Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Passivity Based Robust Controller Design for 6-DOF Stewart Platform Manipulator Used in 4D Ultrasound Imaging

A thesis Submitted to the School of Electrical and Computer Engineering of Addis Ababa Institute of Technology, School of Graduate Studies, Addis Ababa University in partial fulfillment of the Requirement for the Degree of **Masters of Science in Control Engineering.**

By

Dessale Akele

Advisor: Dereje Shiferaw (Ph.D.)

November, 2020

Addis Ababa, Ethiopia

Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Passivity Based Robust Controller Design for 6-DOF Stewart Platform Manipulator Used in 4D Ultrasound Imaging

A thesis Submitted to the School of Electrical and Computer Engineering of Addis Ababa Institute of Technology, School of Graduate Studies, Addis Ababa University in partial fulfillment of the Requirement for the Degree of **Masters of Science in Control Engineering.**

By

Dessale Akele

Approval by Board of Examiners

Chairman, Dept. Graduate Committee	Signature
Dr. Dereje Shiferaw	
Advisor's Name	Signature
Internal Examiner	Signature
External Examiner	Signature

DECLARATION

I, the undersigned, declare that this thesis is my original work, has not been presented for a degree in this or any other universities, all sources of materials used for this thesis work have been fully acknowledged.

Dessale Akele _____

Name Signature

Place: Addis Ababa Institute of Technology, Addis Ababa University, Addis Ababa

Date of submission November, 2020

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

Dr. Dereje Shiferaw _____

Advisor's Name Signature

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my advisor, Dr. Dereje Shiferaw, for his guidance, support and encouragement during the course of the thesis work. He has been an excellent advisor demonstrating good patience and enormous help throughout my thesis study.

ABSTRACT

Stewart platform manipulators are six degrees of freedom robots having high structural rigidity, positioning accuracy, and are preferred over serial robots for applications such as robotic surgery, robotic ultrasound, and so on. However, the absence of robust controllers, their nonlinear dynamics and uncertainties due to model inaccuracies, parameter variations, and external disturbance, has limited the real application of the manipulators.

Passivity based integral sliding mode control is one of the techniques available to design robust controllers and has been implemented for trajectory tracking of the Stewart platform. However, the robustness of passivity based integral sliding mode control has been achieved at the result of high-frequency switching of discontinuous control signal. The discontinuity in the feedback control is undesirable for practical applications because of chattering which causes the wear and tear of the mechanical actuators. In this thesis, the integral sliding mode control law is modified by replacing the discontinuous control part with super twisting control, and proportional derivative plus control is designed based on the passivation principle so that the closed-loop system becomes asymptotically stable. Particle swarm optimization is used to determine the optimal gain matrices values of passivity-based smooth integral sliding mode controller using an integral time absolute error as an optimization problem. The passivity-based smooth integral sliding mode controller has implemented using MATLAB/Simulink, and the result has been compared with a passivity-based integral sliding mode controller with different operating conditions. The simulation results show that for the case of a platform with load and disturbance together, the tracking error for six legs achieved by passivity based integral sliding mode controller and passivity-based smooth integral sliding mode controller was 0.1712, 0.0052, 0.1557, 0.0290, 0.0021, 0.0145 and 0.0005, 0.0018, 0.0023, 0.0017, 0.0017, 0.0006 meters respectively.

The simulation results have shown that the designed passivity based smooth integral sliding mode controller performs better in chattering elimination and avoiding uncertainties due to load variation and variable disturbance.

Keywords: Passivity based smooth integral sliding mode control, PD+ control, super twisting control, integral sliding mode control, particle swarm optimization algorithm, Stewart platform manipulator

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

Fb	Reference frame attached to the base
Fp	Reference frame attached to the platform
Bi	Center of the i^{th} universal joint at the base of the Stewart platform
Pi	Center of the i^{th} spherical joint at platform of Stewart platform
Ai	The points of the arm/leg joint on the i^{th} servo in frame Fb
bi	Position vector of the center of universal joints Bi in frame Fb
pi	Position vector of the center of spherical joints Pi in frame Fp
Ei	Position vector of the center of spherical joints Pi in frame Fb
pix	x coordinate of the position vector of the center of spherical joints Pi in frame Fp
piy	y coordinate of the position vector of the center of spherical joints Pi in frame Fp
piz	z coordinate of the position vector of the center of spherical joints Pi in frame Fp
bix	x coordinate of the position vector of the center of universal joints Bi in frame Fb
biy	y coordinate of the position vector of the center of universal joints Bi in frame Fb
biz	z coordinate of the position vector of the center of universal joints Bi in frame Fb
rb	Radius of the base
rp	Radius of the platform
m	The position of the origin of the platform frame with respect to the base frame
Po	Origin of frame Fp
Bo	Origin of frame Fb
R_p^b	Orientation matrix of frame Fp with respect to Fb
α_{bi}	Angle used to specify location of joints in the base
α_{pi}	Angle used to specify location of joints in the platform
α	Euler rotation angle specifying roll angle
β	Euler rotation angle specifying pitch angle
γ	Euler rotation angle specifying yaw angle

αsi	Angle of the ith servo arm relative to x-y plane
βsi	Angle of the ith servo arm relative to the x-axis
X	Cartesian space position and orientation of the moveable platform or end effector
K_1,K_2	Gain matrices of super twisting control signal
Kp	Proportional gain
Kd	Derivative gain constant
Ki	Integral gain constant
K_3	Gain matrix of discontinuous control signal
C_1	Gain matrix of integral sliding surface

LIST OF ABBREVIATIONS

PD+	Proportional derivative plus
DOF	Degree of freedom
PID	Proportional integral derivative
PB-ISMC	Passivity based integral sliding mode control
PB-SISMC	Passivity based smooth integral sliding mode control
US	Ultrasound
2D	Two dimensional
3D	Three dimensional
4D	Four dimensional
SP	Stewart platform
SMC	Sliding mode control
SISO	Single input single output
MIMO	Multiple input multiple output
ISMC	Integral sliding mode control
SISMC	Smooth integral sliding mode control
STC	Super twisting control
CAD	Computer aided design
PSO	Particle swarm optimization
LQR	Linear quadratic regulator
CG	Center of gravity
PD+SISMC	proportional derivative plus smooth integral sliding mode controller

CHAPTER ONE

1. INTRODUCTION

1.1. General

1.1.1. Ultrasound Imaging

Ultrasound (US) imaging or ultrasonography is inexpensive and non-ionizing diagnostic imaging technology used in medicine to visualize internal organs, their size, structure and any pathological lesion using high frequency sound waves [1, 2]. In scanning with ultrasound probe, high frequency sound waves are transmitted to the areas of interest and the returning echoes are recorded and the image can be viewed in 2D, 3D even 4D. 4D, as the name signifies four dimensional which provides high resolution three dimensional ultrasound volumes as a sequence over the time as its 4^{th} dimension. In free-hand scanning, the physician, based on her/his experience and knowledge, manipulates the transducer on the patient's organ and mentally transforms the 2D images into a 3D tissue structure and makes the diagnosis as shown in figure 1.1.

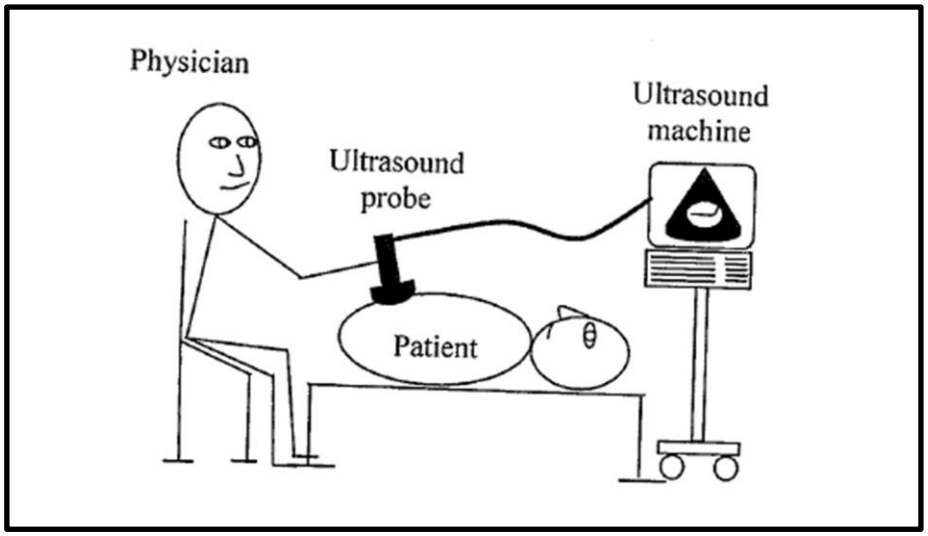


Figure1.1. Conventional ultrasound imaging [3].

Here, in figure 1.1 it can be seen that ultrasonography requires skilled sonographer to hold and manipulate an ultrasound probe through various angles with significant force to obtain quality images. Particularly when treating obese patients, sonographers must apply high forces to complete the procedure efficiently, often resulting in serious injury to the sonographer [3]. To overcome this problem, robotic systems take some of the physical burden from the physician by holding and moving the probe with the appropriate force needed for quality imaging.

1.1.2. Robots Ultrasound

Robots ultrasound can be defined as the combination of ultrasound imaging with a robotic system in medical intervention. With their potential for high precision, dexterity, and repeatability, robots are often uniquely suited for ultrasound applications. Moreover, robots are capable of precisely tracking and manipulating an ultrasound probe, do not suffer from human afflictions such as hand-tremor, fluctuating application forces, which all these make it superior to unassisted, or freehand, techniques. However, robotic systems are complex, which are generally composed of a couple of interconnected systems and have nonlinear and stochastic dynamics which makes the controller design a very challenging task. Therefore, to effectively utilize the advantages of the robot manipulators to the above-mentioned application, a robust and high performance controller is necessary and is the main subject of this thesis. However, the design of a robust and high performance controller needs the study of the mechanical links and joints and their modeling making it the first step in the design of a controller for robotic systems.

There are two basic models that must be derived for any robotic system, the kinematic model and dynamic model. The kinematic model gives the position and velocity of the end effector of the robot as a function of the position and velocity of the joints of the robot, while the dynamic model gives the acceleration of the end effector as a function of the force required at the joints. Both the kinematic and dynamic models depend on the mechanical structure of the robot. Robots can be classified into two based on their mechanical structure, namely serial and parallel robots.

Serial robot: In serial manipulator, the end-effector is linked to the base with a series of linkages and joints all in an open loop chain. The main advantage of these robots is their large work space and their kinematic design which reduce the mathematics of the robot geometry. The drawbacks of these robots are low precision, poor force exertion capability and low payload to weight ratio. Their precision is also very low due to accumulation of the errors in the serially connected links and they suffer from extensive vibration[4]. To solve these problems parallel robots have been proposed.

Parallel robot: A parallel robot is defined as a closed loop mechanism having a fixed base and moveable platform joined by extendable legs or kinematic chains[5]. In parallel manipulators the end effector is connected to the base through multiple links and hence the load is distributed to the links resulting in a higher mass/load ratio. 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. The error in the links will also be averaged out rather than being summed and hence they have a higher precision in positioning. Based on the advantages and disadvantages of parallel robots it can be concluded that the best suitable implementations for such robots include requirements for limited workspace, high accuracy, high agility, and a lightweight and a compact robot. Therefore, they seem perfectly suitable for medical application such as robotic surgery, robotic ultrasound imaging, and so on[4]. One of the most important members of the family of parallel robots is Stewart platform manipulator and has been proposed.

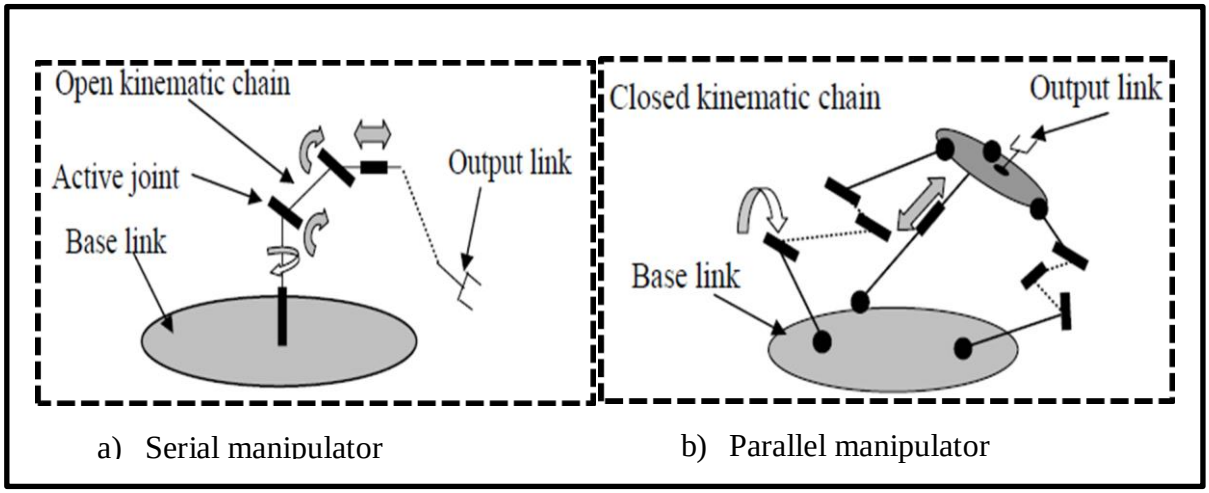


Figure1.2.Robot classification by kinematic chain [4].

1.1.3. Stewart Platform Manipulator

The Stewart platform manipulator has undergone various generalizing modifications from its initial proposal by D. Stewart and as it is understood now, it contains two rigid bodies connected by six extensible legs [6, 7]. The first rigid body is known as the base and is usually fixed while the other one is known as the platform and it is the end effector and is moveable. The extensible legs are joined at the platform by either spherical or universal joints and to the base by universal joints [5]. Six degree of freedom motion is obtained by changing the length of the legs. The Stewart platforms are usually realized with help of prismatic actuators which constitute the length varying legs between the base and the platform but a Stewart platform can be used with any type of prismatic or rotary actuators. Figure 1.3(E) shows a Stewart platform which can be realized with rotary actuators and has been selected in this thesis.

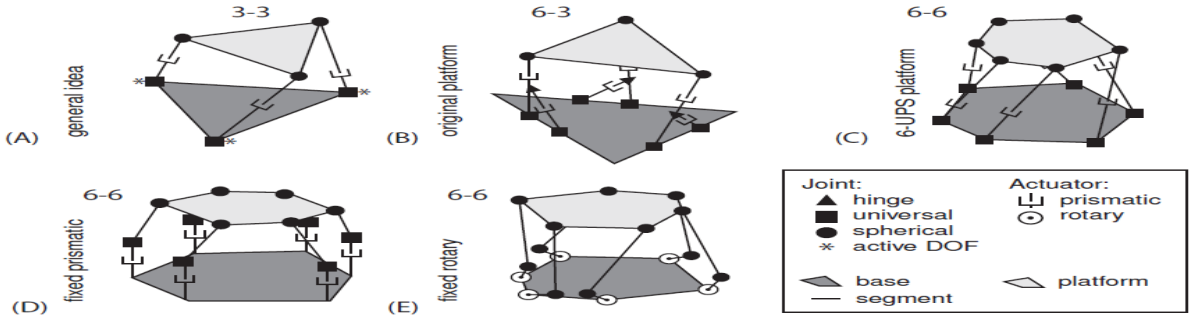


Figure1.3. A diagram depicting several different configuration of Stewart platform [8]

1.4. Literature Review

Jang Ho Cho, Joonho Seo, and Hyun Soo Woo, "Development of Master-Slave Robotic System for Teleoperated Ultrasonography" [9], has been proposed in this paper, the basic structure of the slave robot and master robot is based on Stewart-Gough mechanisms. The workspace of master robot has been analyzed and it covers the work space of the slave robot. However it has not proposed anything about the robustness under external disturbance and unmodeled dynamics of the controller designed.

J. Seo, J. H. Cho, H. Woo, and Y. Lee "Development of Prototype System for Robot-assisted Ultrasound Diagnosis" [10], in this study, the basic structure of the slave robot is designed by Stewart platform and the prototype of the slave robot with US imaging probe developed. Kinematic structure of slave robot and it's the range of motion are explained. The paper is highly focused on to minimize the volume and weight of slave robot as well as satisfies the speed and force for US diagnosis; however it has not proposed anything about optimization of the control effort of the controller designed. In [11] 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. However this control scheme suffers from some problems like high frequency oscillation of the sliding variable near the sliding surface, which is known as chattering, reduction in life of actuator due to the rapid changing discontinuous control action, lack of robustness during reaching phase and sensitivity to unmatched uncertainties[12]. To overcome some of these problems, integral sliding mode control has been proposed by [12-15]. Integral sliding mode [12-15] though is a variant; it eliminates the reaching phase by enforcing the sliding mode throughout the entire system response. The integral sliding mode control is a combination of two controls, a nominal

control which decides the desired performance of the system without disturbance and a discontinuous control which rejects the disturbance and model uncertainty. Even though this method ensures a better performance of the systems the resulting control input becomes discontinuous in nature. This control is not desirable from the implementation point of view because the oscillations caused by the high-frequency switching of discontinuous control will excite the unmodeled dynamics in the closed loop, known as chattering which will lead to unnecessary wear and tear on the actuator components of the system and reduce the life of actuators. Recently a continuous higher order sliding mode control is proposed in which, the benefit of super twisting control and integral sliding mode control is suitably combined [16]. In this study, the ISMC law is modified for more generalized systems with matching disturbance by replacing the discontinuous part of the feedback control with super twisting control.

Ramesh Kumar P., Asif Chalanga, B. Bandyopadhyay, "Smooth integral sliding mode controller for the position control of Stewart platform" [17], in this study, the discontinuous control part is replaced by a twisting algorithm based continuous control hence the proposed controller is continuous in nature due to the combination of two continuous controls. And for a given desired position trajectory of the platform the leg lengths can be calculated using inverse kinematics and corresponding leg velocities can be estimated using a super twisting observer.

In this thesis, a passivity based smooth integral sliding mode controller has been designed and implemented for the Stewart platform manipulator. PD+ has been designed based on the passivity principle, and smooth integral sliding mode control incorporated to enhance system robustness, and an integral time absolute error is formulated as an optimization problem that minimizes the error between the actual and desired leg lengths and optimal gain matrices of the controller are founded using particle swarm optimization algorithm.

1.5. Problem Statement

The most commonly used imaging modality in medicine for observing soft tissues structure non-invasively is ultrasound (US) imaging or ultrasonography. However, traditional ultrasound scanning is conducted by sonographers, which results in musculoskeletal disorders on them. Robotic-assisted ultrasound imaging has been recently used in medical diagnosis to take burdens off the sonographer by holding and moving the probe with the appropriate force needed for quality imaging.

Because of the nonlinear and stochastic dynamics of a robot, the existence of uncertainties due to model inaccuracies, parameter variation and external disturbance, patient movement, and respiratory motion, a robotic-assisted ultrasound imaging system requires the design of a robust and high-performance robot controller. To fulfill this requirement, a controller design technique based on passivity-based integral sliding mode control is implemented for the Stewart platform. The robustness in passivity based integral sliding mode control is achieved as a result of high frequency switching in the discontinuous control signal which is not desirable from an implementation point of view because an oscillation caused by the high-frequency switching of the discontinuous control signal will lead to a dangerous chattering.

Therefore, in this thesis, the integral sliding mode control law is modified by replacing the discontinuous control part with super twisting control, and proportional derivative plus is designed based on the passivation principle so that the closed-loop system asymptotically stable. A particle swarm optimization algorithm is implemented to determine the optimal gain matrices values of the controller using an integral time absolute error as an optimization problem.

1.5. Objective

1.5.1. General Objective

The general objective of this thesis was to apply passivity based robust controller methods such as Proportional Derivative plus Control, Integral Sliding Mode Control, and Super Twisting Control to control 6-DOF Stewart platform manipulator for desired trajectory tracking commands.

1.5.2. Specific Objectives

The specific objectives of this thesis were to:

Study kinematic and dynamic modeling of the Stewart platform manipulator.

Design a rotary Stewart platform manipulator using computer aided design (CAD).

Analyze singularity and workspace of the rotary Stewart platform manipulator to ensure that it has a reasonable workspace volume.

Design a PD+ controller based on passivity analysis and smooth integral sliding mode controller (SISMC) using super twisting control and continuous nominal control.

Determine the optimal gain matrices values of (PB-SISMC) using the PSO algorithm.

Evaluate the performance of the proposed controller using MATLAB/Simulink and to test its robustness against parameter variation and variable disturbance.

1.6. Methodology

The overall block diagram of the robot-assisted 4D ultrasound control system shown in figure 1.4 consists of an ultrasound image controller, robot controller, slave robot that carries the ultrasound probe, and tracking algorithm. However, this thesis focuses on the design of the robot controller, so that the first task to achieve the general objective stated above is studying kinematic and dynamic modeling of the robotic architecture that best fits for medical ultrasound imaging.

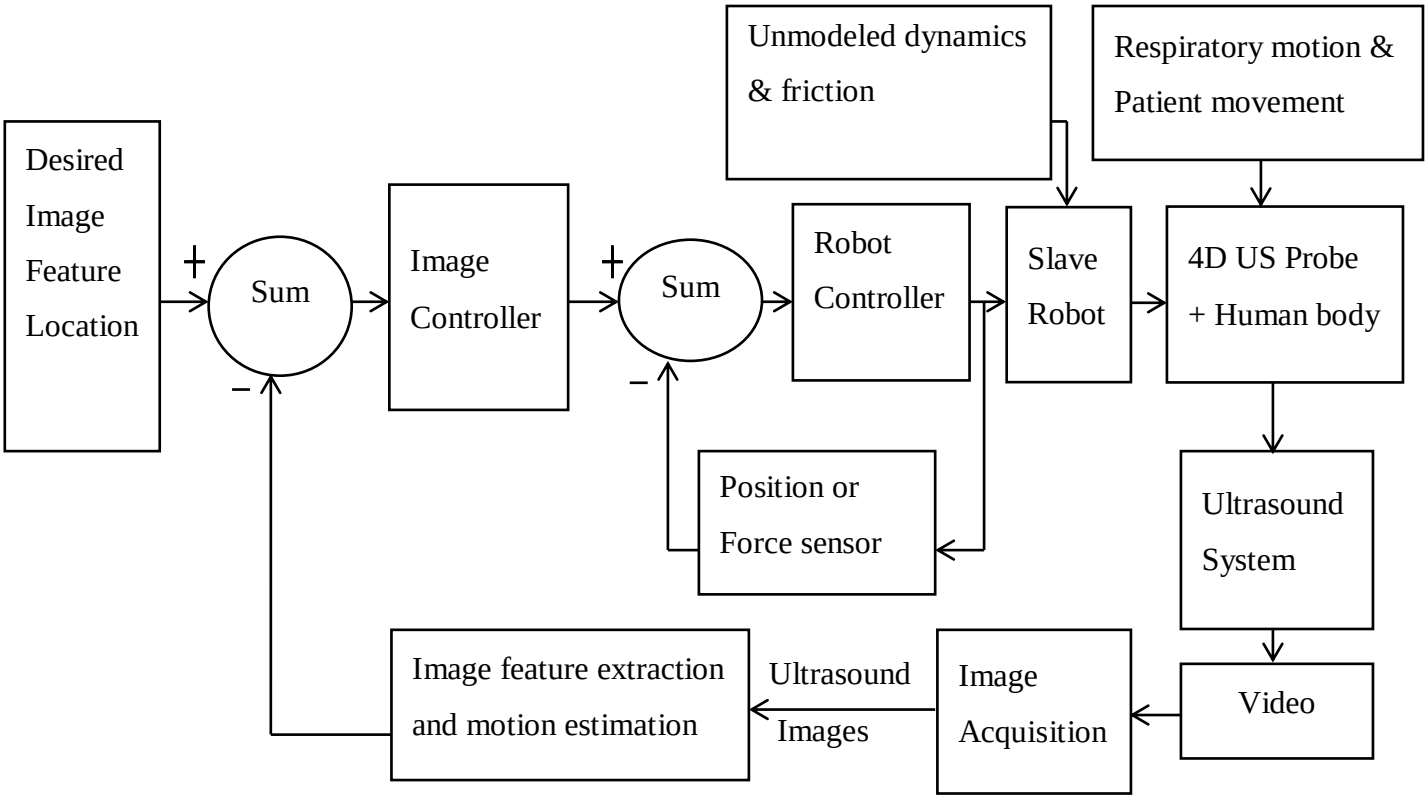


Figure1.4. Block diagram of robot assisted 4D ultrasound control system.

Based on the structural advantages of the manipulator, a slave robot has been designed by six degrees of freedom parallel robot. In particular, the rotary Stewart platform has been used to enhance its potential application in medical ultrasound imaging. The robot controller has been designed based on passivity based smooth integral sliding mode control, and the optimal gain matrices of the controller are obtained using particle swarm optimization, as shown in figure 1.5. Finally, the proposed controller has implemented using MATLAB/Simulink, and the result has been compared with the passivity-based integral sliding mode controller.

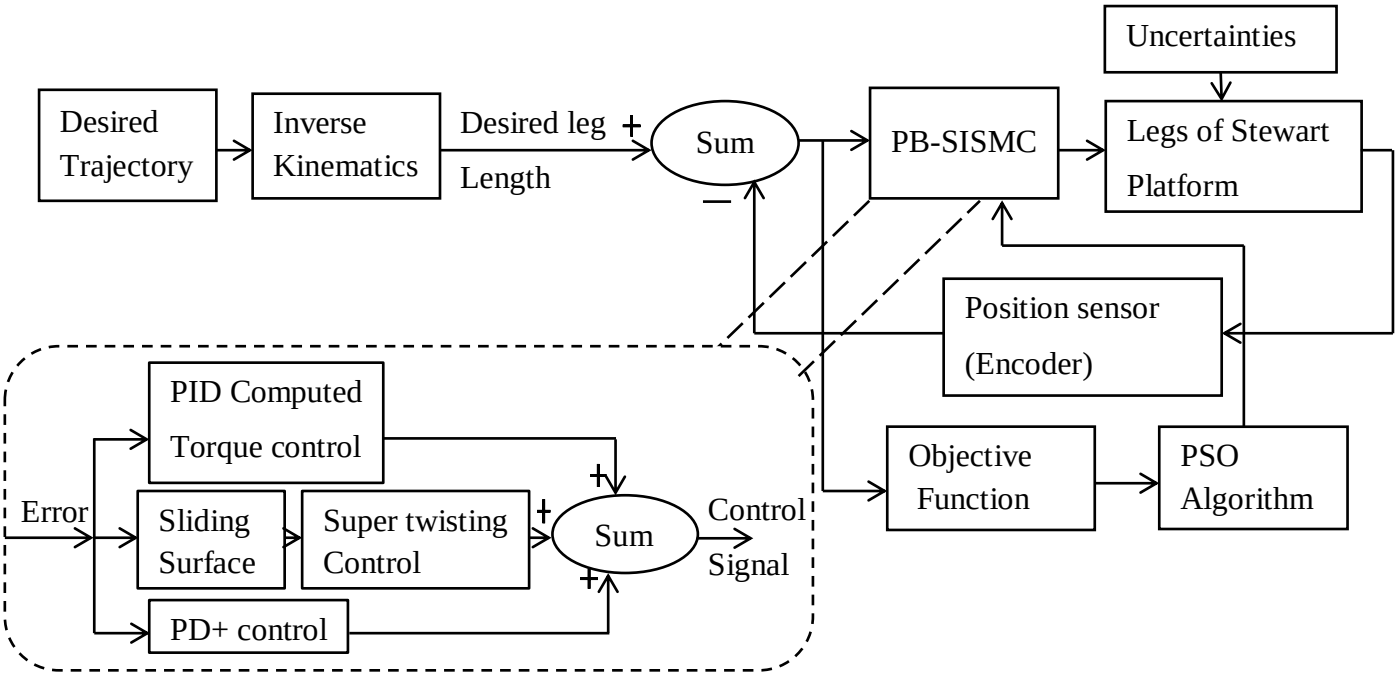


Figure1.5. PSO tuned joint space passivity based smooth integral sliding mode control of Stewart platform manipulator.

Here in figure 1.5, it can be seen that passivity-based smooth integral sliding mode control (PB-SISMC) consists of a nominal control, a super twisting control (STC), and a PD+ control. The controller is designed in joint space and implemented for trajectory tracking of the Stewart platform manipulator.

1.7. Organization of the Thesis

The thesis has been organized into five chapters. Chapter one presents the introduction on ultrasound imaging, robots ultrasound and Stewart platform, problem statement, objective, and methodology. It also gives a state of the art literature reviews on sliding mode control, and passivity based control. Chapter two presents the kinematic and geometric modeling and dynamic modeling of the Stewart platform manipulator. It also deals with the workspace and singularity analysis of the rotary Stewart platform. Chapter three deals passivity based sliding mode controller design for Stewart platform. Chapter four presents the 3D model of the rotary Stewart platform using Solidworks, the performance of the passivity based integral sliding mode controller, and a comparison of the results obtained with the passivity based smooth integral sliding mode controller for the trajectory tracking problems. Chapter five discusses the conclusions drawn and the recommendation on further work for the best control of the Stewart platform manipulator to enhance its application in medical imaging.

CHAPTER TWO

2. MODELING THE STEWART PLATFORM MANIPULATOR

2.1. Kinematic and Geometric Modeling

The Stewart platform, shown in figure 2.1, is composed of six extendable legs that connect the base and the moving platform, a fixed base, and a movable platform to which the probe is attached. It is assumed that the base and the platform are circular with radius rb and rp respectively. Each leg is attached to the base by a universal joint and the platform by a spherical joint.

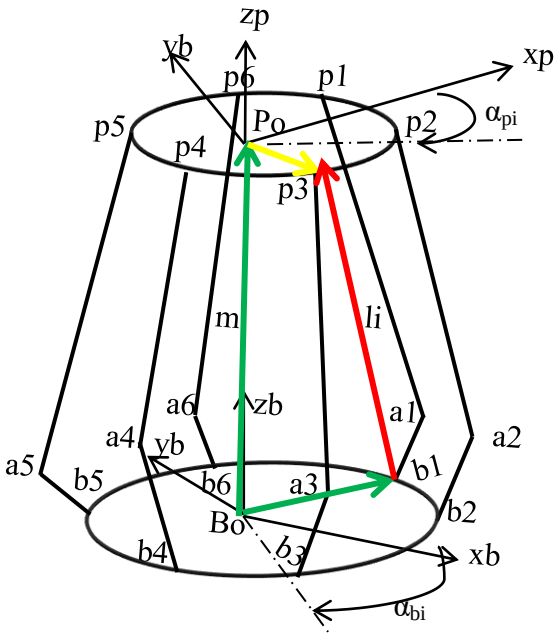


Figure2.1.Coordinate frame assignment for geometric and kinematic description of rotary Stewart platform manipulator.

As shown in figure 2.1, a reference frame Fp (Po-xp,yp,zp) is attached to the robot end effector (platform) and a reference frame Fb (Bo-xb,yb,zb) is fixed to the base. Furthermore, the position of the ith joint on the base point Bi (i=1, 2,. , 6) is denoted by vector, $bi = \begin{bmatrix} bix & biy & biz \end{bmatrix}^T$, i=1,2…6 and the position of the ith joint on the platform point Pi is denoted by vector, $pi = \begin{bmatrix} pix & piy & piz \end{bmatrix}^T$, i=1,2,…6. Let vector, $m = \begin{bmatrix} mx & my & mz \end{bmatrix}^T$ denote the position of point Po with respect to point Bo expressed in frame Fb and let ${}^b_p R$ be the orientation matrix representing the rotation from frame Fb to frame Fp. Further, the position vector of the point Pi expressed in frame Fb, noted Ei, is given by $Ei = m + {}^b_p Rpi$. The Cartesian space position and orientation of the moveable platform or end effector is specified

by $X = \begin{bmatrix} mx & my & mz & \alpha & \beta & \gamma \end{bmatrix}^T$ where mx, my, mz represent the surge, sway and heave motions respectively, whereas α, β, γ are roll, pitch and yaw rotations angles that constitute the orientation matrix ${}^b_p R$. As shown in figure 2.1, the xb axis of frame Fb is selected along the line which bisects the angle b2Bob3, and similarly for the xp axis of frame Fp is selected along the line which bisects the angle p1Pop2. Denote half of the angle b2Bob3 as the base angle α_b, and half of the angle p1Pop2 as the platform angle α_p. These angles are used to define the location of joints on the base and the platform relative to Fb and Fp respectively. Both sets of points are easily found to be

$$bi = \begin{bmatrix} bix & biy & biz \end{bmatrix}^T = \begin{bmatrix} rb\cos(\alpha_{bi}) & rb\sin(\alpha_{bi}) & 0 \end{bmatrix}^T \quad (1.1a)$$

$$pi = \begin{bmatrix} pix & piy & piz \end{bmatrix}^T = \begin{bmatrix} rp\cos(\alpha_{pi}) & rp\sin(\alpha_{pi}) & 0 \end{bmatrix}^T \quad (1.1b)$$

With rb and rp being the radii of the circles on which, respectively, bi and pi lay. The corresponding angles are shown in table 2.1.

Table2.1. Angular coordinates of base and platform attachment points.

i	1	2	3	4	5	6
α_{bi}	30^o	90^o	180^o	210^o	270^o	330^o
α_{pi}	15^o	105^o	135^o	225^o	255^o	345^o

The 6-DOF motions consist the linear motions which consists of longitudinal (surge), lateral (sway), and vertical (heave) motions whereas angular motions are expressed as Euclidean angle rotations with respect to x-axis (roll), y-axis (pitch) and z-axis (yaw) as shown in figure 2.2.

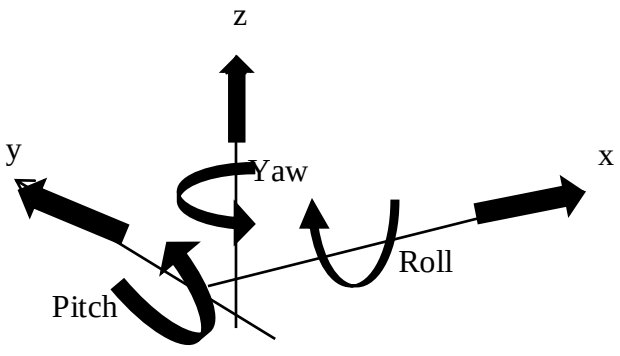


Figure2.2.Linear and angular motions

These six degrees of freedom motion are obtained by changing the length of the six legs. The length of i^{th} leg can be calculated from the vector closure relation shown in figure 2.1 by the green, yellow and red colors and given as

$$\overline{BoBi} + \overline{BiPi} = \overline{BoPo} + \overline{PoPi} \tag{2.2}$$

Where $\overline{BoBi}$ is the vector bi, $\overline{BoPo}$ is the vector m and $\overline{PoPi}$ is the vector pi.

