ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

Passivity-Based Control of Stewart Platform for trajectory tracking

A thesis Submitted to Addis Ababa Institute of Technology, School of Graduate Studies, Addis Ababa University

In partial fulfillment of the requirement for the degree of Master of Science in Electrical Engineering (**Electrical control engineering**)

By

Tariku Sinshaw Tamir

Advisor: Mr. Andinet Negash

March 2015

ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Passivity-Based Control of Stewart Platform for trajectory tracking

By Tariku Sinshaw Tamir

APPROVED BY BOARD OF EXAMINERS

Chairman, Department of Graduate Committee	Signature
Mr. Andinet Negash	
Advisor	Signature
Internal Examiner	Signature
External Examiner	Signature

Abstract

Passivity based control, as one of the tools available to design robust controllers, is introduced for trajectory tracking of the Stewart platform. Since Passivity is a fundamental property of many physical systems which may be roughly defined in terms of energy dissipation and transformation, its inherent input output property quantifies and qualifies the energy balance of a system when simulated by external inputs to generate some outputs. PD+ controller is designed based on passivation principle so that the closed loop system becomes globally uniform and asymptotically stable.

The mathematical model of the Stewart platform, derived from Euler Lagrange equations of motion, is simulated on MALAB/Simulink with the designed controller. So as to get the desired leg-length trajectory, the inverse kinematics formulation is investigated. The mathematical model is verified using "automatic dynamic analysis of mechanical systems" (ADAMS) software.

In the absence of disturbances, the maximum trajectory tracking error is recorded as ($0.006m$) in the time interval between $0\sec$ and $0.8\sec$. Applying unit step disturbance makes the error $0.006m$ after $0.8\sec$, which never be seen in undisturbed system.. The maximum speed of all the six legs trajectory have been found to be $0.43m/s, 0.49m/s, 0.48m/s$, $0.5m/s$, $0.5m/s$ and $0.49m/s$ respectively. More realistic results are observed from ADAMS simulation results.

Key words: passivity based control, PD+, global uniform asymptotically stable, Euler Lagrange equation, ADAMS software, inverse kinematics, MATLAB/Simulink.

Acknowledgment

I am indebted to Mr.Andinet Negash, my advisor, for his stimulating suggestions and for his patience in the long hours of discussion. His deep insight in to the subject matter and his positive criticism throughout the course of this work were of great help to me.

I also wish to express my gratitude to Ahmed Mehamed and my colleagues, academic staffs of Electrical and Computer Engineering department, for their vital comments, criticism and advice for the development of the thesis work. In addition, I would like to thank my senior friends who showed me the way how to gather information and how to write the thesis in its final form.

I am also very grateful to my parents whose confidence in me and encouragement has greatly contributed to the completion of the thesis work.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

DOF	……	Degree of freedom
SP	……	Stewart platform
PBC	……..	Passivity based controller
EL	…….....	Euler Lagrange
PID	…….....	Proportional integral derivative
PD	…….....	Proportional derivative
PD+	……	Proportional derivative plus
DC	……..	Direct current
CAD	……..	Computer aided design
SMC	……..	Sliding mode controller
DAQ	……..	Data acquisition system
PWM	……..	Pulse width modulation
ADAMS	……...	Automatic dynamic analysis of mechanical system
GUAS	…….....	Global uniform asymptotic stability
3D	……	Three - Dimension (al)
AAIT	……...	Addis Ababa institute of technology

List of Notations

$\equiv$	.....	Equivalent to
$\approx$	.....	Approximately equal to
α	……..	Alpha
β	……..	Beta
γ	……..	Gamma
$\rightarrow$	……..	Tends to
$(.)^{-1}$	……..	Inverse
$(.)^{T}$	……..	Transpose
{.}	……..	Coordinate (reference) frame
${}^{A}_{B}R$	……	rotation matrix
det(.)	……...	Determinant
τ	……	Torque
ω	……	Angular speed
$(.)^{*}$	……	Reference signal
J	……	Jacobean, Moment of inertia
μ	……	Micro
$\forall$	…….....	For every (all)

Chapter1

Introduction

1.1.Background of the Project

Unlike open-chain serial robots, parallel manipulators are composed of closed kinematic chains. Serial manipulators can be defined as simple kinematic chains for which all the connection degrees are two, except for the base and the end effectors, such a chain is called an open loop kinematic chain. A closed-loop kinematic chain is obtained when one of the links, but not the base, possesses a connection degree greater than or equal to three, there exist several parallel kinematic chains between base platform and the moving platform. [1]

Serial robots consist of a number of rigid links connected in series so that every actuator supports the weight of the successor links. This serial structure suffers from several disadvantages such as low precision, poor force exertion capability and low payload-to-weight-ratio. Under heavy loads, serial robots cannot perform precision positioning and oscillate at high-speeds. [4]

On the other hand, in parallel manipulators the load is shared by several parallel kinematic chains. This superior architecture provides high rigidity, high payload-to-weight-ratio, high positioning accuracy, low inertia of moving parts and a simpler solution of the inverse kinematics equations over the serial ones. Since high accuracy of parallel robots stems from load sharing of each actuator, there are no cumulative joint errors and deflections in the links. Furthermore, the positioning accuracy of parallel robots is high due to the fact that the positioning error of the platform cannot exceed the average error of each leg's positions. They can provide nanometer-level motion performance. But they have smaller workspace and singularities in their workspace. [4]

Since a parallel structure is a closed kinematics chain, all legs are connected from the origin of the tool point by a parallel connection. This connection allows a higher precision and a higher velocity. Parallel kinematic manipulators have better performance compared to serial kinematic manipulators in terms of a high degree of accuracy, high speeds or accelerations and high stiffness. Therefore, they seem perfectly suitable for industrial high-speed applications, such as

pick-and-place or micro and high-speed machining. They are used in many fields such as flight simulation systems, manufacturing and medical applications. [3]

One of the most popular parallel manipulators is the general purpose 6 degree of freedom (DOF) Stewart Platform (SP) proposed by Stewart in 1965 as a flight simulator. It consists of a top plate (moving platform), a base plate (fixed base), and six extensible legs connecting the top plate to the bottom plate. SP employing the same architecture of the Gough mechanism is the most studied type of parallel manipulators. This is also known as Gough–Stewart platforms in literature. [3]

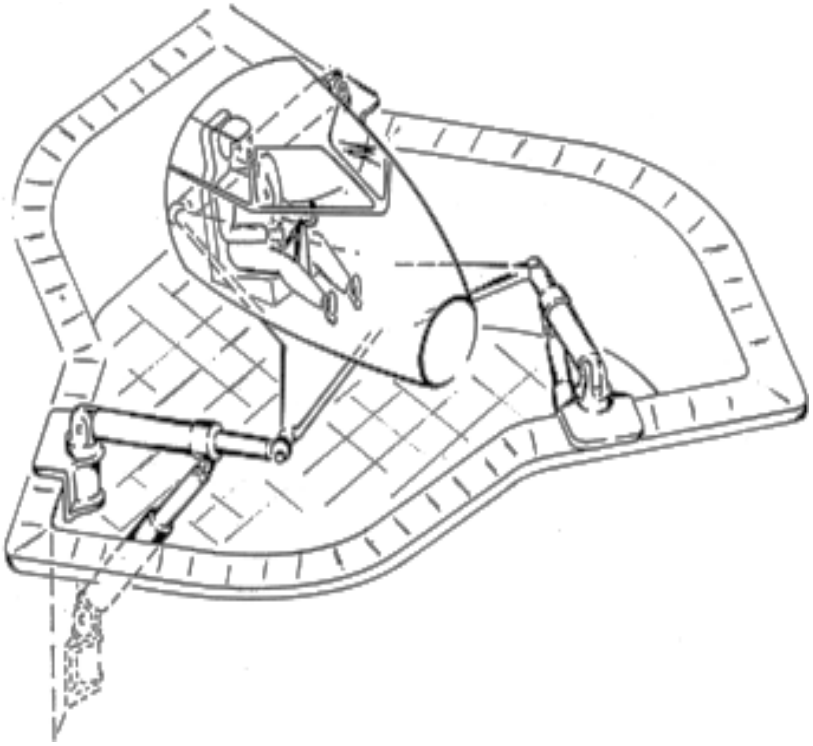


Figure 1.1: Stewart Platform as Flight Simulator. [9]

Different control schemes have been illustrated for the normal functioning of the platform at different application areas. Parallel manipulators suffer from non-linearity, which is difficult to find closed form solution. The analytical methods provide the exact solution; but, they are too complicated because the solution is calculated by solving the high-order polynomial equations. There is also the selection problem of the exact solution among the several ones. In fact there exists no general closed-form solution for the above problem. The Newton Rapson method is known as a simple algorithm for solving nonlinear equations, whose convergence is good, but it

requires much calculation time, and also it sometimes converges to the wrong solution according to the initial values. It can be used to solve the forward kinematic problem. In the methods using the observers, two kinds of observers, linear and nonlinear, are used. The linear observer is based on linearizing the nonlinear terms and leaves the steady-state error. The nonlinear observer has the difficulty to select the observation gain. A neural network based method can be applied to solve the forward kinematics problem as a basic element in the modeling and control of the robot. [4]

While the kinematics of parallel manipulators has been studied extensively during the last two decades, fewer contributions can be found on the dynamics of parallel manipulators. Dynamical analysis of parallel robots, which is very important to develop a model-based controller, is complicated because of the existence of multiple closed-loop chains. Different modeling approaches have been illustrated by different researchers. Euler-Lagrange method, virtual work principles, Lagrange formulation and analyzing using screw theory were used for the modeling of robot manipulators. Dynamic equations of a Stewart-Gough platform can be derived based on Lagrange's formulation. [4]

The term passivity based control (PBC) was coined in 1989 in the context of adaptive control of robot manipulators to define a controller methodology whose aim is to render the closed loop system passive. This objective seemed very natural within that context, since the robot dynamics defines a passive map from input torque to output link velocities. As a matter of fact passivity property is inherent to many other physical systems such as electrical and electromechanical. [15]

1.2. Problem Description

Demand of high precision motion systems has been increasing in recent years. Since performance of today's mechanical systems requires high stiffness, fast motion and accurate positioning capability, parallel manipulators have gained popularity. Currently, parallel robots are widely used in several areas of industry such as manufacturing, medicine and defense. [5]

There are many methods for generating the dynamic equations of mechanical systems. Newton-Euler approach, principle of virtual work and Euler-Lagrange (EL) formulation are the typical

ones. All methods generate equivalent sets of equations, but different forms of these equations may be better suited for computation or analysis. We will use Euler-Lagrange approach for our derivation, which relies on the energy properties of mechanical systems to compute the equations of motion. The resulting equations can be computed in closed form, allowing detailed analysis of the properties of the system. [1]

The Euler Lagrange equations are the outcome of a powerful modeling technique, the variation method. EL systems models are, roughly speaking, obtained from the minimization of an energy function. It is therefore expected that they enjoy some energy dissipation properties, in particular that they define passive maps, which one can profitably use for the controller design. [15]

There are different nonlinear controllers for the control of robotic manipulators, such as Computed torque control, feedback linearization, PID controller, passivity based controller, robust controllers.

The mathematical model of the Stewart platform will be analyzed based on energy minimization principle and passivity principle. Therefore, it will be straight forward to use passivity based controller for Stewart platform modeled by Euler Lagrange formulation. Passivity-based controllers also have extra advantage of improving the robustness properties of the Stewart platform amidst un-modeled dynamics due to joint and link flexibility, friction, noise and disturbances. [1]

1.3. Objective of the Thesis

1.3.1. General Objectives

The general objective of the thesis is:

i. To design a passivity-based controller to get set point regulation and desired trajectory tracking for Stewart platform.

1.3.2. Specific Objectives

The specific objectives of the thesis are:

i. To formulate the kinematics equations of the Stewart platform.
ii. To derive the dynamics equations of the Stewart platform and the actuator system
iii. To design PD+ controller based on passivity analysis for both set point regulation and desired trajectory tracking of the Stewart platform.
iv. To demonstrate the performances of the system using MATLAB/Simulink and ADAMS.

1.4. Contribution of the Thesis

The contributions of this thesis are:

i. The kinematics formulated in this thesis will have a great impact for any parallel robot manipulator to get an accurate model easily. Mathematical modeling using Inverse kinematics is a straight forward for parallel robots. Therefore, one can be clear on kinematics analysis of any robotic systems.
ii. Lagrange modeling of the system makes us to have energy concept in the system. Therefore, setting a controller on the basis of passivity will not be quite difficult task. Since the modeling technique has energy conservation and minimization principle, one automatically looks for a controlling mechanism having energy principles, passivity properties.
iii. Passivity based controller is currently a new concept for robotic applications. Many researchers are becoming more interested with this astonishing concept. Working with passivity principle will make system simple and effective, accurate output.
iv. Simulation using MATLAB/Simulink and ADAMS makes the system meaningful. The outputs will have contribution for getting scientific decision to select a controller. This thesis contributes towards mathematical modeling, passivity analysis of the system, design of PD+ controller for desired trajectory tracking of Stewart platform.

1.5. Outline of the Thesis

Chapter 1 presents the background material on general robotic manipulators; both serial and parallel robots are discussed. Problem statement, the objective as well as the contribution of the thesis is presented here. In addition, an extended review on previous research done on passivity-based controller design method and the various approaches used for the control of the Stewart platform are given. Chapter 2 begins by general overview of the Stewart manipulator and introduces the position and orientation concept and ends by introducing different literatures. The kinematics and dynamics of the Stewart platform and actuator dynamics is presented in chapter 3. Chapter 4 is devoted to controller design based on passivity analysis. PD type controller is presented first followed by PD+ controller for set point regulation and trajectory tracking of the Stewart platform. And finally, ADAMS 3-D model of Stewart platform is presented. Chapter 5 is dedicated to result demonstration and analysis of the performance of the passivity based controller. Conclusions, recommendations and suggestion for future work are all presented in chapter 6.

Chapter 2

The Stewart Manipulator

2.1. General Overview

Figure 2.1: solid model of typical Stewart platform manipulator [10]

The Stewart platform is six degrees of freedom robot. It is composed of six prismatic legs connected to the base and moving platform by passive spherical joints and at the middle of these legs are prismatic actuated joints. The design variation of different Stewart platform types dependents only on the placements of the joint on the base and the platform. The figures below are used to classify these different design types. [16]

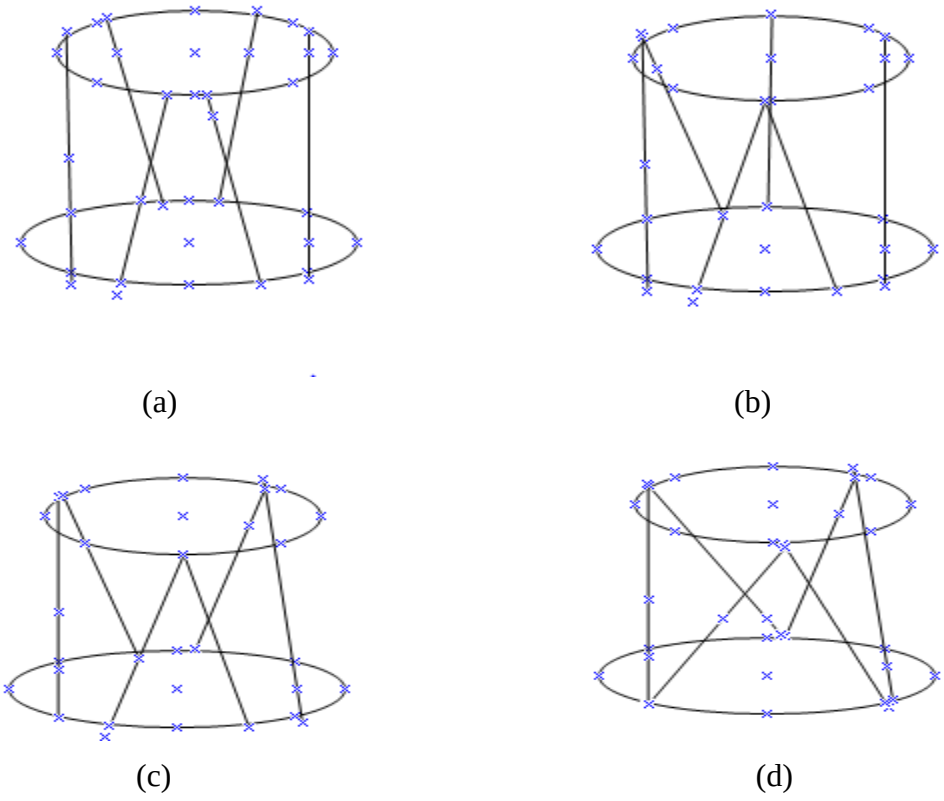


Figure 2.2: (a) 6-6 joint placement, (b) 6-4 joint placement, (c) 6-3 joint placement, (d) 3-3 joint placement [16]

In this thesis, the selected configuration of the Stewart platform is as shown in Figure 2.2 (a). The basis for the selection of the configuration is mainly on the load handling capacity and the overall strength of the platform.

2.2. Literature Review

In [6], passivity based adaptive control of two links Robot Manipulator Electrically Controlled is modeled. The main interest of this contribution is to provide a direct adaptation law in order to improve performances against both robot model parameters uncertainties and disturbances. The stability is checked based on the lyapunov stability theory. Besides, the proposed configuration permits a very remarkable facility of implementation. An adaptive passivity control applied to the two link robot manipulator electrically controlled is proposed. The developed algorithm guarantees the asymptotic stability of the global system, and illustrates its efficiency in

eliminating the disturbances. The results show a good level of tracking accuracy. Moreover, the key advantage of this method lies in its simplicity on one hand and its robustness against parameters uncertainties, joint friction, and disturbances on the other hand.

In [7] is presented practical model based and robust control of parallel manipulators using passivity and sliding mode theory. It provides a practical strategy to realize accurate and robust control for 6 DOFs parallel robots. This paper designs sliding mode controller based on passivity concept for the analysis of the performance of the controlling mechanism. The first part, the passivity formalism offers an excellent framework to design and to tune the closed-loop dynamics, such that the desired behavior is obtained. The basic algorithm is proved to be locally robust towards uncertainties. The second part of the control strategy consists in a sliding mode controller. To keep the practical and computationally efficient implementation, the proposed switching control considers explicitly only the friction model. Here it opts for the so called model-decomposition paradigm and uses additional integral action to improve robustness. The proposed approach is substantiated with experimental results demonstrating the effectiveness and success of the strategy that keeps control setup simple and intuitive.

The application area covers fast handling and light cutting machining tasks with low process forces. As shown in Figure 2.4, the structure comprises of six legs with sharp tip points up ward.

Figure 2.3: CAD model hexapod for machine tool applications [7]

In [9], position control and trajectory tracking of Stewart platform is designed. Firstly, kinematic solution is computed before controller design. A controller is designed to move top platform from initial position to desired position and orientation. It generated required forces for each motor. Position and trajectory control of the platform could be reduced to leg position control after inverse kinematic and path planning algorithms. a PID and sliding mode position Controllers were developed and implemented. These controllers were embedded in a Dspace DS1103 real time controller which is programmable in the Simulink environment. Design details and development stages of the PID and SMC are given from subsystems to main model in Simulink. The real time Dspace representation of the Stewart platform is given in figure 2.5.

Figure 2.4: Stewart platform system [9]

In [10], the paper presents design and development of six DOF motion platforms for vehicle driving simulator. It illustrates the kinematic and dynamics analysis. In this project, PID controller is used and tuned using Ziegler-Nichols and approximation method.

Driving simulators are often used in educational and research purposes. Driving simulators' capability in producing a virtual driving environment resembling real driving condition can be used to train novice drivers before they are exposed to the real world. Aside from that, driving simulators are important in data collection for road safety research, human factor study, vehicle system development and also traffic control device development. These allow designers, engineers as well as ergonomists, to bypass the design and development process of detailed mockups of the automobile interiors for human factor and vehicle performance studies.

Figure 2.5: National advance driving simulator [10]

The motion platform is interfaced with control model in order to perform the 6 - DOF motion cues. The motion platform system is presented in Figure 2.6. First, the desired motion platform positions are fed into the simulation model. The motion platform simulation model then calculates the required actuators length to perform the motion cues. The model sends the input signal and passes through a PID controller to the data acquisition system (DAQ). In the meantime, the simulation model also passes the output data to *SimMechanics* . Communication between the mathematical model and DAQ is established using S Function written in C programming language. The digital signal is converted to analog signal and pulse width modulation (PWM) signal to control the motor driver which drives the DC actuator. The DC actuator position signal is retrieved using potentiometer. The signal is converted to digital signal through the DAQ and filtered with low pass filter before feedback to (PID) as error signal. This completes the closed loop control system.

In [11], trajectory tracking of robot manipulators using linear and nonlinear PD-type controllers is implemented. It presents trajectory tracking of robot manipulators using linear and nonlinear proportional-derivative (PD) type controllers. It is shown that the controller gains which are nonlinear functions of system states can improve tracking performance and prevent actuator

saturation. Furthermore, it is shown that global asymptotic stability of the closed-loop system can be achieved with the proposed PD+ type controllers.

The closed-loop robot system together with a PD controller with gravity compensation has been shown to be globally asymptotically stable provided that the proportional and derivative gains are positive constants. The planned task is to design a linear PD+ type controller applied to the single-link robot manipulator to follow a desired trajectory.

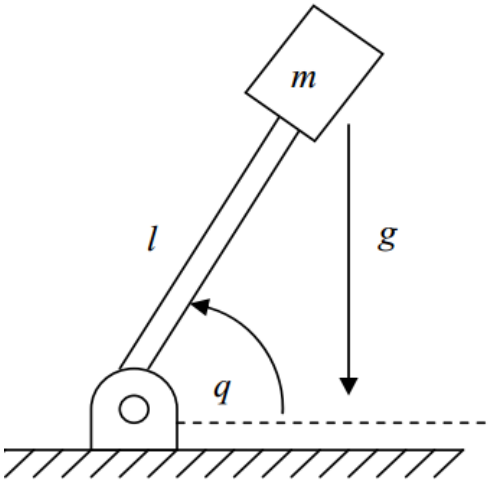


Figure 2.6: single link robot [11]

In [13], design of Stewart platform for wave compensation is introduced. Ship mounted cranes have many applications and are used for several different operations, e.g. to unload ships in minor ports without regular unloading equipment, to offshore operations like transportation of cargo from one ship to another when deep water ports are not available, servicing offshore wind turbines, moving oil rig anchors etc. The Stewart platform is intended as a mounting platform for the ship crane and to move opposite to the ship to keep the crane steady in space, and not relative to the sea to minimize motion in the crane cargo. As great static and dynamics loads are applied to the Stewart platform by the ship crane, a hydraulic actuation system is used for the Stewart platform.

The focus of the paper is to design and optimize the kinematics of the Stewart platform. Secondly it deals with the dimensioning of a hydraulic servo system to the optimized kinematic design.

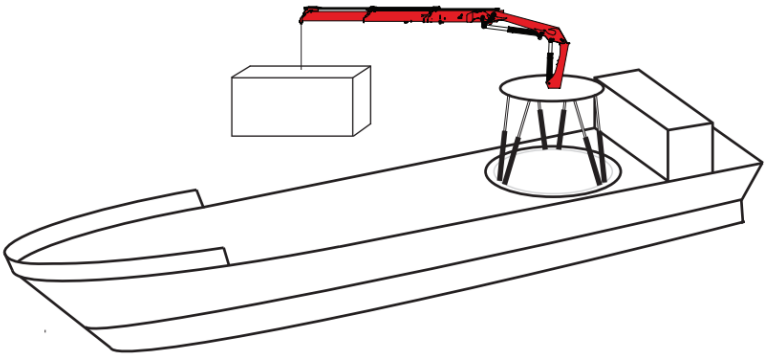


Figure 2.7: ship crane mounted on Stewart platform [13]

Chapter 3

Methodology

3.1. Mathematical Modeling of Stewart Platform

The first stage in mathematical modeling in this particular work is formulating the kinematics of the Stewart platform in such a way that it gives the desired quantity for performance analysis and controlling purpose. The kinematics here used will be inverse kinematics which gives the generalized coordinates/ length of legs/ by taking the end effectors moving platform coordinates as input variables. Secondly, the Jacobean matrix will be derived and analyzed; this matrix will be used in coordinate transformation as well as singularity analysis. Thirdly, the dynamics of the platform and the six legs will be formulated separately. The dynamics of the platform will be modeled by taking the legs and the moving platform separately, and then will be merged together. Lastly, the actuator dynamics will be formulated by considering the electrical and mechanical parts.

The flow chart of passivity-based control of the Stewart platform for trajectory tracking and the controlling mechanism block diagram is shown in Figure 3.1 and Figure 3.2 respectively.

