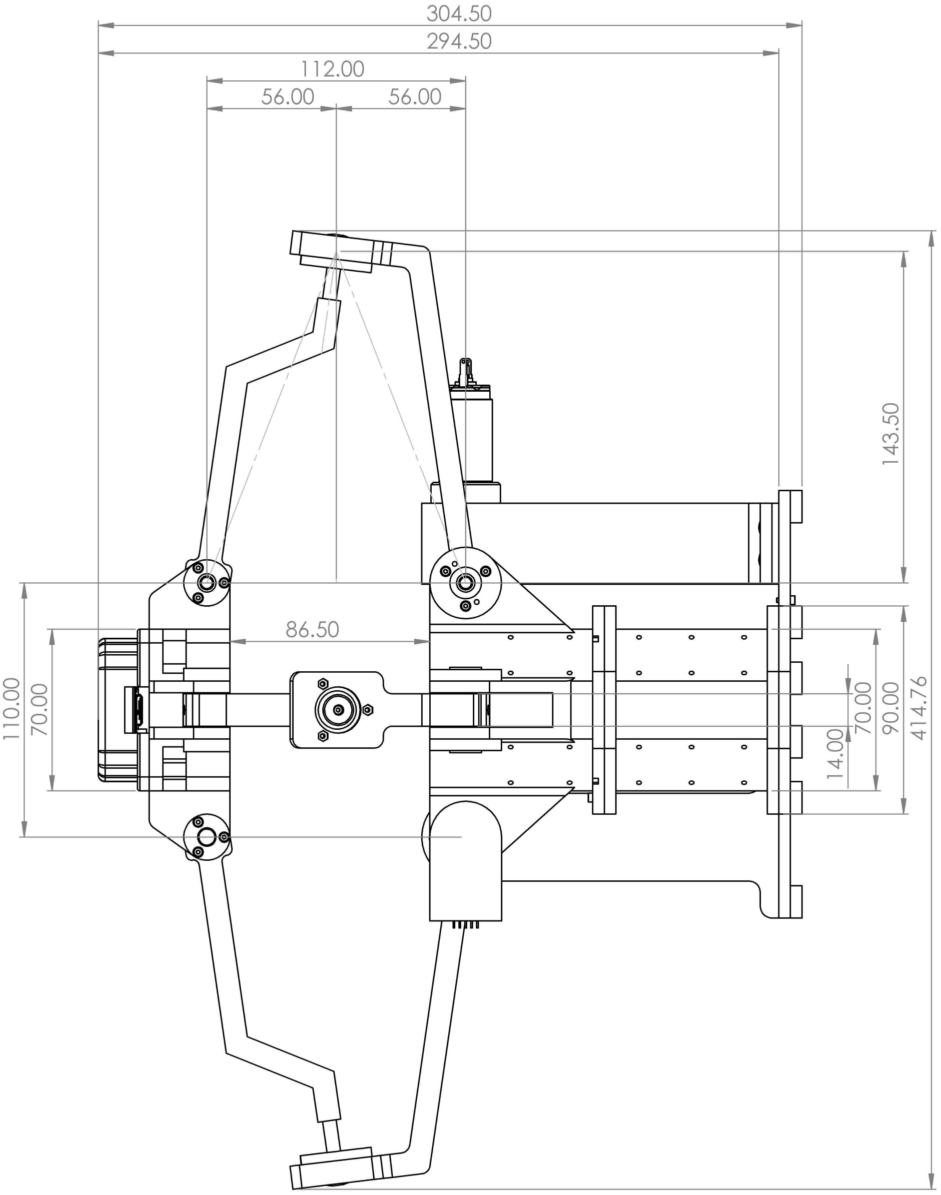


Figure A.1: Side view of mechanism in vertical position