Substituting these symbols and writing all vectors in base reference frame, we get the legs vector, *lvi* which is given by

$$\overline{BiPi} = lvi = m + {}^{b}_{p}Rpi - bi \tag{2.3}$$

Where pi and bi is the position vector for spherical joints and universal joints in the platform and base frames respectively and m is position of center of the platform with respect to base frame and the rotation matrix ${}^{b}_{p}R$ represents the transformation matrix from platform to base.

2.1.1. Rotational Matrix of the Platform

The rotation matrix is defined by the roll, pitch and yaw angles, namely, a rotation of γ (yaw) about the fixed z-axis, $R_z(\gamma)$, followed by a rotation of β (pitch) about the fixed y-axis, $R_z(\beta)$, and a rotation of α (roll) about the fixed x-axis, $R_x(\alpha)$. In this way, the rotation matrix of the platform with respect to the base frame can be obtained as follows:

Let us consider the first rotation γ (yaw) around the z axis and the associated matrix is given as follows.

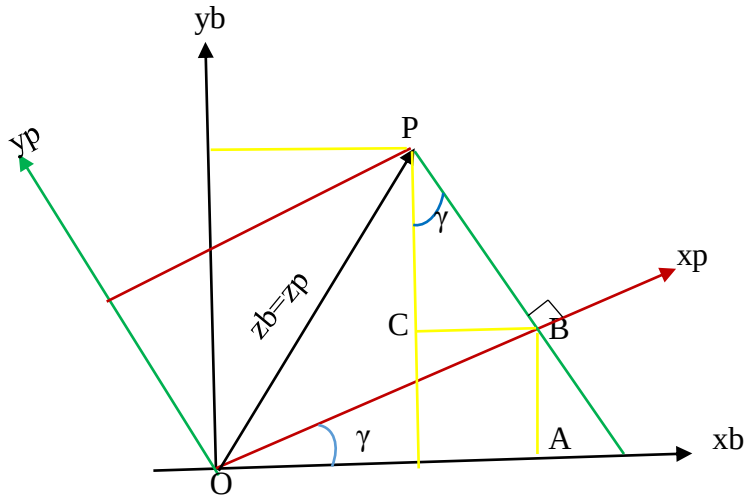


Figure2.3. Rotation about fixed z- axis with angle of γ (yaw)

In this case the distances are xp and yp that go up to the point P, knowing that zp= zb to deduce the counterfoil of rotation R_z (γ) as shown in figure above, such that the dynamic model can be expressed as;

$$P = ixp + jyp + kzp = ixb + jyb + kzb \tag{2.4}$$

$$xb = OA - BC = xp\cos(\gamma) - yp\sin(\gamma) \tag{2.5a}$$

$$yb = AB + PC = xp\sin(\gamma) + yp\cos(\gamma) \tag{2.5b}$$

$$zb = zp \tag{2.5c}$$

Equation (2.5a), (2.5b) and (2.5c) can be rewritten as in matrix form

$$\begin{bmatrix} xb \\ yb \\ zb \end{bmatrix} = R(\gamma)\begin{bmatrix} xp \\ yp \\ zp \end{bmatrix} \tag{2.6}$$

$$\begin{bmatrix} xb \\ yb \\ zb \end{bmatrix} = \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{bmatrix} xp \\ yp \\ zp \end{bmatrix} \tag{2.7}$$

Hence, a rotation of γ (yaw) about the fixed z-axis, Rz(γ) ,can be given as

$$Rz(\gamma) = \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.8}$$

Also let us assume the second rotation β (pitch) around the y axis and the associated matrix is given as follows.

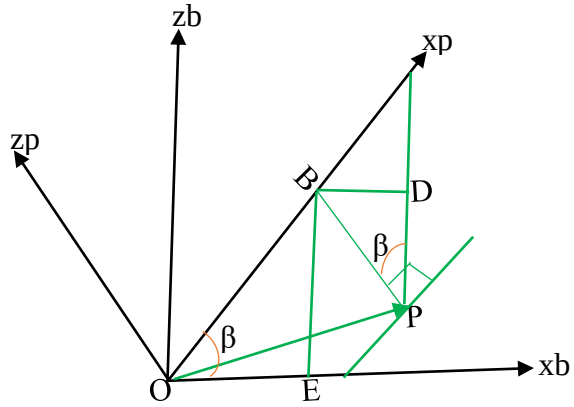


Figure2.4. Rotation about fixed y- axis with angle of β (pitch)

In this case the distances are xp and zp that go up to the point P, knowing that yb = yp to deduce the counterfoil of rotation R_y (β) as shown in figure above, such that the dynamic model can be expressed as;

$$xb = OE + BD = xp\cos(\beta) + zp\sin(\beta). \tag{2.9a}$$

$$yb = yp \tag{2.9b}$$

$$zb = BE - DP = zp\cos(\beta) - xp\sin(\beta) \tag{2.9c}$$

Equation (2.9a), (2.9b) and (2.9c) can be rewritten as in matrix form

$$\begin{bmatrix} xb \\ yb \\ zb \end{bmatrix} = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix} \begin{bmatrix} xp \\ yp \\ zp \end{bmatrix} \tag{2.10}$$

Therefore a rotation of β (pitch) about the fixed y-axis, Rz(β), can be given as

$$Ry(\beta) = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix} \tag{2.11}$$

Again, let us assume the third rotation α (roll) around the x axis and the associated matrix is given as follows.

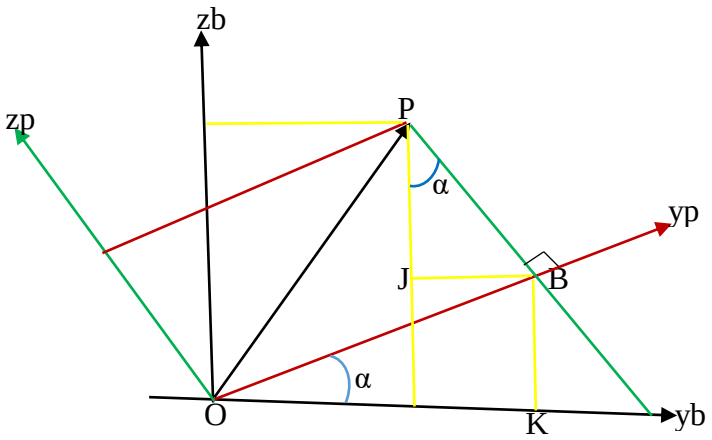


Figure2.5. Rotation about fixed y- axis with angle of α (roll)

In this case the distances are yp and zp that go up to the point P, knowing that xb = xp to deduce the counterfoil of rotation Rx (α) as shown in figure above, such that the dynamic model can be expressed as;

$$xb = xp \tag{2.12a}$$

$$yb = OK - Jb = yp\cos(\alpha) - zp\sin(\alpha) \tag{2.12b}$$

$$zb = KB + JP = yp\sin(\alpha) + zp\cos(\alpha) \tag{2.12c}$$

Equation (2.12a), (2.12b) and (2.12c) can be rewritten as in matrix form

$$\begin{bmatrix} xb \\ yb \\ zb \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{bmatrix} xp \\ yp \\ zp \end{bmatrix} \tag{2.13}$$

Therefore a rotation of α (roll) about the fixed x-axis, Rx (α), can be given as

$$Rx(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix} \tag{2.14}$$

And by combining three matrices in equation (2.8), (2.11) and (2.14), we obtain the combined orientation matrix ${}^{b}_{p}R$ of the end effector referred to the base frame.

$$ {}^{b}_{p}R = Rz(\gamma)Ry(\beta)Rx(\alpha) \tag{2.15} $$

$$ {}^{b}_{p}R = \begin{pmatrix} C(\gamma) & -S(\gamma) & 0 \\ S(\gamma) & C(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} C(\beta) & 0 & S(\beta) \\ 0 & 1 & 0 \\ -S(\beta) & 0 & C(\beta) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & C(\alpha) & -S(\alpha) \\ 0 & S(\alpha) & C(\alpha) \end{pmatrix} \tag{2.16} $$

$$ {}^{b}_{p}R = \begin{pmatrix} C(\gamma)C(\beta) & C(\gamma)S(\alpha)S(\beta) - C(\alpha)S(\gamma) & S(\alpha)S(\gamma) + C(\alpha)C(\gamma)S(\beta) \\ C(\beta)S(\gamma) & C(\alpha)C(\gamma) + S(\alpha)S(\beta)S(\gamma) & -S(\alpha)C(\gamma) + C(\alpha)S(\beta)S(\gamma) \\ -S\,\mathrm{in}(\beta) & C(\beta)S(\alpha) & C(\beta)C(\alpha) \end{pmatrix} \tag{2.17} $$

Where S refers to angle sine and C refers to angle cosine respectively. The angle α, β, and γ are Euler angles. From leg vector equation given in equation 2.3, we get the magnitude of $\overline{BiPi}$ which is the i[th] leg length. Hence the inverse kinematic equation, which gives the length of the leg i for a given platform position and orientation, is given by

$$ \left\| \overline{BiPi} \right\| = li = \left\| lvi \right\| = \left\| m + {}^{b}_{p}Rpi - bi \right\| = \sqrt{(BiPi)^{T}(BiPi)} \tag{2.18} $$

This means that if the desired position and orientation of the platform is given, then the length of each leg can be uniquely determined. The converse of (2.18) gives the forward kinematics which is to calculate the position m and orientation angles of the platform for a given combination of leg lengths. This is a nonlinear and coupled and it has no direct solution.

The solution of (2.18) which is the inverse kinematics problem of Stewart platform provide the length of each leg for a given desired position and orientation of the movable platform. If the leg length of each leg is driven by rotary actuators, as in[8], the inverse kinematics problem for rotary 6 DOF Stewart platform, needs further calculation to determine each servo angle as follows.

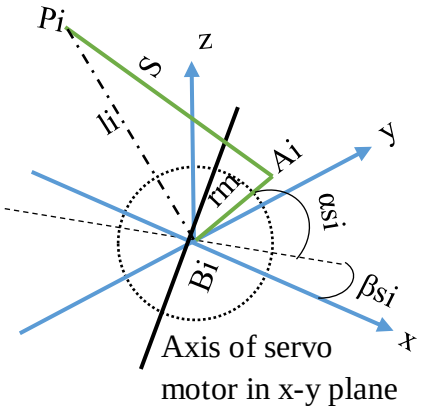


Figure2.6. Representation of servo arm/leg combination

Where: rm is servo arm length

Ai is the points of servo arm/leg joint on the i^{th} servo in the base reference frame with coordinates given as;

$$Ai = \begin{bmatrix} xia \\ yia \\ zia \end{bmatrix}$$

Bi is the points of rotation of the servo arms in the base reference frame with the coordinates given as;

$$Bi = \begin{bmatrix} xib \\ yib \\ zib \end{bmatrix}$$

Pi is center of the i^{th} spherical joint in the platform reference frame with coordinates given as;

$$Pi = \begin{bmatrix} xip \\ yip \\ zip \end{bmatrix}$$

S is length of fixed leg i, li is length of the i^{th} leg given in (2.18), αs is angle of servo arm in x-y plane and βs is angle of servo arm relative to the x-axis, where the shaft axis lies in the x-y plane where z = 0.

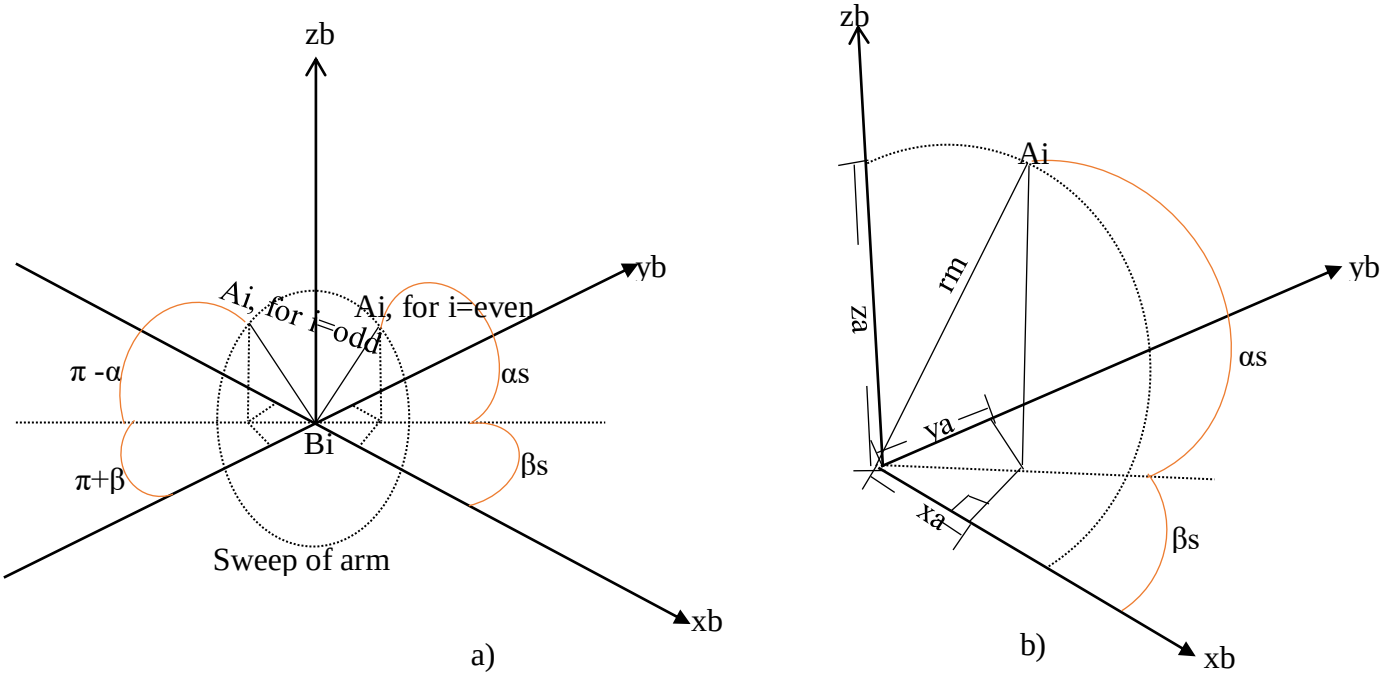


Figure2.7.Angular rotation of the servo motor with respect to the base coordinate axes xb, yb and zb

a) Even and odd arrangement of servo motor with respect to base reference frame

b) Points of the arm/leg joint on the servo with coordinates xa, ya and za in base reference frame

The angle of servo arm relative to the x-axis is shown in table 2.2.

Table2.2. Angle of servo arm relative to the x-axis

i	1	2	3	4	5	6
βsi	90^o	270^o	210^o	30^o	330^o	150^o

Point Ai is constrained to be on the servo arm, but the arrangement of the servos means that the odd and even arms are a reflection of each other. So for the even legs we have:

$$xia = rm\cos(\alpha si)\cos(\beta si) + xib \tag{2.19}$$

$$yia = rm\cos(\alpha si)\sin(\beta si) + yib \tag{2.20}$$

$$zia = rm\sin(\alpha si) + zib \tag{2.21}$$

And for the odd legs we have:

$$xia = rm\cos(\pi - \alpha si)\cos(\pi + \beta si) + xib \tag{2.22}$$

$$yia = rm\cos(\pi - \alpha si)\sin(\pi + \beta si) + yib \tag{2.23}$$

$$zia = rm\sin(\pi - \alpha si) + zib \tag{2.24}$$

Using trig identities the equation of odd legs can be rewritten as follows;

$$\sin(\pi - \alpha si) = \sin(\pi)\cos(\alpha si) - \sin(\alpha si)\cos(\pi) = -\sin(\alpha si)$$

$$\cos(\pi - \alpha si) = \cos(\pi)\cos(\alpha si) + \sin(\pi)\sin(\alpha si) = -\cos(\alpha si)$$

$$\sin(\pi + \beta si) = \sin(\pi)\cos(\beta si) + \sin(\beta si)\cos(\pi) = -\sin(\beta si)$$

$$\cos(\pi + \beta si) = \cos(\pi)\cos(\beta si) - \sin(\pi)\sin(\beta si) = -\cos(\beta si)$$

Substituting these values into the equations (2.22), (2.23) and (2.24) we get the equation for odd legs as

$$xia = rm\cos(\alpha si)\cos(\beta si) + xib \tag{2.25}$$

$$yia = rm\cos(\alpha si)\sin(\beta si) + yib \tag{2.26}$$

$$zia = rm\sin(\alpha si) + zib \tag{2.27}$$

Which is the same equations as (2.19), (2.20), and (2.21) for the even legs. When the servo is in its initial position, a 90^o angle is formed between the fixed rigid leg (S) and servo operating arm (rm) then it is possible to calculate the home position leg length that is between points Pi and Bi using the theorem of Pythagoras give as ;

$$li^2 = S^2 + rm^2 \tag{2.28}$$

Where: i=1… 6

$$li^2 = (xip - xib)^2 + (yip - yib)^2 + (zip - zib)^2 \tag{2.29}$$

$$li^2 = (xip^2 + yip^2 + zip^2) + (xib^2 + yib^2 + zib^2) - 2(xipxib + yipyib + zipzib) \tag{2.30}$$

$$rm^2 = (xia - xib)^2 + (yia - yib)^2 + (zia - zib)^2 \tag{2.31}$$

$$rm^2 = (xia^2 + yia^2 + zia^2) + (xib^2 + yib^2 + zib^2) - 2(xiaxib + yiayib + ziazib) \tag{2.32}$$

$$S^2 = (xip - xia)^2 + (yip - yia)^2 + (zip - zia)^2 \tag{2.33}$$

$$S^2 = (xip^2 + yip^2 + zip^2) + (xia^2 + yia^2 + zia^2) - 2(xipxia + yipyia + zipzia) \tag{2.34}$$

Substituting from equations (2.30) & (2.32), we get;

$$S^2 = li^2 - (xib^2 + yib^2 + zib^2) + 2(xipxib + yipyib + zipzib) + rm^2 -$$

$$(xib^2 + yib^2 + zib^2) + 2(xiaxib + yiayib + ziazib) - 2(xipxia + yipyia + zipzia) \tag{2.35}$$

$$li^2 + rm^2 - S^2 = 2(xib^2 + yib^2 + zib^2) + 2xia(xip - xib) +$$

$$2yia(yip - yib) + 2zia(zip - zib) - 2(xipxib\_yipyib + zipzib) \tag{2.36}$$

Substituting for xia, yia and zia from equation (2.19), (2.20) and (2.21), we get;

$$li^2 + rm^2 - S^2 = 2(xib^2 + yib^2 + zib^2) + 2rm\cos(\alpha si)\cos(\beta si)(xip - xib) +$$

$$2xipx - 2xib^2 + 2rm\cos(\alpha si)\sin(\beta si)(yip - yib) + 2yipyib - 2yib^2 +$$

$$2rm\sin(\alpha si)(zip - zib) + 2zipzib - 2zib^2 - 2(xipxib + yipyib + zipzib) \tag{2.37}$$

$$li^2 + rm^2 - S^2 = 2(xib^2 + yib^2 + zib^2) + 2rm\cos(\alpha si)\cos(\beta si)(xip - xib) +$$

$$2xipxib - 2xib^2 + 2rm\cos(\alpha si)\sin(\beta si)(yip - yib) + 2yipyib - 2yib^2 +$$

$$2rm\sin(\alpha si)(zip - zib) + 2zipzib - zib^2 - 2(xipxib + yipyib + zipzib) \tag{2.38}$$

$$li^2 + rm^2 - S^2 = 2rm\cos(\alpha si)\cos(\beta si)(xip - xib) + 2rm\cos(\alpha si)\sin(\beta si)$$

$$(yip - yib) + 2rm\sin(\alpha si)(zip - zib) \tag{2.39}$$

$$li^2 + rm^2 - S^2 = 2rm\cos(\alpha si)(\cos(\beta si)(xip - xib) + \sin(\beta si)(yip - yib)) +$$

$$2rm\sin(\alpha si)(zip - zib) \tag{2.40}$$

$$li^2 + rm^2 - S^2 = M\sin(\alpha si) + N\cos(\alpha si) \tag{2.41}$$

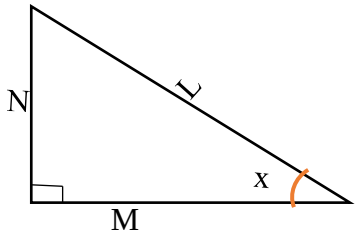


Figure2.8. Triangle for deduction of identity

Consider the right angle triangle shown in figure (2.8) to simplify the mathematical expressions that arise in the kinematics of the plant identity in equation (2.42)

$$\sin(x+\alpha si)=\sin(x)\cos(\alpha si)+\sin(\alpha si)\cos(x) \tag{2.42}$$

$$\sin(x+\alpha si)=\frac{N}{L}\cos(\alpha si)+\frac{M}{L}\sin(\alpha si)$$

$$\tan(x)=\frac{N}{M}, x=\arctan(\frac{N}{M})$$

$$L=\sqrt{N^2+M^2}\sin(x+\alpha si), x+\alpha si=\arcsin(\frac{L}{\sqrt{N^2+M^2}})$$

$$\alpha si=\arcsin(\frac{L}{\sqrt{N^2+M^2}})-x=\arcsin(\frac{L}{\sqrt{N^2+M^2}})-\arctan(\frac{N}{M}) \tag{2.43}$$

Where:

$$L=li^2-S^2+rm^2$$

$$M=2rm(zip-zib)$$

$$N=2rm(\cos(\beta si)(xip-xib)+\sin(\beta si)(yip-yib))$$

For joints with a sufficiently large angular range of motion, the platform can reach the desired position and orientation if a real solution to (2.43) exists for all i.

2.1.2. Kinematics Modeling of the Stewart Platform at Home Position

Home position of the platform is the position of platform at a height of a platform above the base reference frame with no other translational or rotational movement. Thus in order to define the range of movement we have to define some constants such as;

i. Home position vector of the platform with respect to base reference frame

$$Ei0=m+{}^{b}_{p}Rpi \tag{2.44}$$

$$Ei0=\begin{bmatrix} mx0 \\ my0 \\ mz0+h0 \end{bmatrix}+\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{bmatrix} xip \\ yip \\ zip \end{bmatrix}$$

$$Ei0=\begin{bmatrix} xip \\ yip \\ zip+h0 \end{bmatrix} \tag{2.45}$$

Where: $mx0=my0=mz0=0$ and $h0$ is the home position height of the platform above the base reference frame.

ii. Height at home position

From equation (2.28) we have;

$$li^2 = S^2 + rm^2$$

Then the home position leg length can be rewritten as;

$$li0^2 = S^2 + rm^2 = (xip - xib)^2 + (yip - yib)^2 + (zip + h0 - zib)^2 \quad (2.46)$$

$$(zip + h0 - zib)^2 = S^2 + rm^2 - ((xip - xib)^2 + (yip - yib)^2) \quad (2.47)$$

$$h0 = \sqrt{S^2 + rm^2 - ((xip - xib)^2 + (yip - yib)^2)} - zip + zib \quad (2.48)$$

iii. The angle of servo arm at home position

From equation (2.3), we have

$$lvi = m + {}^{b}_{p}R pi - bi$$

Then we get the vector of leg length at home position as

$$li0 = \sqrt{(Ei0 - bi)^2} \quad (2.49)$$

The servo arm angle can be calculated using equation (2.43), so that the home position servo arm angle can be given as

$$\alpha si0 = \arcsin\left(\frac{L0}{\sqrt{N0^2 + M0^2}}\right) - \arctan\left(\frac{N0}{M0}\right) \quad (2.50)$$

Where:

$$N0 = 2rm(\cos(\beta si)(xip - xib) + \sin(\beta si)(yip - yib))$$

$$L0 = 2rm$$

$$M0 = 2rm(zip + h0 - zib)$$

The servos are controlled by a pulse whose duration defines the angle of the arm. Therefore, using the angle of servo arm given in (2.43) and (2.50) for a given servo rate, the pulse width (PW) for each of the servos can be given by

$$PWi = PWi0 + (\alpha si - \alpha si0)r \text{ , for even i and} \quad (2.51a)$$

$$PWi = PWi0 - (\alpha si - \alpha si0)r \text{ , for odd i} \quad (2.51b)$$

Where $PWi0$ is the actual plus width required for the ith servomotor to move the arm to its home position given in µseconds and r is servo rate for the ith servomotor, given in µseconds per radian.

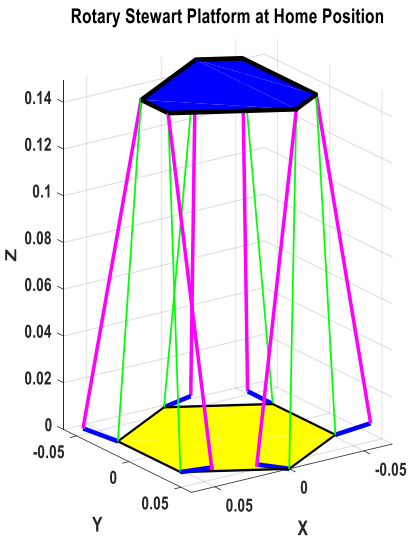


Figure2. 9. 3D simulation of the rotary Stewart platform at home position

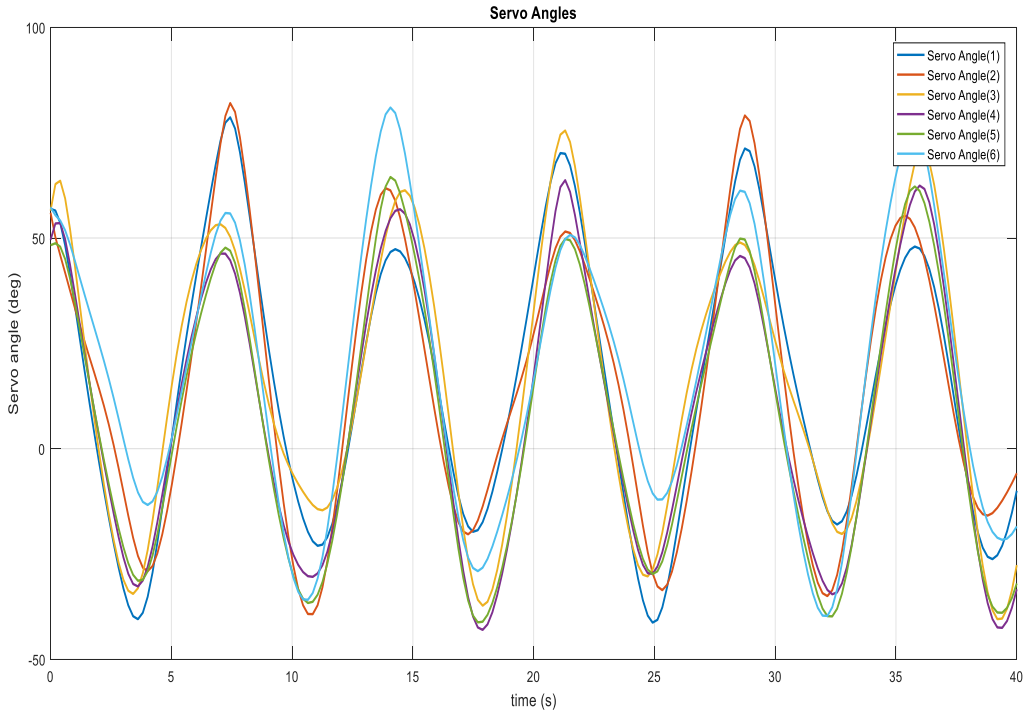


Figure2. 10. Plot of angle of servo arm when platform orientation α=β=γ=0^{o}, with variable position;

$$mx = 0.018(1-\exp^{-\pi t})\cos(0.38\pi t)\text{,m}$$

$$my = 0.019(1-\exp^{-\pi t})\sin(0.18\pi t)\text{,m}$$

$$mz = 0.014\cos(0.28\pi t)\text{,m}$$

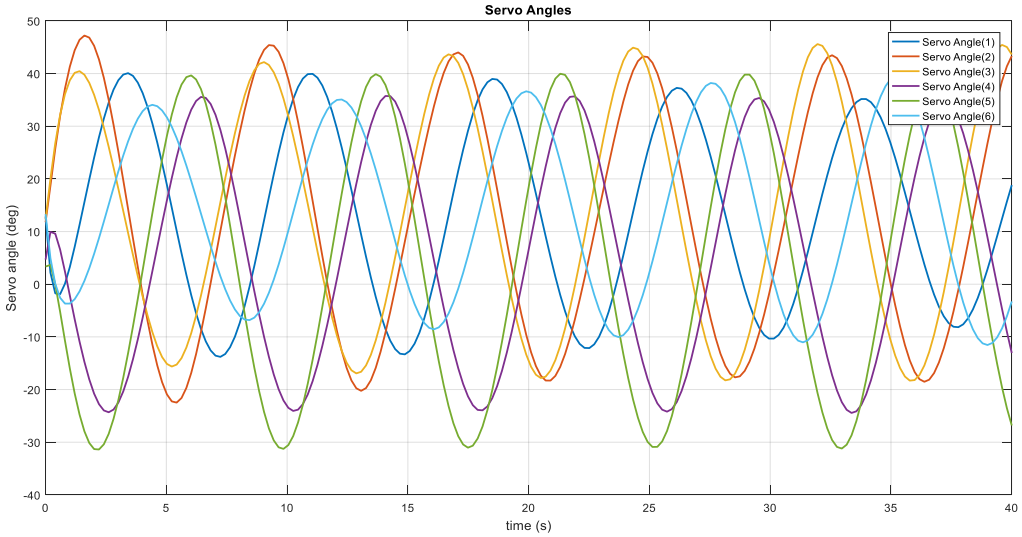


Figure2. 11. Plot of angle of servo arm for platform orientation when platform position mx=my=mz=0m, with variable orientation;

$$\alpha = 0.24(1-\exp^{-\pi t})\sin(0.26\pi t)\text{,rad}$$

$$\beta = 0.15(1-\exp^{-\pi t})\cos(0.26\pi t)\text{,rad}$$

$$\gamma = 0.13(1-\exp^{-\pi t})\cos(0.28\pi t)\text{,rad}$$

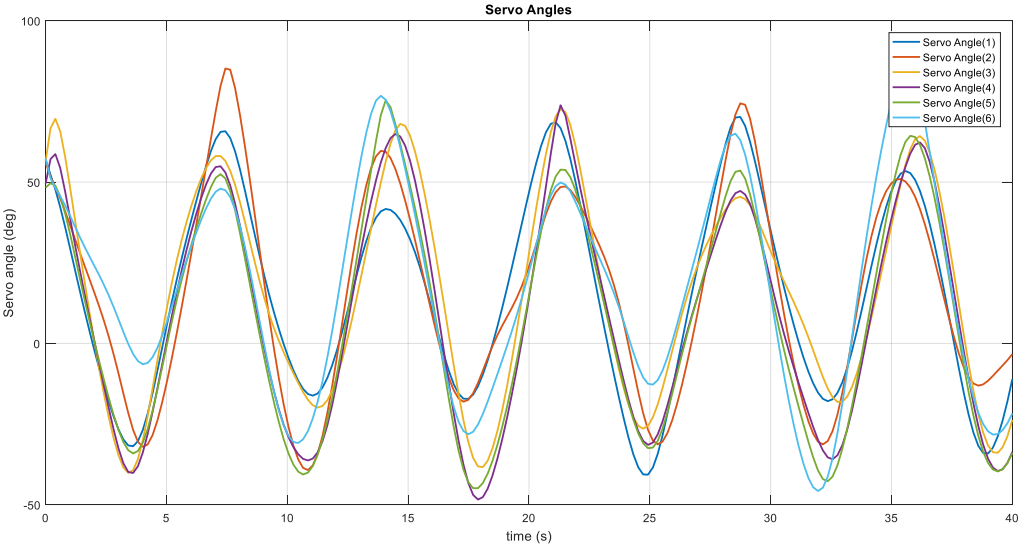


Figure2. 12. Plot of angle of servo arm for platform variable position and orientation;

$$\alpha = 0.018(1-\exp^{-\pi t})\sin(0.26\pi t)\text{,rad}$$

$$\beta = 0.048(1-\exp^{-\pi t})\cos(0.26\pi t)\text{,rad}$$

$$\gamma = 0.015(1-\exp^{-\pi t})\sin(0.78\pi t)\text{,rad}$$

$$mx = 0.018(1-\exp^{-\pi t})\cos(0.38\pi t)\text{,m}$$

$$my = 0.019(1-\exp^{-\pi t})\sin(0.18\pi t)\text{,m}$$

$$mz = 0.014\cos(0.28\pi t)\text{,m}$$

It is clear from equation (2.43) and (2.51), and from figure 2.10-2.12, the pulse width (PW) duration for each of the servos depends on each servo angle. Hence, by adjusting the six leg lengths for a real value of servo angle alpha given in (2.43) and (2.50), we can controlled the servos by a pulse whose duration defines the angle of the arm for a given servo rate and actual plus width.