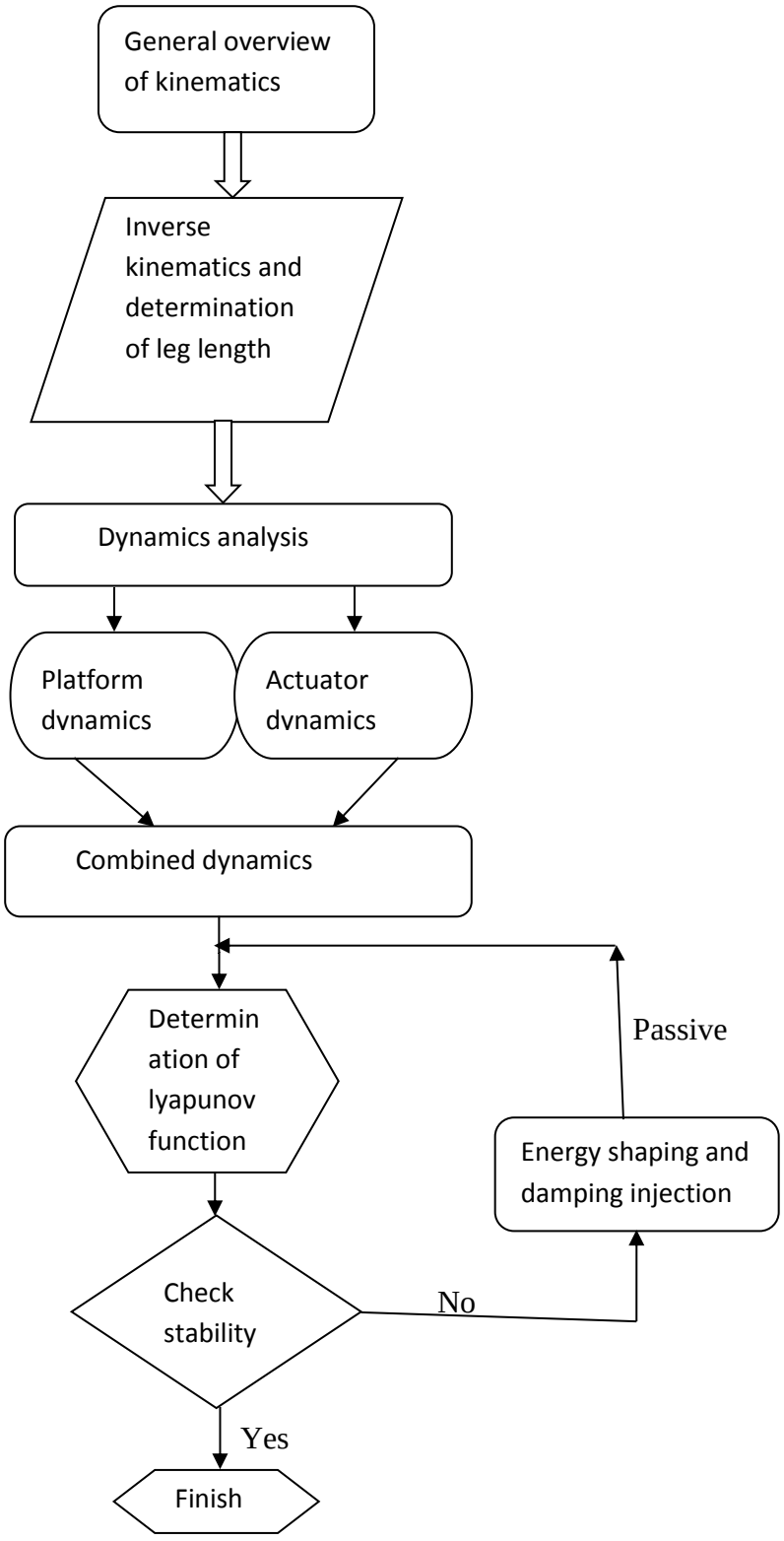


Figure 3.1: Flow chart of passivity based control of Stewart platform for trajectory tracking

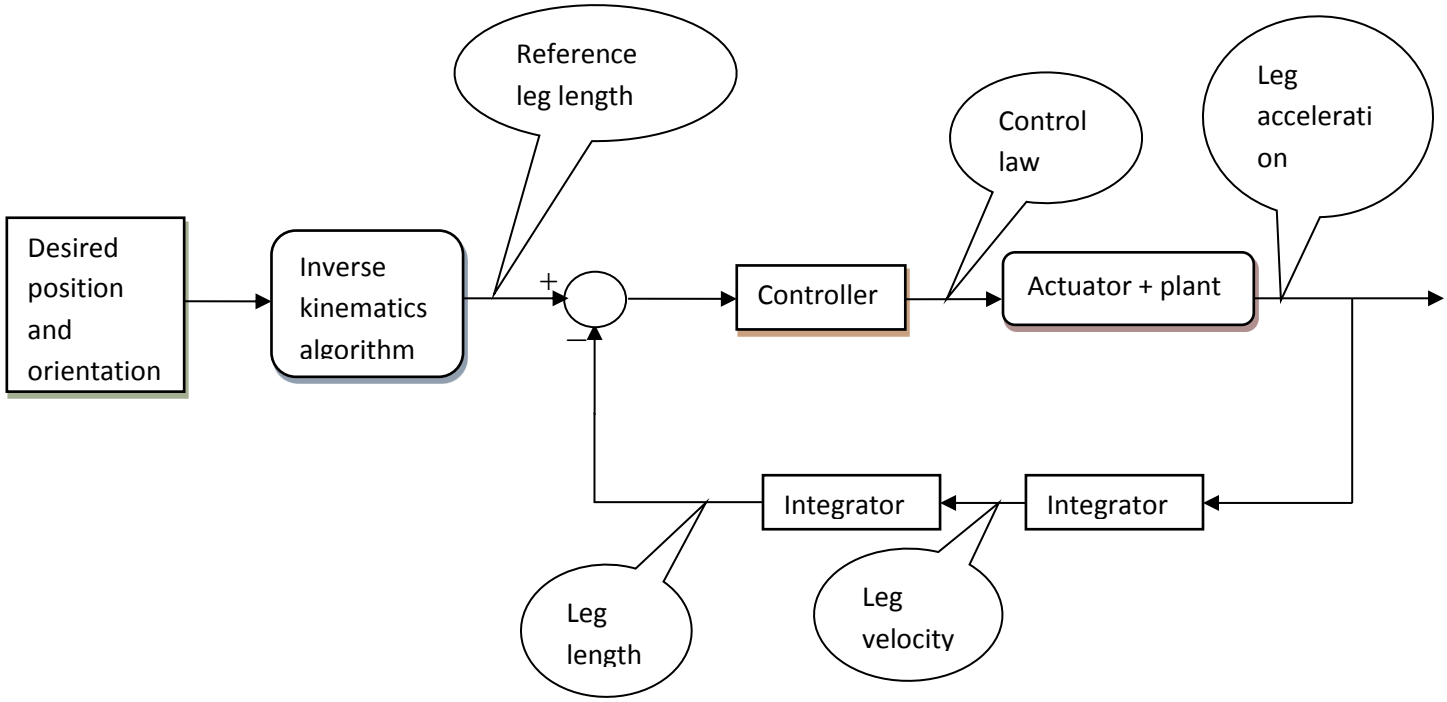


Figure 3.2: outer loop block diagram of Stewart platform

3.2. Kinematics of the Stewart Platform

The SP used in this study is a six DOF parallel mechanism that consists of a rigid moving plate, connected to a fixed base plate through six independent kinematic legs. These legs are identical kinematics chains, which couple the moveable upper and the fixed lower platforms by spherical joints. Each leg contains a precision ball-screw assembly and a DC- motor. Thus, the leg lengths are variable and they can be controlled separately to perform the motion of the moving platform.

The kinematic model of robots describes how the motion of the joints of a robot is related to the motion of the rigid bodies that make up the robot. For the analysis of the kinematics of the manipulator, we need to have the rotation matrix and the position vector.

The rotation matrix is composed of three elementary rotation matrices.

a. Rotation About fixed x- axis with angle of α
b. Rotation About fixed y-axis with angle of β
c. Rotation About fixed z-axis with angle of γ

Let us consider the first rotation (yaw) around the z axis and the associated matrix is given below. We define the rotation matrix $R_z(\gamma)$

$$R_z(\gamma) = \begin{bmatrix} c\gamma & -s\gamma & 0 \\ s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

Also let us assume the second rotation (pitch) around the y axis and the associated matrix is given below. We define the rotation matrix $R_y(\beta)$)

$$R_y(\beta) = \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \tag{3.2}$$

Again, let us assume the third rotation (roll) around the x axis and the associated matrix is given below. We define the rotation matrix $R_x(\alpha)$

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\alpha & -s\alpha \\ 0 & s\alpha & c\alpha \end{bmatrix} \tag{3.3}$$

And by combining three matrices in equation 3.1, 3.2 and 3.3, we obtain the combined orientation matrix ${}^{T}_{B}R$ of the end effector referred to the base frame.

$${}^{T}_{B}R = R_z(\gamma)\, R_y(\beta)\, R_x(\alpha) = \begin{bmatrix} c\gamma & -s\gamma & 0 \\ s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\alpha & -s\alpha \\ 0 & s\alpha & c\alpha \end{bmatrix}$$

$$= \begin{bmatrix} c\beta c\gamma & c\gamma s\alpha s\beta - c\alpha s\gamma & s\alpha s\gamma + c\alpha c\gamma s\beta \\ c\beta s\gamma & c\alpha c\gamma + s\alpha s\beta & s\gamma c\alpha s\beta s\gamma - c\gamma s\alpha \\ -s\beta & c\beta s\alpha & c\alpha c\beta \end{bmatrix}$$

$$\equiv \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{3.4}$$

Where,$\{S(.) = \sin(.)$ and $C(.) = \cos(.)\}$

And the position will be;

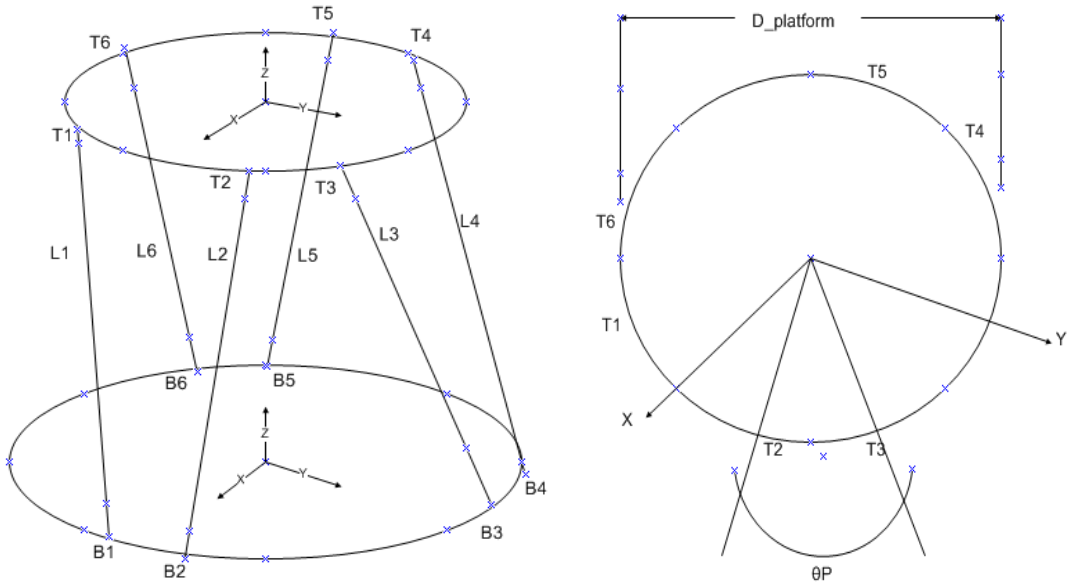
$$P = [P_x \quad P_y \quad P_z]^T \tag{3.5}$$

In case of forward kinematics the generalized coordinates of the manipulator will be given and used to solve the end effectors variables. The second one is inverse kinematics. In this case, the end effectors variable (position and orientation) of the manipulator will be given and used to solve the leg lengths of the manipulator.

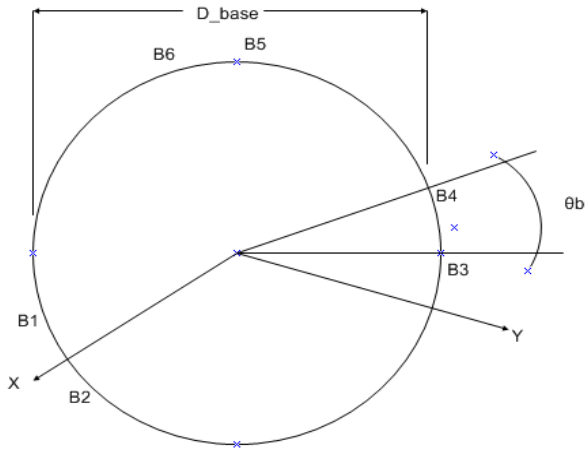
When defining the dimensions of the Stewart platform, the following requirements must be taken into account:

i. The manipulator must be able to operate within the prescribed workspace required by the application.
ii. The kinematic design must allow the prismatic actuators to produce the required forces and velocities in all directions needed for the specific application.
iii. The prismatic actuation forces must be within the limit of what is possible and realistic from a structural point of view.
iv. The length properties of the prismatic actuators must be physical realizable.
v. The usage of stroke length must be as large as possible in order to reduce the size of the prismatic actuators and the manipulator in general. (Provided that identical prismatic actuators are used.)
vi. Avoid collisions of the prismatic actuators however this is only a concern for designs with arbitrary joint placements.

Let us consider the figure shown below.



(a) (b)



(c)

Figure 3.3: (a) the schematic diagram of the SP, (b) the upper plate/moving platform coordinate and attachment points, (c) the lower plate/fixed platform coordinate and attachment points

To clearly describe the motion of the moving platform, the coordinate systems are illustrated in figure 3.3 (a). The coordinate system (B_{xyz}) is attached to the fixed base and the other coordinate

system (T_{xyz}) is located at the center of mass of the moving platform. Points (B_i and T_i) are the connecting points to the base and moving platforms, respectively. These points are placed on fixed and moving platforms (Figure 3.3 (a)). Also, the separation angles between points (T_2 and T_3, T_4 and T_5, T_1 and T_6) are denoted by Θ_p as shown in Figure 3.3 (b). In a similar way, the separation angles between points (B_1 and B_2, B_3 and B_4, B_5 and B_6) are denoted by Θ_b as shown in figure 3.3 (c).

From Figure 3.3(b), the location of the i^{th} attachment point (T_i) on the moving platform can be found (Equation 3.6). From figure 3.3(c), by using the same approach, the location of the i^{th} attachment point (B_i) on the base platform can be also obtained from Equation 3.7. R_p And R_b are the radii of the moving platform and fixed base respectively.

$$GT_i = \begin{bmatrix} GT_{xi} \\ GT_{yi} \\ GT_{zi} \end{bmatrix} = \begin{bmatrix} R_p \cos(\lambda_i) \\ R_p \sin(\lambda_i) \\ 0 \end{bmatrix}, \lambda_i = \frac{i\pi}{3} - \frac{\theta_p}{2}, i = 1,3,5 \text{ And } \lambda_i = \lambda_{i-1} + \theta_p, i = 2,4,6 \tag{3.6}$$

$$B_i = \begin{bmatrix} B_{xi} \\ B_{yi} \\ B_{zi} \end{bmatrix} = \begin{bmatrix} R_b \cos(\upsilon_i) \\ R_b \sin(\upsilon_i) \\ 0 \end{bmatrix}, \upsilon_i = \frac{i\pi}{3} - \frac{\theta_b}{2}, i = 1,3,5 \text{ And } \upsilon_i = \upsilon_{i-1} + \theta_b, i = 2,4,6 \tag{3.7}$$

The pose of the moving platform can be described by a position vector, P and a rotation matrix, ${}^T_B R$. Thus, by using the rotation matrix and the position vector combining with the platform geometry dependent connecting points of the moving platform and fixed base, we can determine the length of the i^{th} link in vector form.

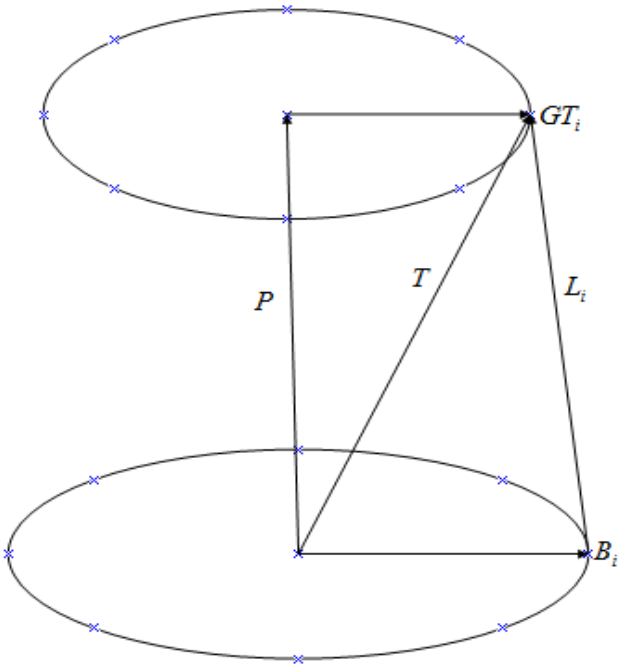


Figure 3.4: kinematic diagram for determination of leg length

From figure 3.4, one can formulate the vector equation described by:

$$T = P + {}^{T}_{B}RGT_i \tag{3.8}$$

And also

$$L_i = T - B_i \tag{3.9}$$

Therefore, we can give the final expression of the length of the i^{th} legs/links as:

$$L_i = {}^{T}_{B}RGT_i + P - B_i, i = 1..... 6 \tag{3.10}$$

When the position and orientation of the moving platform are given:

$$X_{p-o} = [p_x \quad p_y \quad p_z \quad \alpha \quad \beta \quad \gamma]^T$$

The length of each leg may be computed as the following.

$$L_i^{\,2} = (P_x - B_{xi} + GT_{xi}R_{11} + GT_{yi}R_{12})^2 + (P_y - B_{yi} + GT_{xi}R_{21} + GT_{yi}R_{22})^2 + (P_z + GT_{xi}R_{31} + GT_{yi}R_{32})^2 \tag{3.11}$$

Once the position and orientation of the platform is given the leg length will be automatically solved by using equation (3.11). By writing some script on MATLAB the solution can be solved easily or most recommended way is using MATLAB/Simulink. Constructing the model in such a way that it takes input arguments, and outputs the desired quantity is presented. The script is written in parallel with the MATLAB/Simulink model to solve the link length of the platform. Figure 3.5 is the Simulink model which solves the length of each leg vector in vector form.

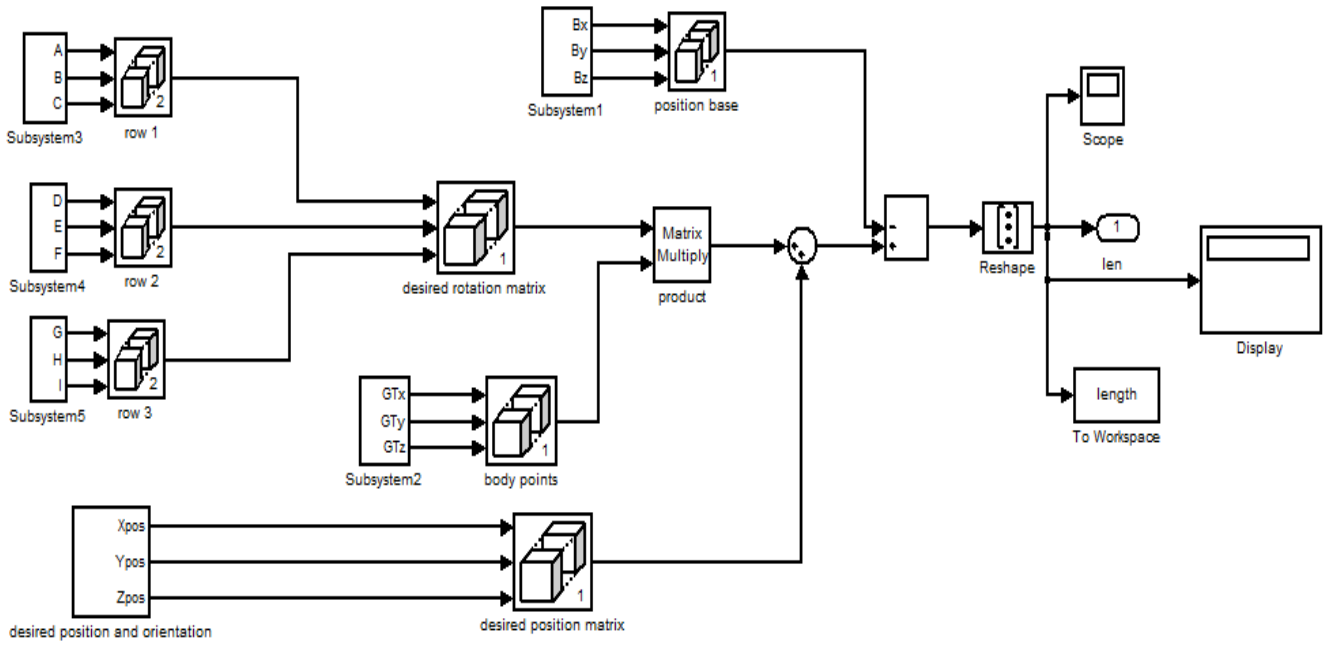


Figure 3.5: MATLAB/Simulink model of the inverse kinematics of the Stewart platform

3.3. Trajectory Tracking of the Stewart Platform

In order to move the robot along a specified path, a trajectory planning algorithm will be developed. Thus, it can be determined start and stop times of the motion besides the desired position and orientation inputs. Also, motors are synchronized with each other during the motion. Kane's transition function is used for trajectory planning algorithm of the Stewart platform.

$$Y(t) = y_0 + (y_f - y_0)\frac{t - t_0}{t_f - t_0} - \frac{y_f - y_0}{2\pi}\sin(2\pi\frac{t - t_0}{t_f - t_0}) \qquad (3.12)$$

Where y (t) is the position function, y_0 is the initial position y_f is the finish position; t is time, t_0 is initial time and t_f *is* the finish time. The length of the legs determined above can be used in this equation to get the desired trajectory. The full trajectory planning on MATLAB/Simulink using equation 3.12 have been done and shown below.

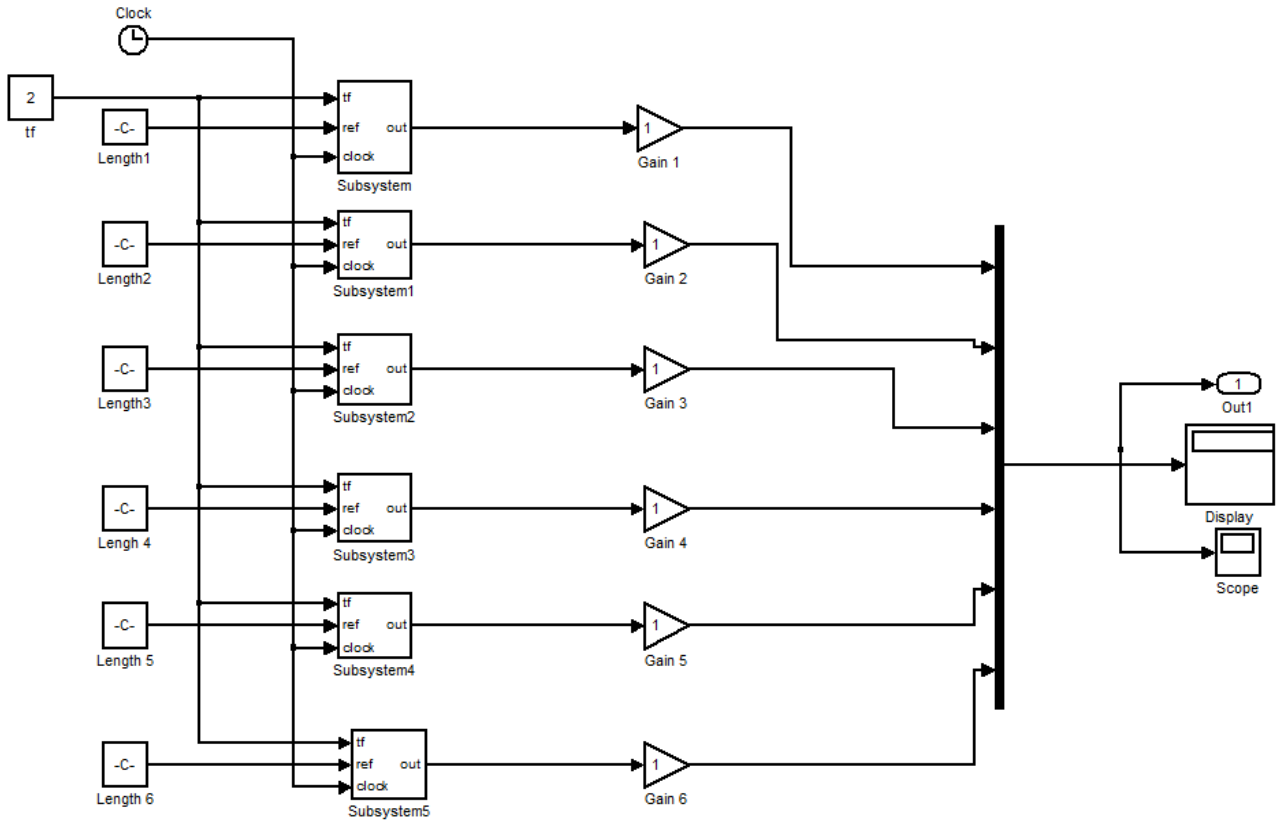


Figure 3.6: reference inputs with trajectory planning

There are six identical subsystems in figure 3.6 which are suited for generating of the desired trajectory by taking input arguments. The subsystems are shown in figure 3.7 below.

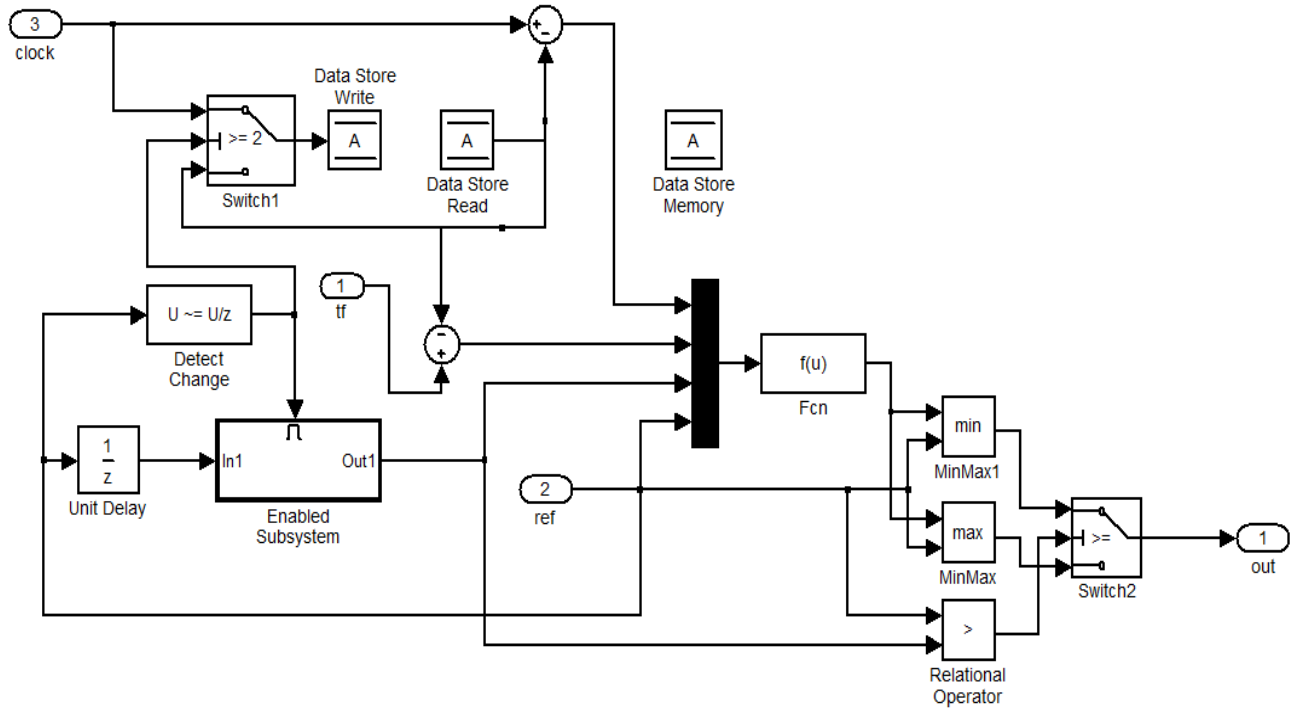


Figure 3.7: trajectory generation sub system

3.4. Jacobian Matrix of Stewart Platform

The Jacobean matrix relates the velocities of the active joints (actuators) to the generalized velocity of the moving platform.

Figure 3.8 shows a schematic view of one of the six legs of the SP manipulator

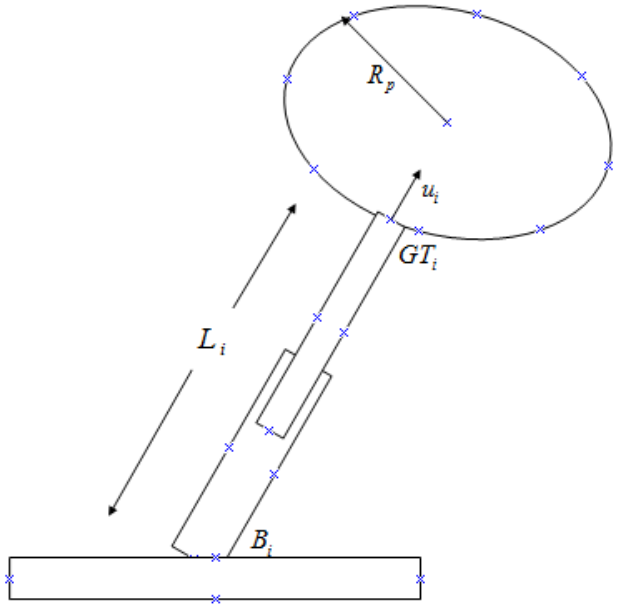


Figure 3.8: schematic view of the i^{th} leg of the parallel manipulator

For the parallel manipulators, the commonly used expression of the Jacobean matrix is given as the following.

$$\dot{L} = J\dot{X} \tag{3.13}$$

Where $\dot{L}$ and $\dot{X}$ are the velocities of the leg and the moving platform respectively.