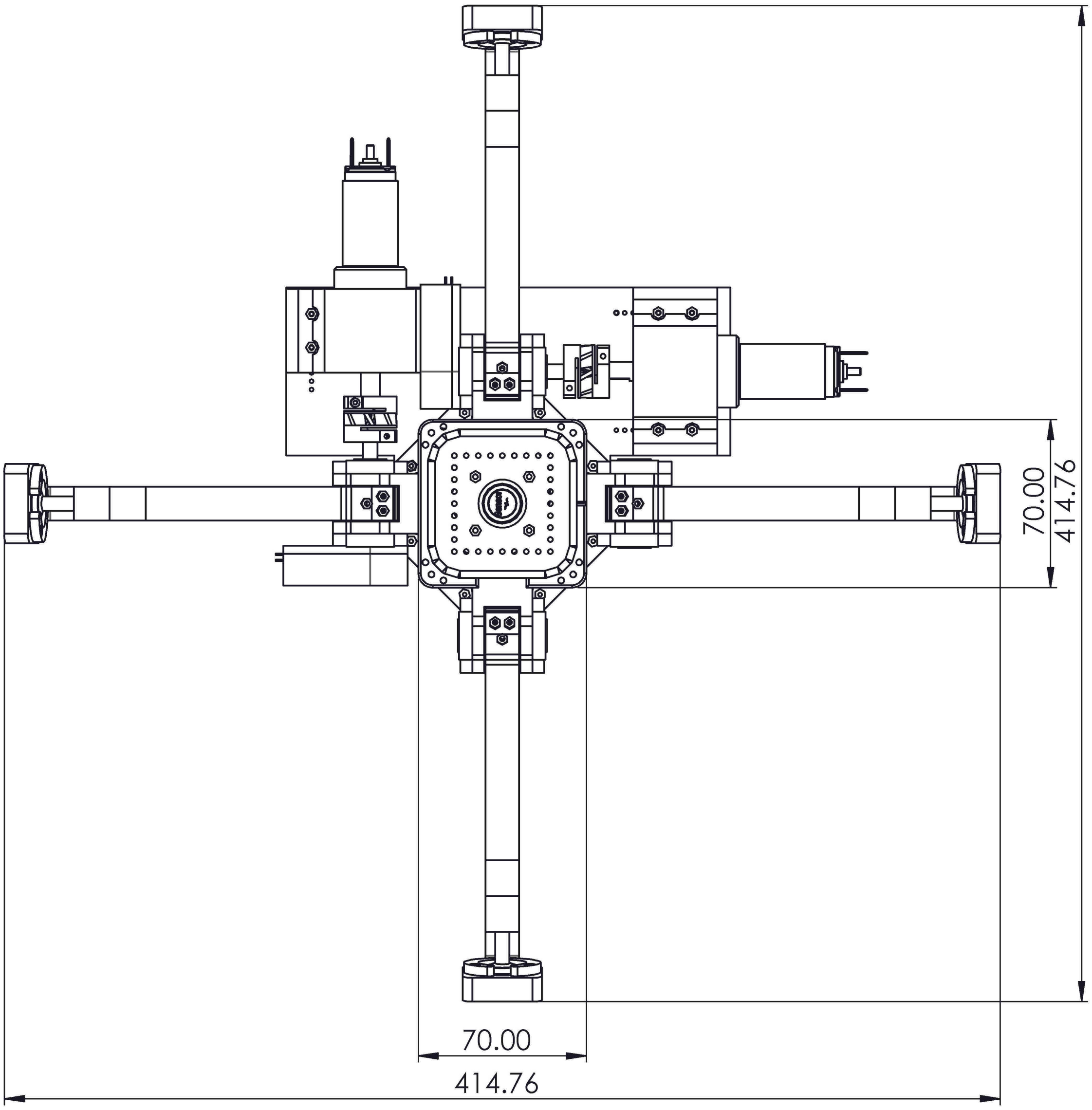


Figure A.2: Side view of mechanism in vertical position