2.1.3. Platform Jacobian

Platform Jacobian matrix is used to determine the platform's singularities, actuator force and platform's position, velocities and acceleration. For parallel manipulators, such as Stewart platform, it is easier to define the Jacobian matrix as a transformation from platform's velocity to the joint velocities. The velocity kinematics is obtained by differentiating (2.18) and it gives the velocity of the legs for a given vector of platform velocities as

$$\dot{q} = \dot{l} = Jp\,\dot{X} \tag{2.52}$$

Where $\dot{X} = \begin{bmatrix} {}^{b}_{p}v & {}^{b}_{p}w \end{bmatrix}^{T}$, is the end effector velocity vector and ${}^{b}_{p}v = \frac{d}{dt}\begin{bmatrix} mx & my & mz \end{bmatrix}^{T}$, is the platform's translational velocity vector, $\dot{q}$ is the six leg velocity vector, ${}^{b}_{p}w$ is the angular velocity vector of end effector, and can be written in terms of derivatives of the platform's rotation angles as

$${}^{b}_{p}w = \begin{pmatrix} \cos(\beta)\cos(\gamma) & -\sin(\gamma) & 0 \\ \cos(\beta)\sin(\gamma) & \cos(\gamma) & 0 \\ -\sin(\beta) & 0 & 1 \end{pmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix}$$

And Jp, is the platform's six by six Jacobian matrix and can be obtained as follows. Since each row of the Jacobian matrix corresponding to one of the platform's six legs, and all the actuators are equivalent, we only need to drive row i, to determine the platform jacobian matrix. Therefore, we can obtain the i^{th} row of Jacobian matrix by differentiating equation (2.18), and substituting into equation (2.3), we get

$$\dot{li} = \frac{1}{2}\frac{2lvi^{T}lvi}{\sqrt{lvi^{T}lvi}} = \frac{lvi^{T}lvi}{li} \tag{2.53}$$

$$\dot{li} = \frac{(Ei - bi)^T ({}^b_p w \times {}^b_p R)\, pi + {}^b_p v)}{li}$$

$$\dot{li} = \frac{(Ei - bi)^T ({}^b_p v) + (Ei - bi)^T ({}^b_p w \times {}^b_p Rpi)}{li}$$

$$\dot{li} = \frac{(Ei - bi)^T ({}^b_p v) + ({}^b_p Rpi \times (Ei - bi))^T ({}^b_p w)}{li}$$

$$\dot{li} = \frac{1}{li}\left[({}^b_p Rpi) + m - bi)^T, ({}^b_p Rpi) \times (m - bi)^T \right] \begin{bmatrix} {}^b_p v \\ {}^b_p w \end{bmatrix} \tag{2.54}$$

Where ${}^b_p \dot{R} = {}^b_p w \times {}^b_p R$, and $a.(b \times c) = (c \times a).b$ have been applied

Comparing equation (2.52) to equation (2.54), we can write the i[th] row of Jacobian matrix as

$$Jp(X) = \frac{1}{li}\left[({}^b_p Rpi + m - bi)^T, ({}^b_p Rpi \times (m - bi))^T \right] \tag{2.55}$$

The first derivative of platform jacobian matrix can be derived using quotient rule as follows

$$\dot{Jp}^T = \frac{1}{li}\begin{bmatrix} {}^b_p w \times ({}^b_p Rpi) + {}^b_p v \\ -{}^b_p v \times ({}^b_p Rpi) + (bi - m) \times ({}^b_p w \times ({}^b_p Rpi)) \end{bmatrix} - \frac{1}{li}\begin{bmatrix} {}^b_p Rpi + m - bi \\ ({}^b_p Rpi) \times (m - bi) \end{bmatrix} \dot{li} \tag{2.56}$$

$$\dot{Jp} = \frac{\begin{bmatrix} {}^b_p w \times ({}^b_p Rpi) + {}^b_p v \\ -{}^b_p v \times ({}^b_p Rpi) + (bi - m) \times ({}^b_p w \times ({}^b_p Rpi)) \end{bmatrix}^T - Jp\,\dot{li}}{li} \tag{2.57}$$

2.1.4. Singularity and Work Space Analysis

2.1.4.1. Singularity Analysis

The physical notion of a singularity in kinematics refers to configurations in which the mechanism's number of degrees of freedom changes instantaneously[18]. For parallel manipulators, the Stewart platform singular configuration results in the gain of one or more degrees of freedom. This means, when the Stewart platform manipulator reaches a singular configuration, it gains one or more uncontrolled degrees of freedom and will rotate and /or translate without a change in the leg lengths. Any method of singularity analysis, it is to be mathematically consistent, should be ascribed to the singularity of the jacobian matrix for inverse kinematics. So one method to compute the singular configuration is to take the determinant of this matrix, implies that when the determinant of the jacobian is equal to zero the manipulator approaches singularities. However, the actual value of the determinant cannot be used as a practical measure of ill-conditioning. For this reason, it is convenient to

use the condition number of the jacobian matrix. It is well known from the singular value decomposition theorem[19]; condition number of a nonsingular square matrix is the ratio of the largest to the smallest singular value.

$$\sigma(Jp) = \frac{\bar{\sigma}(Jp)}{\underline{\sigma}(Jp)} \tag{2.58}$$

Where, Jp is the platform jacobian matrix given in (2.55) $\bar{\sigma}$ and $\underline{\sigma}$ are the maximum and minimum singular value of Jp respectively.

And the singular value decomposition (SVD) [19]of the platform's jacobian matrix is:

$$Jp = U \sum V^T \tag{2.59}$$

Where U and V are orthonormal matrices and $\sum$ is a diagonal matrices of the singular values $\sigma 1, \sigma 2, \ldots\ldots\sigma 6$

Merlet [19], uses a set of geometric rules to establish the constraints on the position and orientation parameters that must be satisfied to obtain the various singular configurations. However, the calculation of such geometric rules is more complicated than that of the condition number of platform jacobian, which is used in the classical method to determine these singular configurations numerically. In this thesis, a graphical description of the platform's singular configuration has been determined by plotting the ratio of the largest to the smallest condition number of platform jacobian while varying two of the platform's six position and orientation parameters. Figure 2.13 shows a plot of singular value ratios versus varying pitch (β) and yaw (γ) for the platform at a nominal position (mx=0.024m, my=0.032m,z=0.086m, and $\alpha=0^o$). The plot has been plotted by computing the singular value ratio of the platform jacobian matrix while varying the platform's yaw and pitch angles between -180^o and 180^o at 3^o intervals.

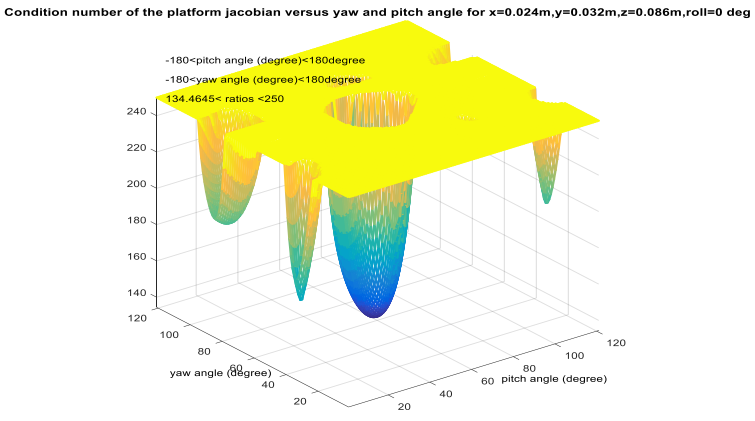


Figure2.13. Singular value ratio versus pitch and yaw

Here, in figure 2.13 it can be seen that there is a singular boundary where the platform is nearing a singular position or orientation. Therefore to prevent loss of control, we have to monitor the ratio of jacobian's singular value by plotting condition number ratios versus two parameters while varying the other four parameters up to their limits and stop the platform before it enters a singular position or orientation.

2.1.4.2. Work Space Analysis

The workspace of the Stewart platform can be defined as the reachable region of the origin of a coordinate system attached to the center of the moving platform[20]. Studies existing in literature on the workspace analysis of parallel mechanisms can be classified into two main groups as positional workspace and orientation workspace. Positional workspace is defined as all possible positions of the platform for a fixed orientation of the mobile platform. Likewise, orientation workspace is defined as all possible orientations of the platform, defined by three orientation angles α, β, γ for a fixed position of the mobile platform. The workspace depends on constraints introduced by joint angle limitations, leg length limitations and interference between the legs. In [21, 22] the effect of leg length and leg interference on the workspace of a particular platform was investigated. The positioning workspace (i.e., the region of three dimensional Cartesian space that can be attained by a manipulator with a given orientation) has been described through methods based on complete discretization the Cartesian space [23, 24]. Methods used for determining the orientation workspace are classified into three main groups by [25] as the discretization method, geometrical method and the Jacobian matrix technique. Therefore, the discretization method was preferred to be used for determining the workspace of 6-DOF manipulators by incremental movements of the mobile platform, and then solving the inverse kinematics to determine leg lengths. Consider again the inverse kinematic equation, which gives the length of the leg i for a given platform position and orientation, as given by

$$\|BiPi\| = li = \|Ei - bi\| = \sqrt{(BiPi)^T BiPi} \tag{2.60}$$

The leg length constraint is expressed by:

$$l_{\min i} \le l_i \le l_{\max i} \tag{2.61}$$

Where l_{mini}, and l_{maxi}, are the minimum and maximum allowable length of leg i. The workspace is determined by rotating the platform about x, y, z axes with α, β, γ amount, and positioning the platform with mx, my, mz amount. To search points included in and on the boundary of a reachable workspace, using the discretization method a MATLAB script was

written to test whether a set of different positions or orientations represented as points in a 3D plot were achievable by the platform. Once a set of valid points was determined, a boundary was plotted around the points encompassing them and creating a boundary volume. This was then plotted and is shown in figure 2.14- 2.17.

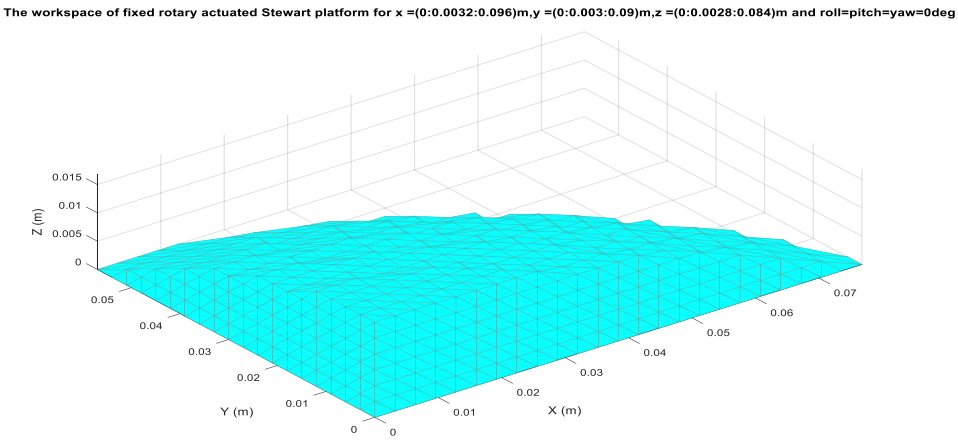
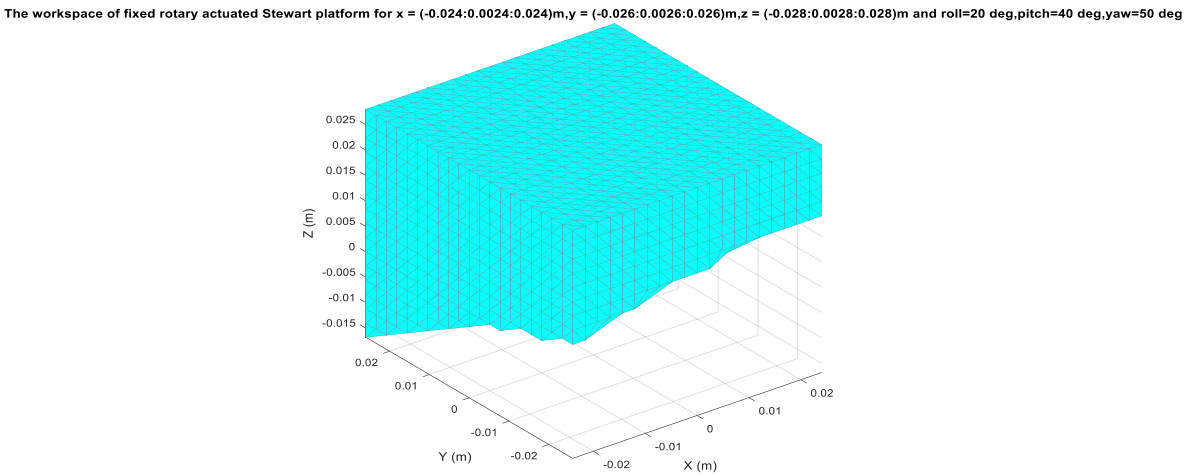


Figure2.13. Translational (positioning) workspace of rotary Stewart platform when α=β= γ=0^o.

It is clear from figure 2.14 the translational workspace was plotted for the range limit of platform positions mx, 0 to 96mm, my, 0 to 90mm and mz, 0 to 84mm when yaw, pitch, and roll of the platform were all kept equal to zero degrees. It can be seen that the central section of the volume is mostly contained within a 0 to 70mm range for the x and 0 to 60mm for y and in the z-direction results in more restrictions on the leg lengths, which make the workspace smaller.



Translational (positioning) workspace of rotary Stewart platform when α=20^o, β=40^o and γ=50^o.

It is clear from figure 2.15 the translational workspace was plotted for the range limit of platform positions -28 to 28mm for mx, my and mz when yaw, pitch, and roll of the platform were all kept equal to 20,40,50 degrees respectively. It can be seen that in the z-direction the points are within the maximum and minimum leg lengths but in x and y results an empty space that represents unreachable workspace.

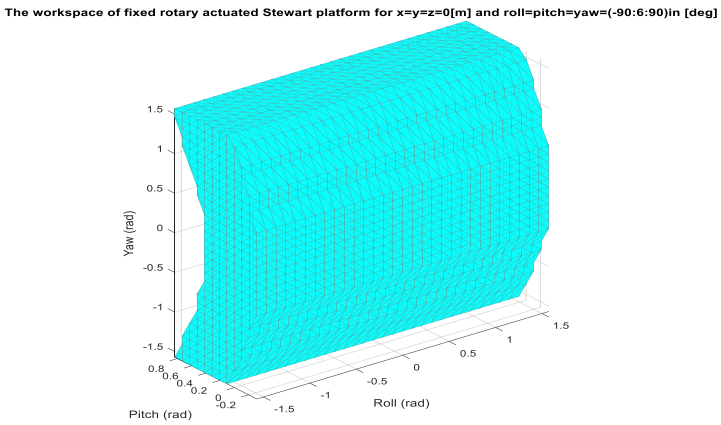


Figure2.14. Orientation workspace of rotary Stewart platform when x=y=z=0mm.

It is clear from figure 2.16 the rotational (orientation) workspace was plotted for the range limit of platform orientations -90^{o} to 90^{o} for a roll (α), pitch (β), and yaw (γ) when platform positions mx, my and mz all kept equal to 0mm. It can be seen that there is small empty space in all platform orientations roll, pitch, and yaw showing that restrictions on leg lengths.

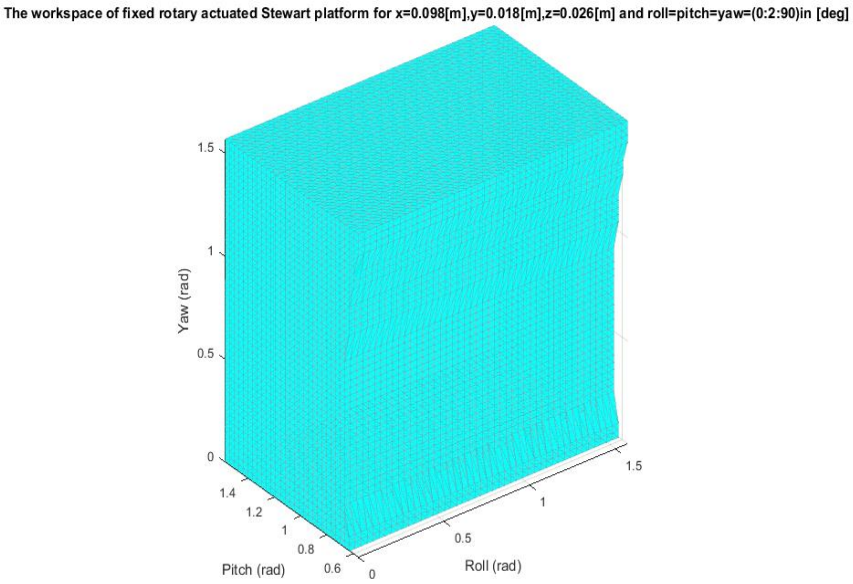


Figure2.15. Orientation workspace of rotary Stewart platform when x=98mm, y=18mm z=26mm.

It is clear from figure 2.17 the rotational (orientation) workspace was plotted for the range limit of platform orientations 0^o to 90^o for roll (α), pitch (β) and yaw (γ) when the platform positions were all kept equal to 98, 18, 26mm respectively. It can be seen that there is small empty space in platform rotation about x-axis showing that restriction on leg lengths.

2.2. Dynamic Modeling of Stewart Platform

The dynamic modeling of a Stewart platform manipulator has been studied by many researchers. The methods used are Lagrangian, Newton Euler and principle of virtual work mentioned by[6, 26]. The Lagrangian formulation gives a closed form equation, which is useful for the design of model based controllers, and it has been used by most authors. Another important point in the formulation of dynamic model of the manipulator is the variables used. The dynamic equation and hence control of the manipulator can be given either in joint space, as a function of the length, velocity and acceleration of legs $(q, \dot{q}, \ddot{q})$ or in task space using the position, velocity and acceleration of generalized vector $X = \begin{bmatrix} mx & my & mz & \alpha & \beta & \gamma \end{bmatrix}^T$ containing Cartesian position of platform center and its orientation. Each approach has its own advantages and disadvantages. The joint space approach is easier from closed loop control point of view as it does not need the use of forward kinematics in feedback loop. In this approach only leg lengths and velocities are needed. The first one can be easily measured by standard sensors and leg velocity can be obtained by differentiating the measured leg lengths and filtering it using low pass filter. But dynamic modeling is complex in joint space because of a nonlinear coordinate transformation needed[27]. On the other hand task space approach is easier for dynamic modeling but it needs the highly nonlinear forward kinematics for feedback controller implementation[28]. Using the Lagrangian method, the actuator torque τ is given in task space as

$$M(X)\ddot{X} + C(X, \dot{X})\dot{X} + G(X) = Jp^T(X)\tau \tag{2.62}$$

Where $X = \begin{bmatrix} mx & my & mz & \alpha & \beta & \gamma \end{bmatrix}^T$ the task space position and orientation of the platform, $M(X)$ is 6x6 manipulator inertia matrix, $C(X, \dot{X})$ is 6x6 is also coriolis and centrifugal force matrix and $G(X)$, is 6x1 gravitational force matrix, $Jp(X)$ is 6x6 manipulator Jacobian matrix and τ is 6x1 the torque vector respectively. Alternatively, writing platform dynamics and leg dynamics separately, the dynamic equation can be rewritten as:

$$M_1(X)+M_2(X)\ddot{X}+(C_1(X,\dot{X})+C_2(X,\dot{X}))\dot{X}+G_1(X)+G_2(X)=Jp^T(X)\tau \quad (2.63)$$

Where: the parameters with subscript 1 are for the platform and parameters with subscript 2 are for leg. This kind of dynamic formulation helps in controller design where the manipulator dynamics is calculated in task space and the leg dynamics is calculated in joint space for improved efficiency. An extended version of this formulation has been suggested by Khalil [26, 29]. In his method, the closed chain is divided into two subsystems, platform and legs. The dynamics of the platform is calculated as a function of the position, velocity and acceleration of the platform $(X,\dot{X},\ddot{X})$ whereas the dynamics of the legs is calculated as a function of the joint position, velocity and acceleration $(q,\ddot{q},\ddot{q})$. Then the actuator torque is calculated as a sum of the two, after projecting them into the active joint space. Hence the active joint torque is given by

$$\tau = Jp^T(u+\sum_{i=1}^{6} J_{vi}{}^T J_i{}^T H_i) \quad (2.64)$$

Where: Jp, is the platform Jacobian matrix, u, is total force and moment exerted on the platform, J_{vi}, is Jacobian matrix that relates the velocity of the origin of the platform frame to the Cartesian velocity transferred from each leg to the platform, J_i is kinematic Jacobian matrix of leg i and it gives the velocity transferred to the platform from each leg i. and H_i is the dynamics of leg i.

There are few assumptions used in the dynamic equations during controller design. These are:

Assumption 1: Manipulator inertia matrix is non-singular

Assumption 2: Manipulator Jacobian matrix is nonsingular throughout the workspace

In addition to the above two assumptions, in the design of robust controllers, there are some assumptions about the uncertainties.

Assumption 3: The uncertainties in the inertia, Coriolis and centrifugal and gravitational matrices are bounded[15].

The dynamic model given in equation (2.62) has not included the actuator dynamics and friction. Thus, the system will have uncertainties because of inertia loading, unmodeled dynamics and friction of actuators. The uncertainties are assumed to have bounds and each term can be expressed as deviations as in equation (2.65) below.

$$M=M_N+\Delta M, C=C_N+\Delta C, G=G_N+\Delta G \quad (2.65)$$

The perturbation ΔM, ΔC and ΔG are assumed to have the following bounds:

$$\|\Delta M\| \le M_N, \|\Delta C\| \le C_N, \|\Delta G\| \le G_N \tag{2.66}$$

Using (2.62) and (2.63), equation (2.64) can be written in state space form as

$$\begin{aligned} \dot{X}_1 &= X_2 \\ \dot{X}_2 &= M_N^{-1}(Jp^T(X)\tau) - C(X_1, X_2)X_1 + G(X_1)) + d \end{aligned} \tag{2.67}$$

Where X_1 is 6x1 state vector of Cartesian space positions and orientations, X_2 is 6x1 state vector of the Cartesian space velocities and d is the lumped uncertainty term given by

$$d = -M_N^{-1}(\Delta M\, \ddot{X} + \Delta C\, \dot{X} + \Delta G) \tag{2.68}$$

The dynamic formulation of Stewart platform in joint space coordinates can be obtained by transforming the dynamics written in the task space coordinates as in (2.67) by substituting the relation below:

$$\begin{aligned} \dot{q} = Jp\,\dot{X} \rightarrow \dot{X} &= Jp^{-1}\dot{q} \\ \ddot{X} &= Jp^{-1}\ddot{q} + \dot{Jp}^{-1}\dot{q} \end{aligned} \tag{2.69}$$

Where $\dot{X} = \begin{bmatrix} {}^b_p v & {}^b_p w \end{bmatrix}^T$, is the end effector velocity vector ${}^b_p v$, is the platform's translational velocity vector and ${}^b_p w$, is the angular velocity vector of end effector and $\dot{q}$ is 6x1 the leg velocity vector. Therefore the dynamic matrix equation of a Stewart platform in the joint space coordinates will be expressed as follows:

$$A(q)\ddot{q} + H(q,\dot{q})\dot{q} + B(q) = \tau + d_j \tag{2.70}$$

Where:

$$A(q) = Jp^{-T}M(X)Jp^{-1}$$

$$H(q,\dot{q}) = Jp^{-T}M(X)\dot{Jp}^{-1} + Jp^{-T}C(X,\dot{X})Jp^{-1}$$

$$B(q) = Jp^{-T}G(X)$$

$$d_j = Jp^{-T}M_N^{-1}(-\Delta MJp^{-1}\ddot{q} - (\Delta M\,\dot{Jp}^{-1} + \Delta CJp^{-1})\dot{q} - \Delta G)$$

Where: q is 6x1 the leg length vector. In order to design any controllers or observer for this system, the dynamic model should be expressed in terms of state space variable as follows:

$$\ddot{q} = A(q)^{-1}\left(\tau - H(q,\dot{q}) \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix} - B(q) + d_j \right) \tag{2.71}$$

Let the system state variables to be defined as:

$$\left\langle \begin{matrix} x_1 = q \\ x_2 = \dot{q} \end{matrix} \middle| \begin{matrix} \dot{x}_1 = x_2 \\ \dot{x}_2 = \ddot{q} \end{matrix} \right\rangle$$

This implies that

$$\begin{aligned} \dot{x} &= f(x) + g(x)u(t) + d(x,t) \\ y &= h(x)x(t) \end{aligned} \tag{2.72}$$

Therefore, the state space equation of Stewart platform in joint space is given as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{pmatrix} 0_{6\times6} & I_{6\times6} \\ 0_{6\times6} & -A(q)^{-1}(H(q,\dot{q})) \end{pmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0_{6\times1} \\ A(q)^{-1} \end{bmatrix} + \begin{bmatrix} 0_{6\times6} \\ A(q)^{-1} \end{bmatrix} u(t) + \begin{bmatrix} 0_{6\times6} \\ A(q)^{-1} \end{bmatrix} d_j$$

$$y = \begin{bmatrix} I_{6\times6}, 0_{6\times6} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{2.73}$$

Where $x = \begin{bmatrix} q^T & \dot{q}^T \end{bmatrix}^T = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$ state vector, u is 6x1 input torque vector, and d_j is the bounded unknown disturbance torque/ uncertainties. The state vector x_1, can be given as

$$x_1 = y = li = \|Ei - bi\| = \sqrt{(BiPi)^T\, BiPi} \tag{2.74}$$

From the general dynamical equation of Stewart platform given in equation (2.70), where $x_1 = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & q_5 & q_6 \end{bmatrix}^T$ are the six leg lengths and $x_2 = \begin{bmatrix} q_7 & q_8 & q_9 & q_{10} & q_{11} & q_{12} \end{bmatrix}^T$ are the corresponding leg velocities, then state equation governing single leg with disturbance can be expressed as

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} q_7 \\ f(t,x_1,x_2) + g(t,x_1,x_2)u_1 + d_1 \end{bmatrix} \tag{2.75}$$

Where $f(t, x_1, x_2)$ and $g(t, x_1, x_2)$ depend upon the geometric parameters of the Stewart platform given in table 2.3.

Table2.3. Geometric specification of Stewart platform[30]

Parameters	Values	Description
Rb	0.8m	Base radius
Rp	0.5m	Platform radius
Lu	0.75m	CG of upper leg from top
Ll	0.15m	CG of lower leg from base
Mp	32kg	Mass of platform
Mul	4kg	Mass of upper leg
Mll	4kg	Mass of lower leg
h	1.5m	Initial height
Ixxp,Iyyp, Izzp	2, 2, 4 kg.m^2	Platform inertia
Ixxl, Iyyl ,Izzl	0.03, 0.03, 0.02 kg.m^2	Lower leg inertia
Ixxu,Iyyu, Izzu	0.75, 0.75, 0.018 kg.m^2	Upper leg inertia

2.3. Actuators Used to Drive Legs

Various actuators have been used to drive Gough-Stewart platform depending up on the size, accuracy or precision required and speed of operation[5, 31]. The most important ones are: hydraulic system, piezoelectric systems and electric motor.

2.3.1. Hydraulic Systems

Hydraulic actuators have high power-to-weight ratio and rapid response and they are used for applications such as flight and other motion simulators and telescopic antenna derives where the payload is very big. Their input output characteristic is not linear; they exhibit high nonlinearity. Moreover, hydraulic actuators resemble velocity source rather than force source and therefore their control is relatively complex than electric motors and their dynamics cannot be simply neglected rather it has to be modeled and used with platform dynamics if good motion tracking is to be obtained [27, 32-34].

2.3.2. Piezoelectric systems

Piezoelectric actuators are used for small sized micro manipulators used for robotic intervention [35], force sensing and pointing applications such as in optical

communication[36]. Generally, piezoelectric actuators offer high bandwidth, good control linearity and high force output with small size.

2.3.3. Electric motors

Various types of electric motors, including AC and DC servo motors and linear direct drive motors have been used to drive Gough-Stewart platform manipulators [8, 28, 37, 38]. In case of AC and DC servo motors, the rotary motion of the electric motor will be converted to linear motion using gear assembly and lead screw. Rotary actuated legs are able to extend and retract by adjusting the angle between two linkages. As opposed to the linear actuator, larger input torques need to be exerted by the motor to exert the same output forces on the ends of the legs. This design also has the characteristic that a small rotation of the motor results in a significant extension of the leg. This means that extensions and retractions of the leg can be much faster if the motors are able to output the required torque. The interesting feature of electric motors is their linear characteristics when they are driven using direct force control (current control) method. Mostly their dynamics can be neglected without having significant effect on controller performance.

CHAPTER THREE

3. CONTROL OF STEWART PLATFORM

A controller in the Stewart platform manipulator has to generate torque signals which will be applied at the legs such that the moveable platform moves in the desired direction at the desired speed. The relationship between force/torque and the acceleration of the center of the platform is nonlinear and coupled, makes the controller design a very challenging task. The two schemes which commonly used to control the Stewart platform are joint space control and task space control.

3.1. Joint Space Control

The joint space control which is considered in this thesis is developed using the information of joint displacement/leg lengths, in such a way the desired task space position, velocity and acceleration of the platform center converted to desired joint leg lengths and close the loop by using measured leg lengths as feedback. The individual legs of a parallel manipulator are considered as independent systems and coupling effects among the legs are considered as disturbances. In this method each leg of the manipulator is controlled as a single input single output (SISO) system. The independent joint space control is widely employed in industrial applications where a multi-link manipulator is divided into several independent links. The performance of this type of controller can be improved by a disturbance observer as done by [39, 40]. The SISO controller implementation is cheap and can be implemented in parallel easily, however this method results in synchronization error as explained by[13, 41].

3.2. Task Space Control

In this approach, the desired task space position, velocity, and acceleration are not converted to the desired leg length rather it is used directly by taking measured or estimated task space position, velocity, and acceleration as feedback. Hence in this approach, the control signal is calculated in task space, and then it will be converted to joint space using the Jacobian matrix. The task space control achieves better performance by addressing the system coupling, which results in a multi-input multi-output (MIMO) system and has the potential to compensate for coupling errors. But the implementation of task space controller involves real-time measurement of position and velocity, which is complex and requires costly sensors, thus restricts the applications of task space controller in real-time applications by[42]. A simplified block diagram of these two approaches is given in figure.3.1 and 3.2.

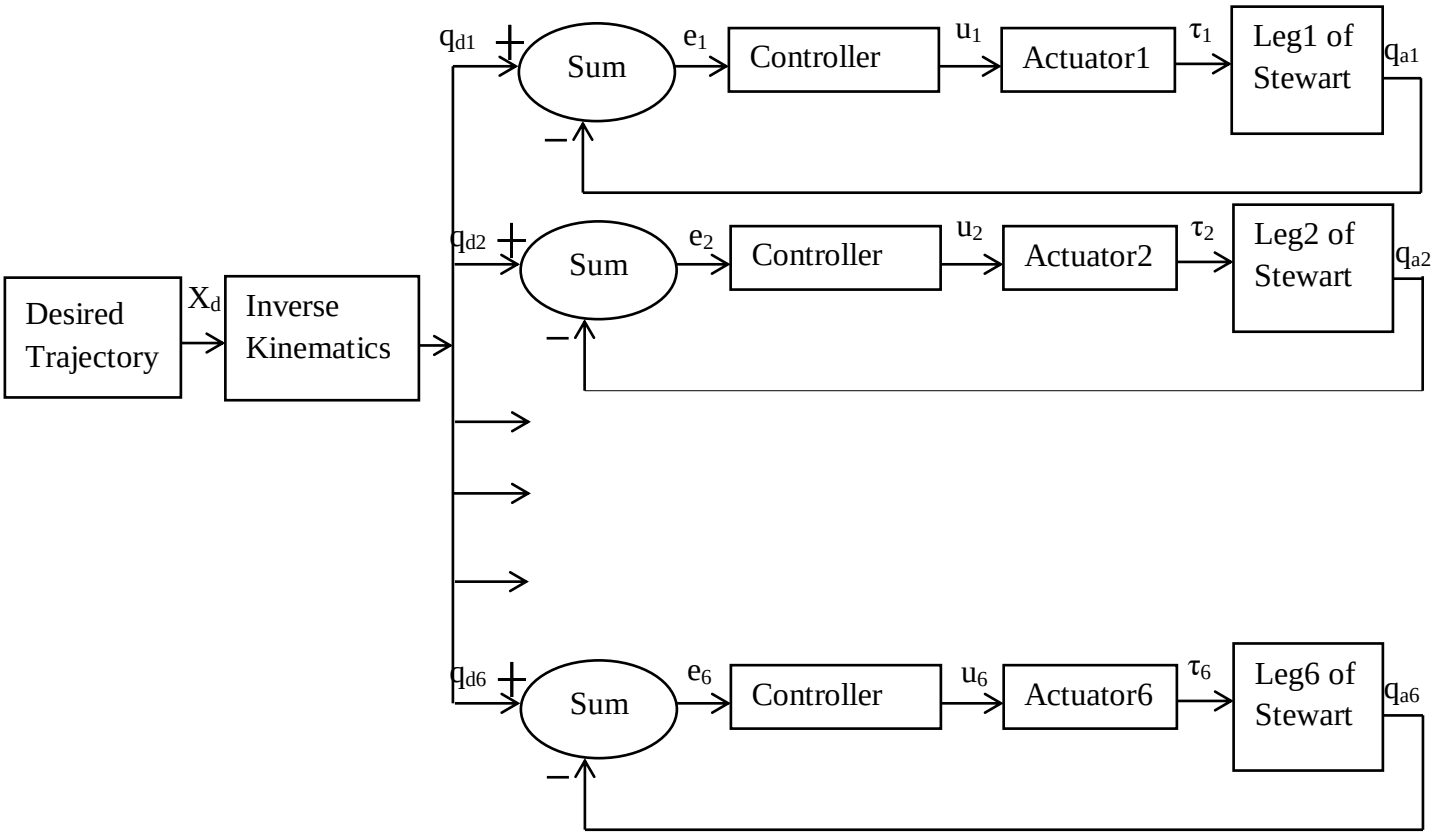


Figure3.1. Joint space trajectory tracking control approach for a Stewart platform manipulator

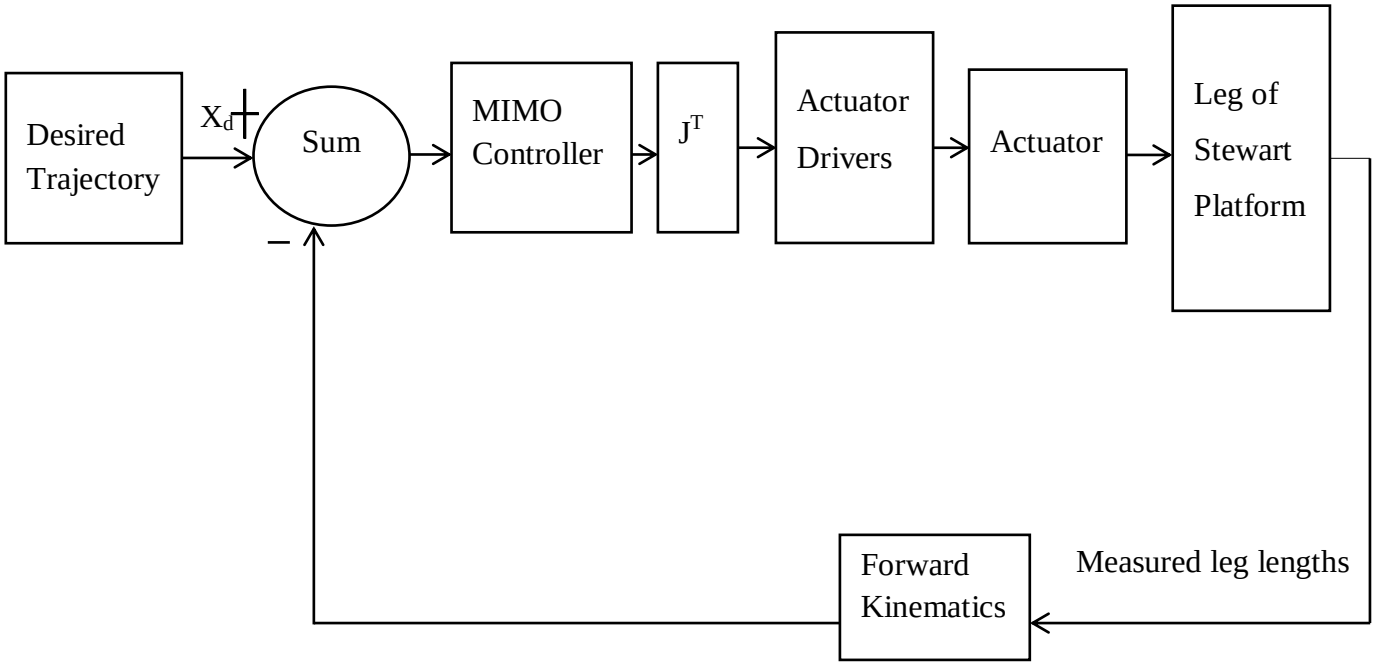


Figure3.2. Task space trajectory tracking control approach for a Stewart platform manipulator

3.3. Passivity Based Control of Stewart Platform in Joint Space

3.3.1. Concept of Passivity Based Control

Passivity is a property presented in many physical systems such as electronic systems, electromechanical systems, and robotics, and so on. The primary idea in passive systems is that the power flowing into the system is not less than the increase in storage. Thus, it cannot store more energy than the energy being supplied to it from the outside, with the difference being the dissipated energy. It is clear from this energy interpretation that the concept of passivity is related to the stability properties of the systems. Viewing a dynamical system as a virtual energy transformation device, which can decompose a complex nonlinear system into several simpler subsystems that, upon careful interconnection, and adds up their energies to determine the overall system's behavior[43]. This allows recasting the control problem as finding a dynamical system and an interconnection pattern such that the overall energy function takes the desired form. This energy shaping approach is the essence of passivity based control (PBC). The term passivity based control (PBC) was coined in 1989 in the context of adaptive control of the dynamic system to make the closed-loop system passive. To define passivity mathematically, consider a dynamic model of a nonlinear system given by

$$\dot{x} = f(x) + g(x)u(t) + d(x,t)$$
$$y = h(x) \tag{3.1}$$

Where $x \in R^n$, is state vector, $y \in R^p$, is output vector, f(x) and g(x) are $nx1$ and nxm dimensional vector valued and matrix valued smooth nonlinear functions, u is mx1 dimensional vector of control signal, d is uncertainty term which includes matched and unmatched uncertainties due to parameter variations and model inaccuracies.