We know that L is a function of the Cartesian coordinate (X_{p-o}).

$$L = F(X_{p-o}) \tag{3.14}$$

When we differentiate the length L with respect to time t, we will get the following expression.

$$\frac{dL}{dt} = \frac{\delta L}{\delta X_{p-o}} \frac{dX_{p-o}}{dt}$$

$$\equiv \dot{L} = J\dot{X}_{p-o} \tag{3.15}$$

Where, $J = \dfrac{\delta L}{\delta X_{p-o}}$

Equation 3.15 can be rewritten to see the relationship between the actuator velocities, $\dot{L}$ and the generalized velocity of the moving platform ($\dot{X}_{p-o}$) and the velocity of the moving platform connection point of the leg as the following.

$$\dot{L} = J\dot{X}_{P-O} = J_I \vec{V}_{TJ} \tag{3.16}$$

Where $\vec{V}_{\mathrm{TJ}}$ is the velocity of the platform connection point of the leg, and given by the following expression.

$$\vec{V}_{TJ} = J_{II}\dot{X}_{p-o} \tag{3.17}$$

Now, combining the above two equations, we will get,

$$\dot{L} = J\dot{X}_{p-o} = J_I J_{II}\dot{X}_{p-o} \tag{3.18}$$

The first Jacobean matrix in the above is given as:

$$\mathrm{J_I} = \begin{bmatrix} \overrightarrow{u_1}^T & .... & .... & .... & .... & 0 \\ 0 & \overrightarrow{u_2}^T & .... & .... & .... & 0 \\ 0 & 0 & \overrightarrow{u_3}^T & .... & .... & 0 \\ 0 & 0 & 0 & \overrightarrow{u_4}^T & .... & 0 \\ 0 & 0 & 0 & 0 & \overrightarrow{u_5}^T & 0 \\ 0 & 0 & 0 & 0 & 0 & \overrightarrow{u_6}^T \end{bmatrix} \tag{3.19}$$

Where, u_i is the unit vector along the axis of the prismatic joint of link i. It can be obtained as follows.

$$\vec{u}_i = \frac{B_i T_j}{|L_i|} = \frac{L_i}{l_i}, j = i \tag{3.20}$$

The second Jacobean matrix will be calculated as follows:

$\mathrm{J_{II}}$ (3.21)

$$= \begin{bmatrix} I_{3X3} & R_y(\beta)\mathrm{S(X)}R_x(\alpha)R_z(\gamma)GT_1 & \mathrm{S(Y)}R_y(\beta)R_x(\alpha)R_z(\gamma)GT_1 & R_y(\beta)R_x(\alpha)\mathrm{S(Z)}R_z(\gamma)GT_1 \\ & & . & \\ & & . & \\ & & . & \\ & & . & \\ I_{3X3} & R_y(\beta)\mathrm{S(X)}R_x(\alpha)R_z(\gamma)GT_6 & \mathrm{S(Y)}R_y(\beta)R_x(\alpha)R_z(\gamma)GT_6 & R_y(\beta)R_x(\alpha)\mathrm{S(Z)}R_z(\gamma)GT_6 \end{bmatrix}$$

Where I_{3x3} denotes the 3x3 identity matrix and S designates the 3x3 skew symmetric matrix associated with the vector $P_x = [P_x \ P_y \ P_z]^T$,

$$S = \begin{bmatrix} 0 & -P_z & P_y \\ P_z & 0 & -P_x \\ -P_y & P_x & 0 \end{bmatrix} \quad (3.22)$$

Therefore, the proposed Jacobean matrix of the Stewart platform manipulator will be defined as:

$$J = J_I J_{II} \quad (3.23)$$

3.5. Workspace Analysis

The workspace of the Stewart platform is defined as the range of allowable end effectors displacement, i.e., the region of three dimensional Cartesian space that can be attained by the end effectors with the given orientation of platform. It is determined by the scale and configuration of the mechanism, constrained by the kinematic limitations. The optimal design of Stewart platform is to choose a geometry for which the resulting workspace spans the desired range of motions. [20]

An important step during the design of a parallel manipulator is the determination of its workspace for a six degree of freedom. Parallel manipulator workspace limitations are due to the bounded range of their linear actuators, mechanical limits on their passive joints and links interference. The computation of the workspace of a parallel manipulator is far more complex than serial link manipulator as its translation ability is dependent up on the orientation of the end effector.

In most of the works dealing with workspace determination it is assumed that the orientation of the end effectors is fixed. In that case only the possible translations of the end effectors are to be determined. Therefore, it can be possible to determine the position of the moving platform of the Stewart platform by considering the orientation of the platform constant value. From Equation (3.10), we can extract the position vector shown as follows.

$$P = L_i - R_{xyz} GT_i + B_i, i = 1.....6 \quad (3.24)$$

Where, L_i is the length of i^{th} leg, which has maximum and minimum value. R_{xyz}, is the rotation matrix which depends on the orientation of the platform. GT_i and B_i are the actuator (leg) connecting points from the upper platform and the base respectively, and are constant values (geometry of the platform). The position calculated from the above formula will have upper and lower limit depend on the length of the actuator (L_{max} and L_{min}). For different values of orientation of the platform, different values of position P will be evaluated.

$$P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \tag{3.25}$$

3.6. Singularity Analysis

A complex kinematic chain consists of a set of rigid bodies connected to each other with joints. The chain is also characterized by a set joint coordinates, denoted here by an n-dimensional vector q, which corresponds to the powered joints and by a set of end effectors coordinates, denoted here by an m-dimensional vector x. Although the number of joint coordinates and end effecter coordinates does not have to be equal, the number of independent joint coordinates and end effecter coordinates will always be the same and therefore, vectors q and x can be reduced or augmented to vectors of the same dimension which will be equal to the degree of freedom of the linkage. [14]

The joint and end effecter coordinate rates are then related through the Jacobian matrix of the chain as previously shown in equation (3.13) and can be rewritten as:

$$\dot{q} = J\dot{x} \tag{3.26}$$

As opposed to the convention used for serial manipulators, the Jacobean matrix is defined here as the one mapping the end effecter coordinate rates in to the joint coordinate rates. The above equation can also be written as:

$$\dot{x} = K\dot{q} \tag{3.27}$$

Where, K =J^{-1}, and J and K being configuration dependent.

Singularities occur in configurations where J is rank deficient. For general complex kinematic chains, there are three types of singularities which have different physical interpretations.

(i) The first type of singularity occurs when $\det(k) = 0$ *and* $\det(J) \to \infty$. The corresponding configuration is one in which the chain reaches either a boundary of its workspace, or an internal boundary limiting different sub regions of the workspace where the number of solutions is not the same. In other words, this type of singularities consists of the set of points where different branches of the inverse kinematic problem meet.

(ii) The second type of singularity occurs when $\det(J) = 0$ and $\det(k) \to \infty$ tends to infinity. This corresponds to configurations in which the chain remains uncontrollable even when all the actuated joints are locked. As opposed to the first one, this types of singularity lies within the workspace of the chain and correspond a point or a set of points where different branches of the direct kinematic chain problem meet.

(iii) The third type of singularity is of a slightly different nature than the first two, since it requires conditions on the linkage parameters. Indeed if some specific conditions on the linkage parameters are satisfied, the chain can reach configurations where the first two types of singularities meet and the Jacobean matrix then becomes indeterminate.

3.7. Dynamics of the Stewart Platform

In this article, we study the dynamic equations of the Stewart platform manipulator. The dynamics of a robot manipulator using a set of nonlinear, second-order; ordinary differential equations which depend on the kinematic and inertial properties of the robot will be described. [1]

For control design purposes, it is necessary to have a mathematical model that reveals the dynamic behavior of a system. We will use a Lagrange analysis for our derivation, which relies on the energy properties of mechanical systems to compute the equations of motion. The resulting equations can be computed in closed form, allowing detailed analysis of the properties of the system.

3.7.1. Euler Lagrange System

An EL system is a system whose motion is described by the EL equations; they are a set of nonlinear ordinary differential equations with a certain specific structure. In modeling physical systems with lumped parameters two basic approaches have been typically used: derivation of the equations of motion using forces laws or application of variation principles to selected energy functions. For simple systems having only elements of the "same nature" the first approach is usually sufficient. For instance, for purely mechanical or electrical systems, Newton's second law and Kirchhoff's laws respectively, yield the desired equations. For systems having "mixed natures", variation approach will be used. The common link between the different subsystems is that all of them transform energy. Therefore, it seems natural to formulate the modeling problem in terms of energy quantities. [15]

The starting point of the variation approach to modeling is the definition of the energy functions in terms of sets of generalized variables. The equations of motion are then derived invoking well-known principles of analytical dynamics, in particular the fundamental Hamilton principle, which roughly speaking states that the system moves along trajectories that minimize the integral of the Lagrange. [15]

In this subsection, the EL formulation is particularly suited for passivity based control (PBC) since it underscores the role of the interconnections between systems and provides us with the storage and dissipation functions, which are the corner stone of the PBC design technique. Therefore, one will be forced to use EL method to derive the dynamic modeling of the SP manipulator. Considering q and τ as the corresponding generalized coordinates and generalized torques, respectively, the general classical equations of the motion can be obtained from the Lagrange formulation:

$$\frac{d}{dt}\frac{\delta L}{\delta \dot{q}} - \frac{\delta L}{\delta q} = \frac{d}{dt}\left(\frac{\delta K(q,\dot{q})}{\delta \dot{q}}\right) - \frac{\delta K(q,\dot{q})}{\delta q} + \frac{\delta P(q)}{\delta q} = \tau \tag{3.28}$$

Where, L is the Lagrange function and given by:

$$\mathrm{L} = \mathrm{K}(\mathrm{q},\dot{q}) - \mathrm{P}(\mathrm{q}) \tag{3.29}$$

Where, $\mathrm{K}(\mathrm{q},\dot{q})$ is the kinetic energy and P(q) is the potential energy

For n- link manipulator, Equation 3.28 can be rewritten as:

$$M(q)\ddot{q}+V(q,\dot{q})\dot{q}+G(q)=\tau \tag{3.30}$$

Where, M (q) is inertia matrix of the manipulator, V (q,$\dot{q}$) is the coriolis/centrifugal vector, G (q) is the gravity matrix of the manipulator, q is the generalized coordinate and τ is the actuator force.

Let the Generalized coordinates q is replaced with Cartesian coordinates X_{p-o}, then Equation (3.30) will be written as:

$$M(X_{p-o})\ddot{X}_{P-O}+V(X_{P-O},\dot{X}_{p-o})\dot{X}_{p-o}+G(X_{p-o})=J^{T}(X_{p-o})\tau \tag{3.31}$$

Where, $\tau = [\tau_1\ \tau_2\ \tau_3\ \tau_4\ \tau_5\ \tau_6]$ are the torques applied by the actuator in leg i.

In order to derive the dynamic equations of the SP manipulator, the whole system is separated into two parts: the moving platform and the legs.

3.7.2. Kinetic and Potential Energies of the Moving/Upper Platform

The characteristics of the upper platform are mass M_{up}, tensor of inertia in the moving frame $I_{(mf)}$, angular velocity in the fixed frame $\Omega_{up(ff)}$ and angular velocity in the moving frame $\Omega_{up(mf)}$. The kinetic energy of the moving platform is a summation of two motion energies since the Moving platform has translation and rotation about three orthogonal axes, (*xyz*)

i. The first One is translation energy occurring because of the translation motion of the center of mass of the moving platform, The translation energy is defined by:

$$k_{up(trans)}=\frac{1}{2}m_{up}(\dot{p}_x^{\,2}+\dot{p}_y^{\,2}+\dot{p}_z^{\,2}) \tag{3.32}$$

Where, m_{up} is the moving/upper platform mass and $[p_x\ \ p_y\ \ p_z]$ are position vectors.

ii. The second one is the energy due to rotational motion of the moving platform around its center of mass, rotational kinetic energy can be written as:

$$k_{up(rot)} = \frac{1}{2}\vec{\Omega}^{T}{}_{up(mf)} I_{(mf)} \vec{\Omega}^{T}{}_{up(mf)} \tag{3.33}$$

Where, $I_{(mf)}$ is the rotational inertia mass, And $\Omega_{up(mf)}$ is the angular velocity of the moving platform and they are given as:

$$I_{(mf)} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \tag{3.34}$$

$$\vec{\Omega}_{up(mf)} = R_z(\gamma)^T R_x(\alpha)^T R_y(\beta)^T \vec{\Omega}_{up(ff)} \tag{3.35}$$

Where,

$\Omega_{up(ff)}$ Denotes the angular velocity of the moving platform with respect to the fixed base Frame, Given the definition of the angles α, β and γ, it is expressed as follows. [15]

$$\vec{\Omega}_{up(ff)} = \dot{\alpha}R_y(\beta)\vec{X} + \dot{\beta}\vec{Y} + \dot{\gamma}R_x(\alpha)R_Z(\gamma)\vec{Z} = \left(\begin{bmatrix} \mathrm{c}(\beta) & 0 & \mathrm{s}(\beta) \\ 0 & 1 & 0 \\ -\mathrm{s}(\beta) & 0 & \mathrm{c}(\beta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{c}(\alpha) & -\mathrm{s}(\alpha) \\ 0 & \mathrm{s}(\alpha) & \mathrm{c}(\alpha) \end{bmatrix} \begin{bmatrix} \mathrm{c}(\gamma) & -\mathrm{s}(\gamma) & 0 \\ \mathrm{s}(\gamma) & \mathrm{c}(\gamma) & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix}$$

Simplifying the above expression yields:

$$\Omega_{up(ff)} = \begin{bmatrix} \mathrm{c}(\beta) & 0 & 0 \\ 0 & 1 & -\mathrm{s}(\alpha) \\ -\mathrm{s}(\beta) & 0 & \mathrm{c}(\alpha) \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} \tag{3.36}$$

In the moving platform coordinate system, the angular velocity of the moving platform is calculated as:

$$\vec{\Omega}_{up(mf)} = \begin{bmatrix} c\gamma & c\alpha s\gamma & -c\alpha c\gamma s\beta - c\alpha s\alpha s\gamma + c\alpha c\beta s\alpha s\gamma \\ -s\gamma & c\alpha c\gamma & -c\alpha c\gamma s\alpha + c\alpha s\beta s\gamma + c\alpha c\beta s\alpha c\gamma \\ 0 & -s\alpha & s^2\alpha + c^2\alpha c\beta \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} \tag{3.37}$$

Then we can obtain;

$$k_{up(rot)} = \frac{1}{2}\begin{bmatrix}\dot{\alpha}\\ \dot{\beta}\\ \dot{\gamma}\end{bmatrix}^T \begin{bmatrix} I_xC^2\gamma + I_yS^2\gamma & (I_x - I_y)C\alpha C\gamma S\gamma & 0 \\ (I_x - I_y)C\alpha C\gamma S\gamma & C^2\alpha(I_xS^2\gamma + I_yC^2\gamma) + I_zS^2\alpha & -I_zS^2\alpha \\ 0 & -I_zS^2\alpha & I_z \end{bmatrix}\begin{bmatrix}\dot{\alpha}\\ \dot{\beta}\\ \dot{\gamma}\end{bmatrix} \tag{3.38}$$

As a result, the total kinetic energy of the moving platform in a compact form is given by:

$$k_{up} = k_{up(trans)} + k_{up(rot)} = \frac{1}{2}m_{up}(\dot{p}_x{}^2 + \dot{p}_y{}^2 + \dot{p}_z{}^2) + \frac{1}{2}\vec{\Omega}^T{}_{up(mf)} I_{(mf)} \vec{\Omega}_{up(mf)}$$

$$= \frac{1}{2}\dot{X}^T{}_{p-o} m_{up}(X_{p-o})\dot{X}_{p-o} = \frac{1}{2}[\dot{P}_x \quad \dot{P}_y \quad \dot{P}_z \quad \dot{\alpha} \quad \dot{\beta} \quad \dot{\gamma}]\, \mathrm{m_{up}} \begin{bmatrix}\dot{P}_x\\ \dot{P}_y\\ \dot{P}_z\\ \dot{\alpha}\\ \dot{\beta}\\ \dot{\gamma}\end{bmatrix}$$

Or

$$k_{up} = \frac{1}{2}\begin{bmatrix}\dot{P}_x\\ \dot{P}_y\\ \dot{P}_z\\ \dot{\alpha}\\ \dot{\beta}\\ \dot{\gamma}\end{bmatrix}^T \begin{bmatrix} m_{up} & 0 & 0 & 0 & 0 & 0 \\ 0 & m_{up} & 0 & 0 & 0 & 0 \\ 0 & 0 & m_{up} & 0 & 0 & 0 \\ 0 & 0 & 0 & I_xC^2\gamma + I_yS^2\gamma & (I_x - I_y)C\alpha C\gamma S\gamma & 0 \\ 0 & 0 & 0 & (I_x - I_y)C\alpha C\gamma S\gamma & C^2\alpha(I_xS^2\gamma + I_yC^2\gamma) + I_zS^2\alpha & -I_zS^2\alpha \\ 0 & 0 & 0 & 0 & -I_zS^2\alpha & I_z \end{bmatrix}\begin{bmatrix}\dot{P}_x\\ \dot{P}_y\\ \dot{P}_z\\ \dot{\alpha}\\ \dot{\beta}\\ \dot{\gamma}\end{bmatrix} \tag{3.39}$$

Where, M_{up} is the 6x6 mass diagonal matrix of the moving platform, and expressed as:

$$m_{up} = \begin{bmatrix} m_{up} & 0 & 0 & 0 & 0 & 0 \\ 0 & m_{up} & 0 & 0 & 0 & 0 \\ 0 & 0 & m_{up} & 0 & 0 & 0 \\ 0 & 0 & 0 & I_xC^2\gamma + I_yS^2\gamma & (I_x - I_y)C\alpha C\gamma S\gamma & 0 \\ 0 & 0 & 0 & (I_x - I_y)C\alpha C\gamma S\gamma & C^2\alpha(I_xS^2\gamma + I_yC^2\gamma) + I_zS^2\alpha & -I_zS^2\alpha \\ 0 & 0 & 0 & 0 & -I_zS^2\alpha & I_z \end{bmatrix} \tag{3.40}$$

Also, potential energy of the moving platform is:

$$p_{up} = m_{up} g p_Z$$

Or

$$p_{up} = [0 \quad 0 \quad m_{up} g \quad 0 \quad 0 \quad 0] \begin{bmatrix} p_x \\ p_y \\ p_z \\ \alpha \\ \beta \\ \gamma \end{bmatrix} \tag{3.41}$$

3.7.3. Kinetic and Potential Energies of the Legs

A precise study of the dynamics of the legs should require decomposition into two parts: the fixed part (to the base) and the moving part. And we should consider them as two bodies with their own inertia. However, this will lead to a complex formulation. As a simplified yet still accurate model, we will consider that each leg can be represented by a moving point (its center of mass G_i) and assume that the mass is concentrated there. This takes into account the motion of the center of gravity due to the change in leg actuator length. [3]

The lower fixed part of the leg is connected to the base platform through a spherical joint, whereas the upper moving part is connected to the moving platform through a spherical joint. As shown in figure 3.9, the center of mass G_i for each part of the leg is considered. G_{1i}, Denotes the center of mass of the fixed part. l_1 and m_1 are the length and the mass of the fixed part respectively and δ is the distance between B_i and G_i for the moving part of the leg, G_{2i} denotes the center of mass. l_2 And m_2 are the length and the mass of the moving part respectively.

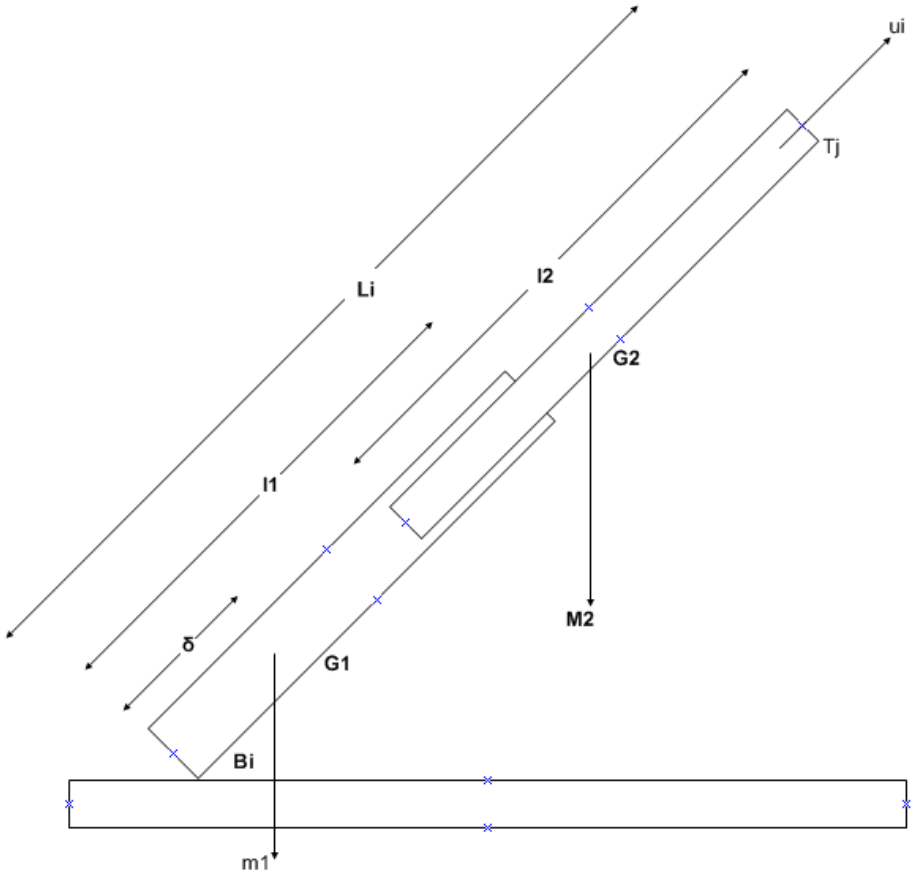


Figure 3.9: leg of the Stewart platform manipulator

To obtain the kinetic and potential energies of a leg ($leg_i, i = 1.....6$), we need to determine the position and velocity of its center of mass G_i. We will consider that each leg can be represented by a moving point (its center of mass G_i).

The position,

$$\vec{B}_i\vec{G}_i = \frac{1}{m_1+m_2}\left[\left(\delta m_1 l_1 - \frac{1}{2}m_2 l_2\right) + m_2 L_i \vec{u}_i\right] \equiv \left[\hat{l} + \frac{m_2}{m_1+m_2}L_i\right]\vec{u}_i \qquad (3.42)$$

Where, $\hat{l} = \dfrac{\delta m_1 l_1 - \left(\frac{1}{2}\right)m_2 l_2}{m_1+m_2}$

The velocity V_{Gi}, of the center of mass G_i is then;

$$\vec{V}_{Gi} = \frac{d\vec{B}_i\vec{G}_i}{dt} = \hat{l}\,\frac{d\vec{u}_i}{dt} + \frac{m_2}{m_1+m_2}\frac{d\vec{B}_i\vec{T}_j}{dt},\ j = i \tag{3.43}$$

Note that;

$$\frac{d\vec{u}_i}{dt} = \frac{d\left(\frac{\vec{B}_i\vec{T}_j}{L_i}\right)}{dt} \equiv -\frac{1}{{L_i}^2}\frac{dL_i}{dt}\vec{B}_i\vec{T}_j + \frac{1}{L_i}\vec{V}_{Tj} \equiv -\frac{1}{L_i}\frac{dL_i}{dt}\vec{u}_i + \frac{1}{L_i}\vec{V}_{Tj} \equiv -\frac{1}{L_i}\left(\vec{V}_{Tj}.\vec{u}_i\right)\vec{u}_i + \frac{1}{L_i}\vec{V}_{Tj}$$

Therefore, equation 3.43 can be rewritten as:

$$\vec{V}_{Gi} = \frac{\hat{l}}{L_i}\left[\vec{V}_{Tj} - \left(\vec{V}_{Tj}.\vec{u}_i\right)\vec{u}_i\right] + \frac{m_2}{m_1+m_2}\vec{V}_{Tj} \tag{3.44}$$

Kinetic energy of the legs may be expressed as;

$$k_{Li} = \frac{1}{2}(m_1+m_2){\vec{V}_{Gi}}^T\vec{V}_{Gi}$$

$$= \frac{1}{2}(m_1+m_2)\left[\begin{array}{l}\left(\frac{\hat{l}^2}{L_i}\right)\left(\vec{V}_{Tj} - \left(\vec{V}_{Tj}.\vec{u}_i\right)\vec{u}_i\right)^T\left(\vec{V}_{Tj} - \left(\vec{V}_{Tj}.\vec{u}_i\right)\vec{u}_i\right) + \left(\frac{m_2}{m_1+m_2}\right)^2{\vec{V}_{Tj}}^T\vec{V}_{Tj} \\ + 2\frac{\hat{l}}{L_i}\left(\frac{m_2}{m_1+m_2}\right)\left(\vec{V}_{Tj} - \left(\vec{V}_{Tj}.\vec{u}_i\right)\vec{u}\right)^T\vec{V}_{Tj}\end{array}\right]$$

Note that;

$$\left(\vec{V}_{Tj} - \left(\vec{V}_{Tj}.\vec{u}_i\right)\vec{u}_i\right)^T\left(\vec{V}_{Tj} - \left(\vec{V}_{Tj}.\vec{u}_i\right)\vec{u}_i\right) = 0$$

Thus, after some calculations one obtains

$$k_{Li} = \frac{1}{2}(m_1+m_2)\left[\left(\frac{\hat{l}}{L_i} + \frac{m_2}{m_1+m_2}\right)^2{\vec{V}_{Ti}}^T\vec{V}_{Tj}\left(\frac{\hat{l}}{L_i}\right)\left(\frac{\hat{l}}{L_i} + \frac{m_2}{m_1+m_2}\right){\vec{V}_{Tj}}^T(\vec{u}_i){\vec{u}_i}^T\vec{V}_{Tj}\right] \tag{3.45}$$

Let us denote;

$$\begin{Bmatrix} h_i = \left(\frac{\hat{l}}{L_i} + \frac{m_2}{m_1 + m_2} \right)^2 \\ k_i = \frac{\hat{l}}{L_i} \left(\frac{\hat{l}}{L_i} + \frac{m_2}{m_1 + m_2} \right) = h_i - \left(\frac{m_2}{m_1 + m_2} \right)^2 \end{Bmatrix}$$

Therefore, equation 3.45 can be rewritten as:

$$k_{Li} = \frac{1}{2}(m_1 + m_2) \left[h_i \left(\vec{V}_{Tj}{}^T \vec{V}_{Tj} - \vec{V}_{Tj}{}^T (\vec{u}_i)(\vec{u}_i)^T \vec{V}_{Tj} \right) + \left(\frac{m_2}{m_1 + m_2} \right)^2 \vec{V}_{Tj}{}^T (\vec{u}_i)(\vec{u}_i)^T \vec{V}_{Tj} \right]$$

Or

$$k_{Li} = \frac{1}{2}(m_1 + m_2) \left[h_i \vec{V}_{Tj}{}^T \vec{V}_{Tj} - k_i \vec{V}_{Tj}{}^T (\vec{u}_i)(\vec{u}_i)^T \vec{V}_{Tj} \right], j = i \tag{3.46}$$

u_i is the vector that gives the direction of the "leg_i". Thus, if V_{Tj} is the velocity of the end of the leg, $V_{Tj}u_i$ corresponds to the speed of the elongation of the leg, that is $V_{Tj}u_i = dL_i / dt$. As a consequence, k_{Li} can be divided into two parts with a physical meaning as follows:

i. The kinetic energy due to the rotation around the fixed point B of the leg (if one supposes that the length of the leg is constant) is given by:

$$k_{Li(rot)} = \frac{1}{2}(m_1 + m_2) \left[h_i \left(\vec{V}_{Tj}{}^T \vec{V}_{Tj} - \vec{V}_{Tj}{}^T (\vec{u}_i)(\vec{u}_i)^T \vec{V}_{Tj} \right) \right], j = i \tag{3.47}$$

ii. The kinetic energy due to the elongation (or the translation motion) of the leg (if one supposes that the direction of the leg is constant) is given by:

$$k_{Li(trans)} = \frac{1}{2}(m_1 + m_2) \left[\left(\frac{m_2}{m_1 + m_2} \right)^2 \vec{V}_{Tj}{}^T (\vec{u}_i)(\vec{u}_i)^T \vec{V}_{Tj} \right], j = i \tag{3.48}$$

Let us now give a compact expression for the kinetic energy of the six legs.

$$k_{Li} = k_{Li(rot)} + k_{Li(trans)} = \frac{1}{2}(m_1 + m_2) \left[\vec{V}_{Tj}{}^T H_i \vec{V}_{Tj} - \dot{L}_i{}^T k_i \dot{L}_i \right]$$

Where, H_i is $6x6$ diagonal matrix of diagonal entry h_i.