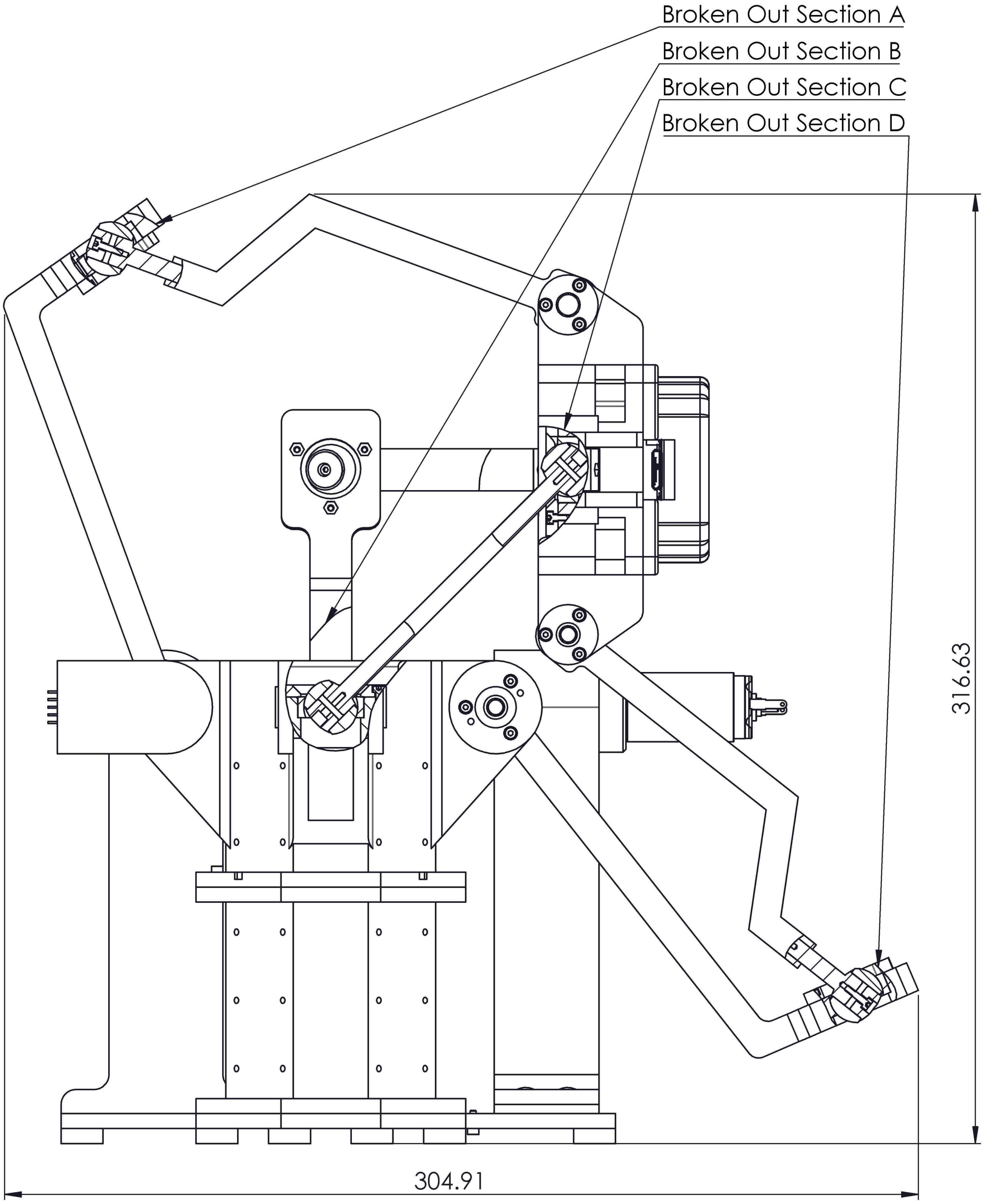


Figure A.3: Side view of mechanism in horizontal position with explanatory breakouts