The system described in (3.1) is said to be passive, if there exists a continuous nonnegative storage functions $V(x(t))$ with $V(0) = 0$ such that for any $x \in R^n$, $y \in R^p$ and $u \in R^m$ the following inequality holds:

$$V(t) - V(0) \le \int_0^t y^T u(\tau) d\tau - d(t) \tag{3.2}$$

Where $d(t)$ is nonnegative function that captures the dissipation effect, e.g., due to resistance or friction, etc. The term on the left side is the stored energy and the first term on the right side is the energy supplied to the system.

3.3.1.1. Design and Analysis of Controller

Consider the dynamic model of the Stewart platform given (2.68) without considering actuator friction and external disturbance given as

$$A(q)\ddot{q}+H(q,\dot{q})\dot{q}+B(q)=\tau \tag{3.3}$$

Where q is vector of joint parameter, i.e. vector of elongations of six legs, A is 6x6 manipulator inertia matrix, H is also 6x6 coriolis and centrifugal torque/force, B is 6x1 gravitational torque/force,τ is 6x1 actuator torque, Then, the corresponding joint space tracking error and its rate vector can be given as

$$\begin{aligned} e &= q_d - q \\ \dot{e} &= \dot{q}_d - \dot{q} \end{aligned} \tag{3.4}$$

Then the error dynamics in state space given as

$$\begin{aligned} \dot{e}_1 &= e_2 \\ \dot{e}_2 &= \ddot{q}_d - A(q)^{-1}(\tau - H(q,\dot{q})\dot{q} - B(q)) \end{aligned} \tag{3.5}$$

Then the control objective is to find a control signal τ that can stabilize the system at $e=\dot{e}=0$ which is not the open loop equilibrium point so that the joint position $q(t)$ asymptotically tracks the trajectory of desired joint position $q_d(t)$. From (3.5), the storage function can be constructed as follows:

$$V(t,e_1,e_2)=\frac{1}{2}e_2{}^T A(q)e_2+\frac{1}{2}e_1{}^T Kpe_1 \tag{3.6}$$

Differentiate the storage energy function given in (3.6) with respect to the time, gives

$$\dot{V}=e_2{}^T A(q)\dot{e}_2+\frac{1}{2}e_2{}^T \dot{A}e_2+e_1{}^T Kp\dot{e}_1 \tag{3.7}$$

$$\dot{V}=e_2{}^T (A(q)\ddot{q}_d-A(q)\ddot{q})+\frac{1}{2}\dot{q}^T \dot{A}\dot{q}+e_1{}^T Kp\dot{e}_1$$

$$\dot{V}=e_2{}^T AA^{-1}(-\tau+H\dot{q}+B)+e_2{}^T A\ddot{q}_d+\frac{1}{2}e_2{}^T \dot{A}e_2+e_1{}^T Kp\dot{e}_1$$

$$\dot{V}=-e_2{}^T\tau-e_2{}^T He_2+e_2{}^T B+e_2{}^T H\dot{q}_d+e_2{}^T A\ddot{q}_d+\frac{1}{2}e_2{}^T \ddot{A}e_2+e_1{}^T Kp\dot{e}_1$$

$$\dot{V}=-e_2{}^T\tau+e_2{}^T B+e_2{}^T H\dot{q}_d+e_2{}^T A\ddot{q}_d+\frac{1}{2}e_2{}^T (\dot{A}-2H)e_2+(e_1{}^T Kp\dot{e}_1)^T$$

$$\dot{V} = e_2^T(-\tau + B + H\dot{q}_d + A(q)\ddot{q}_d + Kpe_1) \tag{3.8}$$

Due to the skew symmetry of $e_2^T(\dot{A} - 2H)e_2$ to stabilize the closed loop system additional input, v is chosen, such that the control signal can be given as

$$\tau = B + H\dot{q}_d + A(q)\ddot{q}_d + Kpe_1 + v \tag{3.9}$$

This makes the storage function negative semi definite as

$$\dot{V} = -e_2^T v \le 0 \tag{3.10}$$

So that we can choose the v as

$$v = Kde_2 \tag{3.11}$$

Where v, is energy shaping and, in such a way that both the kinetic energy and potential energy reshaped to $\frac{1}{2}e_2^T A(q)e_2$ and $\frac{1}{2}e_1^T Kpe_1$ respectively. Then the control signal become

$$\tau = \tau_{pd+} = B + H\dot{q}_d + A(q)\ddot{q}_d + Kpe_1 + Kde_2 \tag{3.12}$$

Which take the form PD+ control for some positive 6x6 diagonal matrices Kp and Kd. PD+ control is one of the simplest control laws that used in the control of robot manipulators with a formal guarantee of the achievement of the motion control objective, globally. The practical implementation of PD+ control requires the exact knowledge of the model of the manipulator, that is, of $B(q), H(q,\dot{q})$, and $A(q)$. In addition it is necessary to know the desired trajectory, $q_d(t)$, $\dot{q}_d(t)$ and $\ddot{q}_d(t)$ as well as to have the measurements $q(t)$, $\dot{q}(t)$ and $\ddot{q}(t)$. Figure 3.3 shows the corresponding block-diagram of the PD+ control for Stewart platform manipulators.

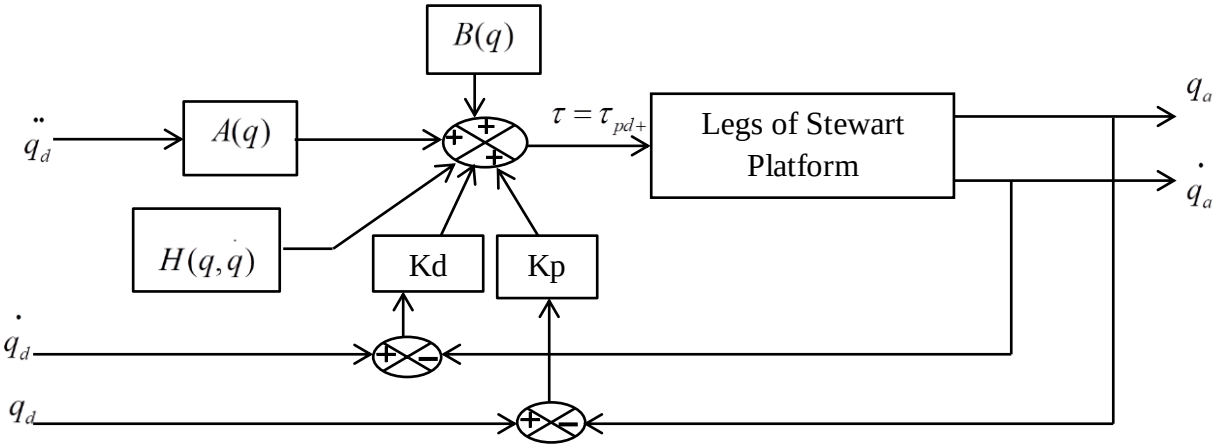


Figure3.3. PD+ control of Stewart platform in joint space

The equation which governs the behavior in closed loop is obtained by substituting the control action τ of the control signal given in (3.12) in the equation of the dynamic model of the Stewart platform given in (3.3) to get

$$A(q)\ddot{e}+H(q,\dot{q})\dot{e}=-Kpe-Kd\dot{e} \tag{3.13}$$

And this can be rewritten in terms of the state $\begin{bmatrix} e^T & \dot{e}^T \end{bmatrix}^T$ in state space as

$$\dot{e}=\dot{e}$$

$$\ddot{e}=A(q)^{-1}(-Kpe-Kd\dot{e}-H(q,\dot{q})). \tag{3.14}$$

From (3.14) it is clear to see that the only equilibrium point of the closed loop equation is the origin $\begin{bmatrix} e^T & \dot{e}^T \end{bmatrix}^T=0$. Therefore, if $q(0)=q_d(0)$ and $\dot{q}(0)=\dot{q}_d(0)$ then $q(t)=q_d(t)$, and $\dot{q}(t)=\dot{q}_d(t)$ for all $t\geq 0$. However, to draw a conclusion for the case when $q(0)\neq q_d(0)$ or $\dot{q}(0)\neq\dot{q}_d(0)$ it is necessary to show the stability of the equilibrium points using the storage function given in (3.6) as follows.

$$V(t,e,\dot{e})=\frac{1}{2}\dot{e}^T A(q)\dot{e}+\frac{1}{2}e^T Kpe \tag{3.15}$$

Taking the time derivative of (3.15), we have

$$\dot{V}(t,e,\dot{e})=\dot{e}^T A(q)\ddot{e}+\frac{1}{2}\dot{e}^T \dot{A}(q)\dot{e}+e^T Kp\dot{e} \tag{3.16}$$

$$\dot{V}(t,e,\dot{e})=\dot{e}^T A(q)A(q)^{-1}(-Kpe-Kd\dot{e}-H(q,\dot{q}))+\frac{1}{2}\dot{e}^T \dot{A}(q)\dot{e}+e^T Kp\dot{e}$$

$$\dot{V}(t,e,\dot{e})=\dot{e}^T(-Kpe-Kd\dot{e}+\frac{1}{2}(\dot{A}(q)-2H(q,\dot{q}))\dot{e}+Kpe) \tag{3.17}$$

After some simplification (3.17) becomes

$$\dot{V}(t,e,\dot{e})=-\dot{e}^T Kd\dot{e}\leq 0 \tag{3.18}$$

It is clear from (3.18) the origin of equilibrium points $\begin{bmatrix} e^T & \dot{e}^T \end{bmatrix}^T=0$ are stable and the state remains bounded using lyapunov theorems. Therefore, based on the passivity theory, a storage function is constructed, and additional input is introduced to lead the closed-loop system stable. It is clear from (3.12), passivity based control usually requires an accurate

system model to analyze the physical property and role of each term of a dynamical system, thus its practical application is somehow limited resulted from the lack of robustness. Particularly for Stewart Platform manipulator in which the relationship between the forces/torques which has to be given at the legs to drive the system in the desired trajectory and the acceleration of the center of the platform is highly nonlinear and coupled. Moreover, the dynamic model of the manipulator has matched uncertainties from parameter variation, actuator friction, and so on. Therefore to effectively utilize the structural advantages of the manipulator, passivity-based robust control has been designed in this thesis to enhance the robustness in the presence of uncertainties due to parameter variations and unmodeled dynamics.

3.4. Passivity Based Sliding Mode Controller Design for Stewart Platform

3.4.1. Need of Sliding Mode Control

There is always a discrepancy between the actual plant and its mathematical model. These discrepancies arise from unknown external disturbance, plant parameters variation, and unmodeled dynamics. Designing a control law that provides the desired performance in presence of these uncertainties/disturbances is a very challenging task. This has led to interests in the development of robust control methods. One of a robust control design technique is sliding mode control. Sliding mode control is a robust design technique, which can withstand external disturbance and parameter variation[14]. Since 1950, it has been in use for control of various types of systems including linear and nonlinear systems, SISO and MIMO systems, discrete time models, large scale and infinite dimensional systems and stochastic systems[44]. Therefore to enhance system robustness by using a sliding mode mechanism the additional input given in (3.10) has been redesigned and given as

$$v = v_{es} + v_{sl} \tag{3.19}$$

Where:

$$v_{es} = Kd\,\dot{e}$$

And v_{sl}, is a sliding mode control signal given by

$$v_{sl} = \tau_{nominal} + \tau_{disco} \tag{3.20}$$

Where $\tau_{nominal}$, is nominal control and τ_{disco}, is a discontinuous control given by

$$\tau_{disco} = K_3 f(s) = K_3 sign(s) \tag{3.21}$$

Where K_3 is gain of switching function, f(s) is switching function and s is a sliding surface

given as

$$s = \left(\frac{d}{dt} + C_1 \right) e = \dot{e} + C_1 e \tag{3.22}$$

Therefore the overall control input for the system given in (2.69) is given as

$$\tau = \tau_{pd+} + v_{sl} \tag{3.23}$$

Where τ_{pd+}, PD+ control given in (3.12), and v_{sl}, is robust control designed using SMC techniques, then substitute PB-SMC given in (3.23) into the derivative of storage function given in (3.8), it gives

$$\dot{V} = -e_2^{\,T} (Kde_2 + K_3 sign(s) + \tau_{nominal}) \leq 0 \tag{3.24}$$

This shows that the total energy in the system decays and the sliding surface converges to zero. The switching gain K_3 is chosen to ensure the convergence of the sliding surface given in (3.22) to zero, while Kd reshapes the system into output strictly passive, such that the closed loop system is stable. Thus, design of additional input v, given in (3.19) leads the system to be passive through energy reshaping control v_{es}, to ensure the stability of the closed loop system and enhance system robustness by using sliding mode technique v_{sl}. However, classical sliding mode control has certain drawbacks. These are: (i) chattering or high frequency oscillation of control signal, (ii) lack of robustness during the sliding phase, (iii) reduced life time of actuators and (iv) is unable to compensate unmatched uncertainties [45]. Integral sliding mode controller is an improvement of the conventional SMC that can solve the reaching phase problem and improve the robustness against unmatched uncertainties. Its basic structure and advantages has been discussed in [12, 13, 15, 30, 46, 47]. The reaching phase problem of conventional SMC is solved by integral SMC because, integral SMC has a nonlinear sliding surface with an integral term that make the system states to be on sliding surface from the initial time [48]. The other important advantage of the integral sliding surface is it helps to achieve a stable sliding dynamics and enables to design a control signal that completely compensates matched uncertainties and minimize unmatched uncertainty to any desired level[14]. This improves robustness to unmatched uncertainty.

3.4.2. Integral Sliding Mode Controller Design

The integral sliding mode control is a combination of two controls, a nominal control which is responsible to achieve desired performance of the system without disturbance and a discontinuous control which rejects the disturbance and uncertainty. Like any other sliding

mode controller design, the design of the controller involves two basic steps: the design of stable sliding surface and design of a control law.

3.4.2.1. Design of Integral Sliding Surface

Let as consider again the dynamic model of a Stewart manipulator given in (2.70)

$$A(q)\ddot{q} + H(q,\dot{q})\dot{q} + B(q) = \tau + d_j \tag{2.25}$$

Where q is vector of joint parameter, i.e. vector of elongations of six legs, A is 6x6 manipulator inertia matrix, H is also 6x6 coriolis and centrifugal torque/force, B is 6x1 gravitational torque/force, τ is the actuator torque and d_j is the disturbance torque. The same assumptions on the mass matrix and manipulator Jacobian as given in chapter two and other chapters are considered. The assumption on the uncertainty of the dynamic parameters is also the same.

The joint space tracking error can be given as

$$\begin{aligned} e &= q_d - q \\ \dot{e} &= \dot{q}_d - \dot{q} \end{aligned} \tag{2.26}$$

Where q is the actual leg length and q_d is the desired leg length. Then, the error dynamics in state space given as

$$\begin{aligned} \dot{e}_1 &= e_2 \\ \dot{e}_2 &= \ddot{q}_d - A_N^{-1}(\tau - H(q,\dot{q})_N \dot{q} - B(q)_N + d_j) \end{aligned} \tag{2.27}$$

Where the nominal joint space parameters are obtained from task space values by using

$$A(q)_N = Jp^{-T} M(X)_N Jp^{-1} \tag{2.28}$$

$$H(q,\dot{q})_N = Jp^{-T} M(X)_N \dot{Jp}^{-1} + Jp^{-T} C(X,\dot{X})_N Jp^{-1} \tag{2.29}$$

$$B(q)_N = Jp^{-T} G(X) \tag{2.30}$$

$$d_j = -Jp^{-T} M_N^{-1}(\Delta M Jp^{-1}\ddot{q} + (\Delta M \dot{Jp}^{-1} + \Delta C Jp^{-1})\dot{q} + \Delta G) \tag{2.31}$$

Comparing equation (2.70) and (3.27)

$$x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \tag{2.32}$$

$$f = \begin{bmatrix} e_2 \\ \ddot{q}_d - A_N^{-1}(\tau - H(q,\dot{q})_N \dot{q} - B(q)_N \end{bmatrix} \tag{2.33}$$

$$g = \begin{bmatrix} 0_{6\times 6} \\ -A_N^{-1} \end{bmatrix} \tag{2.34}$$

And d is

$$d = \begin{bmatrix} 0_{6\times 6} \\ A_N^{-1} \end{bmatrix} d_j \tag{2.35}$$

Then the integral sliding surface for the Stewart platform manipulator is given as [15]

$$s(x,t) = C(x - x(0) - \int_{t_0}^{t} (f(x,m) + g(x,m)\tau_{nominal})dm \tag{2.36}$$

Where $\tau_{nominal}$ is the nominal control torque, $x = \begin{bmatrix} e & \dot{e} \end{bmatrix}^T$ and $x(0)$ is the initial condition of the error dynamics and f and g are as given in (3.33) and (3.34) above.

3.4.2.2. Design of Integral Sliding Mode Control Law

The control law v_{sl}, for ISMC has the following form.

$$v_{sl} = \tau_{nominal} + \tau_{disco} \tag{3.37}$$

Where $\tau_{nominal}$, is nominal control and τ_{disco}, is a discontinuous control given by

$$\tau_{disco} = K_3 f(s) = K_3 sign(s) \tag{3.38}$$

Where K_3 is gain of switching function, f(s) is switching function and s is a sliding surface given in (3.36) as

$$s(x,t) = C(x - x(0) - \int_{t_0}^{t} (f(x,m) + g(x,m)\tau_{nominal})dm \tag{3.39}$$

Where: C and $x(0)$ are the projection matrix and initial condition of the system respectively. The system trajectory always starts from the sliding surface since the initial values of the state are incorporated in the surface. When the system is on the sliding surface the equivalent control is calculated by equating the derivative of the sliding surface to zero.

$$\dot{s} = C\dot{x} - C(f(x,t) + g(x,t)\tau_{nominal}) = 0 \tag{3.40}$$

$$\dot{s} = C(f(x,t) + g(x,t)v_{sl}) + Cd_j - C(f(x,t) + g(x,t)\tau_{nominal}) = 0$$

$$\dot{s} = C(f(x,t) + g(x,t)(\tau_{nominal} + \tau_{disco})) + Cd_j - C(f(x,t) + g(x,t)\tau_{nominal}) = 0$$

$$\dot{s} = Cg(x,t)\tau_{disco} + Cd_j = 0 \tag{3.41}$$

$$-g(x,t)\tau_{disco} = d_j \tag{3.42}$$

Hence the equivalent control signal becomes

$$\tau_{eq} = \tau_{nominal} + A_N d_j \tag{3.43}$$

Where $\tau_{nominal}$ is the nominal control given in (3.49), with this nominal control the sliding mode dynamics of the system becomes

$$\dot{e}_1 = e_2$$

$$\dot{e}_2 = -Kpe_1 - Kde_2 - Ki\int e_1 dt \tag{3.44}$$

It is clear from (3.44) when the system is on the sliding mode the uncertainty is compensated by the discontinuous control and the sliding mode dynamics is stable. But the original control which is applied to the plant given in (2.68) also contains discontinuous control which will lead to chattering phenomenon. The chattering is highly undesirable for practical point of view implementation of integral sliding mode control. To overcome this problem and improve utility of sliding mode controller a PD+ smooth integral sliding mode controller is proposed in this thesis.

3.4.2.3. Design of Nominal Controller

The nominal control $\tau_{nominal}$, is designed to drive the platform to the desired position for finite time when the system is free from disturbances. This can be any linear control, PID, LQR (linear quadratic regulator), state feedback, optimal control, time-varying control, adaptive control, etc. In this thesis, PID computed torque control is considered as the nominal control, which is given as follows:

Consider again the error dynamics in state space given in (3.27) as

$$\dot{e}_1 = e_2$$

$$\dot{e}_2 = \ddot{q}_d - A_N^{-1}(\tau - H(q,\dot{q})_N \dot{q} - B(q)_N + d_j) \tag{3.45}$$

Now define the control input as

$$u = \ddot{q}_d - A(x_1)^{-1}(\tau_{nominal} - H(x_1,x_2)x_2 - B(x_1)) \tag{3.46}$$

And the disturbance as

$$d = -A_N^{-1} d_j \tag{4.47}$$

The control signal in (3.46) can be selected as

$$u = -Kpe_1 - Kde_2 - Ki\int e_1 dt \tag{4.48}$$

Therefore, the nominal control signal is given as

$$\tau_{nominal} = A(x_1)(\ddot{q}_d - u) + H(x_1, x_2)x_2 + B(x_1) \tag{3.49}$$

Then substitute the control signal given in (3.48) in to (3.49), then nominal control signal becomes

$$\tau_{nominal} = A(x_1)(\ddot{q}_d + Kpe_1 + Kde_2 + Ki\int e_1 dt) + H(x_1, x_2)x_2 + B(x_1) \tag{3.50}$$

And the error dynamics given as

$$\dot{e}_2 = -Kpe_1 - Kde_2 - Ki\int e_1 dt \tag{3.51}$$

Let $\dot{\varepsilon} = e_1$, then the error dynamics can be rewritten as

$$\dot{e}_2 = -Kpe_1 - Kde_2 - Ki\varepsilon \tag{3.52}$$

This can be written in state space as

$$\begin{bmatrix} \dot{\varepsilon} \\ \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -Ki & -Kp & -Kd \end{pmatrix} \begin{bmatrix} \varepsilon \\ e_1 \\ e_2 \end{bmatrix} \tag{3.53}$$

Then the closed loop characteristic polynomial is given by

$$P_{cl}(s) = |sI - A| = \left| \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{pmatrix} - \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -Ki & -Kp & -Kd \end{pmatrix} \right| \tag{3.54}$$

$$P_{cl}(s) = \left| \begin{pmatrix} s & -1 & 0 \\ 0 & s & -1 \\ Ki & Kp & s + Kd \end{pmatrix} \right|$$

$$P_{cl}(s) = s^3 + Kds^2 + Kps + Ki \tag{3.55}$$

Where $Kp = diag(Kpi)$, $Kd = diag(Kdi)$, and $Ki = diag(Kii)$, i=1,2……6.

Hence PID gain can be chosen such that the characteristic polynomial must be Hurwitz, i.e. $0 < Kii < KpiKdi$

3.4.3. Smooth Integral Sliding Mode Controller Design

3.4.3.1. Need of Continuous Control

As we see from (3.44), the uncertainty is compensated and the sliding mode dynamics is stable, but the control input becomes discontinuous. Therefore, this control is not desirable from the implementation point of view because the oscillations caused by the high-frequency switching of discontinuous control will excite the unmodeled dynamics in the closed loop, known as chattering which lead to the necessity of continuous control. To replace the discontinuous control part of integral sliding mode control super twisting control has been introduced. Thus, the control input for the system given in (2.68) is $v_{sl} = \tau_{pd+} + \tau_{nominal} + \tau_{stc}$, where $\tau_{nominal}$ is the nominal control given in (3.49) and τ_{stc} is super twisting algorithm based second order continuous control instead of discontinuous control for the existing integral sliding mode control. Therefore, the overall control input for the system given in (2.68) is PD+ smooth integral sliding mode control, PD+ control is designed based on passivation principle so that the closed loop system becomes globally uniform and asymptotically stable. Furthermore, a great robustness can be provided through SISMC employed by PB-SISMC.

3.4.3.2. Design and Analysis of Controller

Consider again the system in (2.68), but now the control input for the plant is given by

$$v_{sl} = \tau_{nominal} + \tau_{stc} \tag{3.56}$$

Where $\tau_{nominal}$ is the nominal control given in (3.49) and τ_{stc} is super twisting control (STC) which is a continuous control and can be represented mathematically as [49].

$$\begin{aligned} \tau_{stc} &= K_1 |s|^{\frac{1}{2}} sign(s) + w \\ \dot{w} &= K_2 sign(s) \end{aligned} \tag{3.57}$$

Where K_1, K_2 and are diagonal matrices and s is the sliding surface given in (3.36)

$$s(x,t) = C(x - x(0) - \int_{t_0}^{t} (f(x,m) + g(x,m)\tau_{nominal})dm \tag{3.58}$$

Where $\tau_{nominal}$ is the nominal control torque, $x = \begin{bmatrix} e & \dot{e} \end{bmatrix}^T$ and $x(0)$ is the initial condition of the error dynamics and f and g are as given in (3.33) and (3.34) and C is the projection matrix. The integral sliding surface is chosen such that the system trajectories start from the sliding surface and if disturbance comes into the picture τ_{stc} becomes active and disturbance becomes

compensated. Differentiate the sliding surface s given in (3.58) with respect to time and equal to zero yields

$$\dot{s} = C\dot{x} - C(f(x,t) + g(x,t)\tau_{nominal}) = 0 \tag{3.59}$$

$$\dot{s} = C(f(x,t) + g(x,t)v_{si}) + Cd_j - C(f(x,t) + g(x,t)\tau_{nominal}) = 0$$

$$\dot{s} = C(f(x,t) + g(x,t)(\tau_{nominal} + \tau_{stc})) + Cd_j - C(f(x,t) + g(x,t)\tau_{nominal}) = 0$$

$$\dot{s} = Cg(x,t)\tau_{stc} + Cd_j = 0 \tag{3.60}$$

It is clear from (3.60) knowing the bound of $\left|\dot{d}_j\right| \le Km$ s and $\dot{s}$ converges to zero in finite time for appropriate selection of gains K_1 and K_2 which are 6x6 diagonal matrices [49]. The magnitude of Km required to achieve stability is

$$Km \ge \left\| -Jp^{-T} M_N^{-1} (\Delta M J p^{-1} \ddot{q} + (\Delta M \dot{J} p^{-1} + \Delta C J p^{-1}) \dot{q} + \Delta G) \right\| \tag{3.61}$$

And also from (3.60) the disturbance can be estimated as

$$\dot{s} = Cg(x,t)\tau_{stc} + Cd_j = 0$$

$$d_j = -g(x,t)\tau_{stc} = A_N^{-1}\tau_{stc} \tag{3.62}$$

It is clear from (3.62), when the system is on sliding mode the disturbance can be estimated successfully as $d_j = -A_N^{-1}\tau_{stc}$ and the effect of disturbance will canceled out. The overall control action becomes continuous due to the combination of two continuous control results in chattering free control output. However the knowledge of $\dot{s}(t_0)$ is necessary and since $\dot{s}$ is immeasurable, to start the second order sliding mode from $t = t_0 = 0$, the initial value of s and $\dot{s}$ must be zero. In case of non-zero initial value of disturbance the sliding mode will start after some finite time $t \ge t_0$ which can be designed as small as possible by appropriate selection of gains K_1 and K_2. Once s and $\dot{s}$ become zero then, it remains zero forever in spite of disturbance.

For a nominal controller given in (3.45) above

$$\tau_{nominal} = A(x_1)(\ddot{q}_d + Kpe_1 + Kde_2 + Ki\int e_1 dt) + H(x_1, x_2)x_2 + B(x_1) \tag{3.63}$$

Then, for some positive diagonal matrices Kp, Kd and Ki then the sliding dynamics of the system becomes,

$$\dot{e}_1 = e_2$$

$$\dot{e}_2 = -Kpe_1 - Kde_2 - Ki\int e_1 dt \tag{3.64}$$

This shows a stable sliding dynamics. And the integral sliding surface given in (3.58) is implemented after the expressions (3.33)-(3.35) are substituted and after simplification it becomes

$$s = C_1(e = -e(0) - \int \dot{e}_n \, d\tau) + \dot{e} - e(0) + \int (Kpe + Kd\dot{e} + Ki\int ed\tau)dw \tag{3.65}$$

And the overall control input for the system given in (2.68) is given by

$$\tau = \tau_{pd+} + \tau_{nominal} + \tau_{stc} \tag{3.66}$$

Where τ_{pd+} is given by (3.12), $\tau_{nominal}$ is given by (3.63) and τ_{stc} is given by (3.57).

The block diagram representation of the proposed control system is given in figure 3.4. In the figure, $q_d i$ is the desired leg length of leg i, $\dot{q}_d i$ is the desired leg velocity of leg i and $\ddot{q}_d i$ is desired leg acceleration of leg i. Similarly $q_a i$, is the actual leg length of leg i and $\dot{q}_a i$ is the actual leg velocity of leg i.

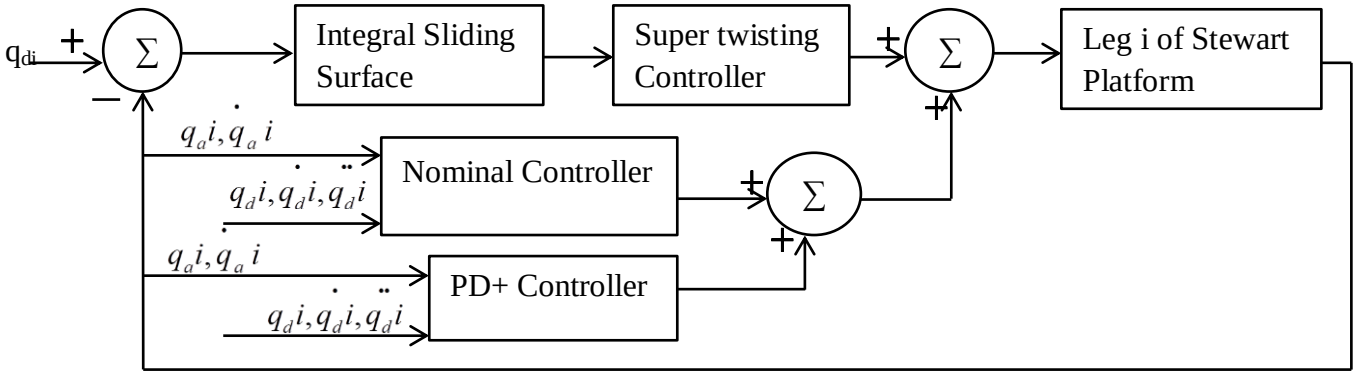


Figure3.4. Block diagram representation of joint space PD+ smooth integral sliding mode control of Stewart platform.

3.5. Particle Swarm Optimization Algorithm

3.5.1. Introduction to PSO

Particle swarm optimization is one of the most popular nature-inspired metaheuristic optimization algorithm developed by James Kennedy and Russell Eberhart in 1995[50]. The principle of PSO optimization technique is to search for the best solution in n-dimension in

the search space. That can be done by a collaborative share between each particle in a swarm and it is used for optimizing nonlinear control systems for its satisfactory results.

Particle swarm optimization (PSO) is inspired by social and cooperative behavior displayed by various species to fill their needs in the search space. The algorithm is guided by personal experience (Pbest), overall experience (Gbest), and the present movement of the particles to decide their next positions in the search space. Further, the experiences are accelerated by two factors c1and c2, and two random numbers generated between [0, 1] whereas the present movement is multiplied by an inertia factor w [51, 52]. The initial population (swarm) of size N and dimension D is denoted as $X = \begin{bmatrix} X_1 & X_2 & . & . & . & X_N \end{bmatrix}^T$, where T denotes the transpose operator. Each individual (particle) $X_i (i = 1, 2, ...N)$ is given as $X_i = \begin{bmatrix} X_{i,1} & X_{i,2} & . & . & . & X_{i,N} \end{bmatrix}$. Also, the initial velocity of the population is denoted as $V = \begin{bmatrix} V_1 & V_2 & . & . & . & V_N \end{bmatrix}^T$. Therefore, the velocity of each particle $X_i (i = 1, 2, ...N)$ is given as $V_i = \begin{bmatrix} V_{i,1} & V_{i,2} & . & . & . & V_{i,N} \end{bmatrix}$. The index i, varies from 1 to N whereas the index j varies from 1 to D. The detailed algorithms of various methods are described below for completeness.

$$V_{i,j}^{K+1} = wV_{i,j}^{K} + c_1 r_1 (P_{best\,i,j}^{\ K} - X_{i.j}^{K}) + c_2 r_2 (G_{best\,i,j}^{\ K} - X_{i,j}^{K}) \tag{3.67}$$

$$X_{i,j}^{K+1} = X_{i,j}^{K} + V_{i,j}^{K+1} \tag{3.68}$$

Where,

D is dimension index (number of decision variable), N, is number of particles in the group K, is pointer of iteration, w, c_1 and c_2 is inertia, cognitive and social factor respectively and r_1, r_2 are uniform random numbers in the interval (0 1). $X_{i,j}^{K}$, is current position of particle 'i' at iteration, $V_{i,j}^{K}$ velocity of particle at iteration i, $P_{best\,i,j}^{\ K}$ is best previous position of the ith particle and $G_{best\,i,j}^{\ K}$, is best particle among all the particles in the swarming population.

In equation (3.67) the first term represents the effect of the inertia of the particle, the 2nd term represents the particle memory influence, and the 3rd term represents the swarm (society) influence, and $P_{best\,i,j}^{\ K}$ represents personal best jth component of ith individual whereas, $G_{best\,i,j}^{\ K}$ represents jth component of the best individual of population up to iteration K. The search mechanism of PSO in multidimensional search space is shown in figure 3.5.