The total kinetic energy of the six legs k_L can be expressed as:

$$k_L = \sum_{1}^{6} k_{Li} \equiv \frac{1}{2}(m_1 + m_2)\left[\begin{bmatrix}\vec{V}_{T1}\\ \vec{V}_{T2}\\ \vec{V}_{T3}\\ \vec{V}_{T4}\\ \vec{V}_{T5}\\ \vec{V}_{T6}\end{bmatrix}^T H \begin{bmatrix}\vec{V}_{T1}\\ \vec{V}_{T2}\\ \vec{V}_{T3}\\ \vec{V}_{T4}\\ \vec{V}_{T5}\\ \vec{V}_{T6}\end{bmatrix} - \begin{bmatrix}\dot{L}_1\\ \dot{L}_2\\ \dot{L}_3\\ \dot{L}_4\\ \dot{L}_5\\ \dot{L}_6\end{bmatrix}^T K \begin{bmatrix}\dot{L}_1\\ \dot{L}_2\\ \dot{L}_3\\ \dot{L}_4\\ \dot{L}_5\\ \dot{L}_6\end{bmatrix}\right] \tag{3.49}$$

Where,

$$H = \begin{bmatrix} h_1+h_2 & 0 & 0 & & & & & & & & & \\ 0 & h_1+h_2 & 0 & & & & & & & & & \\ 0 & 0 & h_1+h_2 & & & & & & 0 & & & \\ & & & \ddots & & & & & & & & \\ & & & & h_3+h_4 & 0 & 0 & & & & & \\ & & & & 0 & h_3+h_4 & 0 & & & & & \\ & & & & 0 & 0 & h_3+h_4 & & & & & \\ & & & & & & & \ddots & & & & \\ & & & & & & & & h_5+h_6 & 0 & 0 & \\ & & 0 & & & & & & 0 & h_5+h_6 & 0 & \\ & & & & & & & & 0 & 0 & h_5+h_6 & \\ & & & & & & & & & & & \ddots \end{bmatrix} \tag{3.50}$$

And

$$K = \begin{bmatrix} K_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & K_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & K_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & K_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & K_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & K_6 \end{bmatrix} \tag{3.51}$$

Let us now introduce the Jacobeans J_1 and J_2. One can write

$$\begin{bmatrix} \dot{L}_1 \\ \dot{L}_2 \\ \dot{L}_3 \\ \dot{L}_4 \\ \dot{L}_5 \\ \dot{L}_6 \end{bmatrix} = J_1 \begin{bmatrix} \vec{V}_{T1} \\ \vec{V}_{T2} \\ \vec{V}_{T3} \\ \vec{V}_{T4} \\ \vec{V}_{T5} \\ \vec{V}_{T6} \end{bmatrix} \equiv J_1 J_2 \begin{bmatrix} \dot{P}_x \\ \dot{p}_y \\ \dot{p}_z \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix}$$

Thus, one finally obtains:

$$k_L = \frac{1}{2} \dot{X}_{p-o}{}^T M_{legs}(X_{p-o}) \dot{X}_{p-o} \tag{3.52}$$

Where,

$$M_{legs} = (m_1 + m_2)\left[J_2{}^T \left[H - J_1{}^T K J_1\right] J_2\right] \tag{3.53}$$

This implies, total inertia matrix,

$$M = M_{up} + M_{legs} \tag{3.54}$$

Potential energy of the legs P_{legs} can be derived in a similar fashion.

$$p_{legs} = (m_1 + m_2) g \sum_1^6 \vec{B}_i \vec{G}_i . \vec{Z}$$

$$\equiv (m_1 + m_2) g \sum_1^6 \left[\hat{l} + \frac{m_2}{m_1 + m_2} L_i \right] \left(\vec{u}_i \vec{Z}\right) \tag{3.55}$$

From equation 3.55, $\vec{u}_i = \left(\frac{\vec{B}_i \vec{T}_j}{L_i} \right), \vec{u}_i . \vec{Z} = \frac{Z_{Tj}}{L_i}$

Where, $Z_{Tj} = p_z + \vec{Z} . \vec{G}_{Tj(ff)}$

One can finally write,

$$p_{legs} = (m_1 + m_2) g \sum_1^3 \left[\hat{l} \left(\frac{1}{L_{2i}} + \frac{1}{L_{2i-1}} \right) + \frac{2m_2}{m_1 + m_2} \right] \left(p_Z + Z_{Tj}\right) \tag{3.56}$$

Where,

$$Z_{Tj} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}^T R_Z(\gamma)^T R_x(\alpha)^T R_y(\beta)^T G_{Tj} \tag{3.57}$$

Where, G_{Tj} is the attachment point of the leg with the upper plate (moving platform) and is dependent on the geometry of the platform.

Once the total kinetic energy and potential energy of Stewart platform is derived, the dynamics can be obtained from Lagrange equation stated in equation 3.28. Therefore, equation 3.31 is the desired dynamics equation for the Stewart platform. We have to calculate the terms $M(X_{p-o}), V(X_{p-o}, \dot{X}_{p-o})$ and $G(X_{p-o})$.

More precisely,

$$\begin{aligned} M(X_{p-o}) &= M_{up} + M_{legs} \\ V(X_{p-o}, \dot{X}_{p-o}) &= V_{up} + V_{legs} \\ G(X_{p-o}) &= G_{up} + G_{legs} \end{aligned} \tag{3.58}$$

From the expression of the kinetic energies of the platform and the legs, we directly have an expression of the coefficient $M(X_{p-o})$. From equation 3.40, we have,

$$M_{up} = \begin{bmatrix} M_u & 0 & 0 & 0 & 0 & 0 \\ 0 & M_u & 0 & 0 & 0 & 0 \\ 0 & 0 & M_u & 0 & 0 & 0 \\ 0 & 0 & 0 & I_x C^2\gamma + I_y S^2\gamma & (I_x - I_y) C\alpha C\gamma S\gamma & 0 \\ 0 & 0 & 0 & (I_x - I_y) C\alpha C\gamma S\gamma & C^2\alpha (I_x S^2\gamma + I_y C^2\gamma) + I_z S^2\alpha & -I_z S^2\alpha \\ 0 & 0 & 0 & 0 & -I_z S^2\alpha & I_z \end{bmatrix}$$

And also, from equation 3.53, we know,

$$M_{legs} = (m_1 + m_2)\left[J_2{}^T \left[H - J_1{}^T K J_1\right] J_2\right]$$

The compact expression of the total inertia matrix of the Stewart platform has been given in equation 3.54.

The gravity matrix can be derived from the potential energy of the platform. The upper platform and the legs gravity matrix are presented separately.

$$\left\{ \begin{aligned} G_{up} &= \frac{\delta p_{up}(X_{p-o})}{\delta X_{p-o}} \\ &\equiv \begin{bmatrix} 0 \\ 0 \\ M_{up} g \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned} \right\} \tag{3.59}$$

Similarly,

$$\begin{aligned} G_{legs}(k) &= \frac{\delta p_{legs}(X_{p-o})}{\delta X_{p-o}(k)} \\ &\equiv (m_1 + m_2) g \sum_{1}^{3} \left[\hat{l} \left[\frac{1}{{L_{2i}}^2} \frac{\delta L_{2i}}{\delta X_{p-o}(k)} + \frac{1}{{L_{2i-1}}^2} \frac{\delta L_{2i-1}}{\delta X_{p-o}(k)} \right] \right] (p_z + Z_{Ti}) \\ &\quad + (m_1 + m_2) g \sum_{1}^{3} \left[\hat{l} \left[\frac{1}{L_{2i}} + \frac{1}{L_{2i-1}} \right] + \frac{2m_2}{m_1 + m_2} \right] \frac{\delta (p_Z + Z_{Tj})}{\delta X_{p-o}(k)} \end{aligned} \tag{3.60}$$

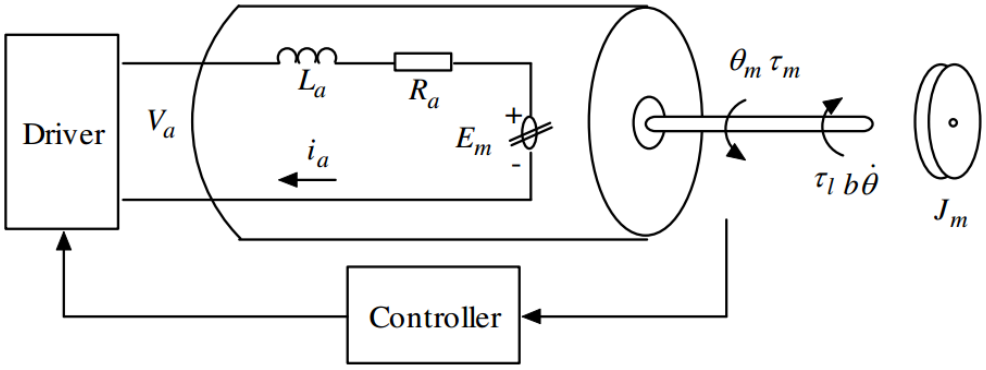
Where,

$$Z_{Tj} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}^T \left[{R_{Z(\gamma)}}^T {R_{x(\alpha)}}^T {R_{y(\beta)}}^T G_{Tj(mf)} \right]$$

Note that the Coriolis/centrifugal term can be neglected for slow motion system. Therefore we can neglect this term in the dynamics of the Stewart platform. And the full attention will be on the inertia matrix and gravity matrix.

3.8. Dynamics of Actuators

The leg system of the Stewart platform is composed of dc motor, precision linear bearing & ball screw and coupling elements. The DC motor model is given below.



. Figure 3.10: Circuit diagram of the DC motor

The symbols represent the following variables; θ_m is the rotor position (radian), τ_m is the produced torque by the motor (Nm), τ_l is the load torque, V_a is the armature voltage (V), L_a is the armature inductance (H), R_a is the armature resistance (Ω), E_m is the reverse EMF (V), k_b is the reverse EMF constant, k_T is torque constant. Let the DC motor be modeled in both transfer function and Euler Lagrange model.

i. Transfer function model

From figure 3.10, one can write the voltage loop equation as follows.

$$\left\{ \begin{array}{l} L_a \dfrac{di_a}{dt} + R_a I_a = V_a - E_m \\ where, E_m = k_b \omega \end{array} \right\} \tag{3.61}$$

Take the Laplace transform of the above equation and we reach at;

$$I_a = \frac{V_a - k_b \omega}{R_a + L_a s} \tag{3.62}$$

And, the motor torque equation may be expressed as follows;

$$J_m \frac{d\omega}{dt} + b_m \omega + \tau_L = \tau_m \tag{3.63}$$

Where,

$$\tau_m = k_T I_a \tag{3.64}$$

Take the Laplace transform of equation 3.63 and write the angular speed of the motor in terms of the other quantities;

$$\begin{Bmatrix} J_m s\omega + b_m \omega + \tau_L = k_T I \\ \omega = \dfrac{k_T I - \tau_L}{Js + b_m} \end{Bmatrix} \tag{3.65}$$

By using equation 3.62 and equation 3.65, one can draw the block diagram shown below.

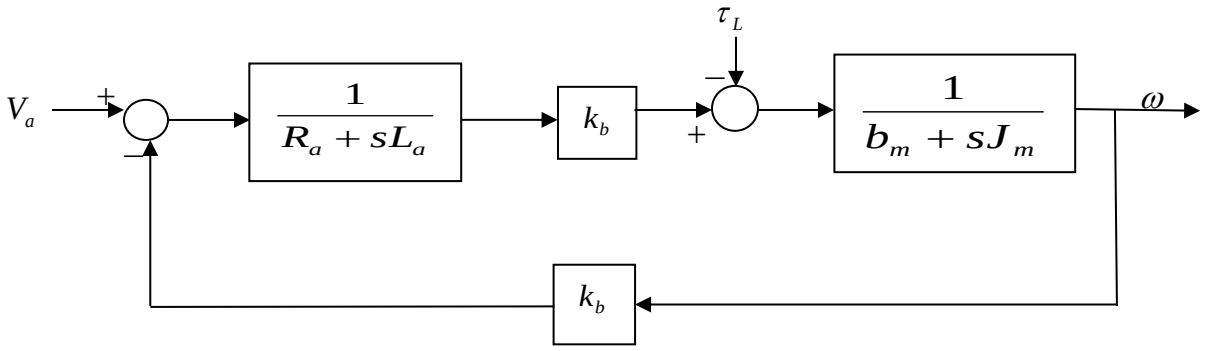


Figure 3.11: open loop speed control of DC motor

ii. Euler Lagrange model

The DC motor can be also modeled using Euler Lagrange formulation. Once its mathematical model is derived, it is possible to integrate with the Stewart platform dynamics. The leg length is proportional to the angular displacement of the motor by the constant $\frac{p}{2\pi n}$. Therefore, the angular velocity of the motor is given as;

$$\omega = \frac{2\pi n}{p} \dot{L} \tag{3.66}$$

Where, L is the length of the leg.

The actuator dynamics can be written in matrix form as;

$$\tau_m = M_a \ddot{L} + N_a \dot{L} + K_a F \tag{3.67}$$

With;

$$\left\{ \begin{array}{l} M_a = \dfrac{2\pi}{np}(J_s + n^2 J_m) I_{6x6} \\ N_a = \dfrac{2\pi}{np}(b_s + n^2 b_m) I_{6x6} \\ K_a = \dfrac{p}{n2\pi} I_{6x6} \end{array} \right\} \tag{3.68}$$

Where M_a, N_a and K_a are the inertia matrix, viscous damping coefficient matrix and gain matrix of the actuator, respectively. Also, J_s and J_m are the mass moment of inertia of the ball screw and motor, b_s and b_m are the viscous damping coefficient of the ball screw and motor, p and n are the pitch of the ball screw and the gear ratio. τ_m And F are the vectors of motor torques and the forces applied by the actuators.

Since the dynamics of the platform is derived in the moving platform coordinates (Cartesian Space, X_{p-o}), Equation 3.68 should be expressed in Cartesian space and the combined dynamics may be presented as follows.

$$\tau_m = \overline{M}(X_{p-o})\ddot{X}_{p-o} + \overline{N}(X_{p-o}, \dot{X}_{p-o})\dot{X}_{p-o} + \overline{G}(X_{p-o}) \tag{3.69}$$

The terms in the equation above are obtained from joint space terms and the Jacobean matrix.

$$\left\{ \begin{array}{l} \overline{M}(X_{p-o}) = K_a J^{-T} M(X_{p-o}) + M_a J \\ \overline{N}(X_{p-o}, \dot{X}_{p-o}) = K_a J^{-T} N(X_{P-O}, \dot{X}_{p-o}) + N_a J + M_a \dot{J} \\ \overline{G}(X_{p-o}) = K_a J^{-T} G(X_{p-o}) \end{array} \right\} \tag{3.70}$$

Once inertia matrix, coriolis/centrifugal matrix and gravity matrix, are determined, it can be possible to have a model which shall be ready for computer simulation. The next task will be modeling the dynamics of the Stewart platform using MATLAB/Simulink.

By combining equation 3.62, 3.66 and 3.69, one can develop the open loop block diagram which may be used in MATLAB/Simulink modeling for simulation purpose in performance analysis of the system.

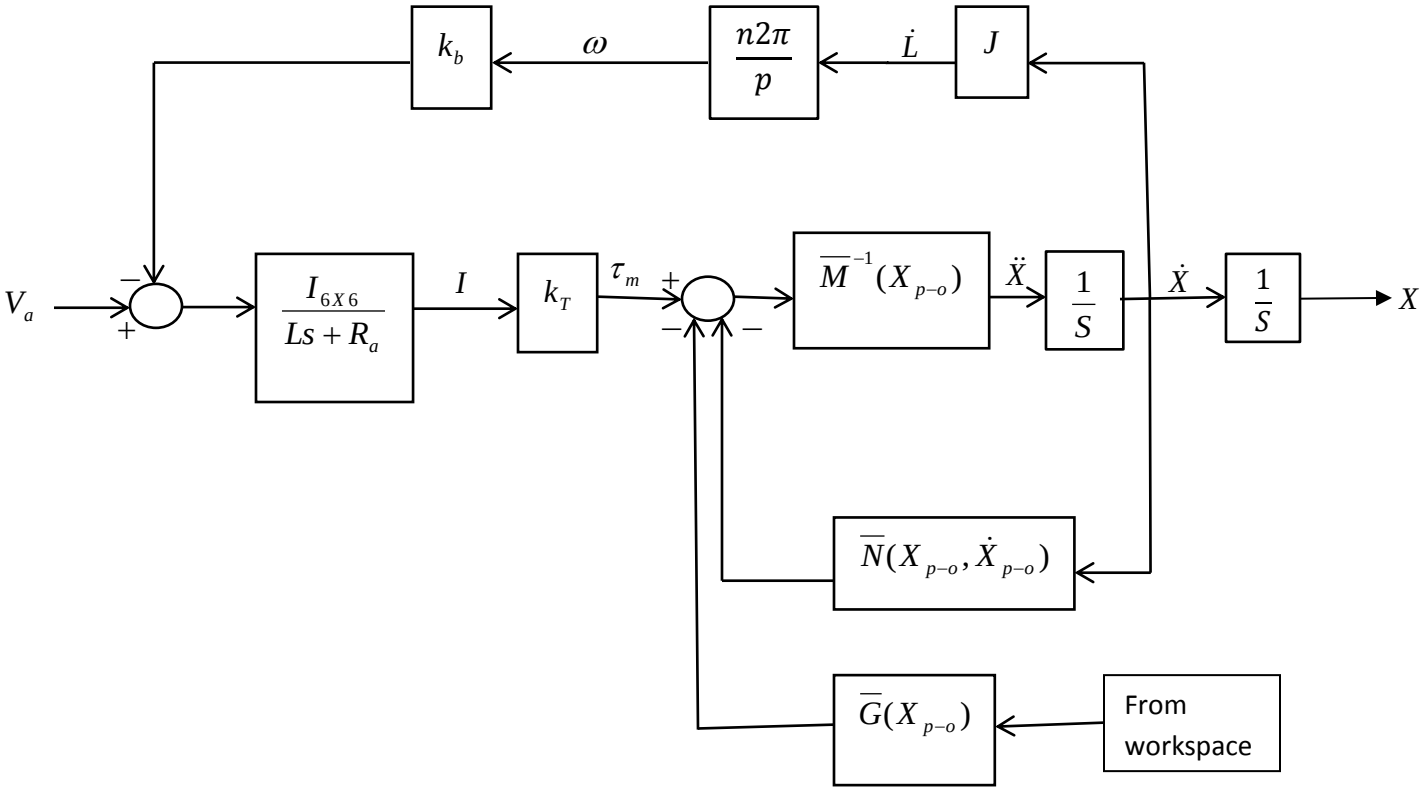


Figure 3.12: complete dynamics block diagram of Stewart platform.

The open loop system of the Stewart platform with actuator dynamics is modeled on MATLAB/Simulink with the desired trajectory of legs as a reference input which has been evaluated from the inverse kinematics of the previous section. The following diagram shows the complete dynamics of the Stewart platform with reference leg length trajectory block.

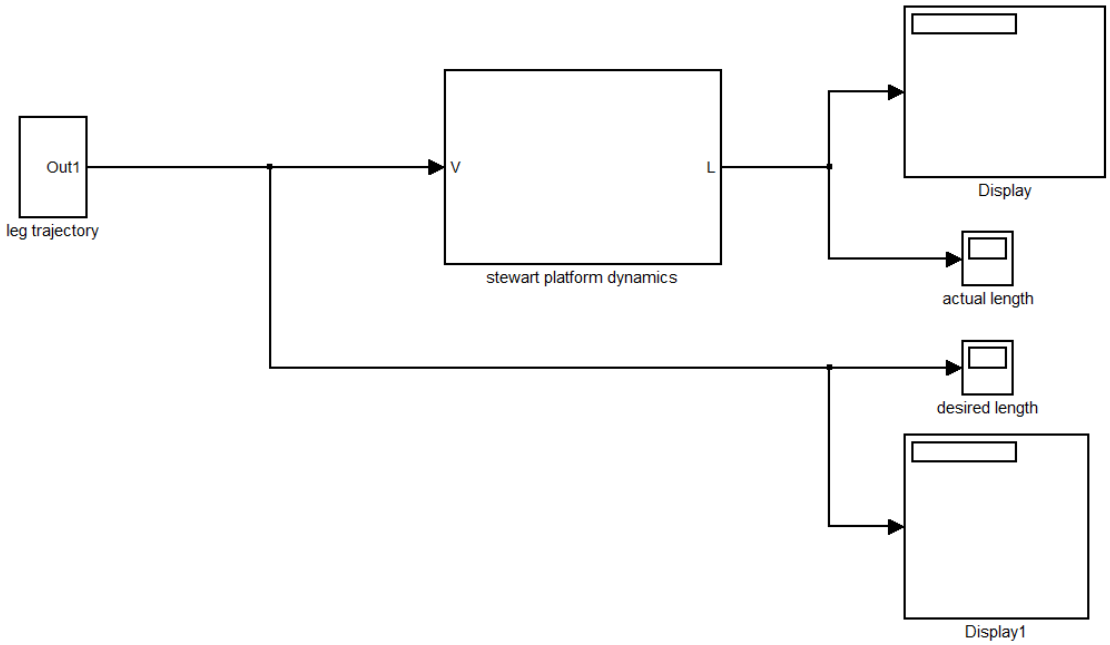


Figure 3.13: open loop MATLAB/Simulink model of the Stewart platform

Figure 3.13 comprises of two subsystems, dynamics of the Stewart platform and reference leg trajectory. The reference leg trajectory has been illustrated and shown on the kinematics of the Stewart platform section. The dynamics of the Stewart platform is shown as below.

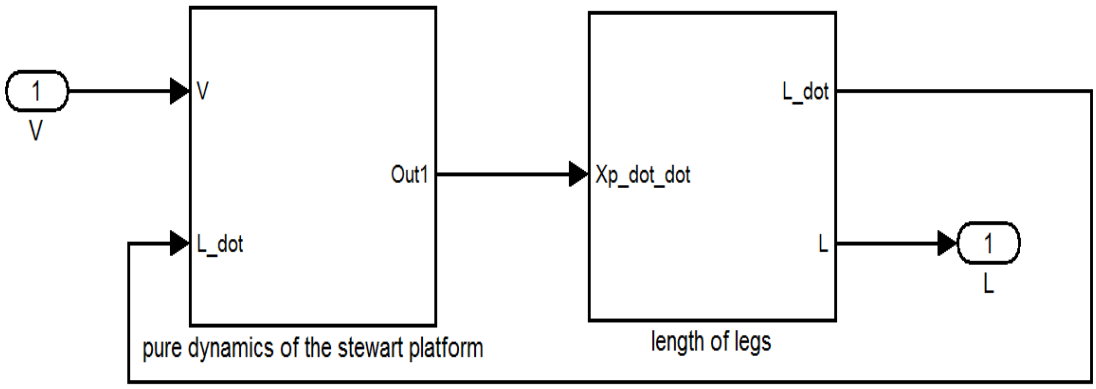


Figure 3.14: Stewart platform dynamics subsystem

Similarly, The Stewart platform dynamics has two subsystems shown below.

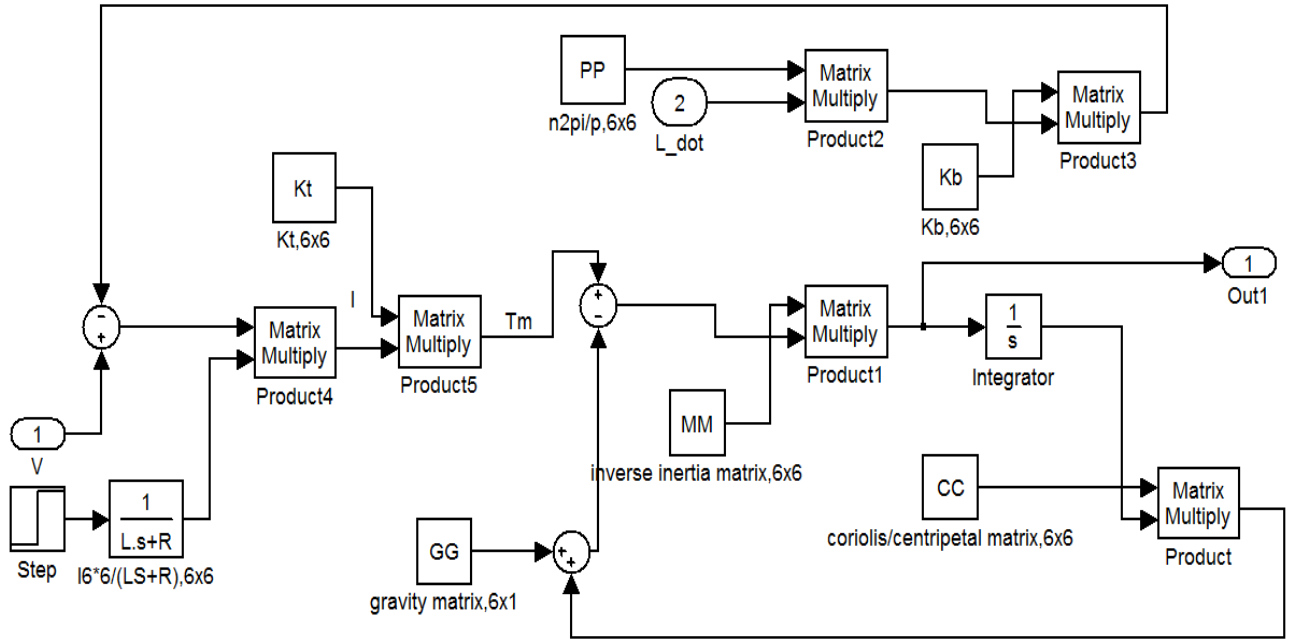


Figure 3.15: dynamics model of the Stewart platform

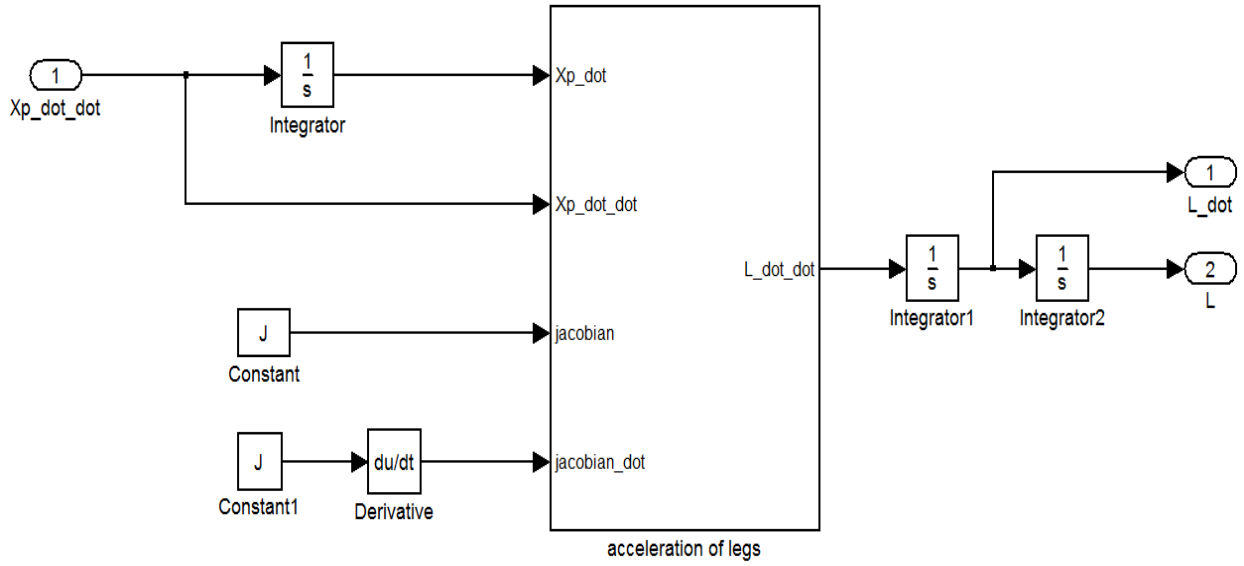


Figure 3.16: block diagram determining length of the legs

Under figure 3.16, there is a subsystem which computes the acceleration of the legs.