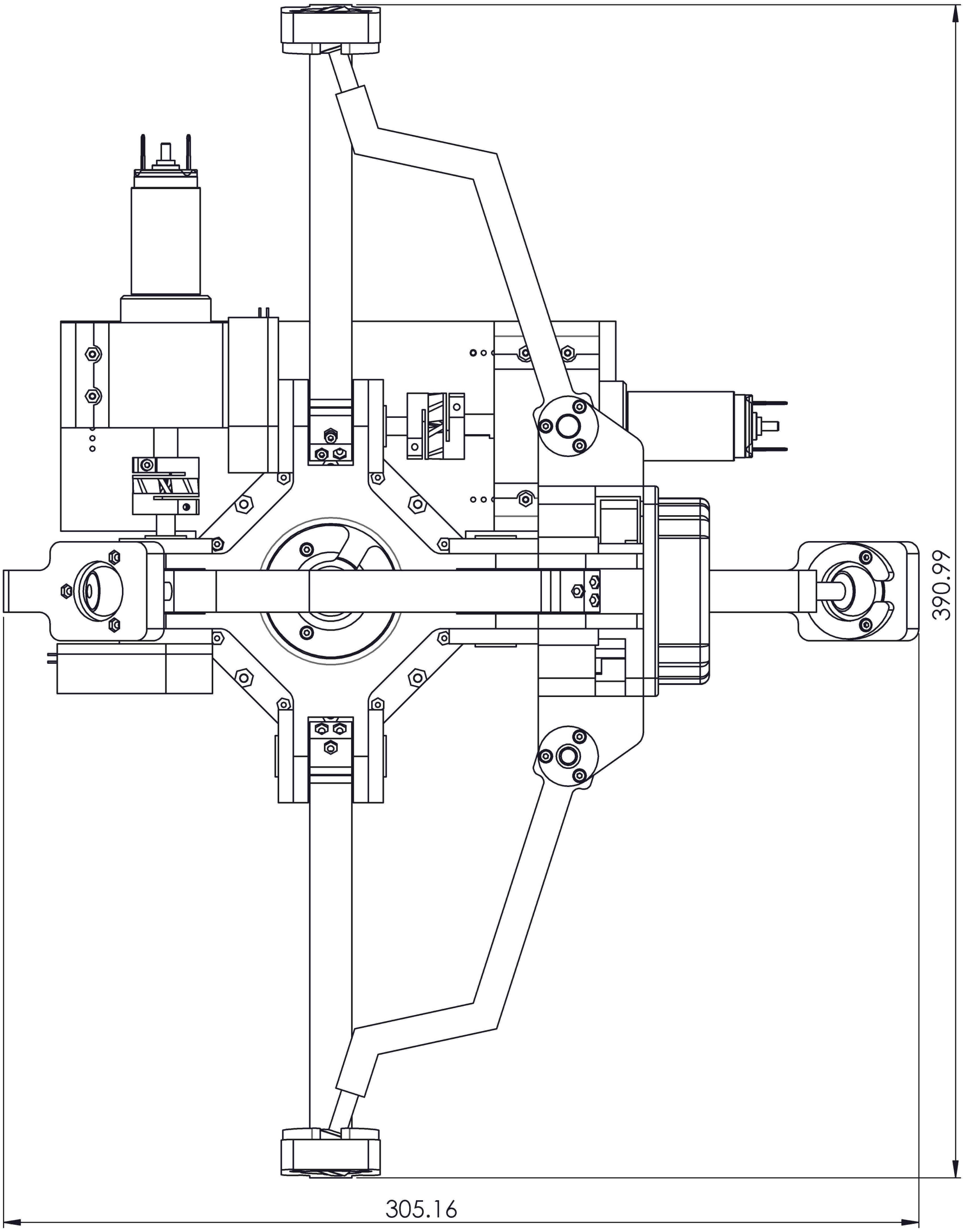


Figure A.4: Top view of mechanism in horizontal position

Appendix B

PSEUDOCODE FOR MOTION CONTROLLER

Implemented code handles the calculations which can be seen at Eqns. B.1 under one sampling time whenever its timer interrupt subroutine is called. This discrete controller code is written according to calculations made in Chapter 6

$$\begin{aligned}
\theta(k) &= 7.226 \times 10^{-3}\theta_{in} \\
I(k) &= 642.1312 \times 10^{-6} I_{in} \\
e_{theta}(k) &= \theta^{*}(k) - \theta(k) \\
I^{*}(k) &= \quad I^{*}(k-1) + 146.296 e_{\theta}(k) \\
&\quad\; -292.032 e_{\theta}(k-1) + 145.737 e_{\theta}(k-2) \\
e_{I}(k) &= I^{*}(k) - I(k) \\
V_a^{*}(k) &= V_a^{*}(k-1) + 8.064 e_{I}(k) - 0.143 e_{I}(k-1) \\
T_{ontime}(k) &= K_a V_a^{*}(k) \\
k &= k+1
\end{aligned}$$

where

θ_{in} = Encoder position reading;

I_{in} = Current transducer sensor reading ;

T_{ontime} = PWM on time;