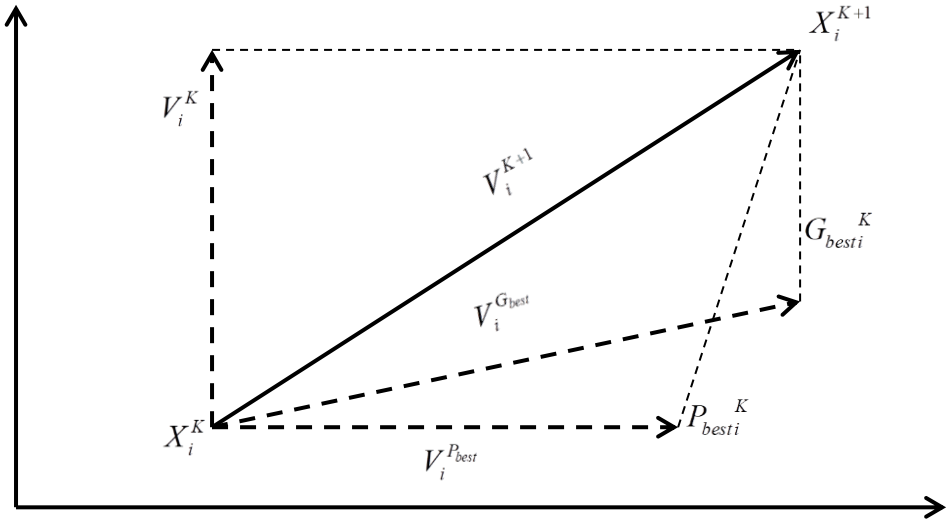


Figure3.5. PSO search mechanism in multidimensional search space.

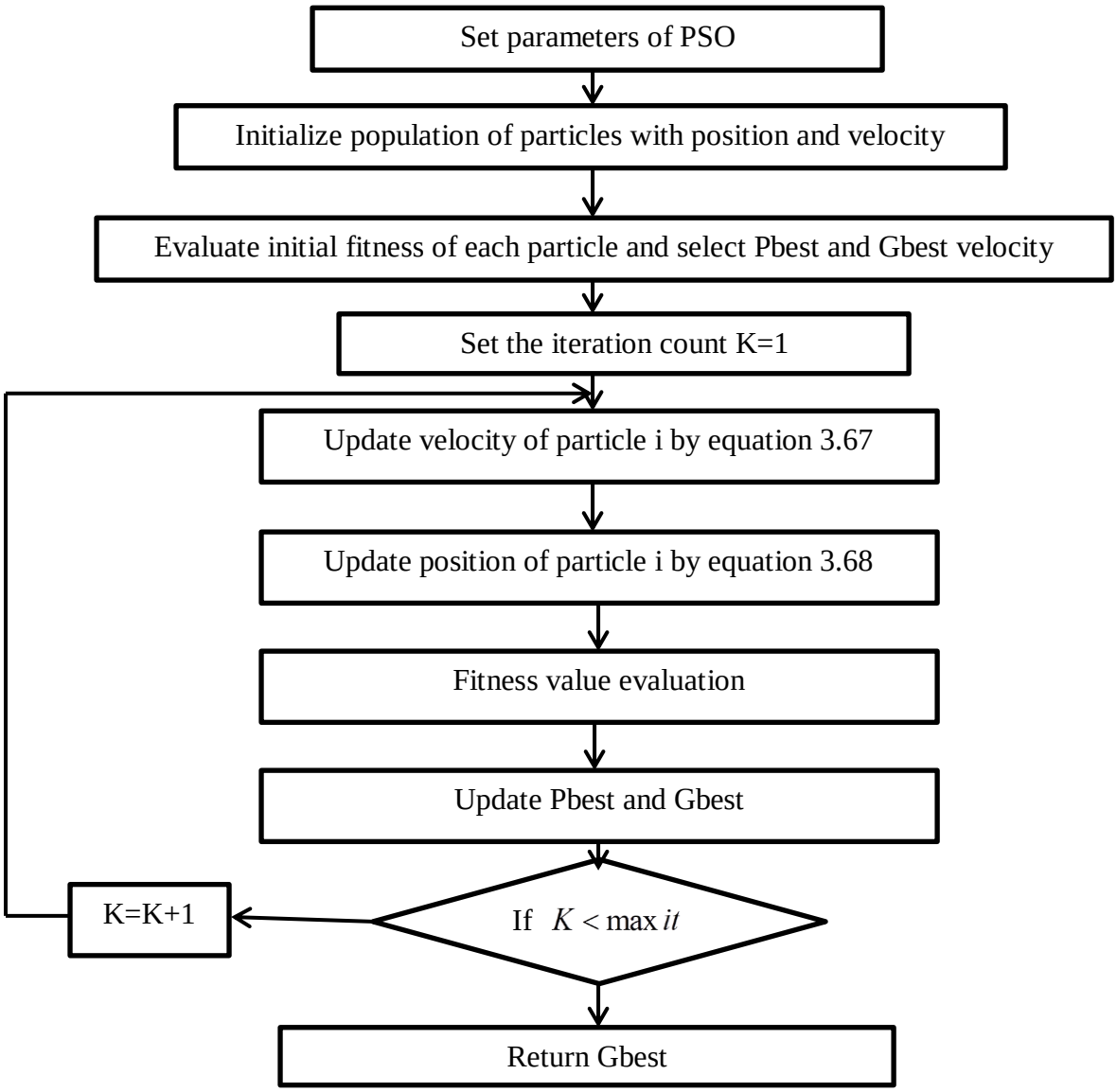


Figure3.6. PSO flow chart

3.5.2. Parameter Selection of PSO Algorithm and Fitness Function

The most common parameters used are [52]:

- Inertial weight: 0.9 to 0.4
- Acceleration factors ($c1 and\ c2$): 1.5 to 2.5
- Population size: 10 to 100
- Maximum iteration (Maxite): 50 to 10000
- Initial velocity: 10 % of position

3.5.2.1. Fitness Function

The convergence of the optimization algorithms towards the global optimal solution is characterized by the fitness function. Hence, a poorly selected fitness function may give a completely wrong result. The performance index given in (3.69) is selected here to minimize the absolute error between desired and actual leg length with time.

$$F_{cost} = w\sum_{i=1}^{6}\int_{0}^{t}\left|e_i(t)\right|dt \tag{3.69}$$

Where w_1 is the weighting factor and $e_i(t)$ is the tracking error for leg i, given as

$$e_i(t) = q_{di}(t) - q_{ai}(t) \tag{3.70}$$

Where $q_{ai}(t)$ is the actual length and $q_{di}(t)$ is the desired length of leg i. In such a way, my aim is to find the optimum value of the controller gains that minimizes the coupling error between the legs.

And the parameters used in the PSO algorithm for tuning of PD+ SISM controller gains are given in table 3.1.

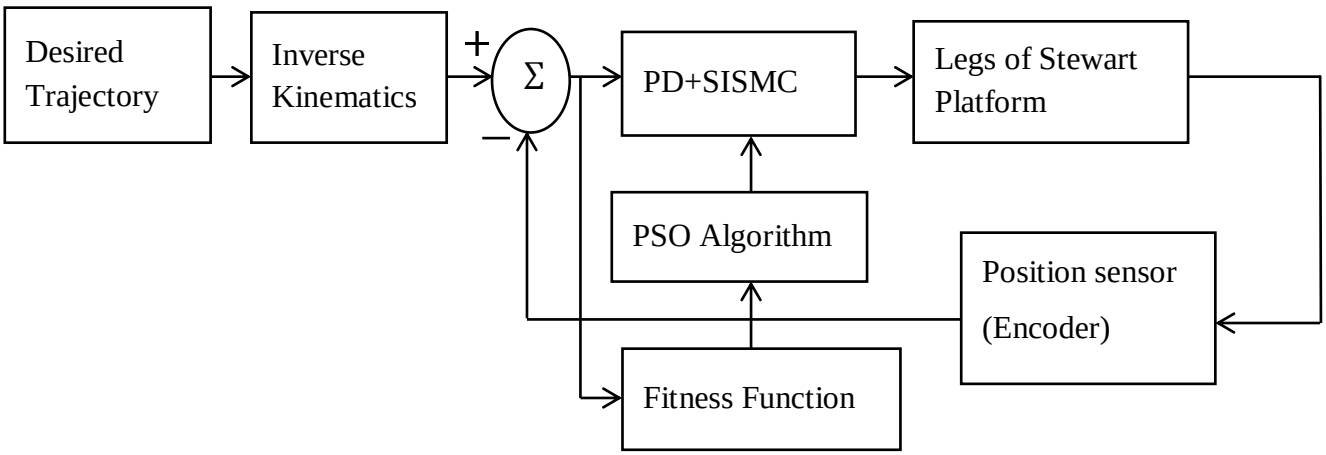


Figure3.7. The block diagram of PSO based PD+SISMC of Stewart platform in joint space

Table3.1. Selected design parameters for PSO algorithm

No.	Design Parameter	Value
1	Max number of iterations	50
2	population(swarm size)	20
3	Inertia coefficient(w)	1
4	damping coefficient	0.9
5	Personal acceleration coefficient(c1)	1.5
6	Social acceleration coefficient(c2)	1.5

With these parameters, the optimal solutions obtained using the intelligent method particle swarm optimization is given in table 4.1 of chapter four.

CHAPTER FOUR

4. SIMULATION RESULT AND DISCUSSION

In this chapter, detailed simulation results are briefly discussed, and the performance of the passivity-based smooth integral sliding mode controller has compared with the passivity-based integral sliding mode controller.

4.1. Multi-Body Modeling of Stewart Platform

Stewart platform manipulator is a parallel kinematics manipulator; possesses highly nonlinear and coupled dynamics. So that to design a controller for the Stewart platform, needs manipulating complicated equations that modeled the physical components used to solve the mechanical equations and then solve these equations using complex numerical integration techniques. With the advent of computational tools such as SimMechanics, in conjunction with Simulink and MATLAB, it is possible to easily model and simulate the robot mechanics together with the control system. SimMechanics is a block diagram modeling environment, which uses the standard Newtonian dynamics of forces and torques for the design and simulation of rigid body mechanisms and their motions, and supports Solidworks in order to specify mechanical models.

4.1.1. Rotary Stewart Platform CAD Modeling

Computer-aided design (CAD) tools allow us to model machines geometrically as collections of parts or assemblies. The block diagram approach does not include full geometric information, nor do CAD assemblies typically incorporate controllers or allow us to perform dynamic simulations. With CAD translation, we can combine the power of CAD and SimMechanics. The translator transforms geometric CAD assemblies into Simulink block diagram models. The intermediary between a CAD assembly and its SimMechanics model is an XML file in a Physical Modeling format. With SolidWorks 2017 program, a typical rotary Stewart platform given in figure 4.1 has been modeled, with a geometric parameter given in table 4.1.

Table4.1. Rotary Stewart platform 3D design parameters

Parameter	Value	Description
Rb	60mm	Base plate radius
Rp	50mm	Top plate radius

S	149mm	Length of the i^{th} rigid leg.
Rm	40mm	Length of servo arm

Mass and inertia of the manipulator parts depends on the material type, so for more flexibility the materials are chosen to have little weight. Each part of the rotary Stewart platform has been modeled with the appropriate dimension and material type separately, then the parts are assembled into a sub assembly and assembly by applying mate or constraint properties between two parts, which later becomes a joint in SimMechanics model which determines the of motion between two rigid parts and define the number of degree of freedom (DOF) s between parts. The actuators are connected to theses joints to enable the platform to move.

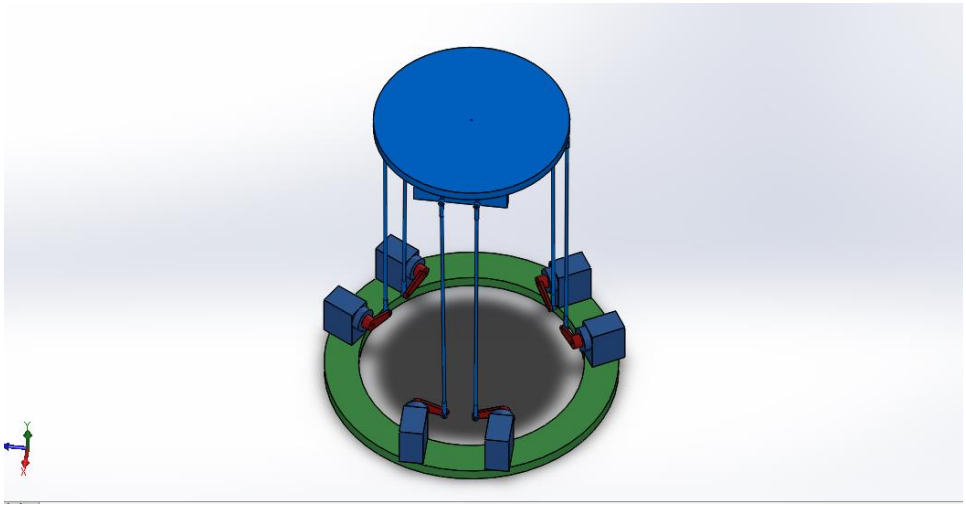


Figure4. 1. 3D model of rotary Stewart platform in Solid Works

4.1.2. Dynamic Model of Stewart Platform in SimMechanics

In this thesis, for the simulation study of the performance of the controller, a typical 6-6 geometry Stewart platform with the geometric parameters given in table2.1 and 2.3 of chapter two is implemented using the SimMechanics toolbox of MATLAB. Figure 4.2 shows the complete Simulink model of passivity-based smooth integral sliding mode control of the Stewart platform manipulator in joint space. It consists of the Stewart platform block, sliding surface block, controller design block, and desired position trajectory block. The Simulink model of the controller subsystem is modeled using the mathematical expression given in chapter three. And the Simulink model of the desired leg length trajectory subsystem is modeled by the mathematical equation presented in chapter two, and the details of each subsystem block model are given in appendix A.

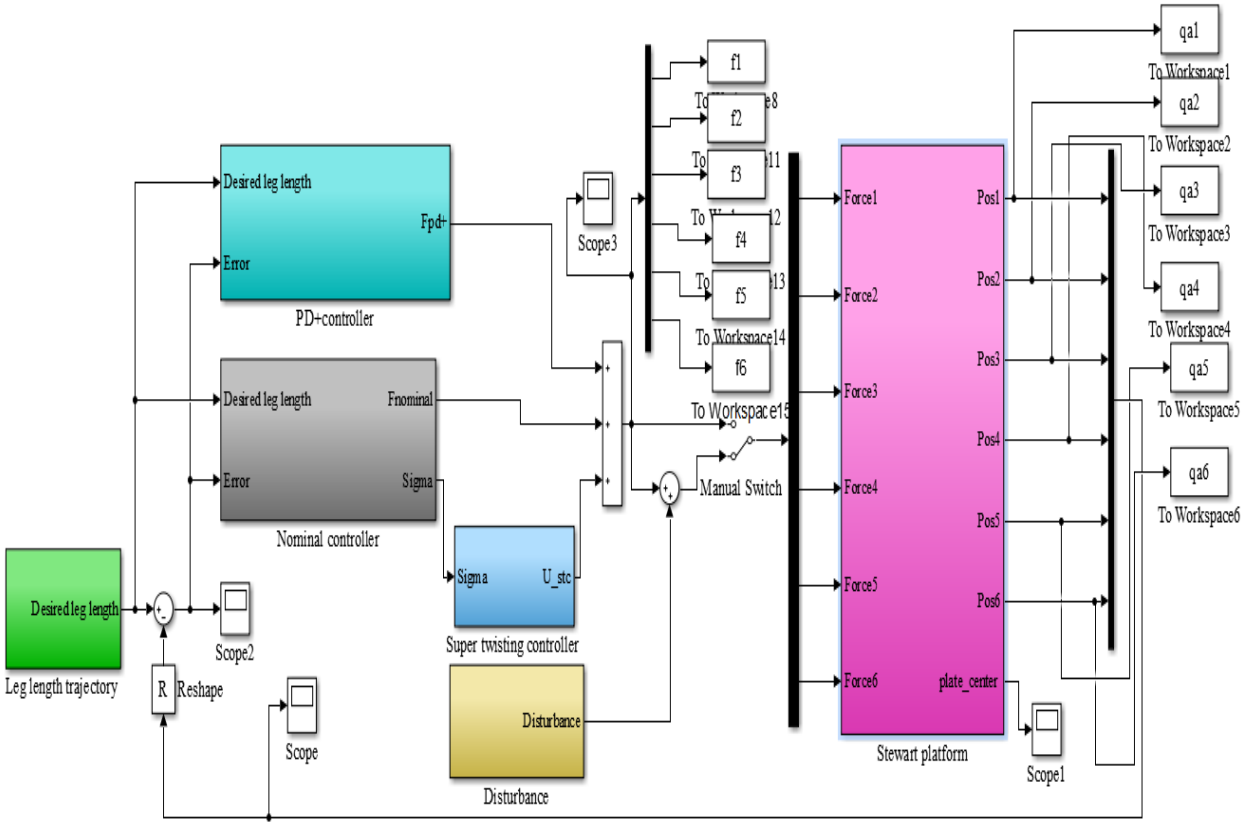


Figure4. 2.The complete Simulink model of passivity based smooth integral sliding mode control of the Stewart platform in joint space.

4.2. Reference Trajectory Generation

The reference trajectory for the mobile platform is generated by the desired leg length trajectory sub system block. In this model block, translational and rotational desired position signals are a fast trajectory having heave motion, circular motion in X-Y plane and angle twists as given by[32] below.

$$mx = 0.5(1-\exp^{-\pi t})\cos(1.88\pi t), m$$

$$my = 0.5(1-\exp^{-\pi t})\sin(1.88\pi t), m$$

$$mz = 3+\frac{0.02}{1+0.9t}\sin(\frac{2\pi t(0.1+5.9t)}{10.5})+\frac{\pi}{24}, m$$

$$\alpha = 0, \deg$$

$$\beta = 0.5(1-\exp^{-\pi t})\sin(0.86\pi t), rad$$

$$\gamma = 0.3(1-\exp^{-\pi t})\sin(0.74\pi t), \deg \qquad (4.1)$$

As can be seen from equation (4.1) the reference trajectory uses constant, exponential function and sinusoidal function. It is possible to implement another reference trajectory by

replacing these references with other functions or blocks. Moreover, it is possible to use space mouse to generate reference trajectory. In other word it is possible to make the platform track user controlled trajectory in real time.

4.3. Controller Design

The controller block takes the difference between the desired and actual leg lengths as controller inputs, and generates a response to this leg length with the force output to actuate the prismatic joints between the lower and upper legs. Thus, it accepts the leg trajectory and imposes force on the platform by actuating the prismatic joint. One of the important issues on the Stewart platform controller design is the selection of control law and the optimal choice of controller gain values. The control law is proportional derivative plus and smooth integral sliding mode control law given in (3.66) and rewritten as

$$\tau = B + H(q,\dot{q})\ddot{q}_d + A(q)\dddot{q}_d + Kpe + Kd\dot{e} + A(q)(\dddot{q}_d + Kpe + Kd\dot{e} + \ldots$$

$$Ki\int edt) + H(q,\dot{q})\dot{q} + B(q) + K_1|s|^{\frac{1}{2}}sign(s) + \int K_2 sign(s)d\tau \qquad (4.2)$$

Hence, to choose the elements of K_1, K2, Kp, Kd, and C_1 matrices values optimally, PSO algorithm used and applied for Stewart platform. Figure4.3. shows PSO tuned passivity based smooth integral sliding mode control system.

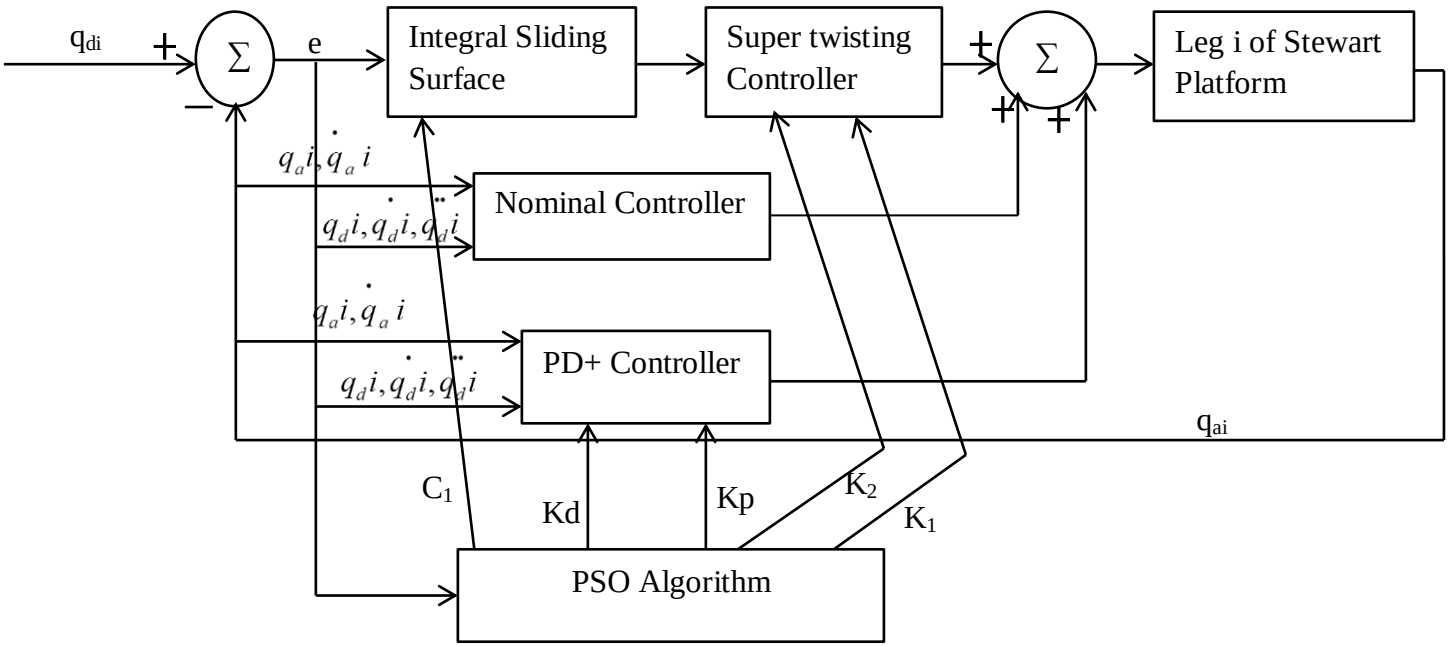


Figure4. 3. PSO tuned joint space PB-SISMC of single leg.

The input to the block is not a single error, but it is a weighted sum of the errors from other legs. The PSO algorithm tunes the parameters of the PB-SISMC based on the fitness function

expressed in chapter three, and the particle swarm optimization algorithm is implemented by writing a code in MATLAB, which was a very hard task. After applying the particle swarm optimization I get the values of the controller gain matrices as shown in table 4.2.

Table4.2. Optimal parameters of PSO tuned PB-SISM controller

Controllers gain matrices	Values
PD+ controller gains(K_p, K_d)	$Kp = 10^3 \begin{bmatrix} 5.6984 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5.9788 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.0404 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4.9645 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4.5675 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5.5535 \end{bmatrix}$
	$Kd = 10^3 \begin{bmatrix} 2.5124 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6.9510 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.4536 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5.0050 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.0591 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4.4609 \end{bmatrix}$
Super twisting control gain matrices (K_1,K_2)	$K_1 = 10^3 \begin{bmatrix} 5.0349 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5.0349 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5.0349 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5.0349 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5.0349 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5.0349 \end{bmatrix}$
	$K_2 = 10^3 \begin{bmatrix} 0.5498 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5498 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5498 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5498 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5498 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5498 \end{bmatrix}$

Integral sliding surface gain matrix (C_1)

$$C_1 = 10^3 \begin{bmatrix} 0.1888 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2885 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.3418 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.3672 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1797 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.4352 \end{bmatrix}$$

Switching function gain matrix (K_3)

$$K_3 = 10^5 \begin{bmatrix} 4.7813 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4.9262 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5.2667 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5.8491 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7.5941 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7.7119 \end{bmatrix}$$

And the gains of the nominal controller used for the simulation are

$$Kp = diag\begin{bmatrix} 9.9492 & 9.9492 & 9.9492 & 9.9492 & 9.9492 & 9.9492 \end{bmatrix} 10^6$$

$$Ki = diag\begin{bmatrix} 0.8687 & 0.8687 & 0.8687 & 0.8687 & 0.8687 & 0.8687 \end{bmatrix} 10^3$$

$$Kd = diag\begin{bmatrix} 8.9161 & 8.9161 & 8.9161 & 8.9161 & 8.9161 & 8.9161 \end{bmatrix} 10^4$$

Figure 4.4 shows how the fitness function is minimized as the iteration increases in the PSO algorithm and the simulation results are given in the next sections.

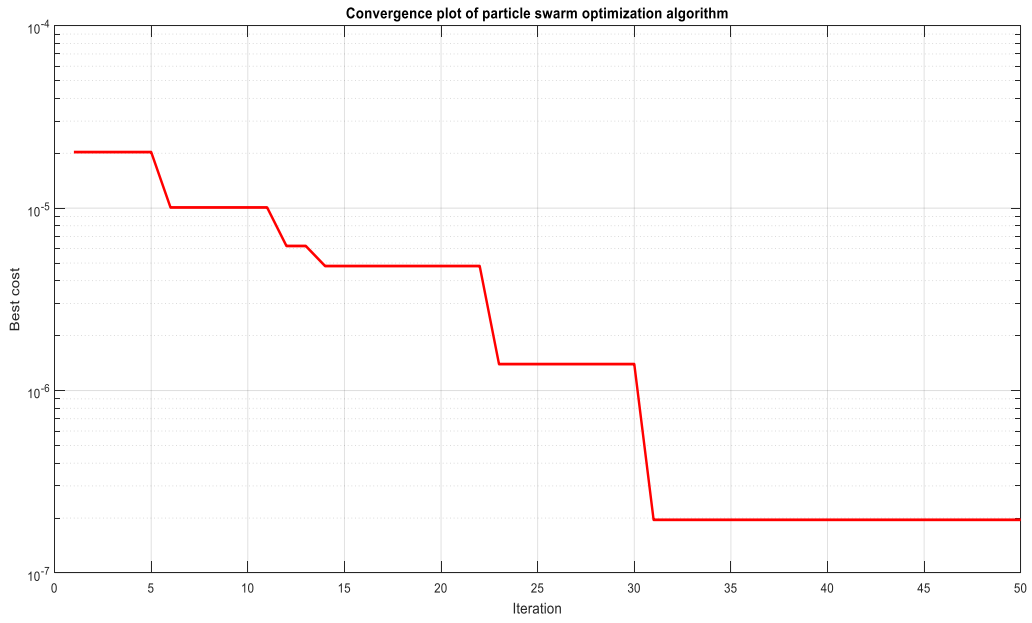


Figure4. 4. Convergence plot of PSO using Semilogy

4.4. Tracking Performance of the Controllers without Load and Disturbance

The reference trajectory given in (4.1) is taken as a reference signal to demonstrate the performance of trajectory tracking capability of the controllers, and different cases have been considered to test the performance of the designed PB-SISMC and PB-ISMC.

Case i. Figure 4.5-4.10 shows the integral sliding surface for each leg achieved with both PB-SISMC and PB-ISMC. From the comparison of the two plots, the sliding surface for each leg enforced by PB-ISMC stays almost at zero with a sliding accuracy of 0.0005, 0.0019, 0.0022, 0.0004, 0.0008, and 0.0025 meters. However, the controller exhibits too much chattering in the control signal, as shown in figure 4.11-4.16. These results show that the proposed PB-SISMC is much better in chattering elimination.

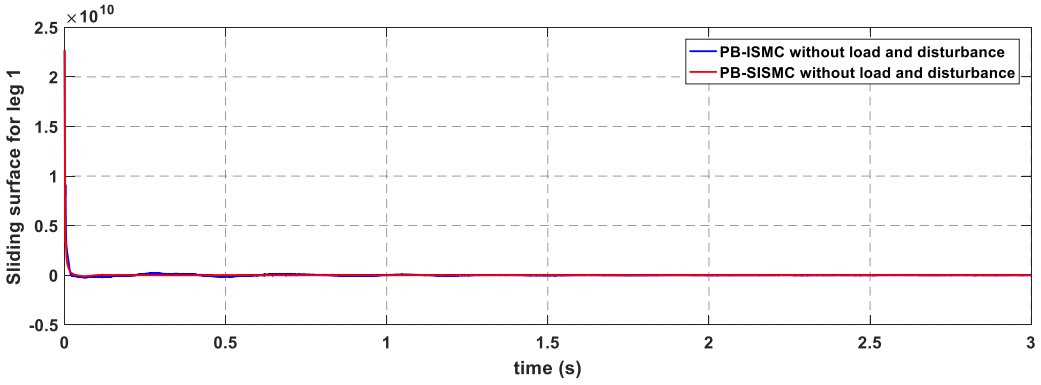


Figure4.5. Sliding surface for leg 1, without load and disturbance

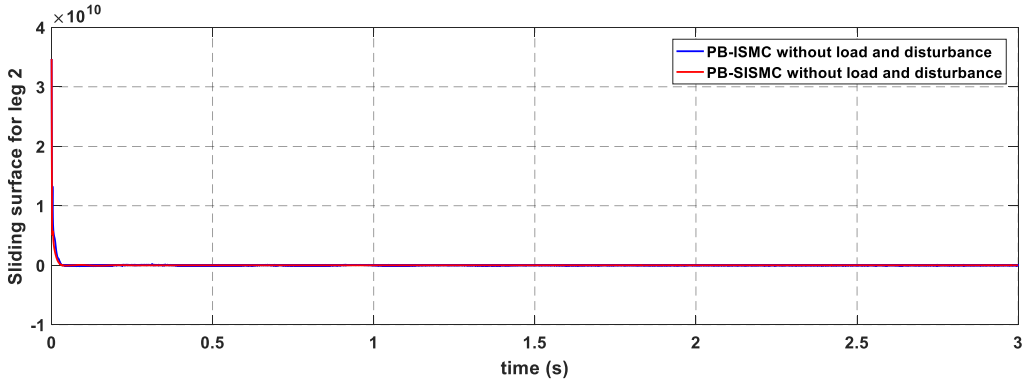


Figure4.6. Sliding surface for leg 2, without load and disturbance

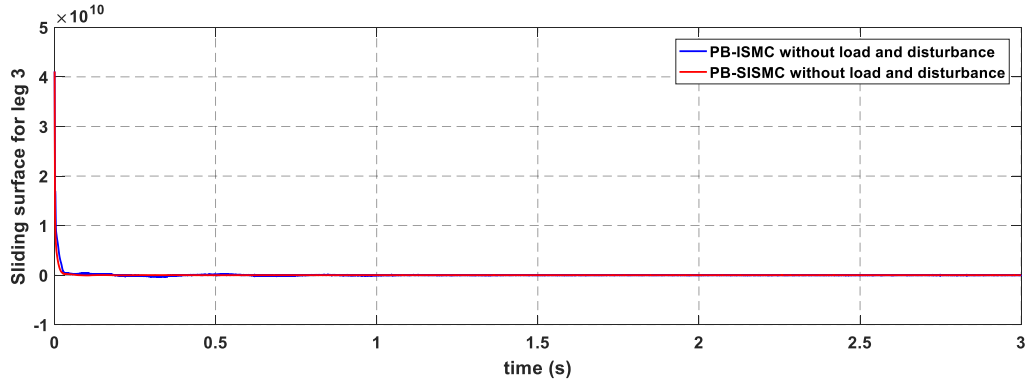


Figure4.7. Sliding surface for leg 3, without load and disturbance

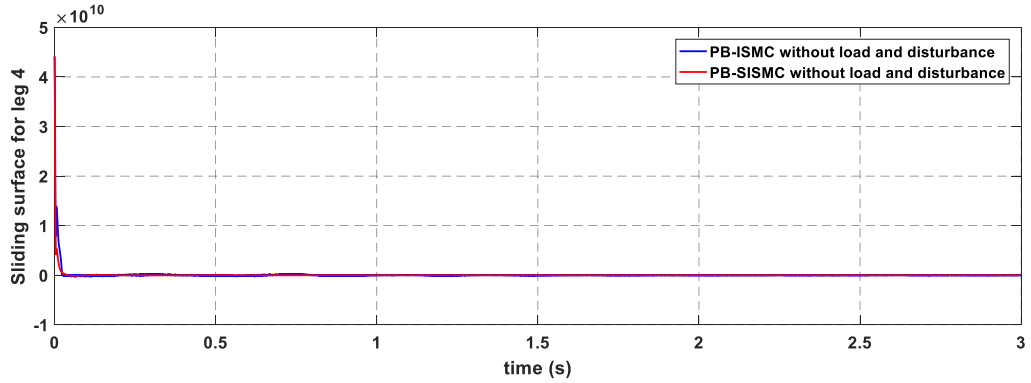


Figure4.8. Sliding surface for leg 4, without load and disturbance

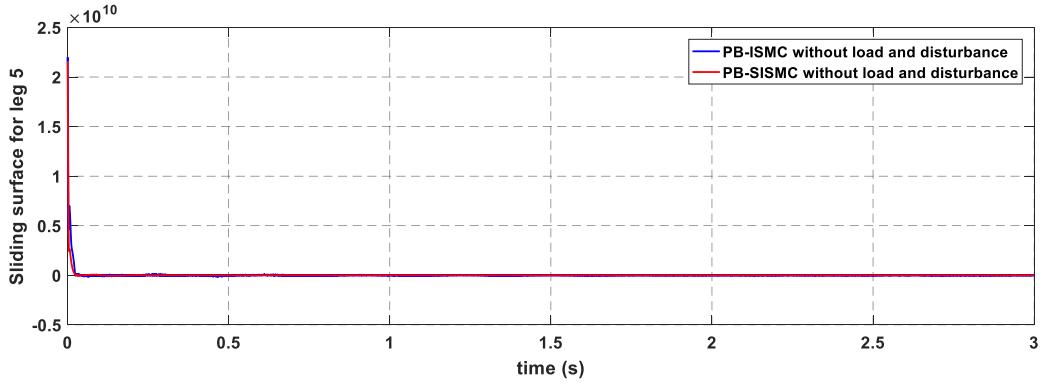


Figure4.9. Sliding surface for leg 5, without load and disturbance

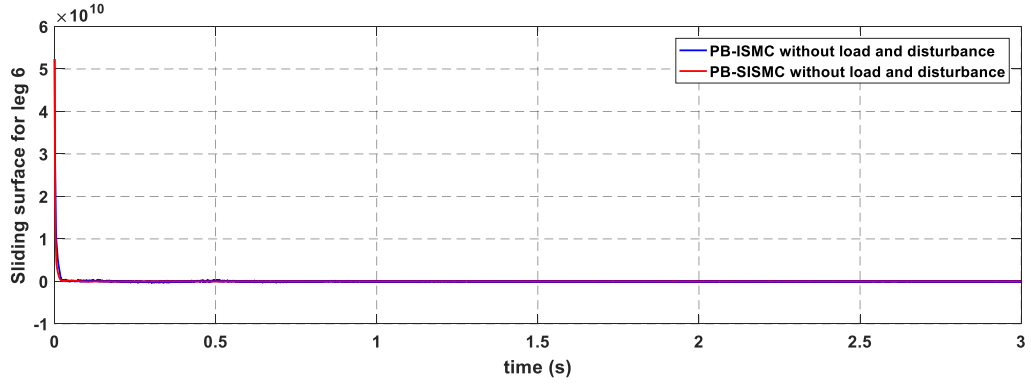


Figure4.10. Sliding surface for leg 6, without load and disturbance

Case ii. The required control force with both PB-ISMC and PB-SISMC for achieving the desired trajectory of each leg is shown in figure 4.11-4.16. From the comparison of the two plots, PB-ISMC exhibits too much chattering in the control signal of each leg and a fixed magnitude proportional to the switching gain K_3, reducing K_3 decreases robustness, and increasing K_3 causes the magnitude of the control signal raise. On the other hand, PB-SISMC performs better in chattering elimination as shown in figure 4.17, which is used for comparison.