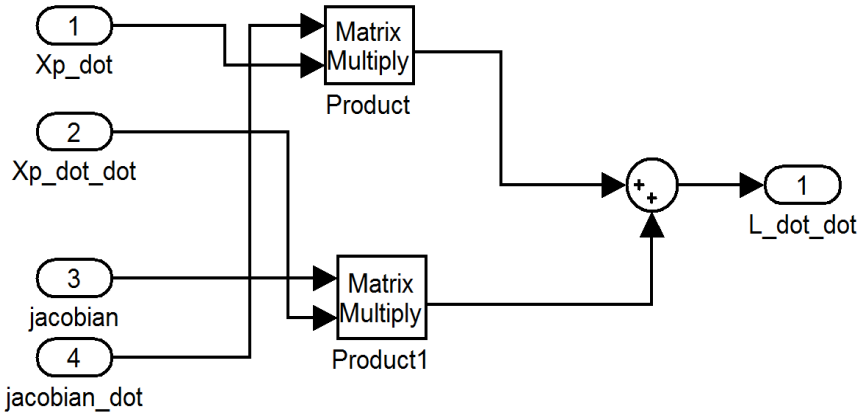


Figure 3.17: acceleration of the legs

3.9. Passivity Based Controller

Passivity based controller is used for achieving stability of nonlinear systems by making the system passive, because passive systems are by nature asymptotically stable systems. In addition it has a robustness property.

Passivity is therefore related to the property of stability in an input output sense, that is, we say that the system is stable if bounded "input energy" supplied to the system, yields bounded output energy. [15]

Since the aim in PBC is to render the closed loop system passive, the main property used in PBC is the fact that the interconnection of passive systems is passive. Conversely, passive systems can be decomposed into passive "subsystems". Thus in this philosophy the controller may be designed as a passive system. It is natural that PBC is most suitable for electrical and electro mechanical systems such as power converters, electrical machines, etc.

3.9.1. Passivity of EL Systems

There are three basic properties of Euler Lagrange systems for the design of PBC: [15]

- It is known that EL systems define passive maps. This is the most fundamental property for PBC since it identifies the output which is easy to control and provides a storage function that will typically motivate the desired closed-loop storage function.
- It is possible to introduce the decomposition of the Euler Lagrange system dynamics into interconnected passive subsystems to simplify the controller design.
- It preserves interconnection, i.e., the action of the controller may have EL structure. In this way the properties of the closed loop system are still captured by the energy and dissipation functions, which are simply the sum of the corresponding functions of the plant and the controller.

Chapter 4

Controller Design

4.1. Parameters Setting of the Stewart Platform

By setting the parameters, like, attachment points on the moving and base platforms and system constants (platform and motor constants), the performance of the controller can be observed. The attachment points GT_i and B_i on the moving and base platform are shown in the following table:

Table 4.1: Attachment points on the moving platform

Coordinate	GT_1(m)	GT_2(m)	GT_3(m)	GT_4(m)	GT_5(m)	GT_6(m)
X	0.0641	0.0641	0.0278	-0.0919	-0.0919	0.0278
Y	-0.0691	0.0691	0.0901	0.0209	-0.0209	-0.0901
Z	0	0	0	0	0	0

Table 4.2: Attachment points on the base platform

Coordinate	B_1(m)	B_2 (m)	B_3 (m)	B_4 (m)	B_5 (m)	B_6 (m)
X	0.1451	0.1451	-0.0544	-0.0906	-0.0906	-0.0544
Y	-0.0209	0.0209	0.1361	0.1152	-0.1152	-0.1361
Z	0	0	0	0	0	0

The system constants are also given in the following table:

Table 4.3: Stewart platform constants [9]

Parameter	Value
m_{up}	1.1324 (Kg)
m_1	0.4279 (Kg)
m_2	0.1228 (Kg)
l_1	0.22 (m)
l_2	0.05 (m)
R_p	0.09433988 (m)
R_b	0.14663308 (m)
$I_{(mf)}$	$\begin{bmatrix} 0.0025 & 0 & 0 \\ 0 & 0.0025 & 0 \\ 0 & 0 & 0.0050 \end{bmatrix} kg.m^2$

Table 4.4: DC motor constants [9]

Parameter	Value
R_a	$7.10(\Omega)$
L_a	$265e-6(H)$
k_b	$2.730e-3(v/rev/\min)$
k_t	$26.10e-3(Nm/A)$
n	$1(rad/rad)$
J_m	$0.58e-6(Kg.m^2)$
J_s	$0.002091e-3(Kg.m^2)$
b_m	$0.0016430e-3(N.s/rad)$
b_s	$0.11796e-3(N.s/rad)$
p	$0.001(m)$

4.2. Outer Loop Controller Design

The figure below shows the feedback interconnection of the plant and the controller. Where, u is the control law and y is the output of the plant.

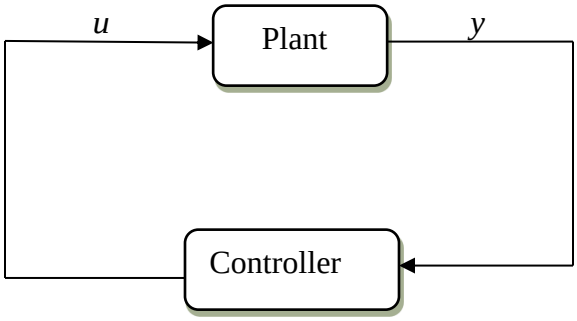


Figure 4.1: feedback interconnection of plant and controller

The controller designed in this thesis is PD+ controller on the basis of the passivity properties of the EL system, which mainly focuses on the conservation of energy. Design of controllers for nonlinear system can be analyzed using input output stability approach. The controller which is going to be designed here with will be based on the passivity theorem, input output stability approach.

The map $u \to y$ is passive if there exists a state function $H(x)$, bounded from below, and a non-negative function $d(t) \geq 0$ such that

$$\int_0^t u^T(s) y(s) ds = H(x(t)) - H(x(0)) + d(t) \tag{4.1}$$

The term on the left side is the energy supplied to the system, the first and the second term on the right side represents the stored energy and dissipated energy respectively.

The passivity property described in equation (4.1) is sufficient to solve regulation tasks in mechanical systems, where the PBC only needs to modify the potential energy and the dissipation function. However, to study tracking problems or treat electrical or electromechanical systems, we need a stronger property. In this case a desired behavior should be imposed, not only on generalized coordinate (q), but on derivative of generalized coordinate $(\dot{q})$ as well, which in

its turn translates into the need for modifying the kinetic energy. The PBC approach may be viewed as an extension of the well-known energy-shaping plus damping injection technique. [15]

For this particular problem we can concentrate our attention on the potential energy and the dissipation functions and proceed through a stage design process. First, an energy shaping stage, where we modify the potential energy of the system in such a way that the "new" potential energy function has a global and unique minimum in the desired equilibrium. Second, damping injection stages where we now modify the dissipation function to ensure asymptotic stability. The assumption of full actuation allows us to assign any arbitrary potential energy function, while the availability of the full-state trivializes the task of damping injection.

Lyapunov Stability analysis is the best approach for realizing the performance of systems which have non linearity by nature; it mostly focuses on the total stored energy of the system called the lyapunov function having a capability of attaining global asymptotic stability of the system. Energy shaping and damping injection are the extension of attaining stability in the sense of lyapunov.

4.2.1. Design of PD+ Controller

PD controller will be used for set point regulation of Stewart platform. It concerns the global asymptotic stabilization via energy shaping plus damping injection of the equilibrium $[q,\dot{q}]^T = [q^*,0]^T$, with q^* a constant vector,

Consider the equation below:

$$M(q)\ddot{q} + V(q,\dot{q})\dot{q} + G(q) = u \tag{4.2}$$

Equation (4.2) can be rewritten in state space representation:

Let a state vector **x** be

$$x = \begin{pmatrix} q \\ \dot{q} \end{pmatrix} \tag{4.3}$$

And the input vector is (u) and suppose the output vector is **q**, then we can obtain the following expression.

$$\left\{ \begin{pmatrix} \dot{q} \\ \ddot{q} \end{pmatrix} = \begin{pmatrix} \dot{q} \\ -M^{-1}(V\dot{q}+G) \end{pmatrix} + \begin{pmatrix} 0 \\ M^{-1} \end{pmatrix} (u) \right\} \qquad (4.4)$$
$$y = [I \;\; 0] \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$$

Let the state-feedback control law be given as:

$$u = -\frac{\delta V_C(q)}{\delta q} - \frac{\delta F_C(\dot{q})}{\delta \dot{q}} \qquad (4.5)$$

The function $V_C(q)$ is such that the potential energy of the closed loop system $V_d(q) = V(q) + V_C(q)$ has unique global minimum at $q = q*$. And the dissipation function, $F_C(\dot{q})$ satisfies $\frac{\delta F_C(\dot{q}=0)}{\delta \dot{q}} = 0$ and $\dot{q}^T \frac{\delta F_C(\dot{q})}{\delta \dot{q}} > 0$ for all $\forall q \neq 0$. Under this condition, the equilibrium $[q, \dot{q}]^T = [q^*, 0]^T$ is globally asymptotically stable (GAS). [15]

Our design problem is to stabilize the Stewart platform at a constant equilibrium $[q, \dot{q}]^T = [q^*, 0]^T$. As proposed above we will seek to modify the potential energy and the dissipation function of the system, leaving untouched the kinetic energy, since it plays no role on the stability properties of the equilibrium. That is we want the closed loop system to be an EL system with EL parameters $(T(q,\dot{q}), V_d(q), F_d(\dot{q})$ where $T(q,\dot{q})$ is the kinetic energy of the system.

Since we know that a minimum potential energy corresponds to a stable equilibrium point, the "new" potential energy function should have a global and unique minimum at the desired position. A natural candidate is then:

$$V_d(q) = \frac{1}{2} K_P \tilde{q}^2 \qquad (4.6)$$

Where, $k_P > 0$ and $\tilde{q} = q - q^*$.

Similarly, the desired Rayleigh dissipation functions can be chosen as:

$$F_d(\dot{q}) = \frac{1}{2} K_d \dot{q}^2 \qquad (4.7)$$

Where, $K_d > 0$

These choices lead to the control:

$$u = \frac{\delta(V(q) - V_d(q))}{\delta q} - \frac{\delta F_C(\dot{q})}{\dot{q}}$$

$$\equiv g(q) - K_P\tilde{q} - K_d\dot{q} \tag{4.8}$$

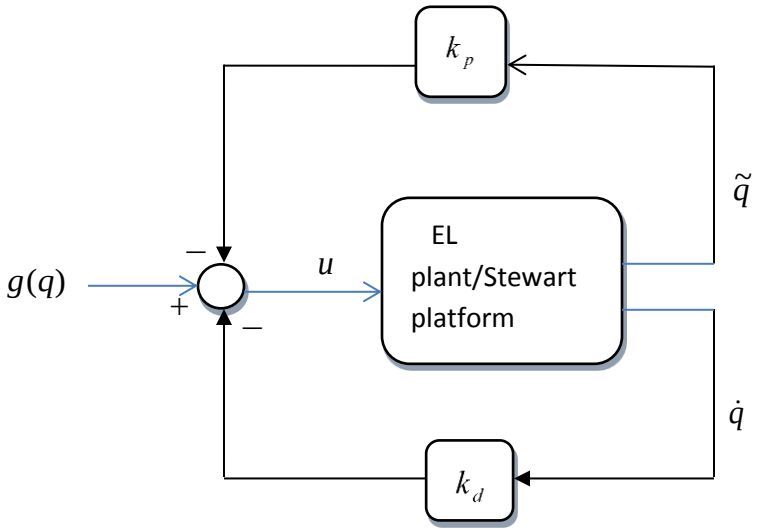


Figure 4.2: PD controller for set point regulation

In the above block diagram, energy shaping and damping injection have a role for the structural make-up of the system. There are two positive definite constants, K_P and K_d, and the gravity matrix component $g(q)$. In the designed controller, there is gravity cancellation, therefore, this kind of controller is generally called PD controller with gravity cancellation.

The PD+ controller is one of the first results guaranteeing global tracking for rigid- joint robots that is global uniform asymptotic stability (GUAS) of the origin $(\tilde{q}, \tilde{\dot{q}}) = (0,0)$. It is the natural extension to tracking control, of the energy shaping plus damping injection PD plus gravity cancellation used for regulation. Therefore, the whole system with the PD+ controller will be expressed as follows. The values of k_p and k_d are presented in Appendix C.

$$M(q)\ddot{q} + V(q,\dot{q})\dot{q} + G(q) = -K_P\tilde{q} - K_d\dot{q} \tag{4.9}$$

As shown in the above equation, since, the control law is designed not to have the gravity term cancelation, the effect of gravity term will be observed on the system. It is known that dominating instead of cancelling the nonlinear term g(q) enhances the robustness of the system Vis-à-vis parametric uncertainties. [15]

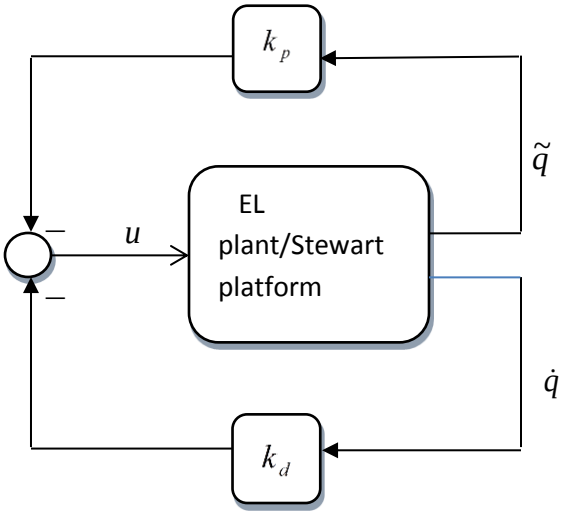


Figure 4.3: PD+ controller for trajectory tracking

The stable equilibrium of an EL system corresponds to the minima of its potential energy function. By setting an appropriate lyapunov function, the internal stability of the system can be checked. The stability of the whole system is guaranteed in sense of lyapunov as presented in appendix A.

The Stewart platform dynamics with controller is modeled using MATLAB/Simulink, which is the extension of equation (3.13), open loop dynamics of the system. The only added part is the control law in to the whole modeled system. The following block diagram shows the Stewart platform dynamics with controller.

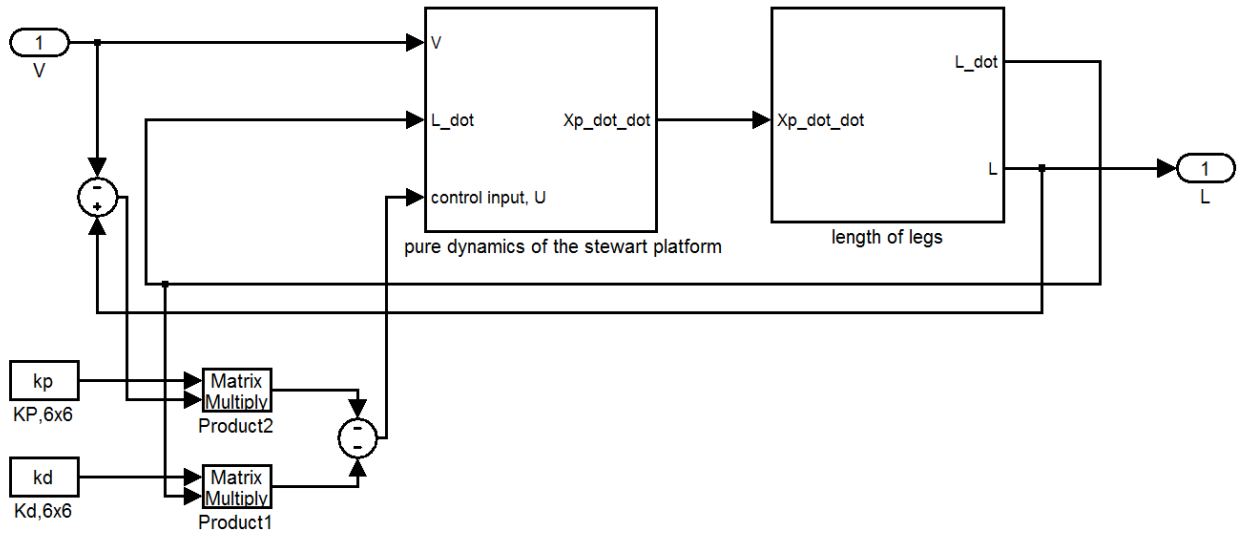


Figure 4.4: complete dynamics of Stewart platform with control input

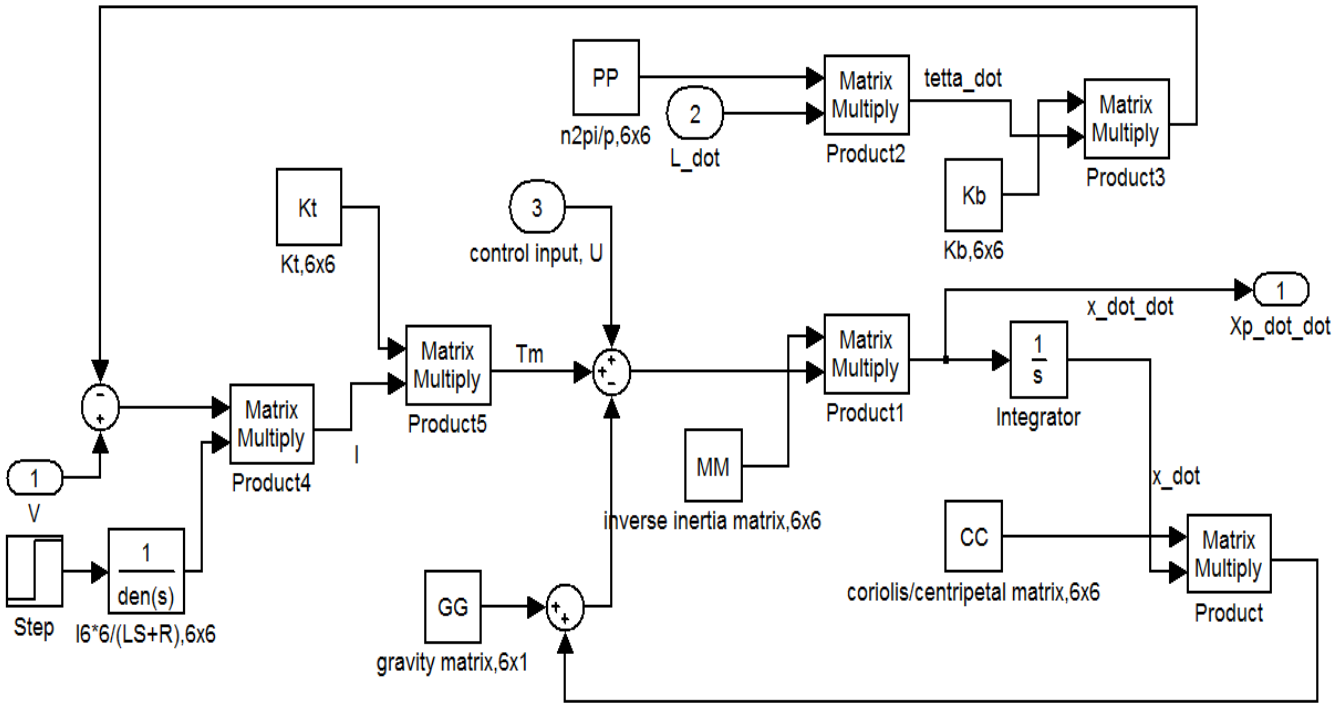


Figure 4.5: Stewart platform dynamics subsystem with control input

4.3. ADAMS modeling of Stewart platform

The diagram in figure (4.6) describes the four step process used in combining MATLAB controls with the Stewart platform in ADAMS.

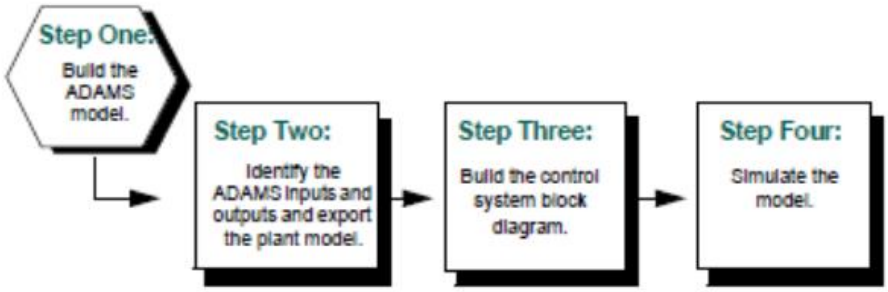


Figure 4.6: Modeling steps

- **Step One**: The first step in working with ADAMS/Controls is to build the ADAMS model. The model should be complete and include all necessary geometry, constraints, forces, and measures.
- **Step Two**: identifying inputs and Outputs. The outputs describe the variables that go to the controls application (the output from the ADAMS model is the input to the MATLAB controls system). The inputs describe the variables that comeback into ADAMS (the outputs of the MATLAB controls system) and, therefore, complete a closed loop between ADAMS and MATLAB. The inputs and outputs are configured as variables. Finally, the plant is imported from ADAMS to MATLAB for control and simulation.
- **Step Three**: Building the Block Diagram - the control system is built with MATLAB.
- **Step Four**: Simulate the Model - Simulate the combined mechanical model and control system.

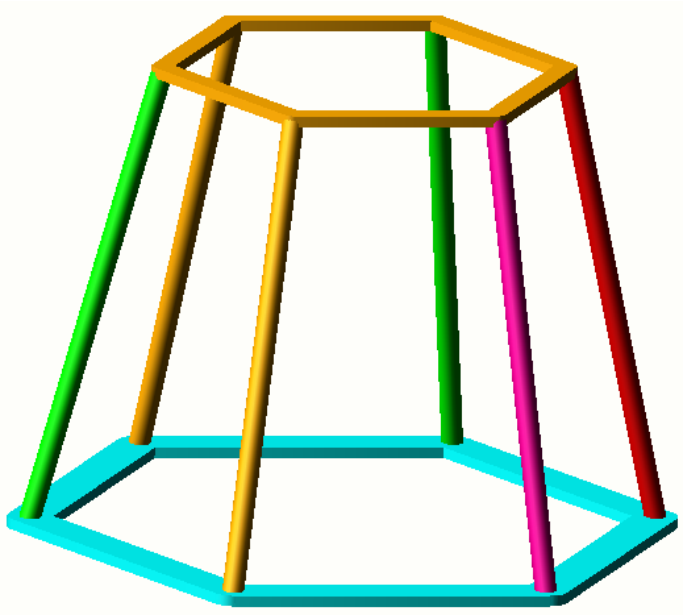


Figure 4.7: ADAMS model of Stewart platform

The built model is shown in figure (4.7), the following components are used:

1. Geometric links
   a. Upper platform (six links)
   b. Six upper links (upper legs)
   c. Six lower links (lower legs)
   d. Base (six links)
2. Connectors: twelve spherical and 6 prismatic type connectors are used to connect different links, they are used to connect:
   a. Upper legs and platform, by spherical joints
   b. Upper legs and lower legs, by prismatic joints
   c. Lower leg and base, by spherical joints
3. Forces
   a. Force 1 – input force for motor 1
   b. Force 2 – input force for motor 2
   c. Force 3 – input force for motor 3
   d. Force 4 – input force for motor 4

e. Force 5 – input force for motor 5
f. Force 6 – input force for motor 6

4. System elements – these are the state variables of the model
   a. SF_1 – state variable for force 1
   b. SF_2 – state variable for force 2
   c. SF_3 – state variable for force 3
   d. SF_4 – state variable for force 4
   e. SF_5 – state variable for force 5
   f. SF_6 – state variable for force 6
   g. SL_1 – state variable for length 1
   h. SL_2 – state variable for length 2
   i. SL_3 – state variable for length 3
   j. SL_4 – state variable for length 4
   k. SL_5 – state variable for length 5
   l. SL_6 – state variable for length 6
5. Measuring elements, these elements are used to measure the angular displacement of each motor.
   a. SL_1 measure – measures the length of leg 1
   b. SL_2 measure – measures the length of leg 2
   c. SL_3 measure – measures the length of leg 3
   d. SL_4 measure – measures the length of leg 4
   e. SL_5 measure – measures the length of leg 5
   f. SL_6 measure – measures the length of leg 6
6. Sensors: in each of the legs, sensors are added to constraint the maximum and minimum of leg length, twelve sensors are used.

The final setup of the Simulink simulation after exporting the ADAMS model is shown in figure 4.8 below.

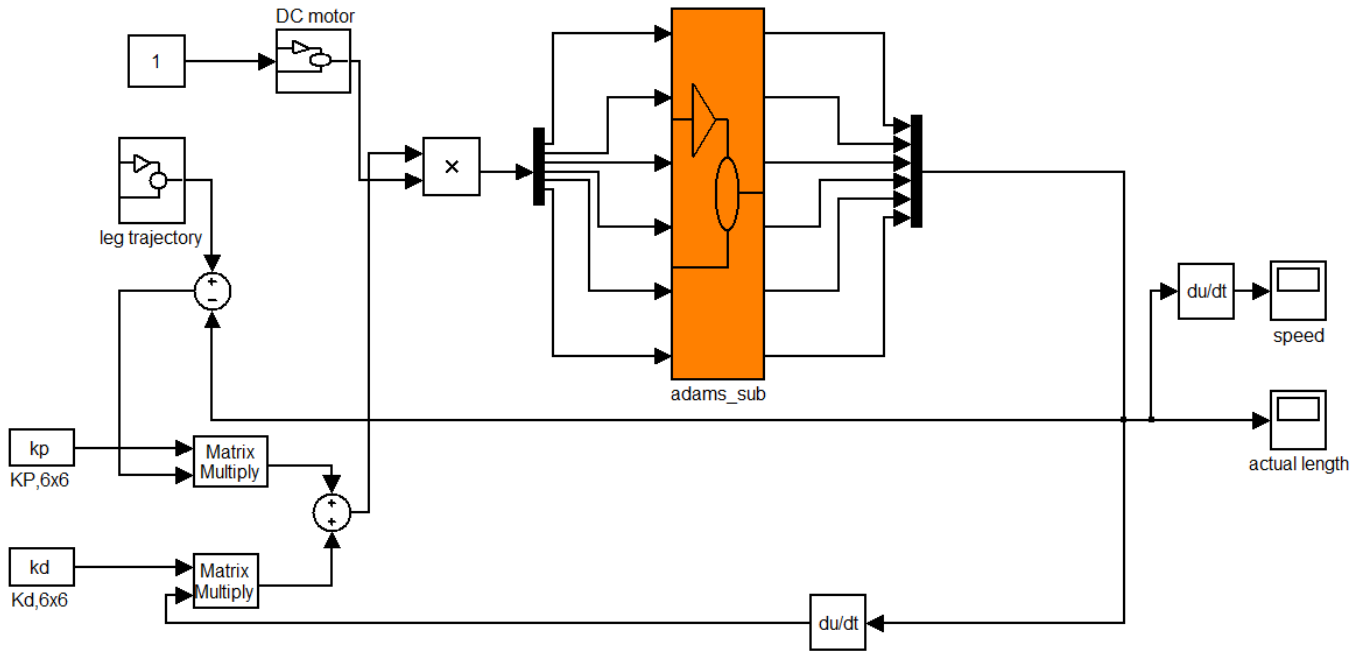


Figure 4.8: ADAMS model and Simulink control

Chapter 5

Simulation Results and Discussions

5.1. Performance Analysis of Outer Loop

The initial values of position and orientation, as well as the leg lengths are given as follows:

$$x = 0m$$
$$y = 0m \quad \text{and} \quad \alpha = \beta = \gamma = 0(rad)$$
$$z = 0.21m$$

And also, $L_1 = L_2 = L_3 = L_4 = L_5 = L_6 = 0.22m$

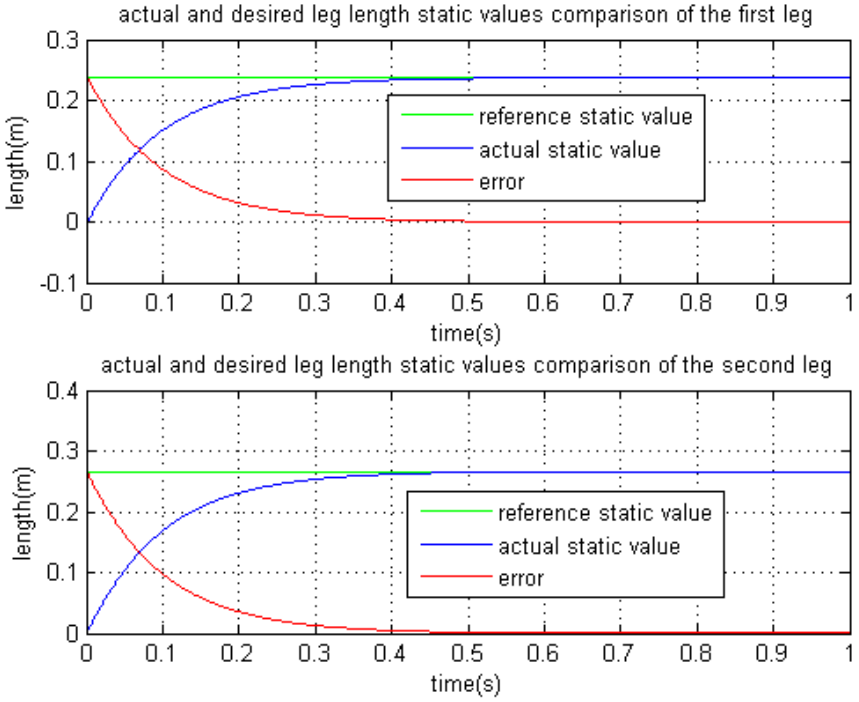
The physical makeup of the Stewart platform contains joints which are not modelled mathematically. Each joint will have mass, and then results friction as well as gravity. These components will have effect on the performance of the system. The designed controller guarantees to minimize this effect. Therefore we need to represent the effects with unit step disturbances which will be applied on the system as external disturbances.