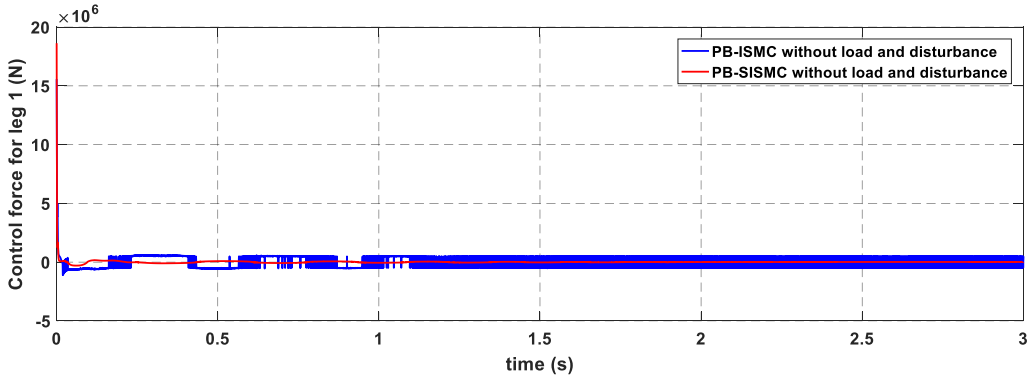


Figure4.11. Control force for leg 1, without load and disturbance

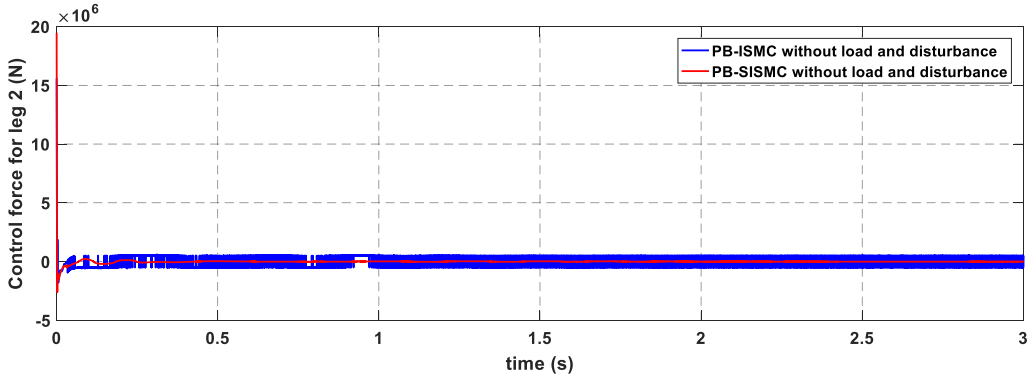


Figure4.12. Control force for leg 2, without load and disturbance

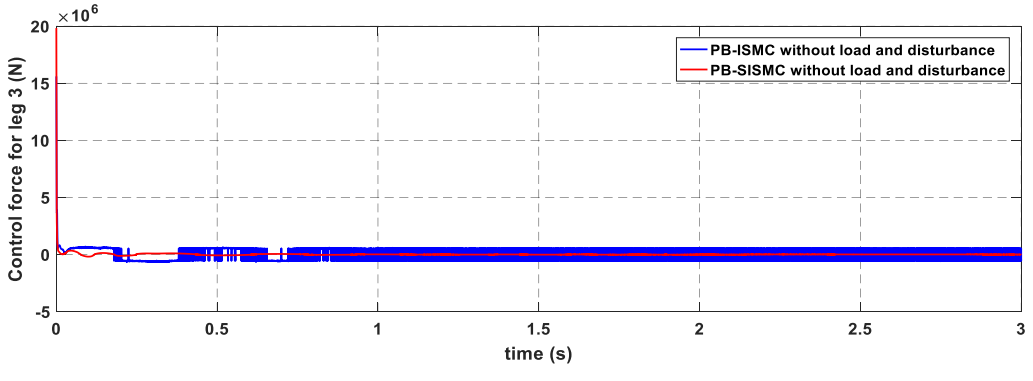


Figure4.13. Control force for leg 3, without load and disturbance

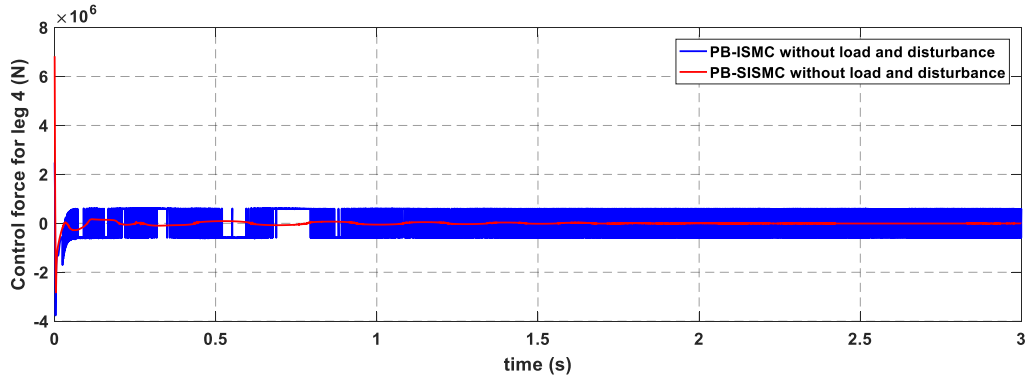


Figure4.14. Control force for leg 4, without load and disturbance

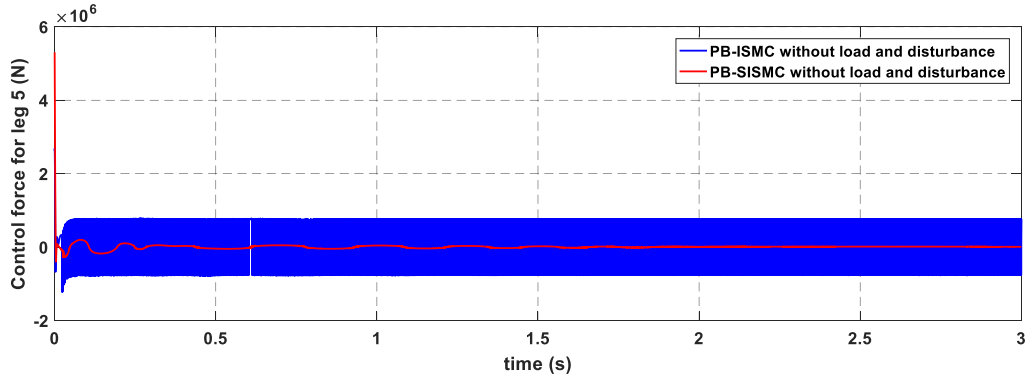


Figure4. 15. Control force for leg 5, without load and disturbance

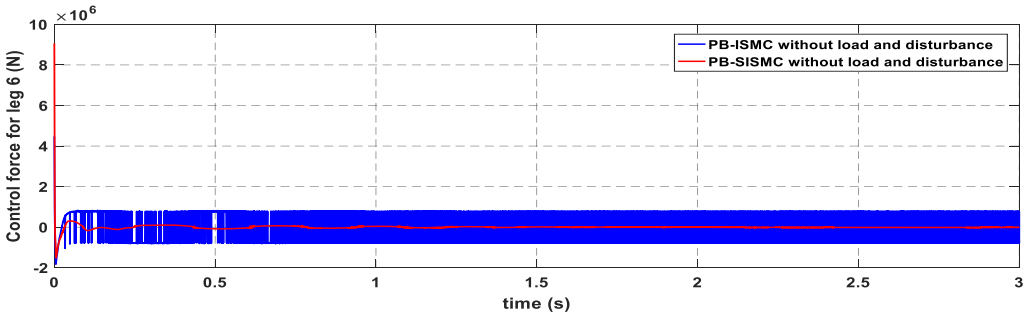


Figure4. 16. Control force for leg 6, without load and disturbance

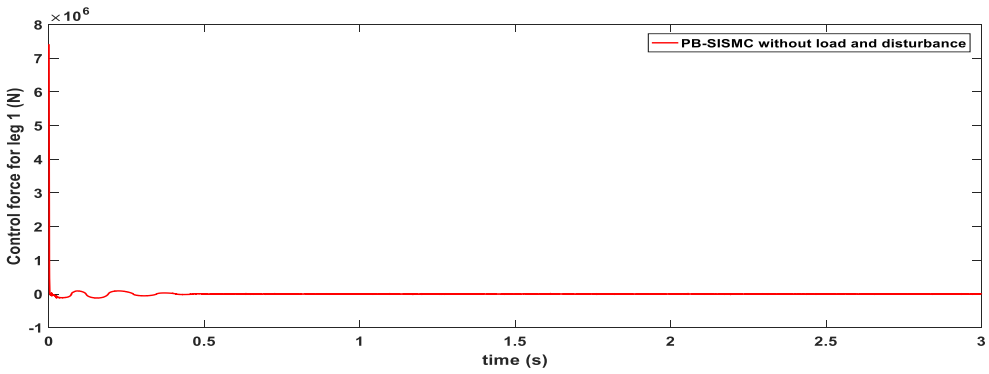
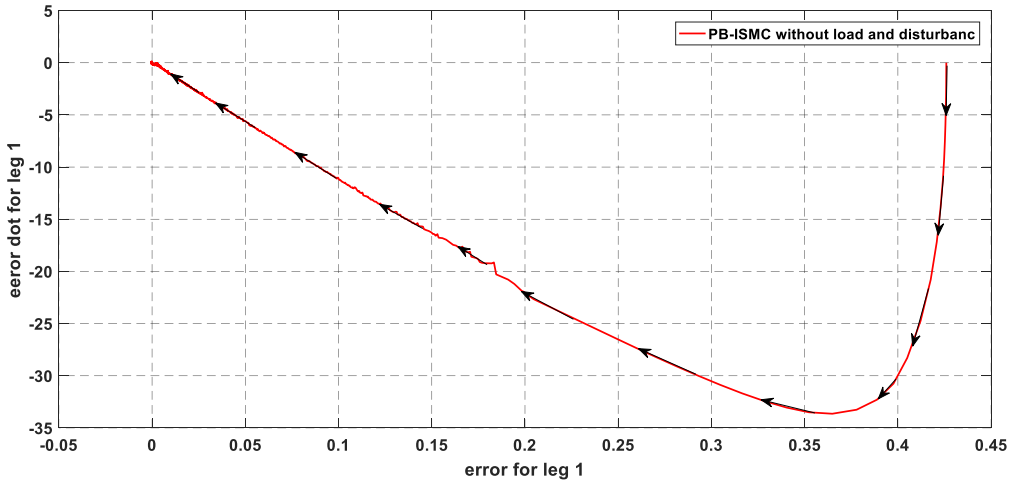
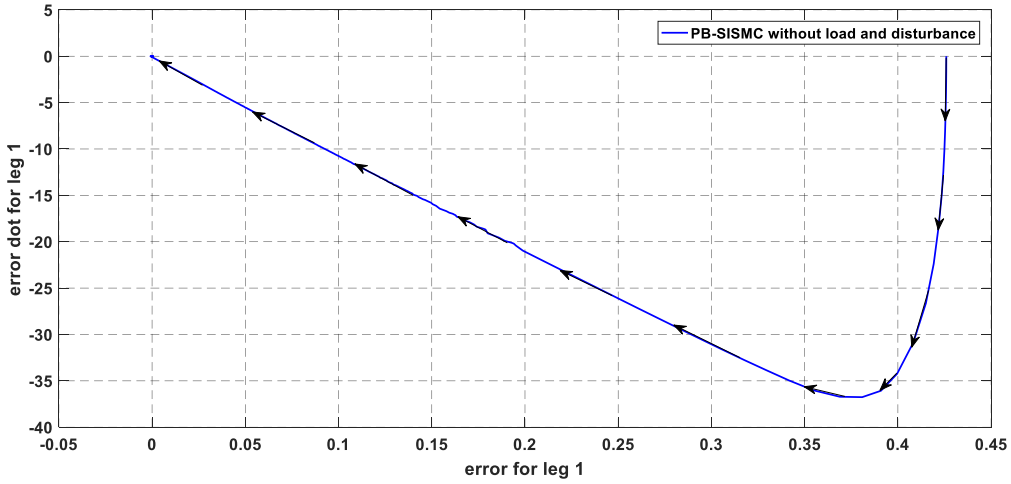


Figure4. 17.Control force for leg 1 with PB-SISMC without load and disturbance

Case iii. Phase portrait of the error and its first time derivative in phase plane for leg 1 with both PB-ISMC and PB-SISMC is shown in figure 4.18. From the comparison of the two subplots, PB-ISMC exhibits chattering in the control signal and a closer observation of the darker region around the origin of the phase plane shows that the sliding dynamics of the system governed by PB-ISMC is stable. And the system trajectory will continue in persisting motion of zigzag path around the origin for an indefinite time. On the other hand, PB-SISMC is superior in terms of chattering elimination.



(a) Phase portrait of error and its first time derivative by PB-ISMC without load and disturbance



(b) Phase portrait of error and its first time derivative by PB-SISMC without load and disturbance

Figure4.18. Comparison of the phase portrait of error and its first time derivative designed by (a) PB-ISMC (b) PB-SISMC

Case iv. The simulation result for leg 1 actual versus desired leg length achieved by PB-SISMC and PB-ISMC and the tracking error for each leg with both PB-ISMC and PB-SISMC are shown in figure 4.19, 4.20 and figure 4.21-4.26. From figure 4.20 and figure 4.21-4.26, it is seen that the actual length of leg 1 reached the desired length of leg 1 almost at 0 seconds. But there is a small steady-state error, such that the actual length does not precisely track the desired leg length as compared to the response and the tracking error achieved by PB-SISMC. As the results are summarized in table 4.3, the trajectory tracking the performance of PB-SISMC is much better than PB-ISMC. The table shows the trajectory tracking performance of the two controllers with no load and disturbance.

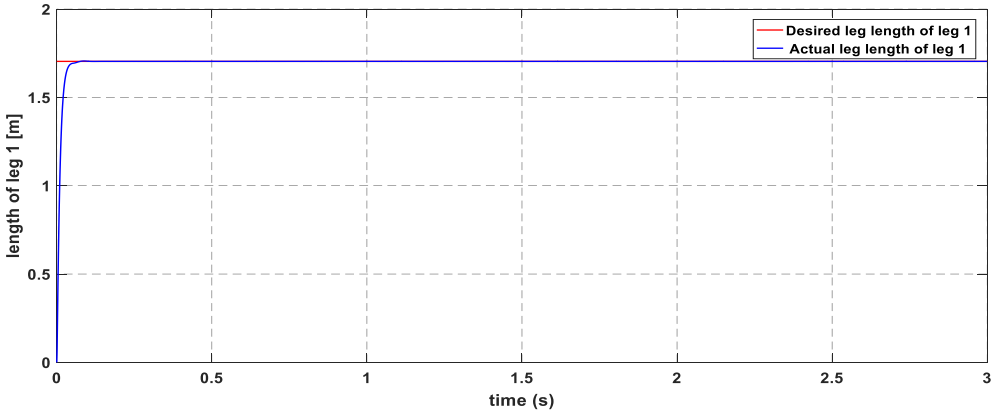


Figure4. 19. The response of the single leg (leg 1) of Stewart platform by PSO tuned PB-SISMC without load and disturbance

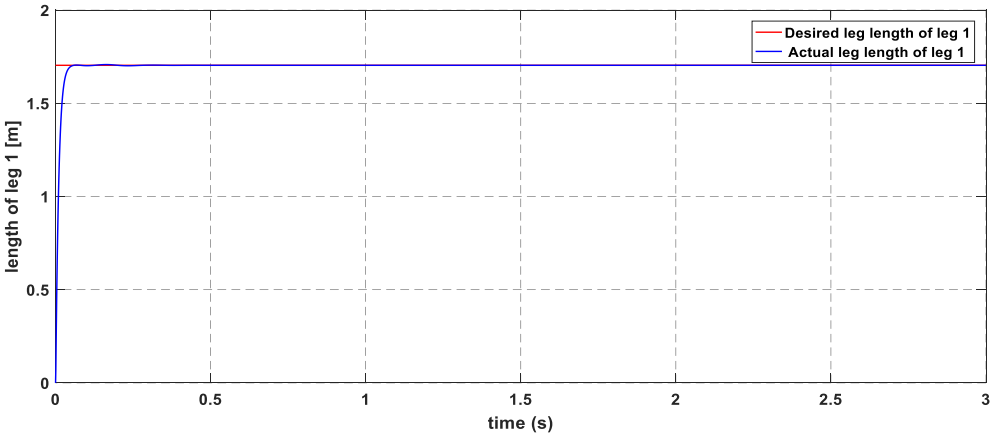


Figure4. 20. The response of the single leg (leg 1) of Stewart platform by PSO tuned PB-ISMC without load and disturbance

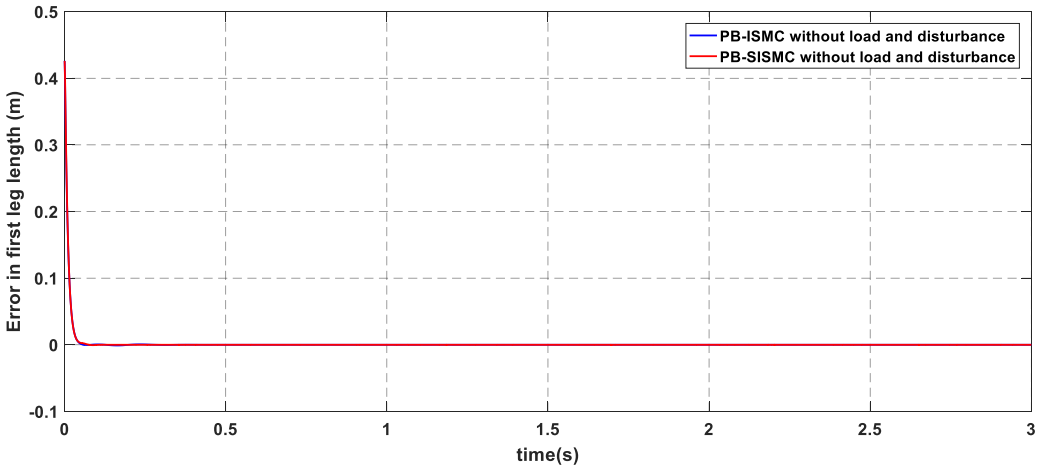


Figure4.21. Tracking error for leg 1, without load and disturbance

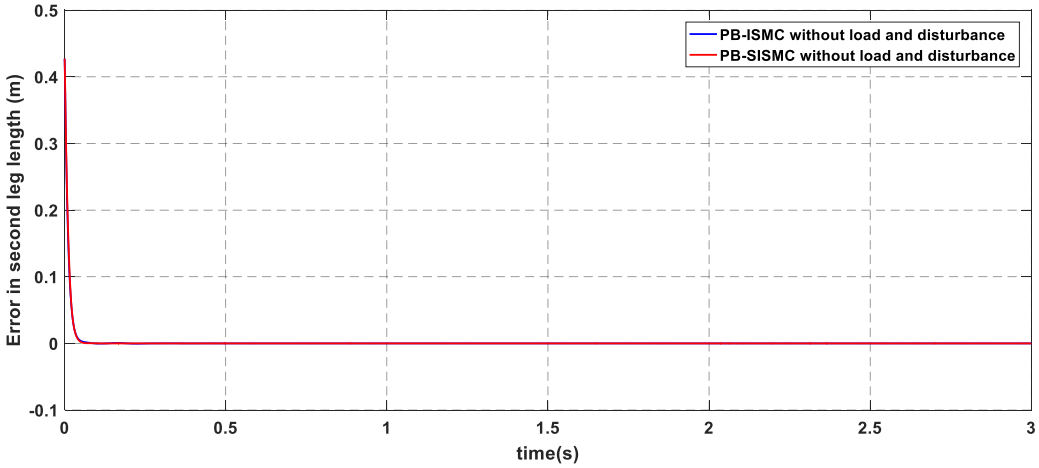


Figure4. 22. Tracking error for leg 2, without load and disturbance

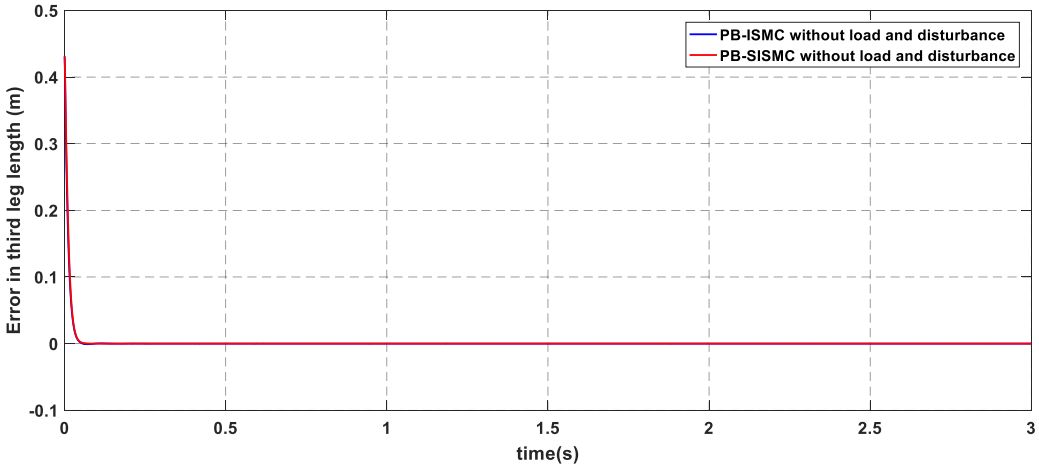


Figure4. 23. Tracking error for leg 3, without load and disturbance

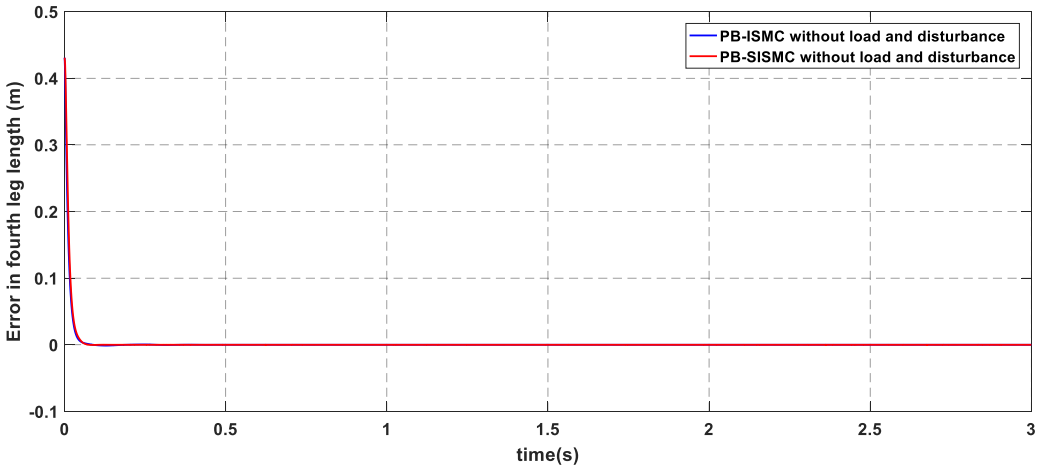


Figure4. 24. Tracking error for leg 4, without load and disturbance

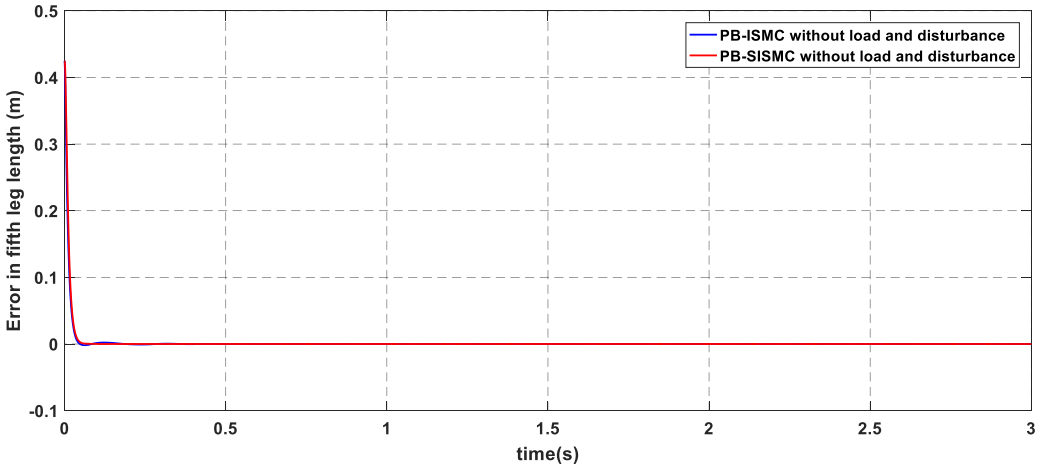


Figure4. 25. Tracking error for leg 5, without load and disturbance

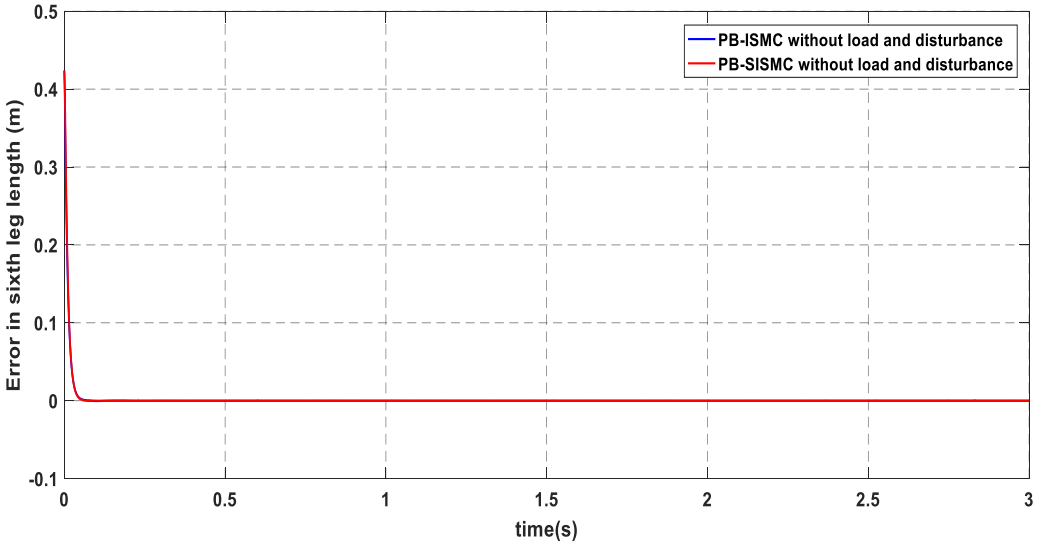


Figure4. 26. Tracking error for leg 6, without load and disturbance

Table4.3. Absolute error for leg i without load and disturbance

Controller	Absolut tracking error for leg i in meter, without load and disturbance					
i	1	2	3	4	5	6
PB-ISMC	0.0001256	0.0001310	0.0001921	0.0003027	0.0001808	0.0001624
PB-SISMC	0.0000143	0.0001173	0.0001116	0.0001255	0.0001511	0.0001161
Difference	0.0001113	0.0000137	0.0000805	0.0001772	0.0000297	0.0000462

4.5. Tracking Performance of the Controllers with Load

The most important characteristic of the controller is its robustness against parameter variation and viscous and coulomb frictions and external disturbance. Robustness against parameter uncertainty is studied by loading the platform with the mass of 200kg.

Case i. The plot of the integral sliding surface of each leg, with load achieved by PB-SISMC and PB-ISMC, is shown in figure 4.27-4.32. From the comparison of the two plots, the sliding surface of leg four, five, and six enforced by PB-ISMC relatively stays almost at 0 for the whole simulation time. However, the controller exhibits too much chattering in the control signal as shown in figure 4.33-4.38. On the other hand, the sliding surface for each leg achieved by PB-SISMC stays at zero for the whole simulation time with sliding accuracy of 0.0001, 0.0001, 0.0007, 0.0004, 0.0001, and 0.0001 meters.

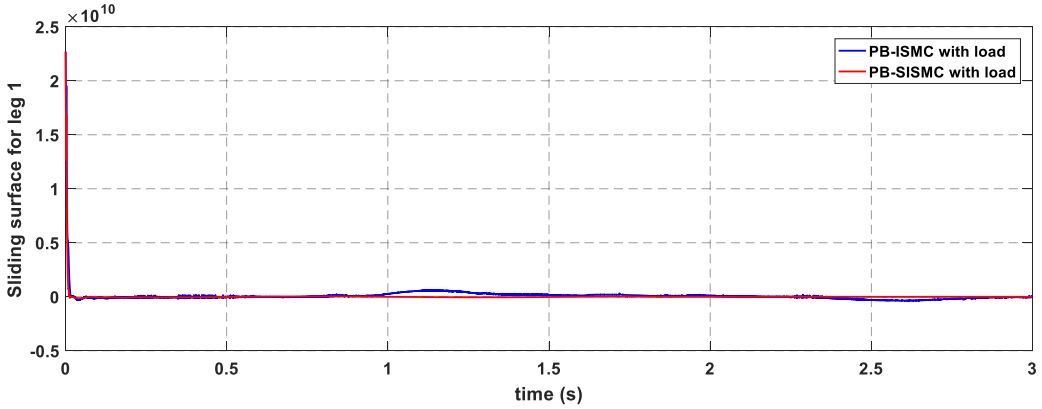


Figure4.27. Sliding surface for leg 1, with load

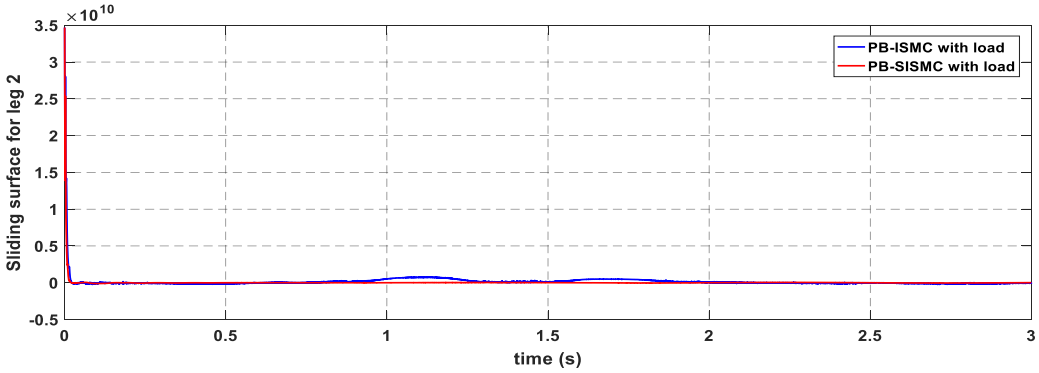


Figure4.28. Sliding surface for leg 2, with load

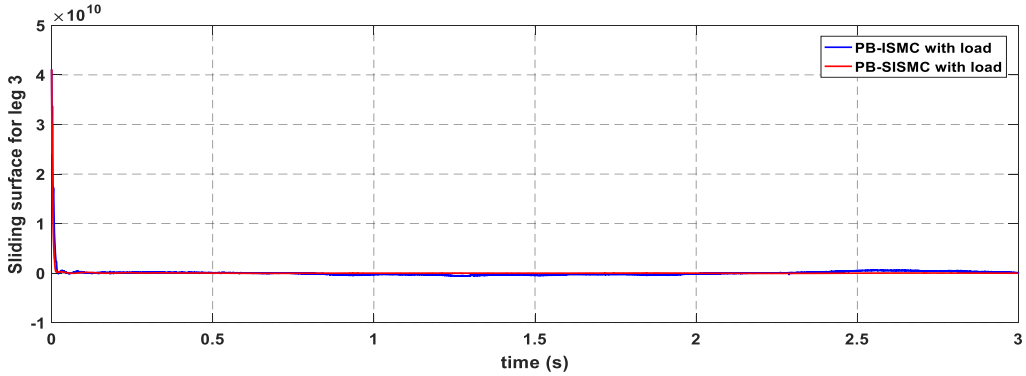


Figure4.29. Sliding surface for leg 3, with load

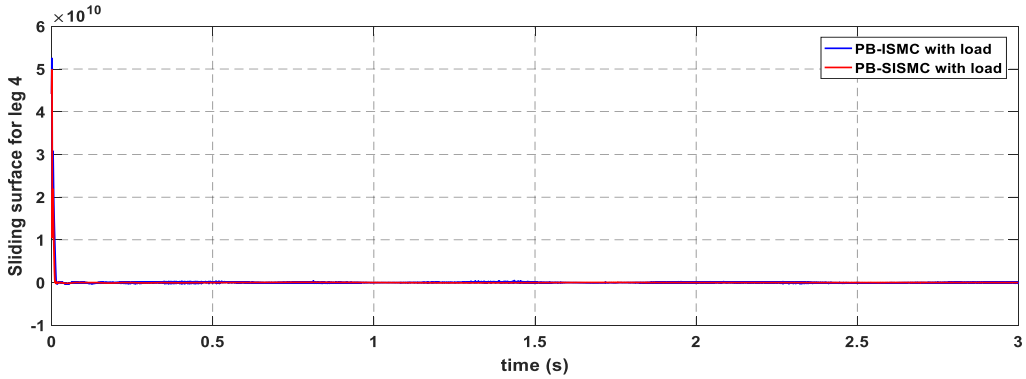


Figure4.30. Sliding surface for leg 4, with load

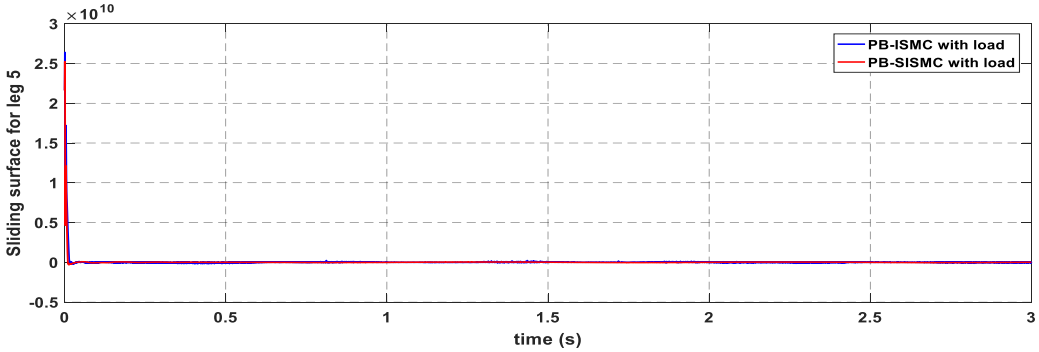


Figure4.31. Sliding surface for leg 5, with load

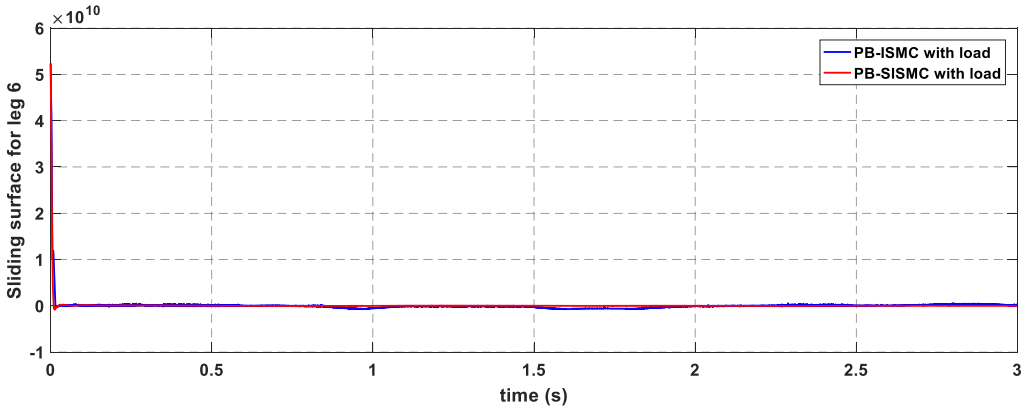


Figure4.32. Sliding surface for leg 6, with load

Case ii. Figure 4.33-4.38 shows the plot of the required control force of each leg, with a load for achieving the desired trajectory with both PB-SISMC and PB-ISMC. From the comparison of the two plots, it is clear that PB-ISMC exhibits too much chattering, this is due to the discontinuous control part of the ISMC, and the magnitude of oscillation is proportional to the switching gain K3. On the other hand, the control input with PB-SISMC is continuous, and hence the dangerous chattering is avoided as shown in figure 4.39.

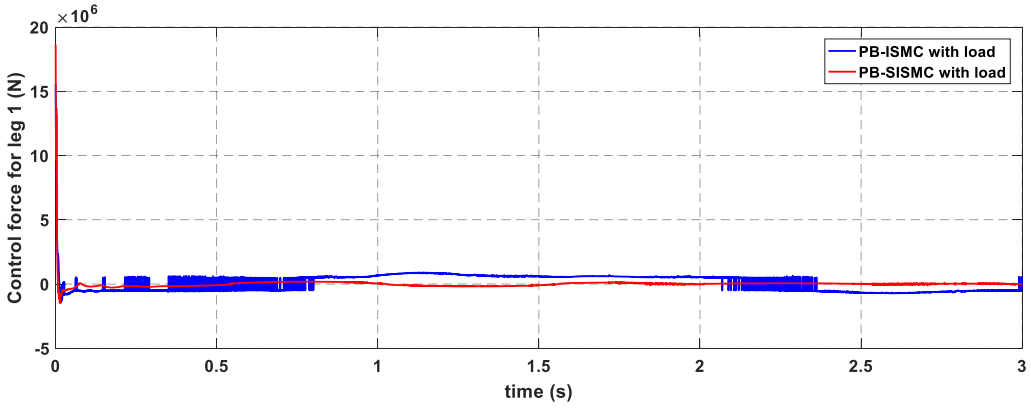


Figure4.33. Control force for leg 1, with load

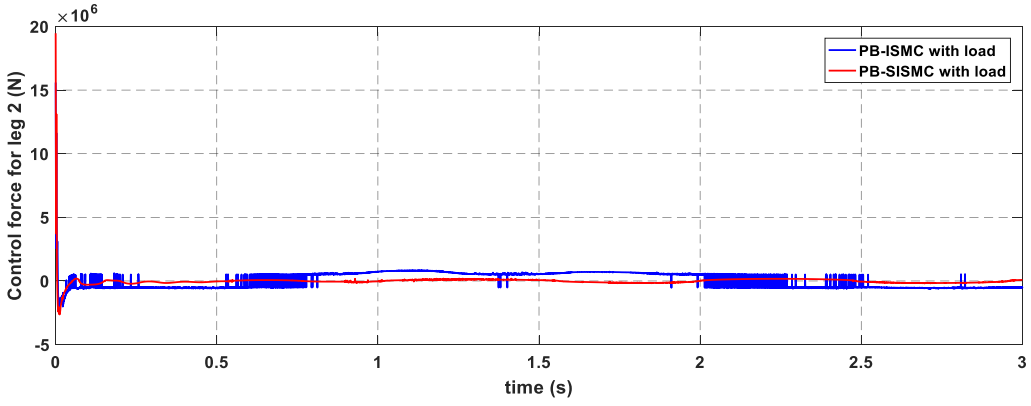


Figure4.34. Control force for leg 2, with load

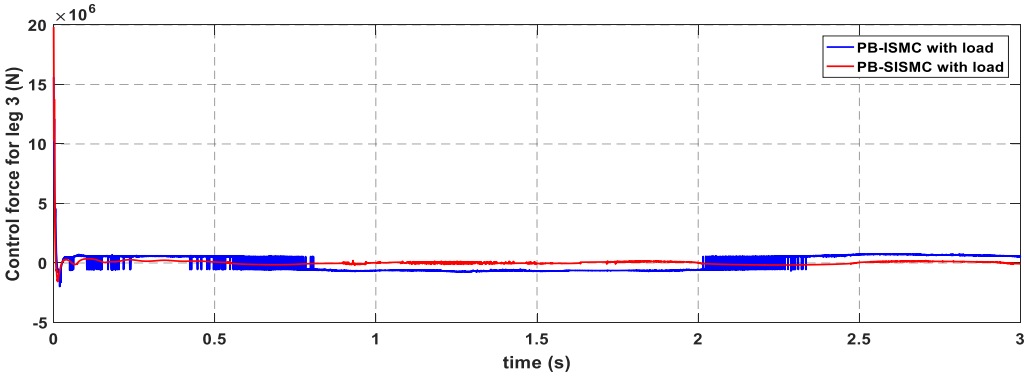


Figure4.35. Control force for leg 3, with load

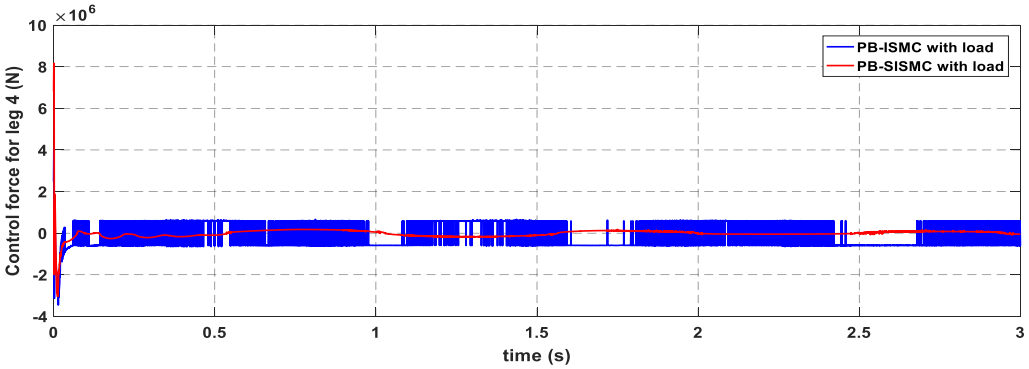


Figure4.36. Control force for leg 4, with load

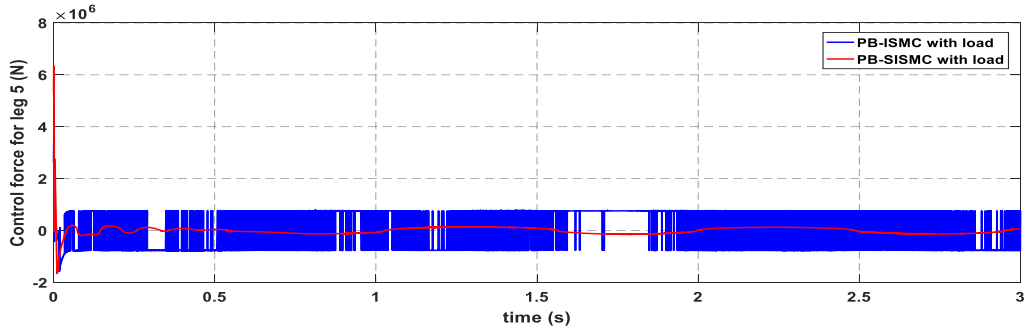


Figure4.37. Control force for leg 5, with load

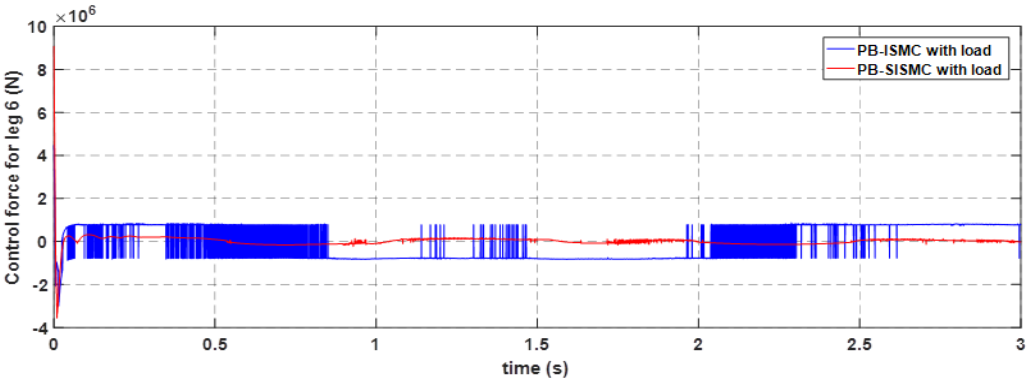


Figure4.38. Control force for leg 6, with load

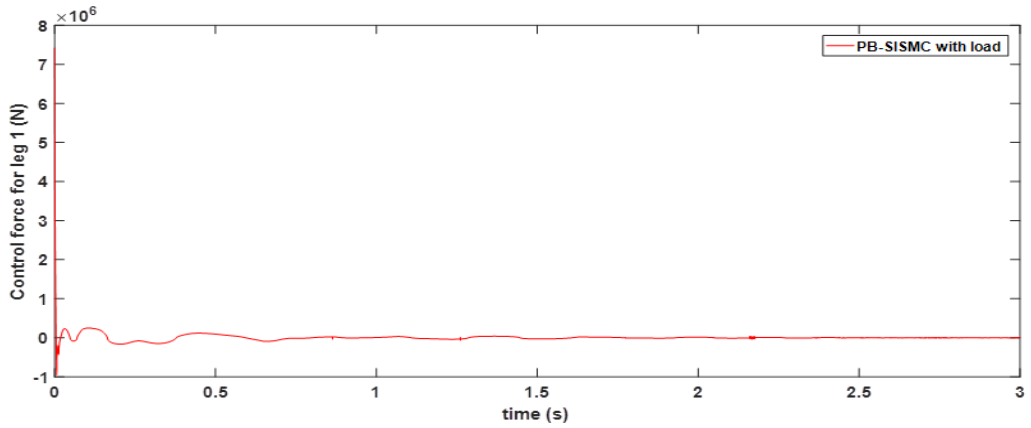
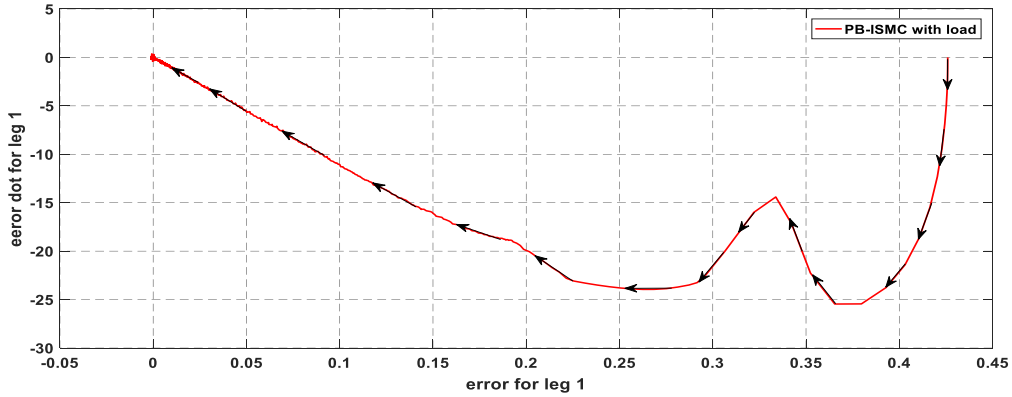
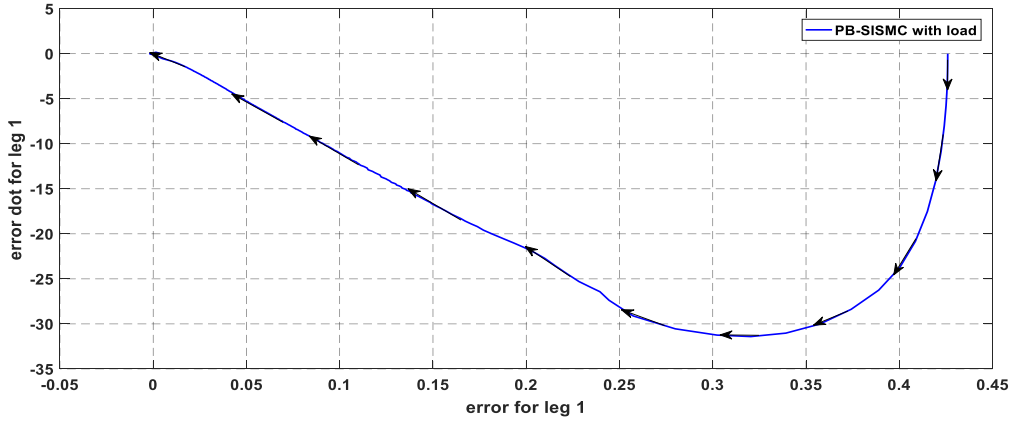


Figure4. 39. Control force for leg 1 with PB-SISMC with load

Case iii. Phase portrait of the error and its first time derivative for leg 1, with load achieved with both PB-ISMC and PB-SISMC is shown in figure 4.40. From the comparison of the two subplots, the PB-ISMC exhibits chattering in the control signal and closer observation of the darker region around the origin of the phase plane and finally reach the origin, this shows that the sliding dynamics of the system governed by PB-ISMC is stable and the system trajectory will continue in a persisting motion of zigzag path around the origin for an indefinite time.



(a) Phase portrait of error and its first derivative by PB-ISMC with load



(b) Phase portrait of error and its first derivative by PB-SISMC with load

Figure4.40. Comparison of the phase portrait of the error and its first derivative designed by (a) PB-ISMC (b) PB-SISMC

Case iv. Figure 4.41 and 4.42 show the simulation result of leg 1 actual versus desired leg length with PB-SISMC and PB-ISMC. From figure 4.42 and figure 4.43-4.48, it is seen that the actual length of leg 1 reached the desired length of leg 1 almost at 0 seconds, but there is a steady-state error that the actual leg length does not precisely track the desired leg length as compared to the response and the tracking error achieved by PB-SISMC. As the results are summarized in table 4.4, the trajectory tracking of PB-SISMC is much better than PB-ISMC even with parameter variation. The table shows the tracking performance of the two controllers with load and without disturbance.

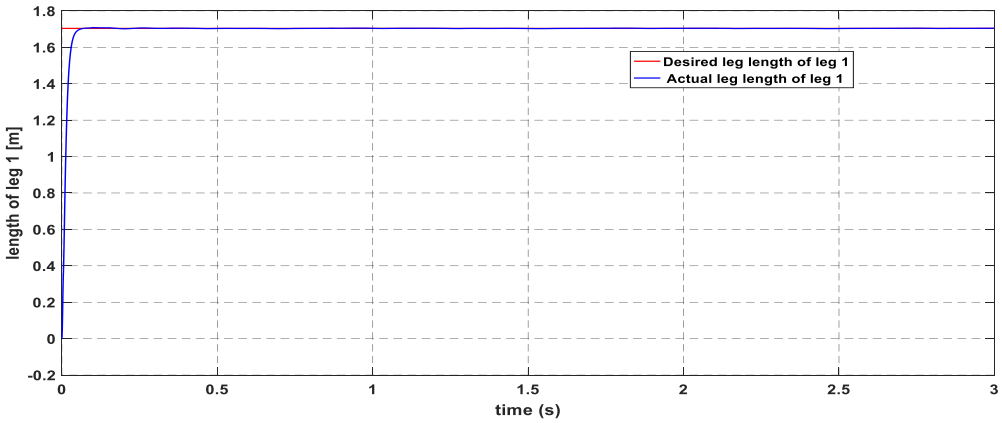


Figure4. 41. The response of the single leg (leg 1) of Stewart platform by PSO tuned PB-SISMC with load

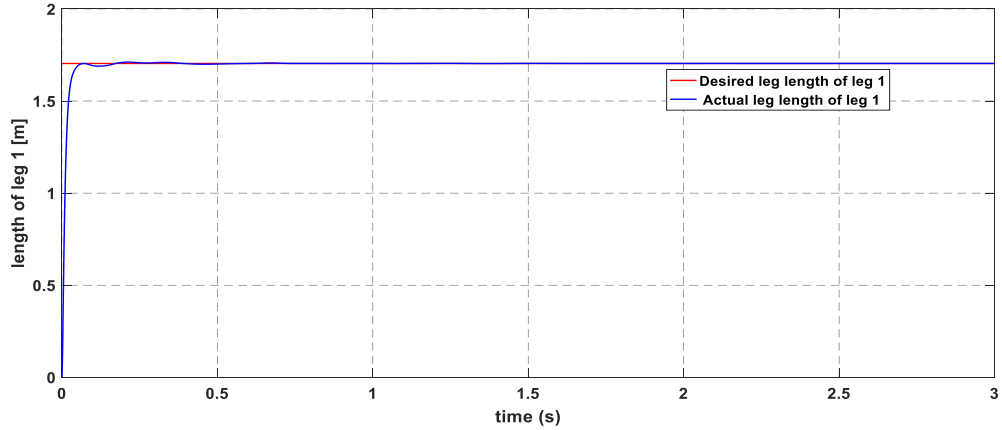


Figure4. 42. The response of the single leg (leg 1) of Stewart platform by PSO tuned PB-ISMC with load

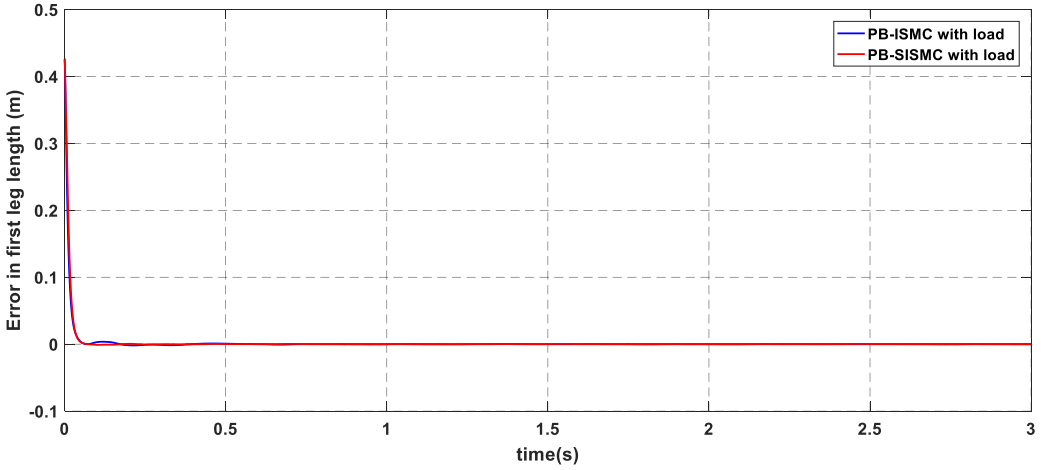


Figure4. 43. Tracking error for leg 1, with load

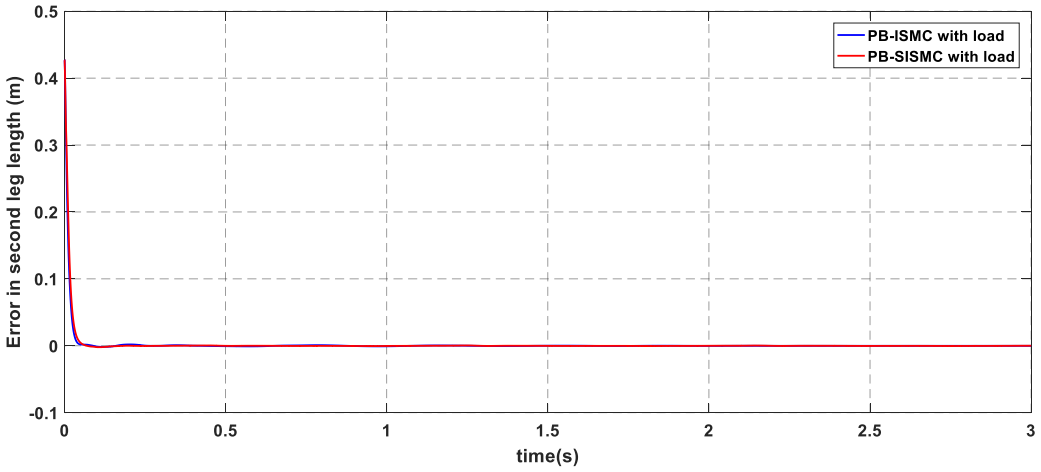


Figure4. 44. Tracking error for leg 2, with load

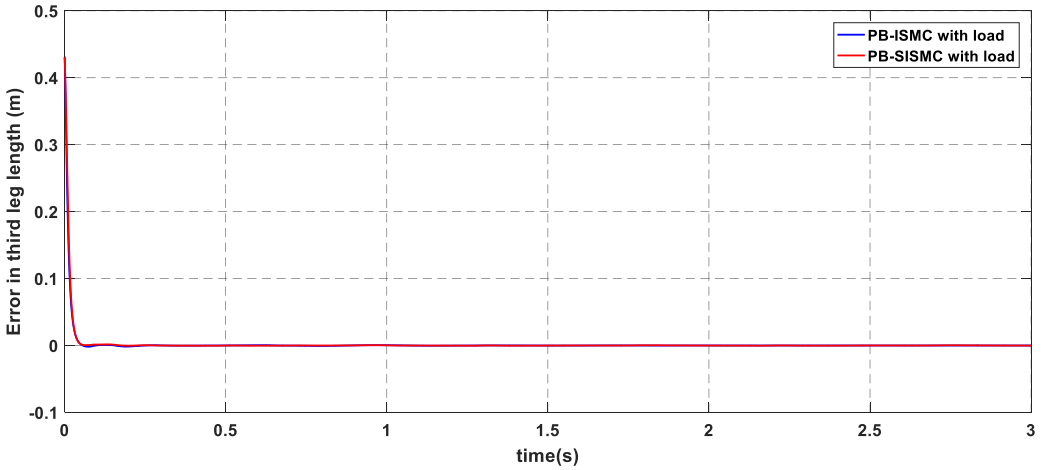


Figure4. 45. Tracking error for leg 3, with load

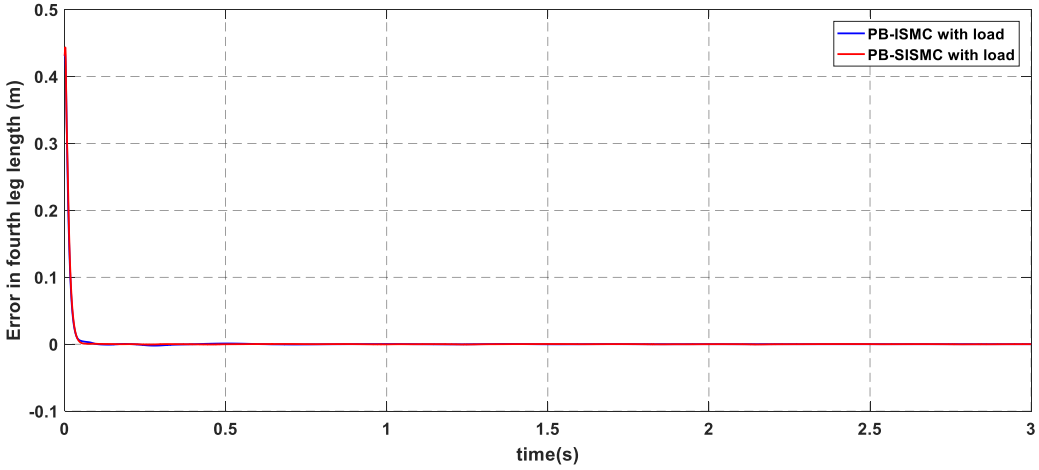


Figure4. 46. Tracking error for leg 4, with load

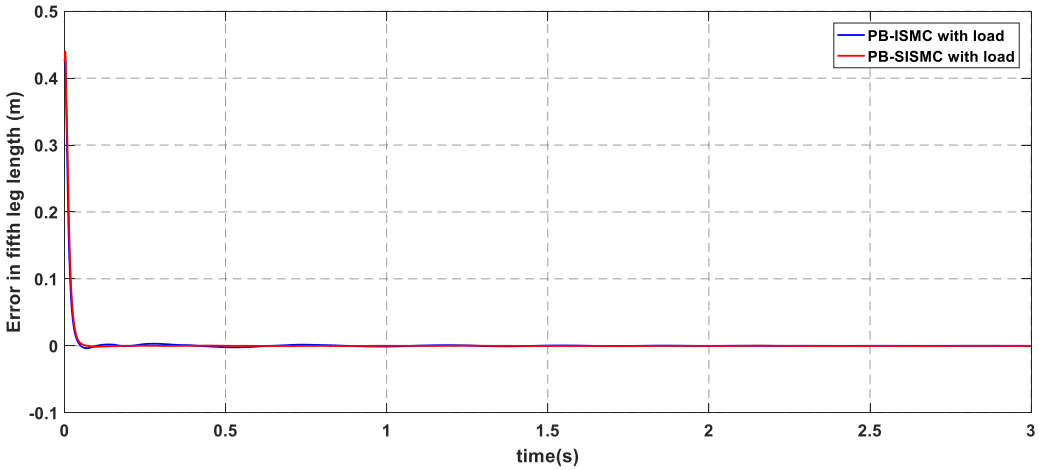


Figure4. 47. Tracking error for leg 5, with load

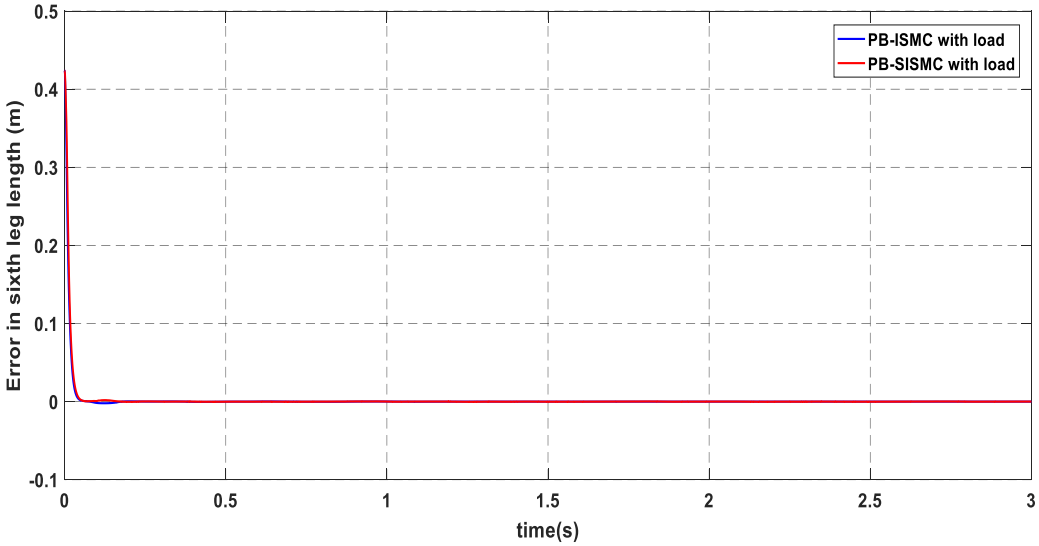


Figure4. 48. Figure4.48. Tracking error for leg 6, with load

Table4.4. Absolute error for leg i with load

Controller	Absolut tracking error for leg i in meter , with load and without disturbance					
i	1	2	3	4	5	6
PB-ISMC	0.0159	0.0183	0.0268	0.0049	0.0068	0.0182
PB-SISMC	0.0006	0.0031	0.0018	0.0011	0.0041	0.0003
Difference	0.0153	0.0152	0.0249	0.0038	0.0027	0.0179

4.6. Tracking Performance of the Controllers with Load and Disturbance

To test for robustness against disturbance and parameter variation the variable disturbance of $10^4\left[2+18\sin(20t)\right]$ is applied to the input of the Stewart platform model block as a matched disturbance and the platform is loaded with the mass of 200kg.

Case i. The plot of the integral sliding surface for each leg achieved by PB-SISMC and PB-ISMC is shown in figure 4.49-4.54. From the comparison of the two plots, the sliding surface for leg four and five enforced by PB-ISMC stays almost at 0 for the whole simulation time but not for the remaining legs. However, the corresponding controller exhibits too much chattering in their control signal, as shown in figure 4.55-4.60. And from the comparison of the two plots, it is clear that the sliding surface of each leg achieved by PB-SISMC stays almost at zero with a sliding accuracy of 0.0007, 0.0008, 0.0003, 0.0001, 0.0004, and 0.0008 meters. This result shows that the integral sliding surface for the six legs achieved by PB-SISMC is converged to zero quickly, even with load and varying disturbance.

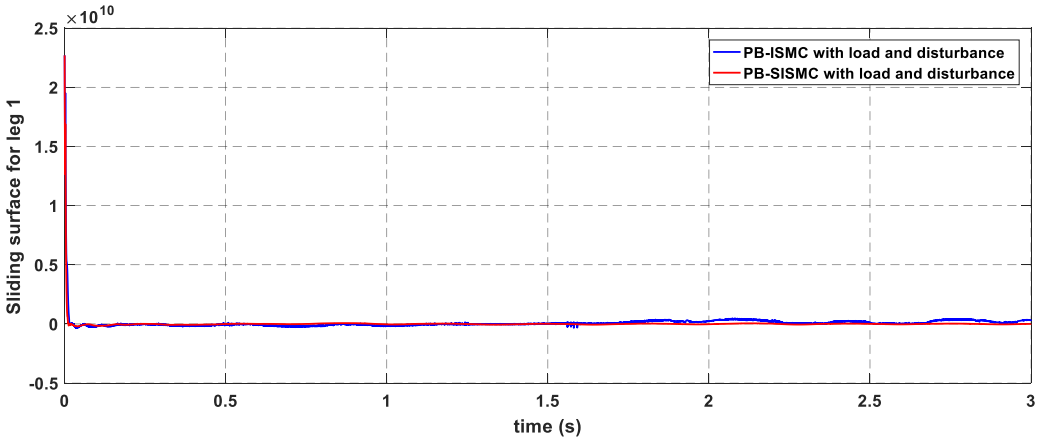


Figure4.49. Sliding surface for leg 1, with load and disturbance

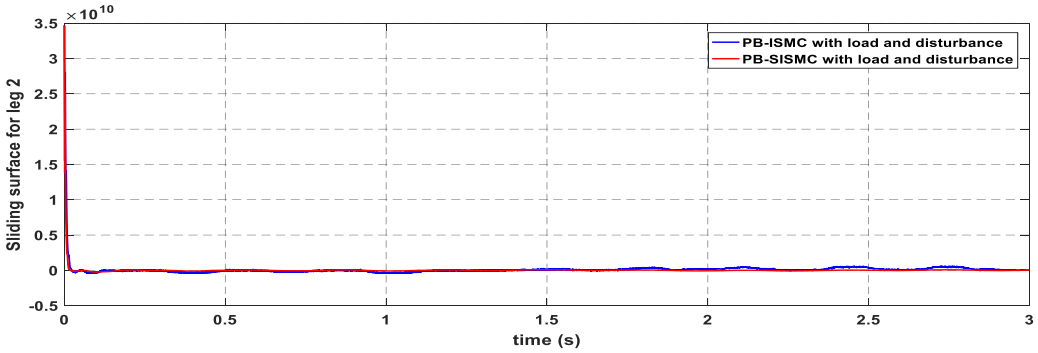


Figure4.50. Sliding surface for leg 2, with load and disturbance

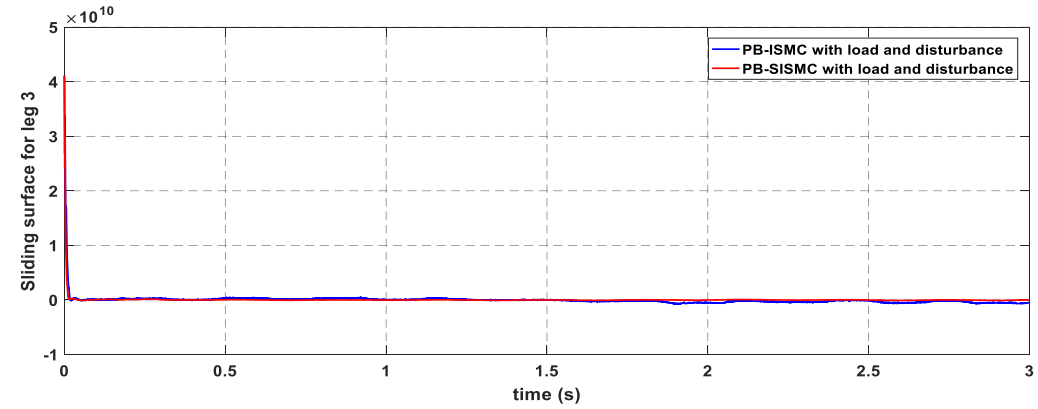


Figure4.51. Sliding surface for leg 3, with load and disturbance

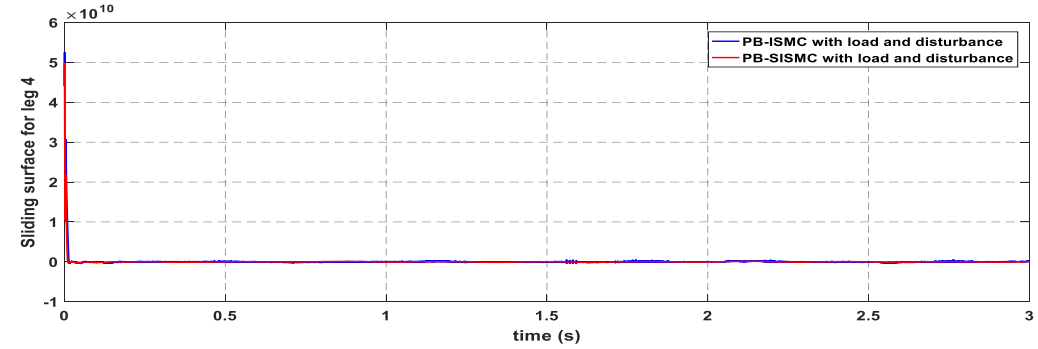


Figure4.52. Sliding surface for leg 4, with load and disturbance

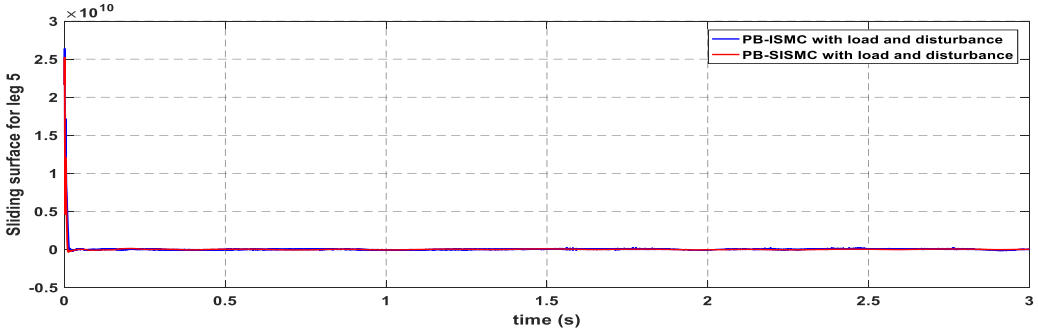


Figure4.53. Sliding surface for leg 5, with load and disturbance

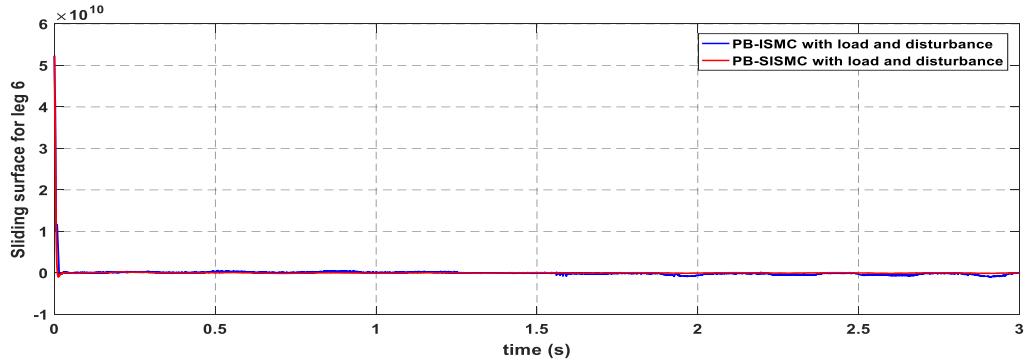


Figure4.54. Sliding surface for leg 6, with load and disturbance

Case ii. Figure 4.55-4.60 shows the evaluation of the control force, with load and disturbance together for achieving the desired trajectory of each leg with both PB-ISMC and PB-SISMC. From the comparison of the two plots, PB-ISMC exhibits too much chattering but the magnitude of oscillation smaller than oscillation without load and disturbance. These results show that the increase in robustness raises the switching gain, and the decrease in robustness reduces the magnitude of switching gain. On the other hand, PB-SISMC performs much better than PB-ISMC in chattering elimination and avoiding uncertainties due to load variation and variable disturbance.