Let the performance of the designed system be analyzed by setting some desired position and orientation of the upper/moving platform of the Stewart platform for both static and trajectory case.

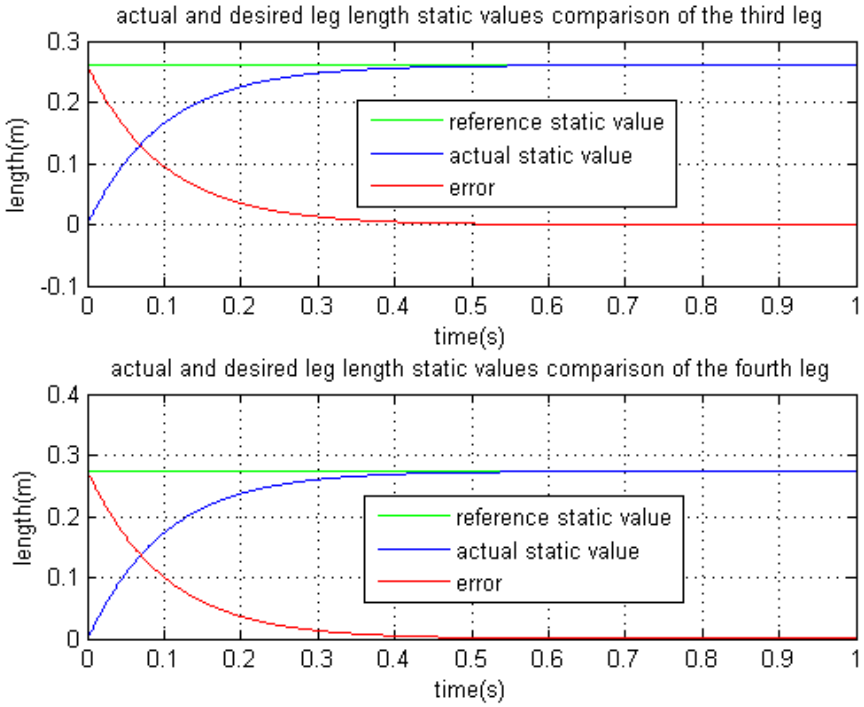
Case1: static reference

$$x = 0.07m$$
$$y = 0.06m \quad \text{and} \quad \alpha = \beta = \gamma = 0(rad)$$
$$z = 0.05m$$

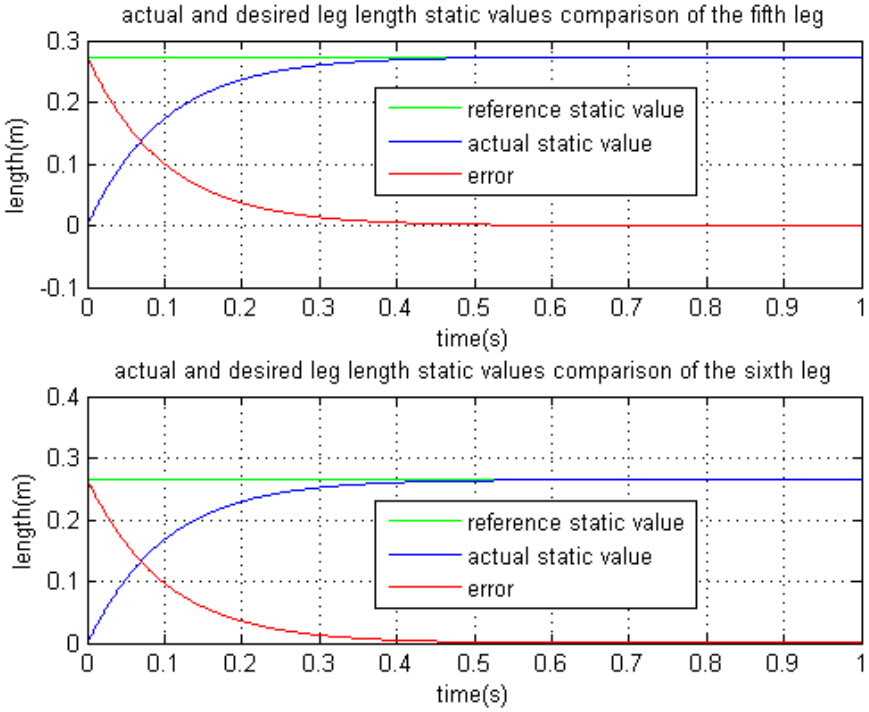
For the listed reference positions and orientations set point values as input for the inverse kinematic algorithm, lets observe the values of all the six leg lengths with in the specified time range ($0 \rightarrow 1\text{sec}$) with a certain disturbances consideration and undisturbed system. The following figures show the output set point values of the controlled system without any external disturbances consideration and with an addition of unit step disturbance.


actual and desired leg length static values comparison of the first leg
length(m)
time(s)
reference static value
actual static value
error
actual and desired leg length static values comparison of the second leg
length(m)
time(s)
reference static value
actual static value
error

(a)


actual and desired leg length static values comparison of the third leg
length(m)
time(s)
reference static value
actual static value
error
actual and desired leg length static values comparison of the fourth leg
length(m)
time(s)
reference static value
actual static value
error

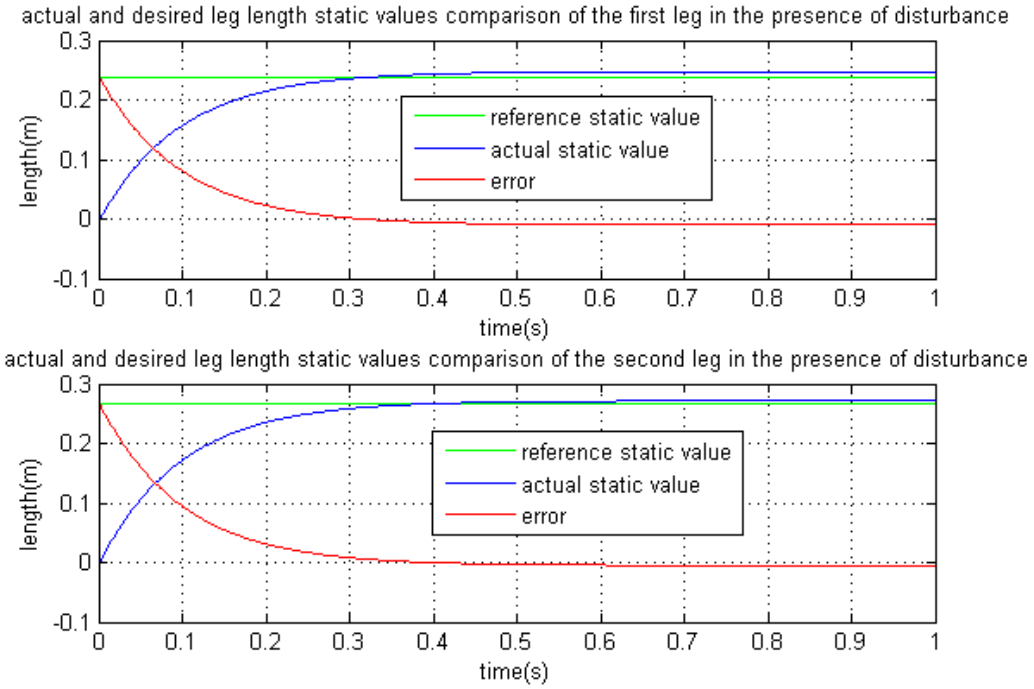
(b)



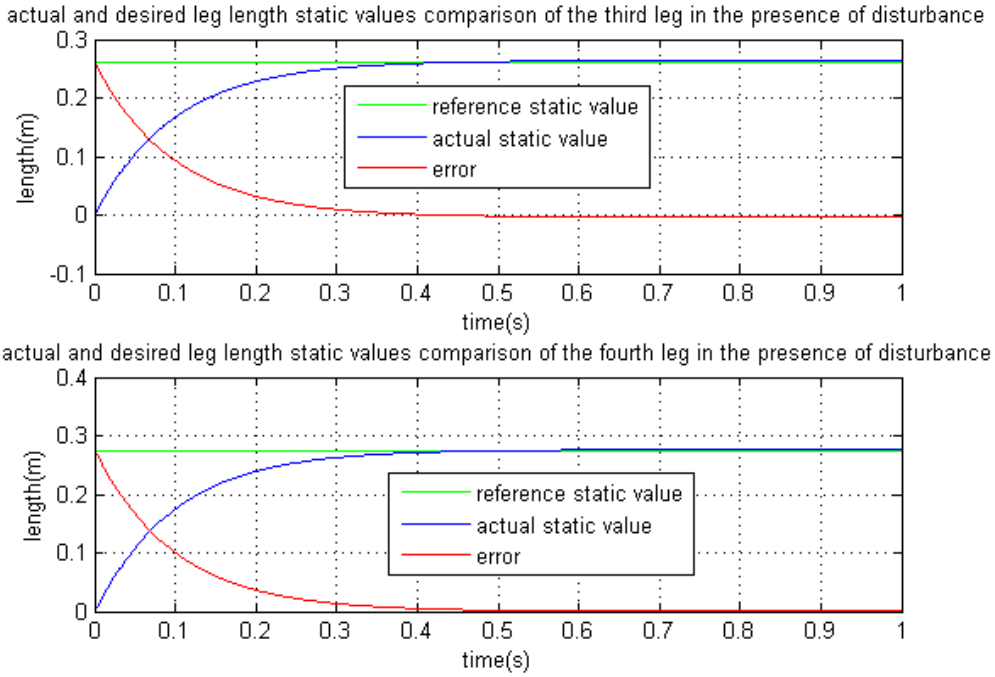
(c)

Figure 5.1: comparison of desired and actual leg length set values (a) first and second leg, (b) third and fourth leg, (c) fifth and sixth leg

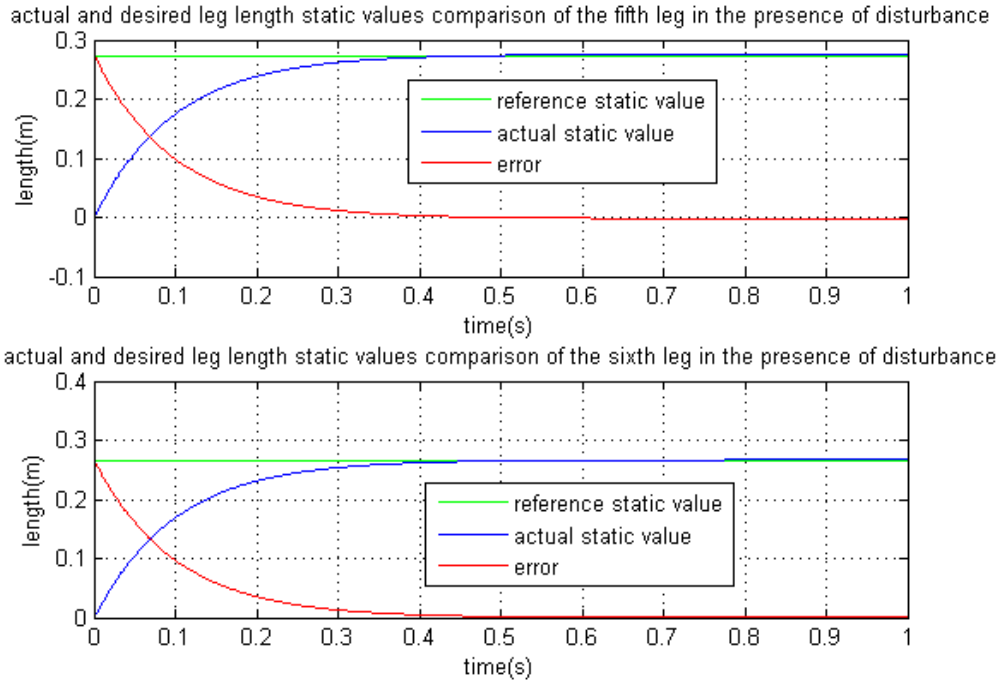
The static values shown in figure 5.1 are without any external disturbances. Let us analysis the performance of the system by adding unit step disturbances to make sure that the controller have a capability of disturbance attenuation. The following figure shows the response of the controlled system in the presence of disturbances.


actual and desired leg length static values comparison of the first leg in the presence of disturbance
length(m)
time(s)
reference static value
actual static value
error
actual and desired leg length static values comparison of the second leg in the presence of disturbance
length(m)
time(s)
reference static value
actual static value
error

(a)


actual and desired leg length static values comparison of the third leg in the presence of disturbance
length(m)
time(s)
reference static value
actual static value
error
actual and desired leg length static values comparison of the fourth leg in the presence of disturbance
length(m)
time(s)
reference static value
actual static value
error

(b)



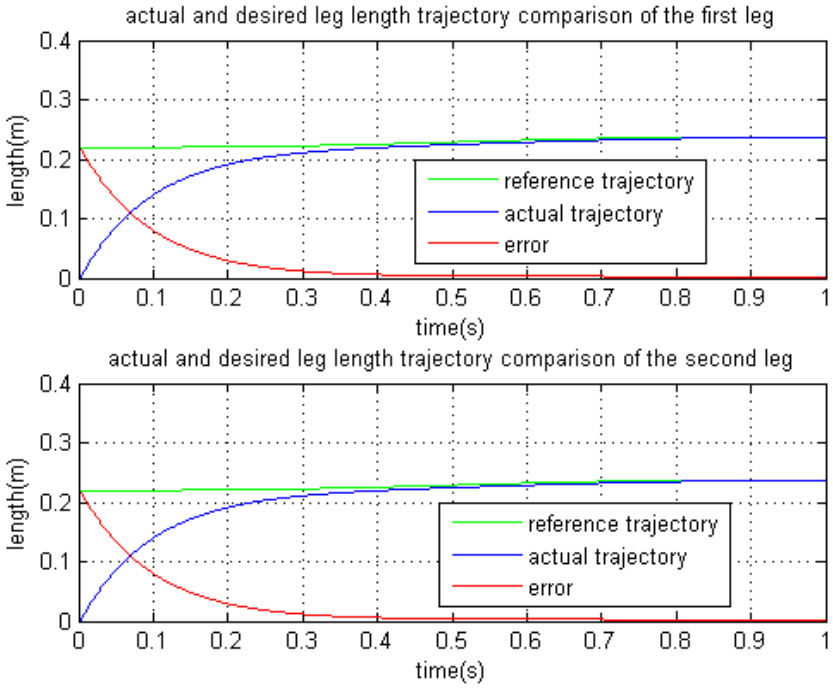
(c)

Figure 5.2: comparison of desired and actual leg length set values in the presence of unit step disturbances (a) first and second leg, (b) third and fourth leg, (c) fifth and sixth leg

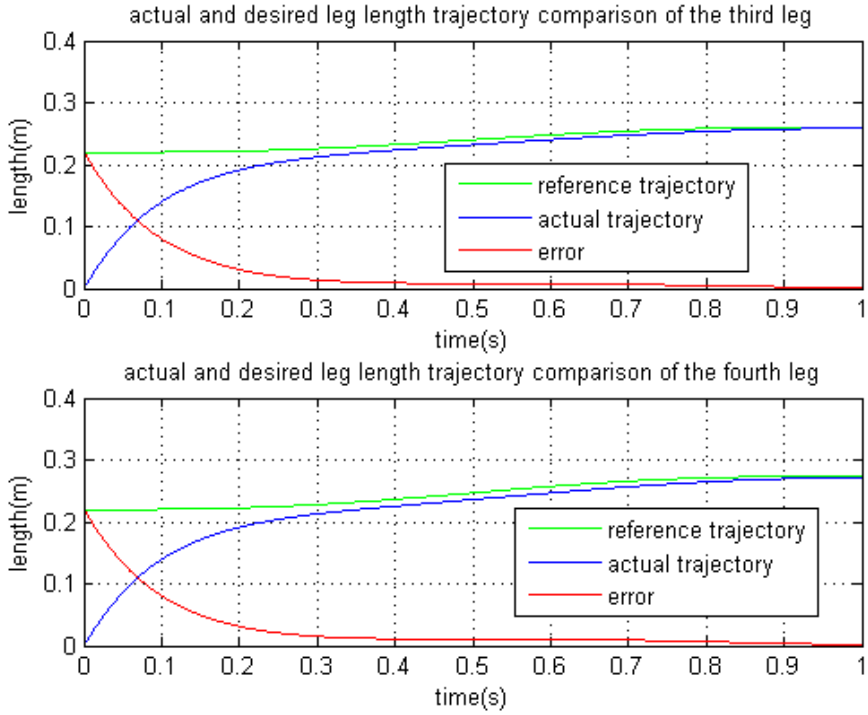
Case2: trajectory reference

$$\begin{bmatrix} x = 0.035t - 0.01\sin(\pi t) \\ y = 0.03t - 0.01\sin(\pi t) \\ z = 0.025t - 0.01\sin(\pi t) \end{bmatrix} m \quad \text{And} \quad \begin{bmatrix} \alpha = 0.08t - 0.03\sin(\pi t) \\ \beta = 0.13t - 0.04\sin(\pi t) \\ \gamma = 0.17t - 0.05\sin(\pi t) \end{bmatrix} (rad)$$

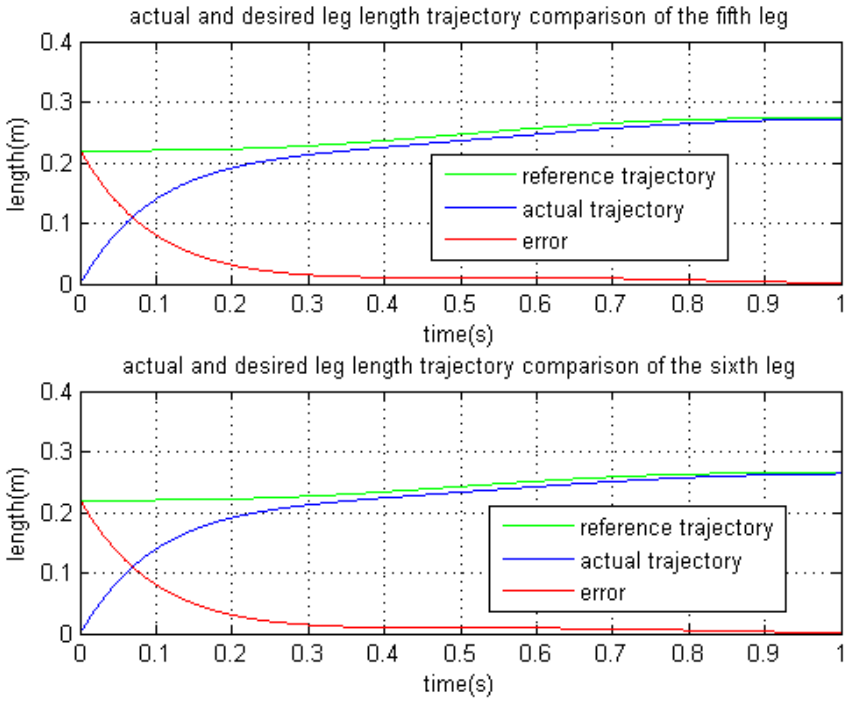
For the listed reference positions and orientations trajectories as input for the inverse kinematic algorithm, lets observe the trajectories of all the six leg length and velocities with in the specified time range ($0 \rightarrow 1\text{sec}$) including disturbances consideration. The following figures show the output trajectory values of the controlled system without any external disturbances consideration and with an addition of unit step disturbances.


actual and desired leg length trajectory comparison of the first leg
length(m)
time(s)
reference trajectory
actual trajectory
error
actual and desired leg length trajectory comparison of the second leg
length(m)
time(s)
reference trajectory
actual trajectory
error

(a)


actual and desired leg length trajectory comparison of the third leg
length(m)
time(s)
reference trajectory
actual trajectory
error
actual and desired leg length trajectory comparison of the fourth leg
length(m)
time(s)
reference trajectory
actual trajectory
error

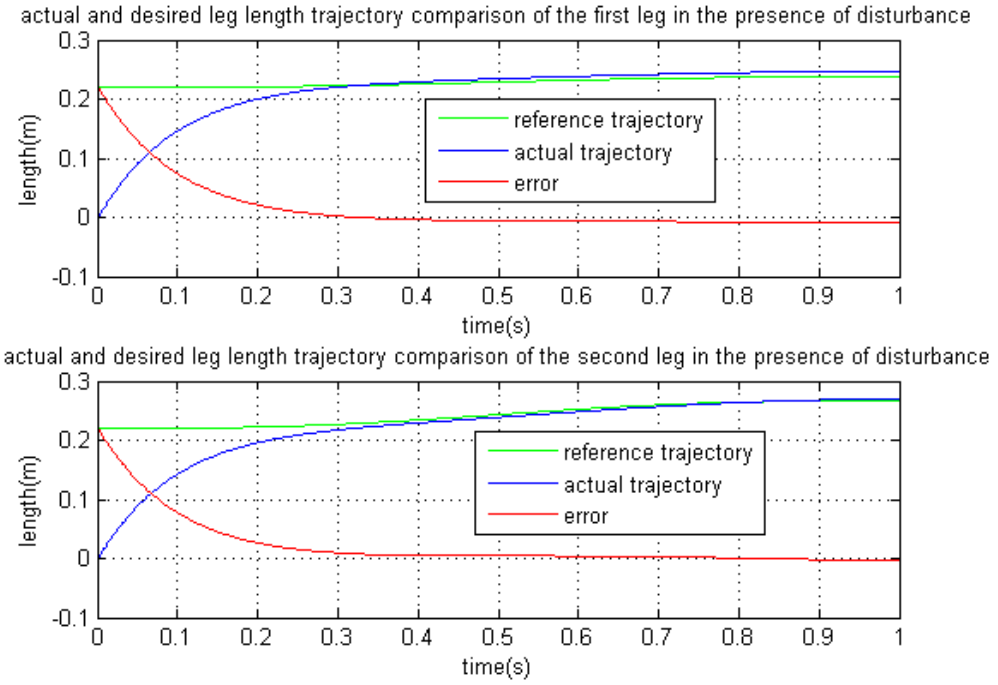
(b)



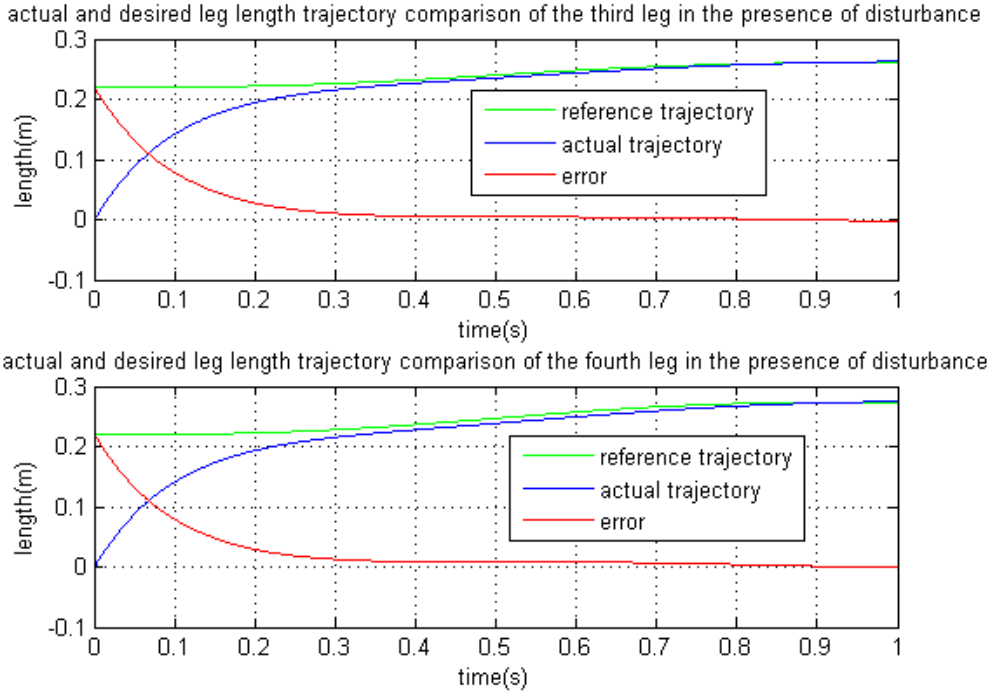
(c)

Figure 5.3: comparison of desired and actual leg length trajectories (a) first and second leg, (b) third and fourth leg, (c) fifth and sixth leg

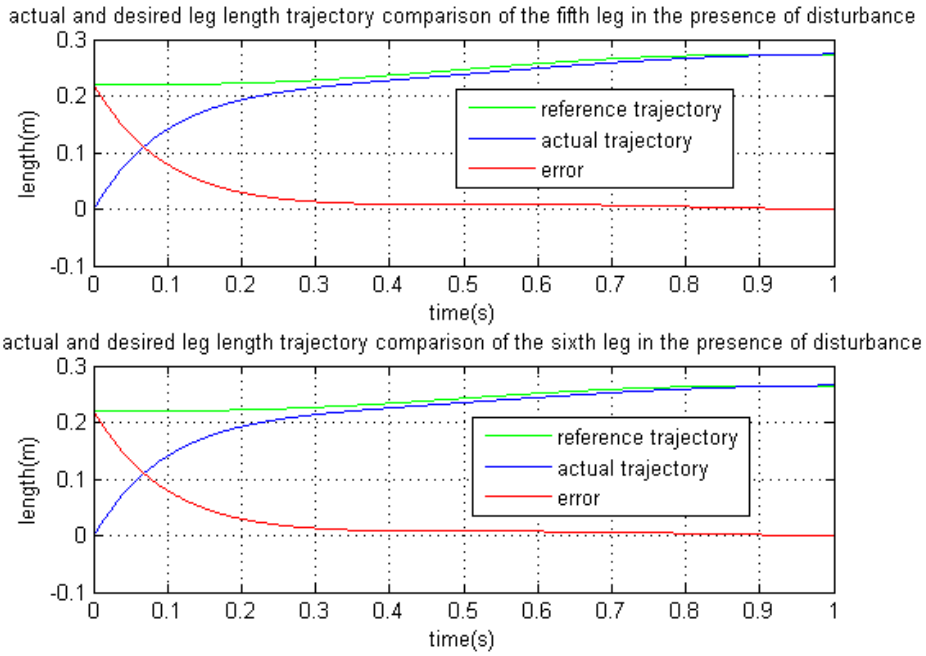
The trajectories shown in figure 5.3 are without any external disturbances. Let us analysis the output trajectories by adding unit step disturbances to make sure that the controller have a capability of disturbance attenuation. The following figure shows the response of the controlled system in the presence of disturbances.


actual and desired leg length trajectory comparison of the first leg in the presence of disturbance
length(m)
time(s)
reference trajectory
actual trajectory
error
actual and desired leg length trajectory comparison of the second leg in the presence of disturbance
length(m)
time(s)
reference trajectory
actual trajectory
error

(a)


actual and desired leg length trajectory comparison of the third leg in the presence of disturbance
length(m)
time(s)
reference trajectory
actual trajectory
error
actual and desired leg length trajectory comparison of the fourth leg in the presence of disturbance
length(m)
time(s)
reference trajectory
actual trajectory
error

(b)



(c)

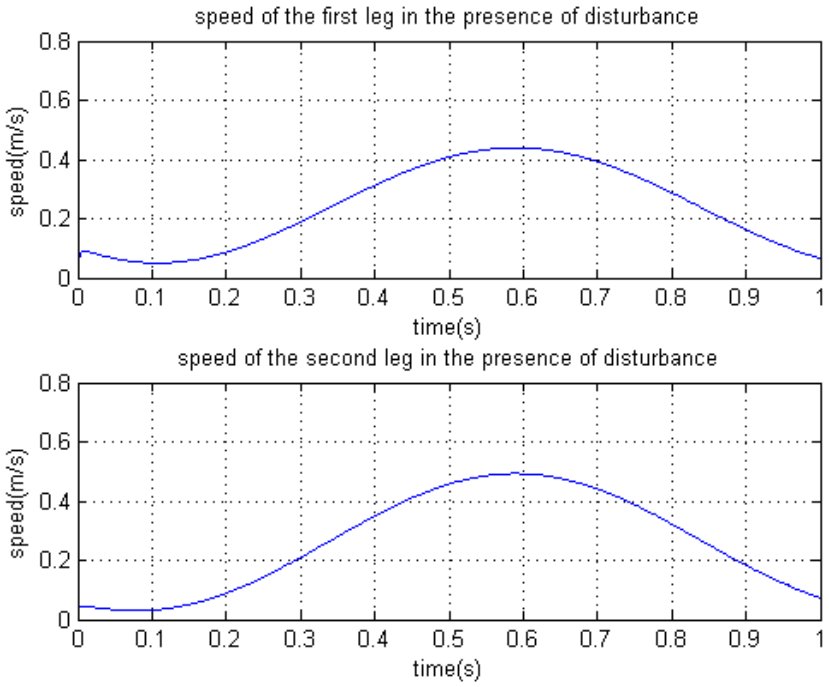
Figure 5.4: comparison of desired and actual leg length trajectories in the presence of disturbances (a) first and second leg, (b) third and fourth leg, (c) fifth and sixth leg

Figure 5.1 shows the output static values of the leg length in the interval between 0sec and 1sec. As shown from the figure, the system tracks the set point value at 0.4sec after some delay. As the time goes from 0sec to 0.4sec, the error decreases and finally becomes approximately zero at 0.4sec. The effect of disturbances on the performance of the controller is illustrated in figure 5.2. From the figure, it is observed that the system will not track the reference set point value exactly after 0.4sec, but with small error of $(0.008m)$ and have the same characteristics with figure 5.1 in the time interval between 0sec to 0.4sec. All the six legs possess similar output characteristics.

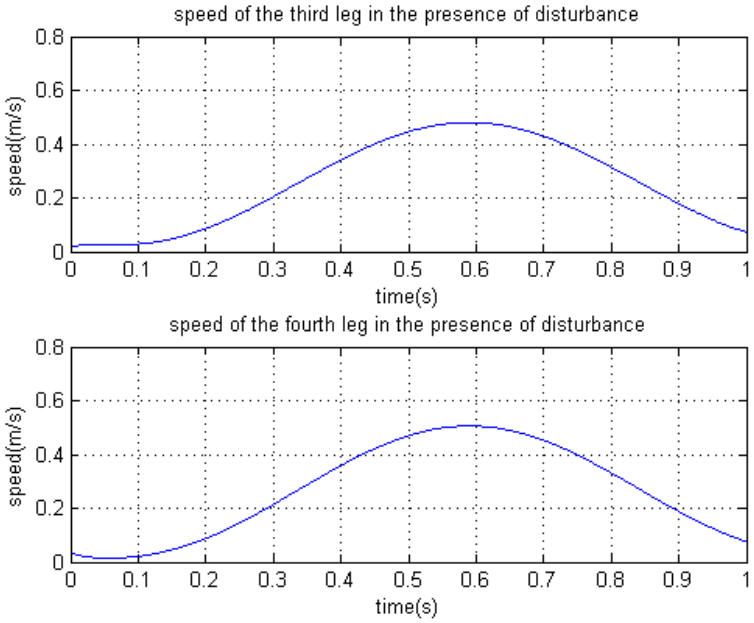
Figure 5.3 shows the output leg length trajectories of all the six legs in the time interval between 0sec and 1sec. As shown from the figure, the system tracks its initial leg lengths value at 0.3sec. The desired trajectory with error of $(0.006m)$ in the time interval between 0.3sec and 0.8sec, and approximately zero errors after 0.8sec is achieved. In figure 5.4, the effect of disturbance on the system is shown. As the figure clears the system tracks the desired trajectory

in similar manner with figure 5.3 in the time interval between 0sec and 0.8sec. and have error of $(0.003m)$ after 0.8sec.

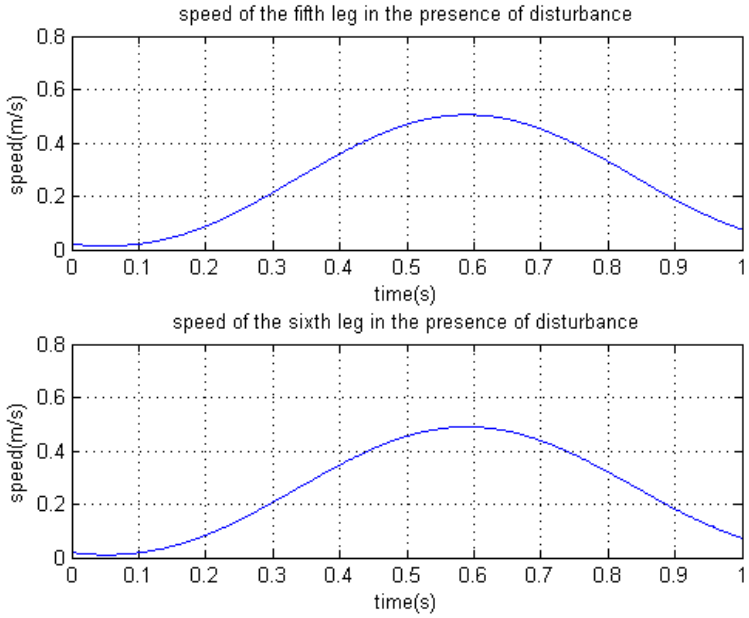
Figure 5.5 presents the legs speed trajectory of all the six legs in the time interval between 0sec and 1sec. As shown from the figure, the motion starts and finishes with nonzero velocity in all the six legs, this is due to limitation of mathematical modeling and it is verified using ADAMS simulation results presented in Figure 5.9. The maximum speed is achieved at 0.6sec in all the legs and have the value of $0.43m/s$, $0.49m/s$, $0.48m/s$, $0.5m/s$, $0.5m/s$ and $0.49m/s$ from the first to the sixth leg respectively. The position and velocity value of all the legs are quite soft changes, and is very important for motors to start and stop more softly.



(a)



(b)



(c)

Figure 5.5: speed of legs motions in the presence of disturbances (a) first and second leg, (b) third and fourth leg, (c) fifth and sixth leg

5.2. ADAMS simulation results

All the simulation results presented above are resulted from the mathematical design of the Stewart platform, and simply simulating it using Simulink blocks. But, to be sure the accuracy of the mathematical modeling, ADAMS 3-D modeling, which presents the physical model of mechanical systems, is needed. Let us see the simulation results taking 'case 2 – trajectory reference' as an input.

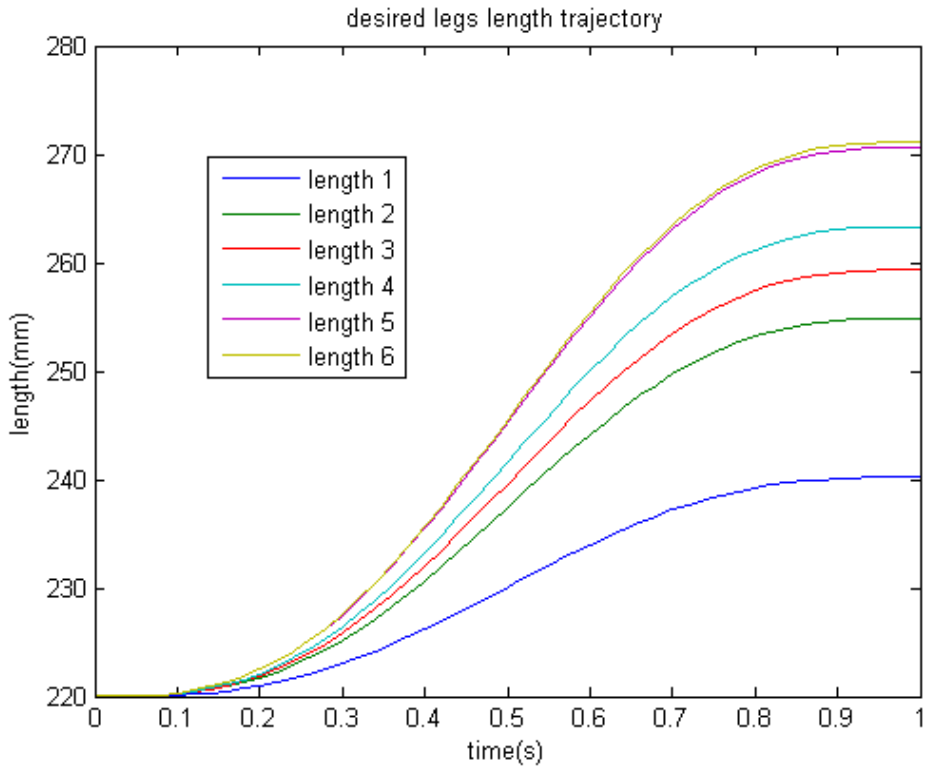


Figure 5.6: desired length trajectories of the six legs

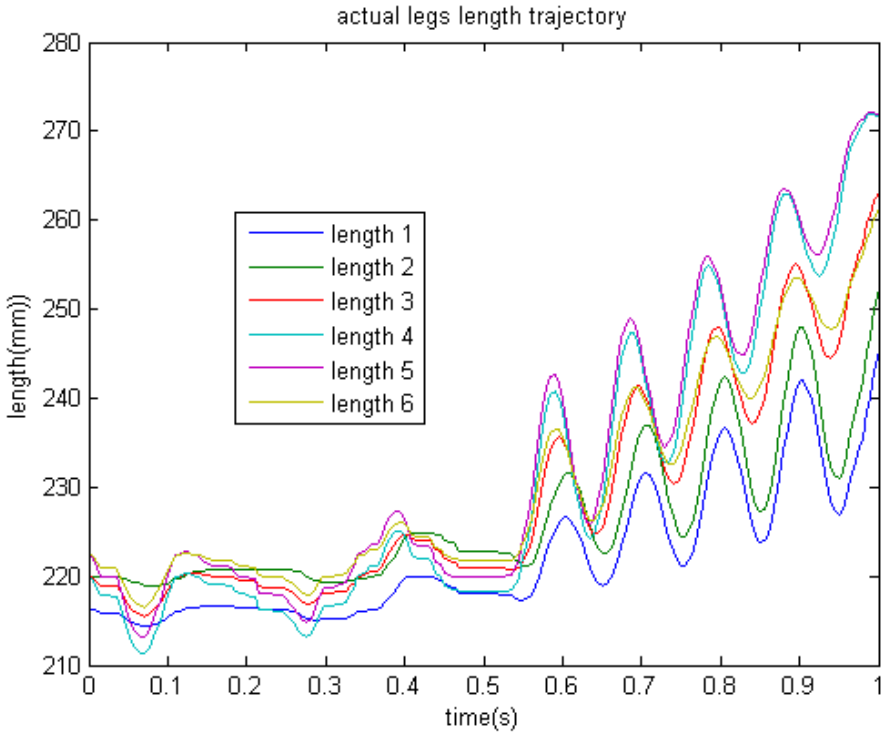


Figure 5.7: actual length trajectories of the six legs

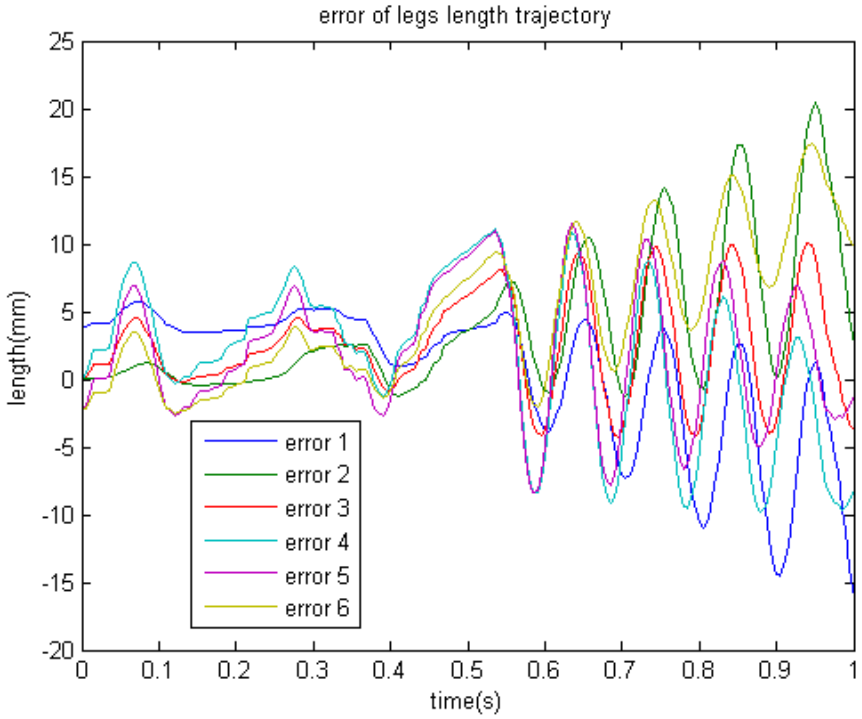


Figure 5.8: trajectory errors of the six legs

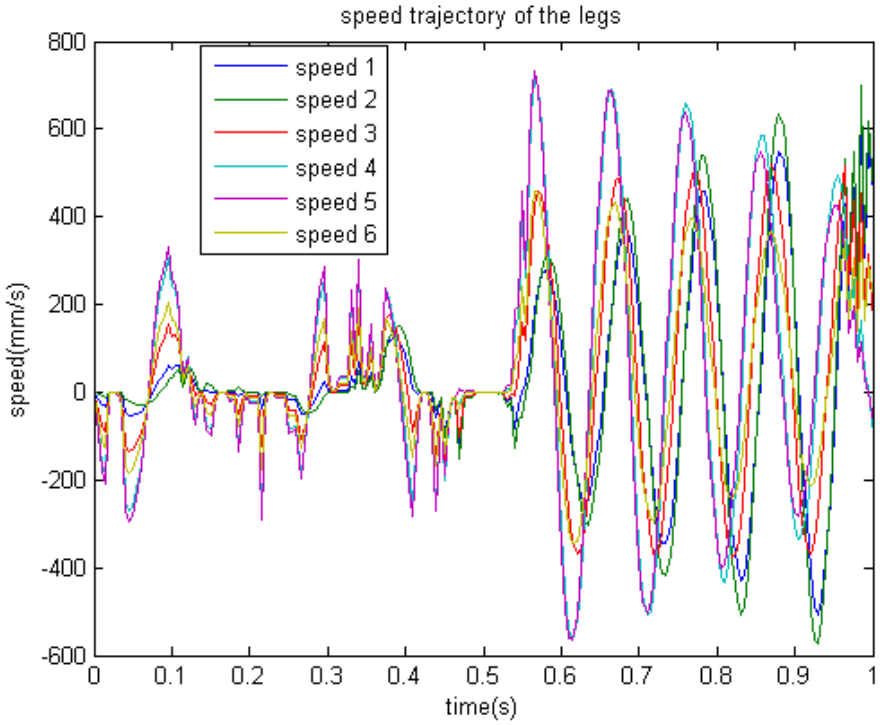


Figure 5.9: speed trajectory of the six legs

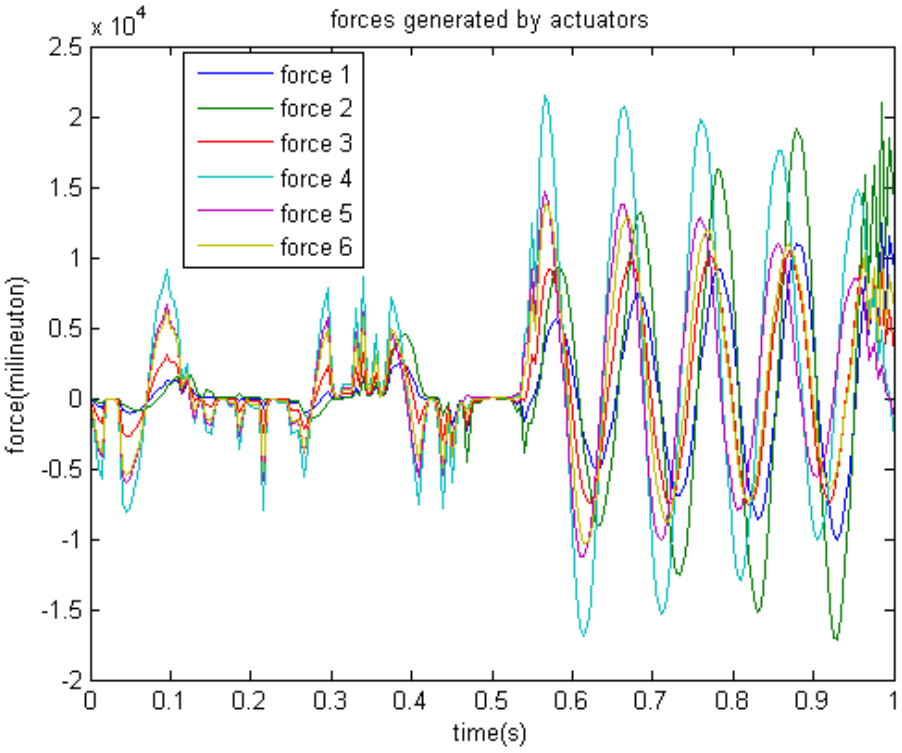


Figure 5.10: actuator forces

The desired leg length trajectory of the Stewart platform in the time interval between 0sec and 1sec is presented in figure 5.6, by taking this as a reference input to the ADAMS 3-D model, the ADAMS simulation generated leg length trajectories as shown in figure 5.7. The actuator forces are calculated by the controller designed in MATLAB/Simulink and presented in figure 5.10, which is bounded from below (-15N) and above (22N). As shown from figure 5.9, the motions start and finish with zero velocity, and have a non-zero trajectory otherwise.

The deviation/error of the leg lengths trajectory is shown in figure 5.8. As the figure clears, almost similar error characteristics is observed in the time interval between 0sec and 0.5sec. After 0.5sec, the error expands due to sensors (used in each of the links) problem. At 1sec, The errors are minimized except the first leg (the upper link of the leg gets inserted to the lower link of the leg) as a result of failure of the sensor (which can't take action when the leg length becomes below the minimum leg length accurately) used in the link.

As can be seen from the ADAMS simulation result, the leg length trajectories possess a certain oscillation pattern, and this is due to friction, link and joint flexibility and other factors, since ADAMS model represent the physical model of the system. Therefore, some deviation from the reference trajectory is observed. Due to sensors used in each of the legs, which limits the upper and lower bound of the leg lengths, zero velocity appeared when the length exceeds the optimum value in some time intervals.

Chapter 6

Conclusions, Recommendations and Future Works

6.1. Conclusion

The PD+ controller, PD controller with gravity compensation, guaranteed tracking for rigid joint robots in the sense of global uniform asymptotic stability (GUAS). The control law is designed in order not to have the gravity term cancelation.

The simulation result shows the effect of the controller on the performance of the system. In the absence of disturbances, the maximum trajectory tracking error is recorded as $(0.006m)$ in the time interval between $0\sec$ and $0.8\sec$. Applying unit step disturbance makes the error $0.006m$ after $0.8\sec$ which never be seen in undisturbed system. The maximum speed from the initial position to final position is recorded as $0.43m/s$, $0.49m/s$, $0.48m/s$, $0.5m/s$, $0.5m/s$ and $0.49m/s$ from the first to the sixth leg respectively. The ADAMS simulation results show the real time performance of the system, having some difference with mathematically modeled results. From the simulation studies, one can generalize that the system meets the desired output.

Finally the stability of the overall system is checked on the basis of lyapunov's direct method, and it is observed that the system is globally uniformly asymptotically stable.

6.2. Recommendation

In this thesis work, the overall system is modeled applying Euler Lagrange equation of motion. Mathematical analysis of robot manipulator, specifically parallel ones are so sophisticated and needs higher nonlinear mathematics investigation. Therefore, it is recommended to use software packages which have modeling capability. ADAMS Simulation Package is a powerful modeling and simulating environment that lets one build, simulate, refine, and ultimately optimize mechanical systems.

6.3. Future Work

In the future, it is recommended to explore further investigation of the passivity principles on the area of applications like, robotic systems, aircraft systems, energy generating systems, generally, in different electro mechanical systems. Besides focusing on passivity principles, it is advised to use an advanced controllers like, adaptive controller and robust controller on the basis of passivation.

Reference

[1] Richard M. Murray, 'Mathematical Introduction to Robotic Manipulation', University of California, Berkeley, 1994

[2] Neil Munro, Ph.d., D.sc. 'Robot Manipulator Control Theory and Practice', second edition, University of Manchester Institute of Science and Technology Manchester, United Kingdom.

[3] Serdar Küçük, 'serial and parallel robot manipulators – kinematics, dynamics, control and optimization', 2012

[4] A. Ghobakhloo, M. Eghtesad and M. Azadi, "Position Control of a Stewart Gough Platform Using Inverse Dynamics Method with Full Dynamics", IEEE publication, 2006.

[5] D.Stewart, "review of stewart platforms, chapter 1".

[6] Nawal Azoui, Lamir Saidi,"passivity based adaptive control of robotic manipulators electrically controlled", international journal of sciences and technology, Vol. 34, 2011.

[7] Houssem Abdellatif, Jens Kotlarski, Tobias Ortmaier and Bodo Heimann, "Practical Model-based and Robust Control of Parallel Manipulators Using Passivity and Sliding Mode Theory".

[8] Philip J.Houdek II, design and implementation issues for stewart platform configuration machine tools, *Faculty of Mechanical Engineering,Boston University,1990.*

[9] Selçuk Kizir and Zafer Bingul, "position control and trajectory tracking of the Stewart platform", Mechatronics Engineering, Kocaeli University Turkey.

[10] Assoc. Prof. *Dr Mohamad Kasim Abdul Jalil,* design and development of 6-dof motion platform for vehicle driving simulator, *Faculty of Mechanical Engineering, Universiti Teknologi Malaysia*

[11] Fan Liu "Trajectory Tracking of Robot Manipulators Using Linear and Nonlinear PD-type Controllers".

[12] M. Velasco-Villa, H. Rodríguez-Cortés, I. Estrada-Sanchez, "Dynamic Trajectory-Tracking Control of an Omni directional Mobile Robot Based on a Passive Approach".

[13] Antinio loria and Henk Nijmeijer, "passivity based control", robotics and automation, vol. XIII.

[14] Clement Gosselin "kinematic analysis, optimization and programming of parallel robotic manipulators", montreal, Canada, 1985.

[15] Romeo Ortega, Antonio Loria, Per Johan Nicklasson and Hebertt Sira-Ramfrez, 'Passivity-based Control of Euler-Lagrange Systems', Mechanical, Electrical and Electromechanically Applications.

[16] Anders Lohmann MadsenMadsen, "design of Stewart platform for wave compensation", master's thesis, 2012.

[17] Kotyczka, Paul. "passivity based trajectory tracking control with predefined local linear error dynamics", American control conference, 2010.

[18] " http://www.coriolis/centrifugal forces.com"

[19] John J. Craig, "introduction to robotics, mechanics and control", third edition.

[20] Youhong Gong, "design analysis of a Stewart platform for vehicle emulator systems", Massachusetts institute of technology, January, 1992.

Appendix

Appendix A: proof of stability of Stewart platform

The performance of the overall system is largely analyzed based on the energy balance of the system and hence its stability can be checked by using lyapunov's direct method.

Let us select the candidate lyapunov function V of the state q with continuous first order derivative so that:

$$V(q(t)) = \frac{1}{2} B\dot{q}^2 + \frac{1}{2} K\tilde{q}^2 \tag{A.1}$$

Where, B and K are positive constants and $\tilde{q}$ is the equilibrium point deviation (given by $q - q^*$ where, q^* is the desired equilibrium point).

It is known that the selected lyapunov function is positive definite. If the time derivative $\dot{V}$ is negative definite, then it is guaranteed that the equilibrium at the origin is globally uniform asymptotically stable in the sense of lyapunov (lyapunov's direct method).

Therefore,

$$\dot{V}(q(t)) = B\dot{q}\ddot{q} + K\dot{q}e, \text{ where, } e = q - q^* \tag{A.2}$$

From equation (4.7), we have $\ddot{q} = M^{-1}(-N\dot{q} - G)$.

This implies,

$$\dot{V}(q(t)) = -BM^{-1}N\dot{q}^2 + (Ke - BM^{-1}G)\dot{q} \tag{A.3}$$

Then by assuming e as a very small number, equation (A.3) can be rewritten as:

$$\dot{V}(q(t)) = -BM^{-1}N\dot{q}^2 - BM^{-1}G\dot{q} \tag{A.4}$$

$$< 0$$

Appendix B: Matlab function for ADAMS/MATLAB interface

```
% Adams / MATLAB Interface
system('taskkill /IM scontrols.exe /F >NUL');clc;
global ADAMS_sysdir; % used by setup_rtw_for_adams.m
global ADAMS_host; % used by start_adams_daemon.m
machine=computer;
datestr(now)
if strcmp(machine, 'SOL2')
   arch = 'solaris32';
elseif strcmp(machine, 'SOL64')
   arch = 'solaris32';
elseif strcmp(machine, 'GLNX86')
   arch = 'linux32';
elseif strcmp(machine, 'GLNXA64')
   arch = 'linux64';
elseif strcmp(machine, 'PCWIN')
   arch = 'win32';
elseif strcmp(machine, 'PCWIN64')
   arch = 'win64';
else
   disp( '%%% Error : Platform unknown or unsupported by Adams/Controls.' ) ;
   arch = 'unknown_or_unsupported';
   return
end
if strcmp(arch,'win64')
   [flag, topdir]=system('adams2014_SE_x64 -top');
else
   [flag, topdir]=system('adams2014_SE -top');
end
if flag == 0
  temp_str=strcat(topdir, '/controls/', arch);
  addpath(temp_str)
  temp_str=strcat(topdir, '/controls/', 'matlab');
  addpath(temp_str)
  temp_str=strcat(topdir, '/controls/', 'utils');
  addpath(temp_str)
  ADAMS_sysdir = strcat(topdir, '');
else
  addpath( 'C:\MSC~1.SOF\ADAMS_~1\2014\controls/win32' ) ;
  addpath( 'C:\MSC~1.SOF\ADAMS_~1\2014\controls/win32' ) ;
  addpath( 'C:\MSC~1.SOF\ADAMS_~1\2014\controls/matlab' ) ;
  addpath( 'C:\MSC~1.SOF\ADAMS_~1\2014\controls/utils' ) ;
  ADAMS_sysdir = 'C:\MSC~1.SOF\ADAMS_~1\2014\' ;
end
ADAMS_exec = '' ;
ADAMS_host = 'pc' ;
ADAMS_cwd ='C:\Users\user\Desktop\modelll'  ;
ADAMS_prefix = 'model11' ;
ADAMS_static = 'no' ;
ADAMS_solver_type = 'C++' ;
if exist([ADAMS_prefix,'.adm']) == 0
   disp( ' ' ) ;
   disp( '%%% Warning : missing ADAMS plant model file(.adm) for Co-
simulation or Function Evaluation.' ) ;
```

```
   disp( '%%% If necessary, please re-export model files or copy the exported
plant model files into the' ) ;
   disp( '%%% working directory.  You may disregard this warning if the Co-
simulation/Function Evaluation' ) ;
   disp( '%%% is TCP/IP-based (running Adams on another machine), or if
setting up MATLAB/Real-Time Workshop' ) ;
   disp( '%%% for generation of an External System Library.' ) ;
   disp( ' ' ) ;
end
ADAMS_init = '' ;
ADAMS_inputs  = 'SF_1!SF_2!SF_3!SF_4!SF_5!SF_6' ;
ADAMS_outputs = 'SL_1!SL_2!SL_3!SL_4!SL_5!SL_6' ;
ADAMS_pinput = 'model11.ctrl_pinput' ;
ADAMS_poutput = 'model11.ctrl_poutput' ;
ADAMS_uy_ids  = [
                    20
                    21
                    22
                    23
                    24
                    25
                    26
                    27
                    28
                    29
                    30
                    31
                ] ;
ADAMS_mode   = 'non-linear' ;
tmp_in  = decode( ADAMS_inputs  ) ;
tmp_out = decode( ADAMS_outputs ) ;
disp( ' ' ) ;
disp( '%%% INFO : ADAMS plant actuators names :' ) ;
disp( [int2str([1:size(tmp_in,1)]'),blanks(size(tmp_in,1))',tmp_in] ) ;
disp( '%%% INFO : ADAMS plant sensors   names :' ) ;
disp( [int2str([1:size(tmp_out,1)]'),blanks(size(tmp_out,1))',tmp_out] ) ;
disp( ' ' ) ;
clear tmp_in tmp_out ;
% Adams / MATLAB Interface
```

Appendix C: Matlab Script for realizing the mathematical model of Stewart platform

```
%some constants
clc
res=7.10;%terminal resistance;
indu=265e-6;%the rotor inductance
pitch=0.001;
gearratio=1;
Ktt=26.10e-3;
Kt=[Ktt 0 0 0 0 0;
   0 Ktt 0 0 0 0;
   0 0 Ktt 0 0 0;
   0 0 0 Ktt 0 0;
   0 0 0 0 Ktt 0;
   0 0 0 0 0 Ktt];%the torque constant
Kbb=2.730e-3;
Kb=[Kbb 0 0 0 0 0;
   0 Kbb 0 0 0 0;
   0 0 Kbb 0 0 0;
   0 0 0 Kbb 0 0;
   0 0 0 0 Kbb 0;
   0 0 0 0 0 Kbb];%back-emf constant
PPP=gearratio*2*3.14/pitch;
PP=[PPP 0 0 0 0 0;
   0 PPP 0 0 0 0;
   0 0 PPP 0 0 0;
   0 0 0 PPP 0 0;
   0 0 0 0 PPP 0;
   0 0 0 0 0 PPP];
%inverse kinematics of the stewart platform
%calculation of rotation matrix by taking orientation inputs(a,b,g)
input('a=')
a=ans/180*3.14;
input('b=')
b=ans/180*3.14;
input('g=')
g=ans/180*3.14;
A= cos(b)*cos(g);
B= cos(g)*sin(a)*sin(b)-cos(a)*sin(g);
C= sin(a)*sin(g)+cos(a)*cos(g)*sin(b);
D= cos(b)*sin(g);
E= cos(a)*cos(g)+sin(a)*sin(b)*sin(g);
F= cos(a)*sin(b)*sin(g)-cos(a)*sin(g);
G= -sin(b);
H= cos(b)*sin(a);
I= cos(a)*cos(b);
Rxyz=[A B C;D E F;G H I]
%position vectors
input('x=')
x=ans;
input('y=')
```

```
y=ans;
input('z=')
z=ans;
ll1=0.22;%length of fixed leg
ll2=0.05;%length of moving leg
r_bass=0.14663308 ;
r_platform=0.09433988 ;
tetta=acos(0.0522932/ll1);%r_bass - r_platform = 0.0522932m
pz=ll1*sin(tetta);%height of the platform
zz=z+pz;
P=[x;y;zz]
%base intial position for leg 1
Bx1=0.1451;
By1=-0.0209;
Bz1=0;
B1=[Bx1;By1;Bz1];
%intial platform position for leg 1
GTx1=0.0641;
GTy1=-0.0691;
GTz1=0;
GT1=[GTx1;GTy1;GTz1];
%intial base position for leg 2
Bx2=0.1451;
By2=0.0209;
Bz2=0;
B2=[Bx2;By2;Bz2];
%intial platform position for leg 2
GTx2=0.0641;
GTy2=0.0691;
GTz2=0;
GT2=[GTx2;GTy2;GTz2];
%intial base position for leg 3
Bx3=-0.0544;
By3=0.1361;
Bz3=0;
B3=[Bx3;By3;Bz3];
%intial platform position for leg 3
GTx3=0.0278;
GTy3=0.0901;
GTz3=0;
GT3=[GTx3;GTy3;GTz3];
%intial base position for leg 4
Bx4=-0.0906;
By4=0.1152;
Bz4=0;
B4=[Bx4;By4;Bz4];
%intial platform position for leg 4
GTx4=-0.0919;
GTy4=0.0209;
GTz4=0;
GT4=[GTx4;GTy4;GTz4];
%intial base position for leg 5
Bx5=-0.0906;
By5=-0.1152;
Bz5=0;
B5=[Bx5;By5;Bz5];
```

```
%intial platform position for leg 5
GTx5=-0.0919;
GTy5=-0.0209;
GTz5=0;
GT5=[GTx5;GTy5;GTz5];
%intial base position for leg 6
Bx6=-0.0544;
By6=-0.1361;
Bz6=0;
B6=[Bx6;By6;Bz6];
%intial platform position for leg 6
GTx6=0.0278;
GTy6=-0.0901;
GTz6=0;
GT6=[GTx6;GTy6;GTz6];
%calculation of the six legs length
l1=Rxyz*GT1+P-B1;
L1=norm(l1,2);
l2=Rxyz*GT2+P-B2;
L2=norm(l2,2);
l3=Rxyz*GT3+P-B3;
L3=norm(l3,2);
l4=Rxyz*GT4+P-B4;
L4=norm(l4,2);
l5=Rxyz*GT5+P-B5;
L5=norm(l5,2);
l6=Rxyz*GT6+P-B6;
L6=norm(l6,2);
LL=[L1;L2;L3;L4;L5;L6]
%workspace calculation
Z_comp5=(ll1+ll2)*sin(tetta);
XY_comp5=-(ll1+ll2)*cos(tetta);
X_comp5=XY_comp5*cos(51.79*3.14/180);
Y_comp5=XY_comp5*sin(51.79*3.14/180);
L_max5=[X_comp5;Y_comp5;Z_comp5];
P_max5=L_max5-Rxyz*GT5+B5%maximum platform position for the given orientation
z_comp5=ll1*sin(tetta);
xy_comp5=-ll1*cos(tetta);
x_comp5=xy_comp5*cos(51.79*3.14/180);
y_comp5=xy_comp5*sin(51.79*3.14/180);
L_min5=[x_comp5;y_comp5;z_comp5];
P_min5=L_min5-Rxyz*GT5+B5%minimum platform position for the given orientation
Z_comp6=(ll1+ll2)*sin(tetta);
XY_comp6=-(ll1+ll2)*cos(tetta);
X_comp6=XY_comp6*cos(68.2*3.14/180);
Y_comp6=XY_comp6*sin(68.2*3.14/180);
L_max6=[X_comp6;Y_comp6;Z_comp6];
P_max6=L_max6-Rxyz*GT6+B6%maximum platform position for the given orientation
z_comp6=ll1*sin(tetta);
xy_comp6=-ll1*cos(tetta);
x_comp6=xy_comp6*cos(68.2*3.14/180);
y_comp6=xy_comp6*sin(68.2*3.14/180);
L_min6=[x_comp6;y_comp6;z_comp6];
P_min6=L_min6-Rxyz*GT6+B6%minimum platform position for the given orientation
Z_comp1=(ll1+ll2)*sin(tetta);
XY_comp1=-(ll1+ll2)*cos(tetta);
```

```
X_comp1=XY_comp1*cos(171.79*3.14/180);
Y_comp1=XY_comp1*sin(171.79*3.14/180);
L_max1=[X_comp1;Y_comp1;Z_comp1];
P_max1=L_max1-Rxyz*GT1+B1%maximum platform position for the given orientation
z_comp1=ll1*sin(tetta);
xy_comp1=-ll1*cos(tetta);
x_comp1=xy_comp1*cos(171.79*3.14/180);
y_comp1=xy_comp1*sin(171.79*3.14/180);
L_min1=[x_comp1;y_comp1;z_comp1];
P_min1=L_min1-Rxyz*GT1+B1%minimum platform position for the given orientation
Z_comp2=(ll1+ll2)*sin(tetta);
XY_comp2=-(ll1+ll2)*cos(tetta);
X_comp2=XY_comp2*cos(188.2*3.14/180);
Y_comp2=XY_comp2*sin(188.2*3.14/180);
L_max2=[X_comp2;Y_comp2;Z_comp2];
P_max2=L_max2-Rxyz*GT2+B2%maximum platform position for the given orientation
z_comp2=ll1*sin(tetta);
xy_comp2=-ll1*cos(tetta);
x_comp2=xy_comp2*cos(188.2*3.14/180);
y_comp2=xy_comp2*sin(188.2*3.14/180);
L_min2=[x_comp2;y_comp2;z_comp2];
P_min2=L_min2-Rxyz*GT2+B2%minimum platform position for the given orientation
Z_comp3=(ll1+ll2)*sin(tetta);
XY_comp3=-(ll1+ll2)*cos(tetta);
X_comp3=XY_comp3*cos(291.8*3.14/180);
Y_comp3=XY_comp3*sin(291.8*3.14/180);
L_max3=[X_comp3;Y_comp3;Z_comp3];
P_max3=L_max3-Rxyz*GT3+B3%maximum platform position for the given orientation
z_comp3=ll1*sin(tetta);
xy_comp3=-ll1*cos(tetta);
x_comp3=xy_comp3*cos(291.8*3.14/180);
y_comp3=xy_comp3*sin(291.8*3.14/180);
L_min3=[x_comp3;y_comp3;z_comp3];
P_min3=L_min3-Rxyz*GT3+B3%minimum platform position for the given orientation
Z_comp4=(ll1+ll2)*sin(tetta);
XY_comp4=-(ll1+ll2)*cos(tetta);
X_comp4=XY_comp4*cos(308.2*3.14/180);
Y_comp4=XY_comp4*sin(308.2*3.14/180);
L_max4=[X_comp4;Y_comp4;Z_comp4];
P_max4=L_max4-Rxyz*GT4+B4%maximum platform position for the given orientation
z_comp4=ll1*sin(tetta);
xy_comp4=-ll1*cos(tetta);
x_comp4=xy_comp4*cos(308.2*3.14/180);
y_comp4=xy_comp4*sin(308.2*3.14/180);
L_min4=[x_comp4;y_comp4;z_comp4];
P_min4=L_min4-Rxyz*GT4+B4%minimum platform position for the given orientation
%<<<<<calculation of the jacobian matrix>>>>>>>.
u1=(l1/L1)';
u2=(l2/L2)';
u3=(l3/L3)';
u4=(l4/L4)';
u5=(l5/L5)';
u6=(l6/L6)';
J1=[u1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 u2 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 u3 0 0 0 0 0 0 0 0 0 0 0;
```

```
   0 0 0 0 0 0 0 0 0 u4 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 u5 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 u6];
S=[0 -z y;z 0 -x;-y x 0];
Iden=[1 0 0;0 1 0;0 0 1];
Rx=[1 0 0;0 cos(a) -sin(a);0 sin(a) cos(a)];
Ry=[cos(b) 0 sin(b);0 1 0;-sin(b) 0 cos(b)];
Rz=[cos(g) -sin(g) 0;sin(g) cos(g) 0;0 0 1];
w1=Ry*S*Rx*Rz*GT1;
w2=S*Ry*Rx*Rz*GT1;
w3=Ry*Rx*S*Rz*GT1;
w4=Ry*S*Rx*Rz*GT2;
w5=S*Ry*Rx*Rz*GT2;
w6=Ry*Rx*S*Rz*GT2;
w7=Ry*S*Rx*Rz*GT3;
w8=S*Ry*Rx*Rz*GT3;
w9=Ry*Rx*S*Rz*GT3;
q1=Ry*S*Rx*Rz*GT4;
q2=S*Ry*Rx*Rz*GT4;
q3=Ry*Rx*S*Rz*GT4;
q4=Ry*S*Rx*Rz*GT5;
q5=S*Ry*Rx*Rz*GT5;
q6=Ry*Rx*S*Rz*GT5;
q7=Ry*S*Rx*Rz*GT6;
q8=S*Ry*Rx*Rz*GT6;
q9=Ry*Rx*S*Rz*GT6;
J2=[Iden w1 w2 w3;Iden w4 w5 w6;Iden w7 w8 w9;Iden q1 q2 q3;Iden q4 q5 q6;Iden q7 q8 q9];
Jac=J1*J2
%determination of inertia matrix and gravity matrix of the platform
Ix=0.0025;
Iy=0.0025;
Iz=0.005;
Mu=1.1324;%mass of the moving/upper platform
m1=0.4279;%mass of the fixed leg
m2=0.1228;%mass of the movable leg
del=ll1/2;
ll=(del*m1*ll1-0.5*m2*ll2)/(m1+m2);
mm=m2/(m1+m2);
h1=(ll/L1+mm)^2;
h2=(ll/L2+mm)^2;
h3=(ll/L3+mm)^2;
h4=(ll/L4+mm)^2;
h5=(ll/L5+mm)^2;
h6=(ll/L6+mm)^2;
k1=h1-(mm)^2;
k2=h2-(mm)^2;
k3=h3-(mm)^2;
k4=h4-(mm)^2;
k5=h5-(mm)^2;
k6=h6-(mm)^2;
hh1=h1+h2;
hh2=h3+h4;
hh3=h5+h6;
%determination of inertia matrix,upper platform
xx=(cos(a))^2*Ix*(sin(g))^2+(cos(a))^2*Iy*(cos(g))^2+Iz*(sin(a))^2;
Mup=[Mu 0 0 0 0 0;
```

```
   0 Mu 0 0 0 0;
   0 0 Mu 0 0 0;
   0 0 0 Ix*(cos(g))^2+Iy*(sin(g))^2 (Ix-Iy)*cos(a)*cos(g)*sin(g) 0;
   0 0 0 (Ix-Iy)*cos(a)*cos(g)*sin(g) xx -Iz*(sin(a))^2;
   0 0 0 0 -Iz*(sin(a))^2 Iz];
%determination of inertia matrix,legs
HH=[hh1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 hh1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 hh1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 hh1 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 hh1 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 hh1 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 hh2 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 hh2 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 hh2 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 hh2 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 hh2 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 hh2 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 hh3 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 hh3 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 hh3 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 hh3 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 hh3 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 hh3];
K=[k1 0 0 0 0 0;
   0 k2 0 0 0 0;
   0 0 k3 0 0 0;
   0 0 0 k4 0 0;
   0 0 0 0 k5 0;
   0 0 0 0 0 k6];
Mlegs=(m1+m2)*J2'*(HH-J1'*K*J1)*J2;
%total inertia matrix
M=Mup+Mlegs
%determination of gravity matrix,upper platform
Gup=[0;0;Mu*9.8;0;0;0];
%determination of gravity matrix,legs
l1x=[1;0;0];
L1x=norm(l1x,2);
l1y=[0;1;0];
L1y=norm(l1y,2);
l1z=[0;0;1];
L1z=norm(l1z,2);
l2x=[1;0;0];
L2x=norm(l2x,2);
l2y=[0;1;0];
L2y=norm(l2y,2);
l2z=[0;0;1];
L2z=norm(l2z,2);
l3x=[1;0;0];
L3x=norm(l3x,2);
l3y=[0;1;0];
L3y=norm(l3y,2);
l3z=[0;0;1];
L3z=norm(l3z,2);
l4x=[1;0;0];
L4x=norm(l4x,2);
```

```
l4y=[0;1;0];
L4y=norm(l4y,2);
l4z=[0;0;1];
L4z=norm(l4z,2);
l5x=[1;0;0];
L5x=norm(l5x,2);
l5y=[0;1;0];
L5y=norm(l5y,2);
l5z=[0;0;1];
L5z=norm(l5z,2);
l6x=[1;0;0];
L6x=norm(l6x,2);
l6y=[0;1;0];
L6y=norm(l6y,2);
l6z=[0;0;1];
L6z=norm(l6z,2);
Rxyzx=[0 cos(g)*cos(a)*sin(b)+sin(a)*sin(g) cos(a)*sin(g)-sin(a)*cos(g)*sin(b);
   0 -sin(a)*cos(g)+cos(a)*sin(b)*sin(g) -sin(a)*sin(b)*sin(g)-cos(g)*cos(a);
   0 cos(b)*cos(a) -sin(a)*cos(b)];
Rxyzy=[-sin(b)*cos(a) cos(g)*sin(a)*cos(b) cos(a)*cos(g)*cos(b);
   -sin(b)*sin(g) sin(a)*cos(b)*sin(g) cos(a)*cos(b)*sin(g);
   -cos(b) -sin(b)*sin(a) -cos(a)*sin(b)];
Rxyzz=[-cos(b)*sin(g) -sin(g)*sin(a)*sin(b)-cos(a)*cos(g) sin(a)*cos(g)-cos(a)*sin(g)*sin(b);
   cos(b)*cos(g) -cos(a)*sin(g)+sin(a)*sin(b)*cos(g) cos(a)*sin(b)*cos(g)+sin(g)*sin(a);
   0 0 0];
l1a=Rxyzx*GT1;
L1a=norm(l1a,2);
l1b=Rxyzy*GT1;
L1b=norm(l1b,2);
l1g=Rxyzz*GT1;
L1g=norm(l1g,2);
l2a=Rxyzx*GT2;
L2a=norm(l2a,2);
l2b=Rxyzy*GT2;
L2b=norm(l2b,2);
l2g=Rxyzz*GT2;
L2g=norm(l2g,2);
l3a=Rxyzx*GT3;
L3a=norm(l3a,2);
l3b=Rxyzy*GT3;
L3b=norm(l3b,2);
l3g=Rxyzz*GT3;
L3g=norm(l3g,2);
l4a=Rxyzx*GT4;
L4a=norm(l4a,2);
l4b=Rxyzy*GT4;
L4b=norm(l4b,2);
l4g=Rxyzz*GT4;
L4g=norm(l4g,2);
l5a=Rxyzx*GT5;
L5a=norm(l5a,2);
l5b=Rxyzy*GT5;
L5b=norm(l5b,2);
l5g=Rxyzz*GT5;
L5g=norm(l5g,2);
l6a=Rxyzx*GT6;
```

```
L6a=norm(l6a,2);
l6b=Rxyzy*GT6;
L6b=norm(l6b,2);
l6g=Rxyzz*GT6;
L6g=norm(l6g,2);
Rx=[1 0 0;0 cos(a) -sin(a);0 sin(a) cos(a)];
Ry=[cos(b) 0 sin(b);0 1 0;-sin(b) 0 cos(b)];
Rz=[cos(g) -sin(g) 0;sin(g) cos(g) 0;0 0 1];
Rzxy=Rz'*Rx'*Ry';
ZT1=[0 0 1]*Rzxy*GT1;
ZT2=[0 0 1]*Rzxy*GT2;
ZT3=[0 0 1]*Rzxy*GT3;
ZT1x=0;
ZT1y=0;
ZT1z=0;
ZT2x=0;
ZT2y=0;
ZT2z=0;
ZT3x=0;
ZT3y=0;
ZT3z=0;
ZT1a=[0 0 1]*Rz'*[0 0 0;0 -sin(a) -cos(a);0 cos(a) -sin(a)]'*Ry'*GT1;
ZT1b=[0 0 1]*Rz'*Rx'*[-sin(b) 0 cos(b);0 0 0;-cos(b) 0 -sin(b)]'*GT1;
ZT1g=[0 0 1]*[-sin(g) -cos(g) 0;cos(g) -sin(g) 0;0 0 0]'*Rx'*Ry'*GT1;
ZT2a=[0 0 1]*Rz'*[0 0 0;0 -sin(a) -cos(a);0 cos(a) -sin(a)]'*Ry'*GT2;
ZT2b=[0 0 1]*Rz'*Rx'*[-sin(b) 0 cos(b);0 0 0;-cos(b) 0 -sin(b)]'*GT2;
ZT2g=[0 0 1]*[-sin(g) -cos(g) 0;cos(g) -sin(g) 0;0 0 0]'*Rx'*Ry'*GT2;
ZT3a=[0 0 1]*Rz'*[0 0 0;0 -sin(a) -cos(a);0 cos(a) -sin(a)]'*Ry'*GT3;
ZT3b=[0 0 1]*Rz'*Rx'*[-sin(b) 0 cos(b);0 0 0;-cos(b) 0 -sin(b)]'*GT3;
ZT3g=[0 0 1]*[-sin(g) -cos(g) 0;cos(g) -sin(g) 0;0 0 0]'*Rx'*Ry'*GT3;
Gx=(m1+m2)*9.8*(ll*(1/L2^2*L2x+1/L1^2*L1x)*(pz+ZT1)+ll*(1/L4^2*L4x+1/L3^2*L3x)*(pz+ZT2)...
+ll*(1/L6^2*L6x+1/L5^2*L5x)*(pz+ZT3))+(m1+m2)*9.8*((ll*(1/L2+1/L1)+2*m2/(m1+m2))*ZT1x...
+(ll*(1/L4+1/L3)+2*m2/(m1+m2))*ZT2x+(ll*(1/L6+1/L5)+2*m2/(m1+m2))*ZT3x);
Gy=(m1+m2)*9.8*(ll*(1/L2^2*L2y+1/L1^2*L1y)*(pz+ZT1)+ll*(1/L4^2*L4y+1/L3^2*L3y)*(pz+ZT2)...
+ll*(1/L6^2*L6y+1/L5^2*L5y)*(pz+ZT3))+(m1+m2)*9.8*((ll*(1/L2+1/L1)+2*m2/(m1+m2))*ZT1y...
+(ll*(1/L4+1/L3)+2*m2/(m1+m2))*ZT2y+(ll*(1/L6+1/L5)+2*m2/(m1+m2))*ZT3y);
Gz=(m1+m2)*9.8*(ll*(1/L2^2*L2z+1/L1^2*L1z)*(pz+ZT1)+ll*(1/L4^2*L4z+1/L3^2*L3z)*(pz+ZT2)...
+ll*(1/L6^2*L6z+1/L5^2*L5z)*(pz+ZT3))+(m1+m2)*9.8*((ll*(1/L2+1/L1)+2*m2/(m1+m2))*ZT1z...
+(ll*(1/L4+1/L3)+2*m2/(m1+m2))*ZT2z+(ll*(1/L6+1/L5)+2*m2/(m1+m2))*ZT3z);
Ga=(m1+m2)*9.8*(ll*(1/L2^2*L2a+1/L1^2*L1a)*(pz+ZT1)+ll*(1/L4^2*L4a+1/L3^2*L3a)*(pz+ZT2)...
+ll*(1/L6^2*L6a+1/L5^2*L5a)*(pz+ZT3))+(m1+m2)*9.8*((ll*(1/L2+1/L1)+2*m2/(m1+m2))*ZT1a...
+(ll*(1/L4+1/L3)+2*m2/(m1+m2))*ZT2a+(ll*(1/L6+1/L5)+2*m2/(m1+m2))*ZT3a);
Gb=(m1+m2)*9.8*(ll*(1/L2^2*L2b+1/L1^2*L1b)*(pz+ZT1)+ll*(1/L4^2*L4b+1/L3^2*L3b)*(pz+ZT2)...
+ll*(1/L6^2*L6b+1/L5^2*L5b)*(pz+ZT3))+(m1+m2)*9.8*((ll*(1/L2+1/L1)+2*m2/(m1+m2))*ZT1b...
+(ll*(1/L4+1/L3)+2*m2/(m1+m2))*ZT2b+(ll*(1/L6+1/L5)+2*m2/(m1+m2))*ZT3b);
Gg=(m1+m2)*9.8*(ll*(1/L2^2*L2g+1/L1^2*L1g)*(pz+ZT1)+ll*(1/L4^2*L4g+1/L3^2*L3g)*(pz+ZT2)...
+ll*(1/L6^2*L6g+1/L5^2*L5g)*(pz+ZT3))+(m1+m2)*9.8*((ll*(1/L2+1/L1)+2*m2/(m1+m2))*ZT1g...
+(ll*(1/L4+1/L3)+2*m2/(m1+m2))*ZT2g+(ll*(1/L6+1/L5)+2*m2/(m1+m2))*ZT3g);
Glegs=[Gx;Gy;Gz;Ga;Gb;Gg];
%gravity matrix,total
Grav=Glegs+Gup
%inertia,gravity and coriolis/centifugal matrix of the moving platform+actuator
Js=0.002091e-3;%moment of inertia of ball screw
Jm=0.58e-6;%moment of inertia of motor
bs=0.11796e-3;%the viscous damping coefficient of the ball-screw
bm=0.0016430e-3;%the viscous damping coefficient of the motor
```

```
Maa=2*3.14/(gearratio*pitch)*(Js+gearratio^2*Jm)*[1 0 0 0 0 0;0 1 0 0 0 0;0 0 1 0 0 0;0 0 0 1 0 0;0 0 0 0 1 0;0 0 0 0 0 1];
Naa=2*3.14*(gearratio*pitch)*(bs+gearratio^2*bm)*[1 0 0 0 0 0;0 1 0 0 0 0;0 0 1 0 0 0;0 0 0 1 0 0;0 0 0 0 1 0;0 0 0 0 0 1];
Kaa=pitch/(gearratio*2*3.14)*[1 0 0 0 0 0;0 1 0 0 0 0;0 0 1 0 0 0;0 0 0 1 0 0;0 0 0 0 1 0;0 0 0 0 0 1];
%inertia matrix,gravity matrix and coriolis matrix of moving platform+actuator
MMM=Kaa*inv(Jac)'*M+Maa*Jac;
MM=inv(MMM)
CC=Naa*Jac
GG=Kaa*inv(Jac)'*Grav
%setting controller gains
kd=diag([10 20 30 40 50 60])
kp=diag([100 200 300 400 500 600])
```

Declaration

I, the under signed here, declare that this thesis is my original work not presented for master of degree at different universities before, and the supporting materials of the thesis have been fully acknowledged.

Tariku Sinshaw Tamir _____

Name signature

Place: Addis Ababa Institute of Technology, Addis Ababa University, Addis Ababa Date of Submission: _____

This thesis has been submitted for examination with my approval as a university advisor.

Mr.Andinet Negash _____

Advisor's Name signature