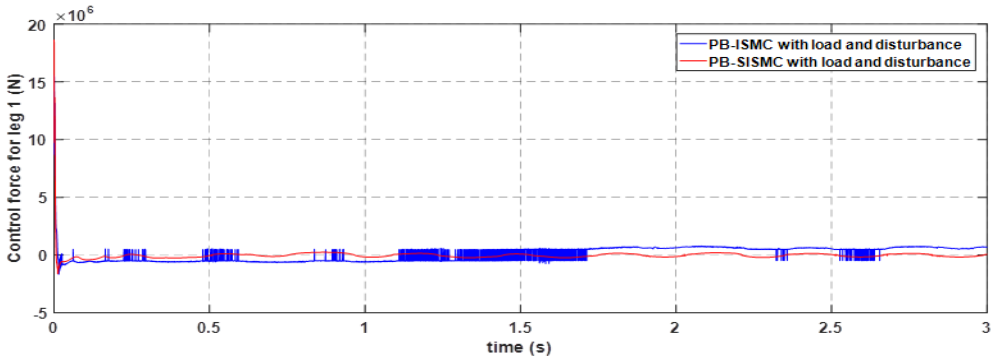


Figure4.55. Control force for leg 1, with load and disturbance

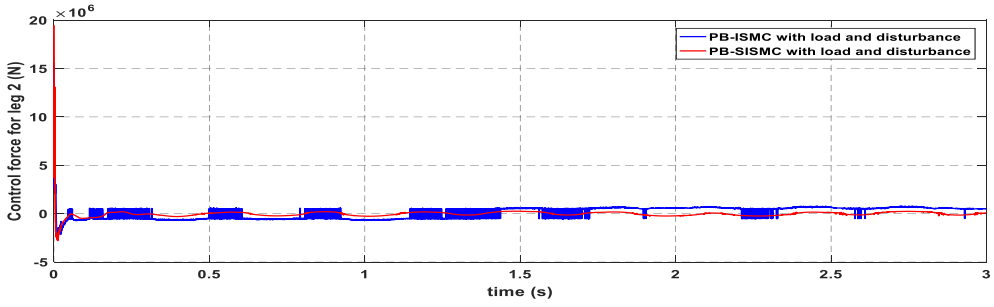


Figure4.56. Control force for leg 2, with load and disturbance

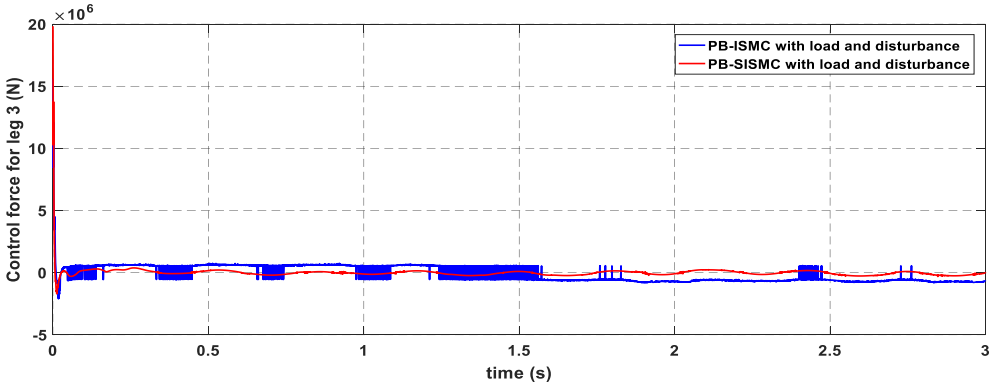


Figure4.57. Control force for leg 3, with load and disturbance

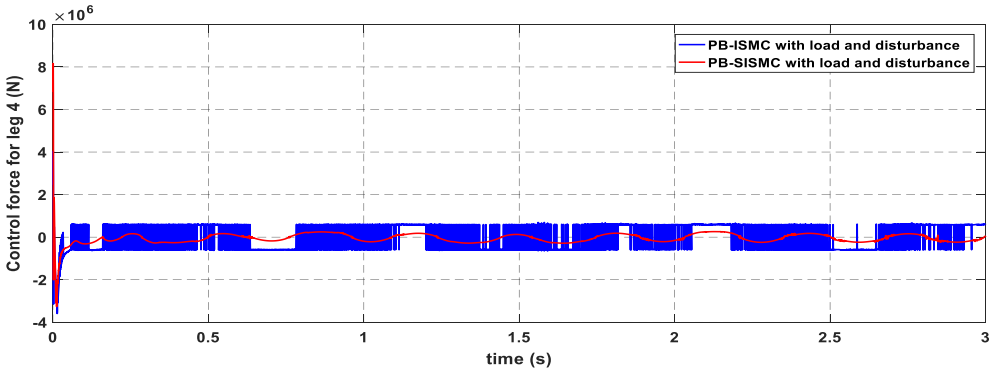


Figure4.58. Control force for leg 4, with load and disturbance

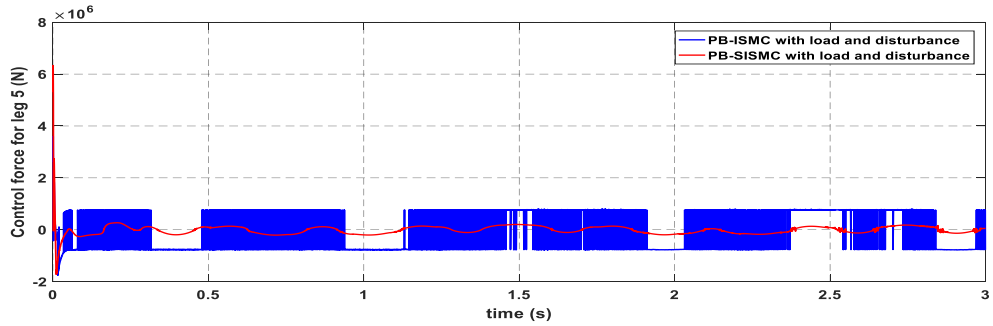


Figure4.59. Control force for leg 5, with load and disturbance

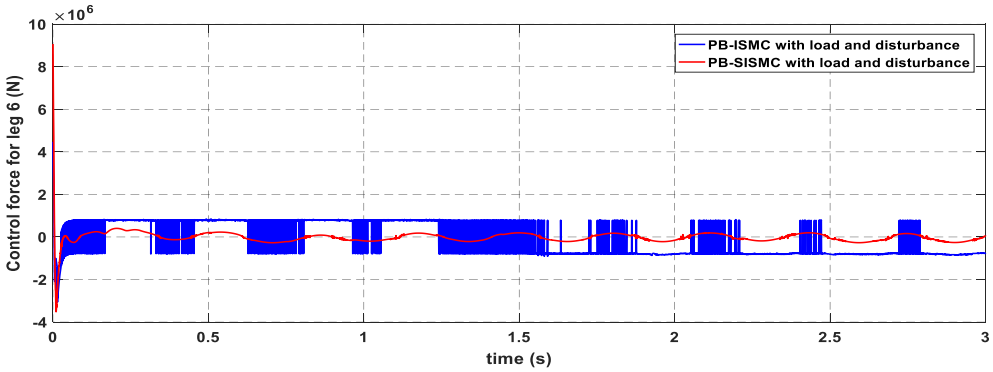
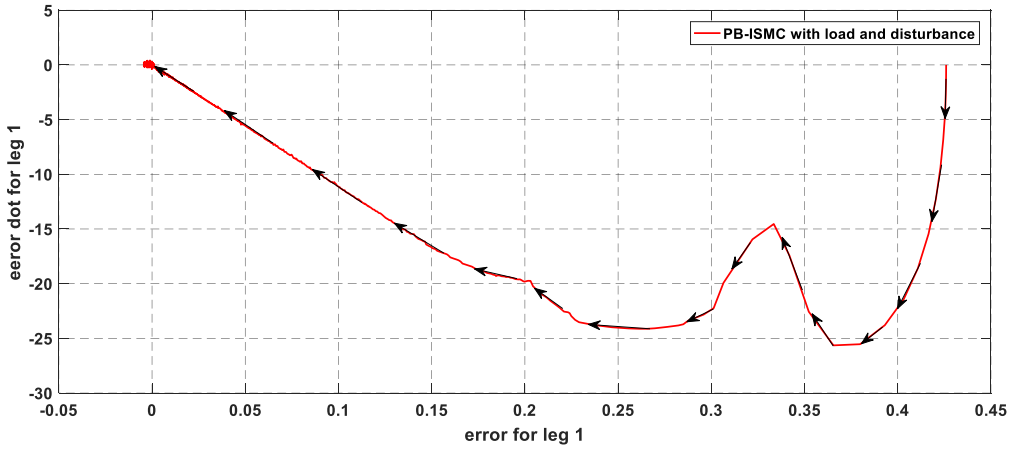
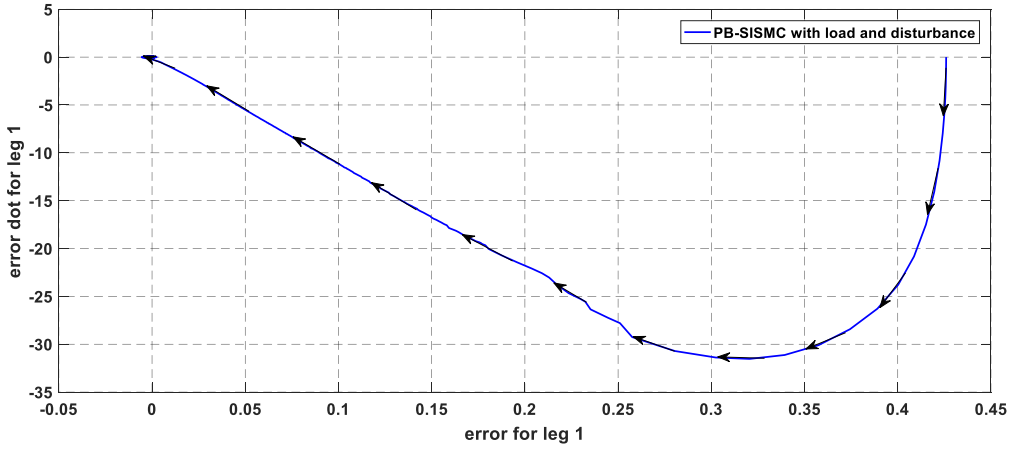


Figure4.60. Control force for leg 6, with load and disturbance

Case iii. Phase portrait of the error and its first time derivative for leg 1, with load and disturbance together achieved by PB-ISMC and PB-SISMC, is shown in figure 4.61. From the comparison of the two subplots, PB-ISMC exhibits chattering in the control signal, and closer observation of the darker region around the origin of the phase plane and finally reach to the origin shows that the sliding dynamics of the system governed by PB-ISMC is stable and the system trajectory down to decay to zero in a persisting motion of zigzag path around the origin for an indefinite time. On the other hand, chattering is avoided with PB-SISMC and the trajectory down to decay to zero without persisting motion of zigzag path around the origin.



(a) Phase portrait of error and its first derivative by PB-ISMC with load and disturbance



(b) Phase portrait of error and its first derivative by PB-SISMC with load and disturbance

Figure4.61. Comparison of the phase portrait of the error and its first derivative designed by (a) PB-ISMC (b) PB-SISMC, with load and disturbance

Case iv. Figure 4.62, 4.63, and 4.64-4.69 show the simulation result of actual leg length versus desired leg length of leg 1 achieved by PB-SISMC, PB-ISMC, and the tracking error for each leg with both PB-ISMC and PB-SISMC, with load and disturbance together. From figure 4.61 and figure 4.64-4.69, it is seen that the actual length of leg 1 reached the desired length of leg 1 almost at 0 seconds. But there is a steady-state error such that the actual leg length does not precisely track the desired leg lengths as compared to the response achieved by PB-SISMC. As the results are summarized in table 4.5, the trajectory tracking the performance of PB-ISMC is much better than PB-ISMC even in parameter variation and variable disturbance. The table shows the tracking performance of the two controllers with load and disturbance together.

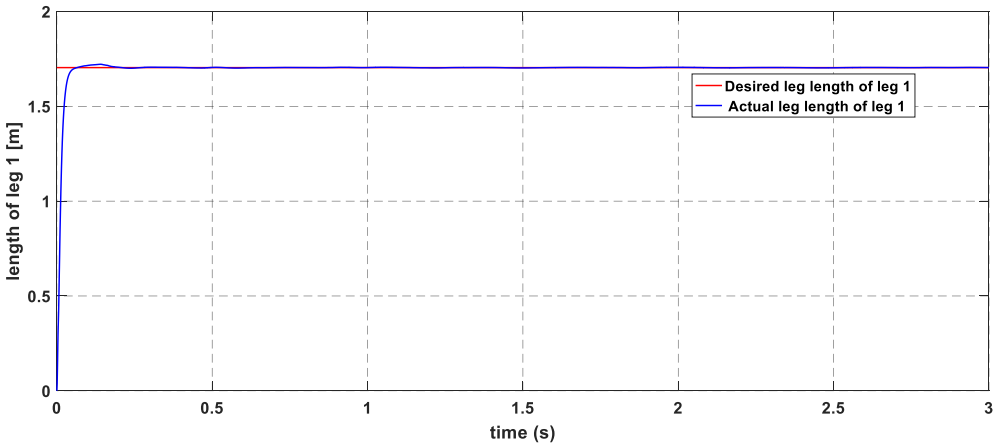


Figure4. 62. The response of the single leg (leg 1) of Stewart platform by PSO PB-SISMC with load and disturbance

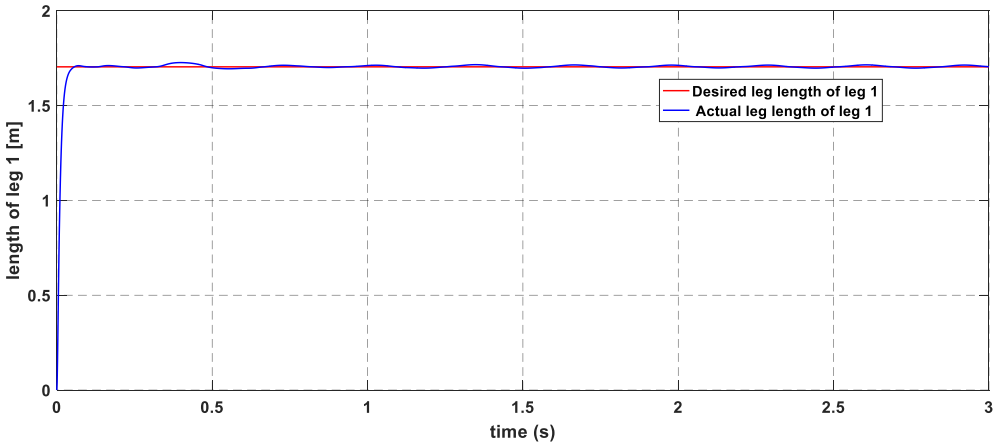


Figure4. 63. The response of the single leg (leg 1) of Stewart platform by PSO tuned PB-ISMC with load and disturbance

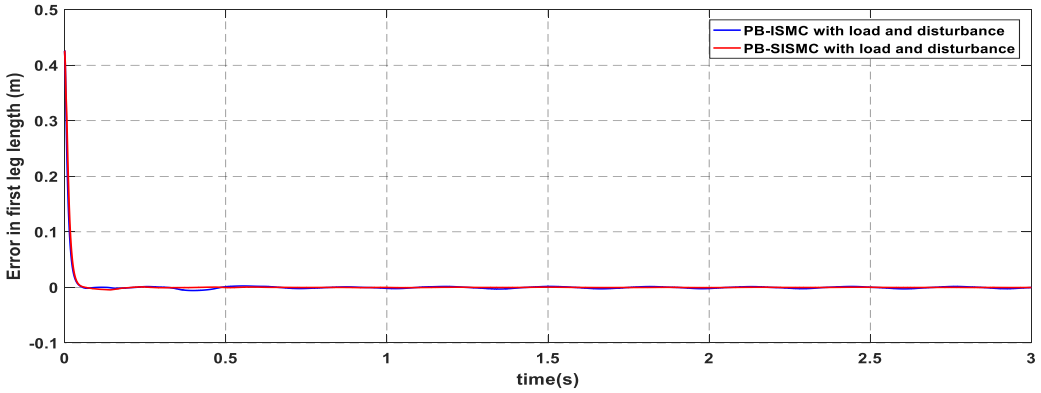


Figure4. 64. Tracking error for leg 1, with load and disturbance

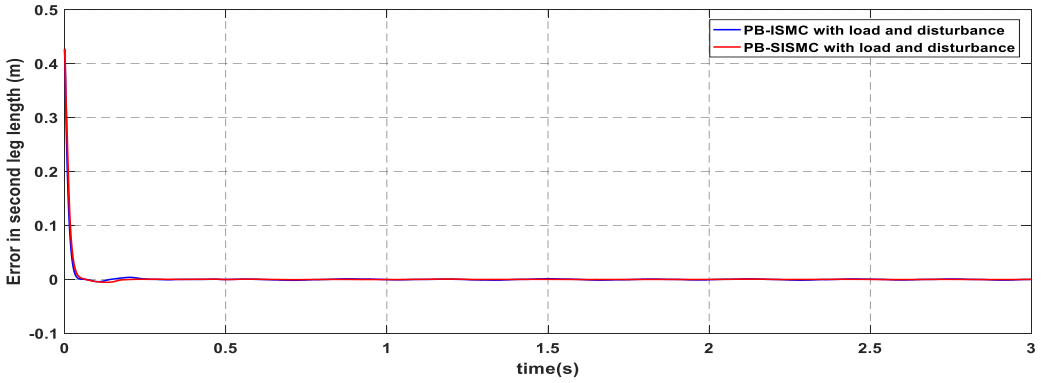


Figure4. 65. Tracking error for leg 2, with load and disturbance

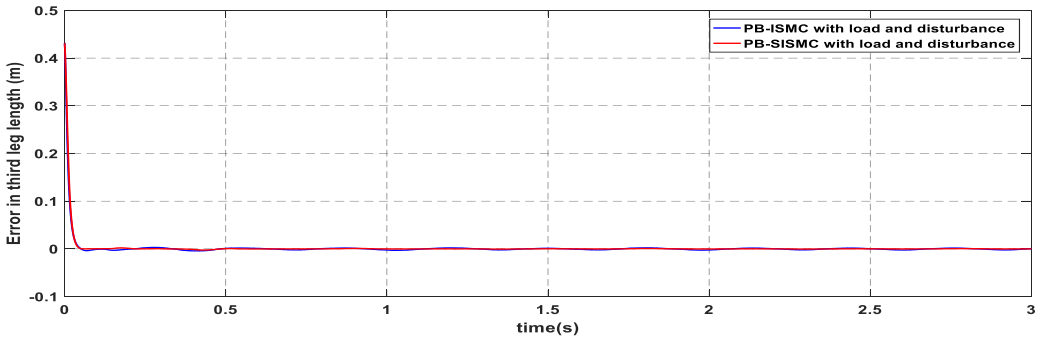


Figure4. 66. Tracking error for leg 3, with load and disturbance

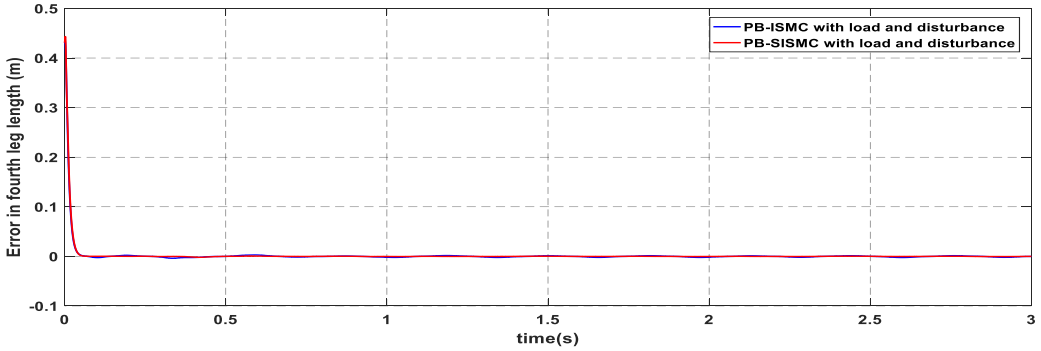


Figure4. 67. Tracking error for leg 4, with load and disturbance

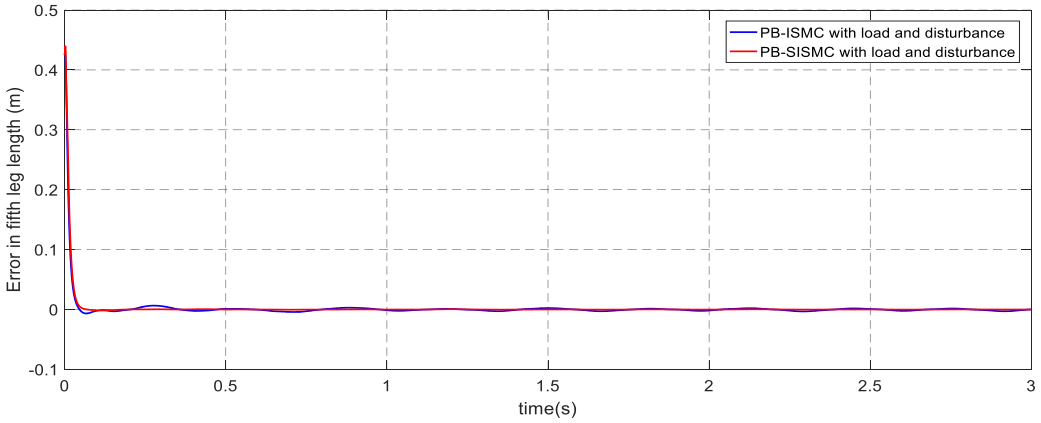


Figure4. 68. Tracking error for leg 5, with load and disturbance

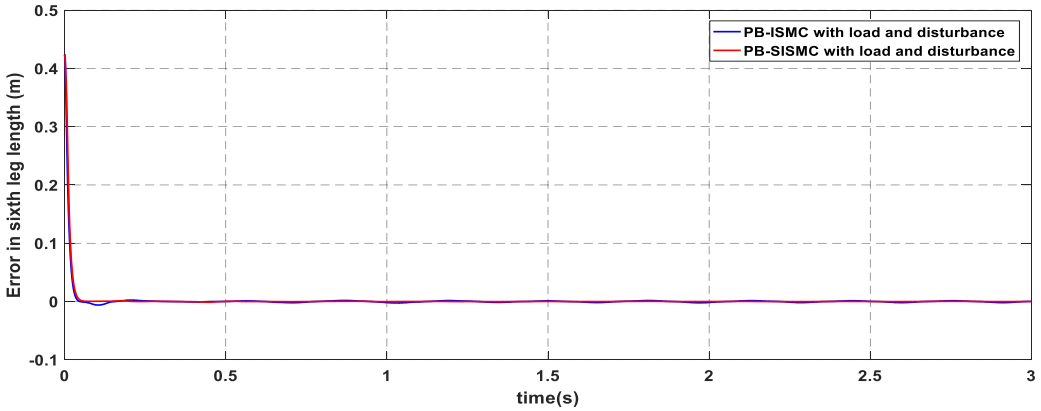


Figure4. 69. Tracking error for leg 6, with load and disturbance

Table4.5. Absolute error for leg i, with load and disturbance

Controller	Absolut tracking error for leg i in meter , with load and without disturbance					
i	1	2	3	4	5	6
PB-ISMC	0.1717	0.0070	0.1580	0.0307	0.0037	0.0151
PB-SISMC	0.0005	0.0018	0.0023	0.0017	0.0017	0.0006
Difference	0.1712	0.0052	0.1557	0.0290	0.0021	0.0145

CHAPTER FIVE

5. CONCLUSIONS AND RECOMMENDATIONS

5.1. Conclusions

In this thesis, proportional derivative plus smooth integral sliding mode controller has been designed for a nonlinear six degree of freedom parallel robot of the Stewart platform manipulator. The integral sliding mode control law modified by replacing the discontinuous control part with super twisting control is continuous. And various evaluation criteria have been chosen to compare the trajectory tracking performance of the controller designed using PB-SISMC with that of PB-ISMC.

The simulation results show that the PB-SISMC avoided the chattering in the control signal of each leg. The chattering observed in the control signal of PB-ISMC is evident in difficulty in the practical implementation of integral sliding mode controllers. For the case of the integral sliding surface using PB-SISMC, it is interesting to see that the sliding surface for each leg converges to zero in finite time and stays near to zero even with load and varying disturbance together with a sliding accuracy of 0.0007, 0.0008, 0.0003, 0.0001, 0.0004 and 0.0008 meters. The trajectory tracking performance of PB-SISMC is better than that of PB-ISMC for all evaluation criteria. The effect of parameter variation in the trajectory tracking performance of the controllers is determined by loading the platform with a mass of 200kg. Even in the presence of parameter variation, it is observed that PB-SISMC performs better in chattering elimination and avoiding uncertainties due to parameter variation. Moreover, the trajectory tracking performance of the controllers are further evaluated with load and variable disturbance of $10^4\left[2+18\sin(20t)\right]$together. The tracking error for each leg achieved by PB-SISMC is converged to zero with tracking accuracy of 0.0005, 0.0018, 0.0023, 0.0017, 0.0017, and 0.0006 meters; however, this cannot achieved using PB-ISMC instead the tracking error increased by 0.1712, 0.0052, 0.1557, 0.0290, 0.0021 and 0.0145 meters for each leg respectively.

The simulation results have shown that the designed passivity based smooth integral sliding mode controller performs much better than that of the passivity based integral sliding mode controller in chattering elimination and avoiding uncertainties because of parameter variation and variable disturbance.

5.2. Recommendations

In this thesis, the proposed controller is designed for the Stewart platform manipulator so that to test its applicability and limitation it has to be further tested with various examples like electromechanical systems, energy generating systems, aircraft systems, and so on. And also it has to be tested in task space control of the Stewart platform manipulator. It is clear that all the controllers have finally to be implemented practically. Therefore as a forward step in the direction of practical implementation, the study needs the ultrasound image serving system and various image feature extraction and tracking algorithms to control the Stewart platform manipulator and to guide the ultrasound probe as a function of its acquired image. And the study of particle swarm optimization based design method in this thesis has to be further tested with various examples to see its applicability and limitations.

REFERENCE

[1] A. M. Priester, S. Natarajan, M. O. J. I. t. o. u. Culjat, ferroelectrics,, and f. control, "Robotic ultrasound systems in medicine," vol. 60, no. 3, pp. 507-523, 2013.

[2] P. N. Wells and H.-D. J. J. o. t. R. S. I. Liang, "Medical ultrasound: imaging of soft tissue strain and elasticity," vol. 8, no. 64, pp. 1521-1549, 2011.

[3] T. Rousseau, N. Mottet, G. Mace, C. Franceschini, and P. J. J. o. U. i. M. Sagot, "Practice guidelines for prevention of musculoskeletal disorders in obstetric sonography," vol. 32, no. 1, pp. 157-164, 2013.

[4] N. Simaan, "Analysis and synthesis of parallel robots for medical applications," Technion-Israel Institute of Technology, Faculty of Mechanical Engineering, 1999.

[5] S. Briot and I. A. J. T. o. t. C. S. f. M. E. Bonev, "Are parallel robots more accurate than serial robots?," vol. 31, no. 4, pp. 445-455, 2007.

[6] B. Dasgupta, T. J. M. Mruthyunjaya, and m. theory, "The Stewart platform manipulator: a review," vol. 35, no. 1, pp. 15-40, 2000.

[7] E. Fichter, D. Kerr, and J. J. P. o. t. I. o. M. E. Rees-Jones, Part C: Journal of Mechanical Engineering Science, "The Gough—Stewart platform parallel manipulator: A retrospective appreciation," vol. 223, no. 1, pp. 243-281, 2009.

[8] F. J. A. i. m. R. Szufnarowski, "Stewart platform with fixed rotary actuators: a low cost design study," no. 4, 2013.

[9] J. H. Cho, J. Seo, and H. S. Woo, "Development of master-slave robotic system for teleoperated ultrasonography," in *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2017, pp. 585-586: IEEE.

[10] J. Seo, J. Cho, H. Woo, and Y. Lee, "Development of prototype system for robot-assisted ultrasound diagnosis," in *2015 15th International Conference on Control, Automation and Systems (ICCAS)*, 2015, pp. 1285-1288: IEEE.

[11] H. Abdellatif, J. Kotlarski, T. Ortmaier, and B. Heimann, *Practical model-based and robust control of parallel manipulators using passivity and sliding mode theory*. London: IntechOpen, 2010.

[12] V. Utkin and J. Shi, "Integral sliding mode in systems operating under uncertainty conditions," in *Proceedings of 35th IEEE conference on decision and control*, 1996, vol. 4, pp. 4591-4596: IEEE.

[13] D. Shiferaw and A. Jain, "Comparison of joint space and task space integral sliding mode controller implementations of a 6-DOF parallel robot," in *Proceedings of the*

11th WSEAS international conference on robotics, control and manufacturing technology, 2011.

[14] F. Castaños and L. J. I. T. o. A. C. Fridman, "Analysis and design of integral sliding manifolds for systems with unmatched perturbations," vol. 51, no. 5, pp. 853-858, 2006.

[15] S. Dereje, M. K. Pattanshetti, A. Jain, R. J. I. J. o. S. a. Mitra, Engineering, and Devolopment, "Genetic algorithm based integral sliding surface design and its application to Stewart platform manipulator control," vol. 5, no. 1, pp. 518-528, 2011.

[16] A. Chalanga, S. Kamal, and B. Bandyopadhyay, "Continuous integral sliding mode control: A chattering free approach," in *2013 IEEE International Symposium on Industrial Electronics*, 2013, pp. 1-6: IEEE.

[17] A. Chalanga and B. J. I. t. Bandyopadhyay, "Smooth integral sliding mode controller for the position control of Stewart platform," vol. 58, pp. 543-551, 2015.

[18] Y. Su, C. Zheng, and B. Duan, "Singularity analysis of a 6 DOF Stewart platform using genetic algorithm," in *IEEE International Conference on Systems, Man and Cybernetics*, 2002, vol. 7, p. 6 pp. vol. 7: IEEE.

[19] J.-P. J. T. I. J. o. R. R. Merlet, "Singular configurations of parallel manipulators and Grassmann geometry," vol. 8, no. 5, pp. 45-56, 1989.

[20] O. Masory and J. J. A. r. Wang, "Workspace evaluation of Stewart platforms," vol. 9, no. 4, pp. 443-461, 1994.

[21] T. Arai, K. Cleary, T. Nakamura, H. Adachi, and K. Homma, "Design, analysis and construction of a prototype parallel link manipulator," in *EEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*, 1990, pp. 205-212: IEEE.

[22] T. J. P. I. Arai, Tokyo, Japan, "Development of parallel link manipulator for underground excavation task," 1991.

[23] K. Cleary and T. Arai, "A prototype parallel manipulator: Kinematics, construction, software, workspace results, and singularity analysis," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, 1991, pp. 566-571: IEEE.

[24] E. F. J. T. I. J. o. R. R. Fichter, "A Stewart platform-based manipulator: general theory and practical construction," vol. 5, no. 2, pp. 157-182, 1986.

[25] J.-P. J. J. o. i. Merlet and r. systems, "Determination of the orientation workspace of parallel manipulators," vol. 13, no. 2, pp. 143-160, 1995.

[26] W. Khalil, O. J. J. o. i. Ibrahim, and r. systems, "General solution for the dynamic modeling of parallel robots," vol. 49, no. 1, pp. 19-37, 2007.

[27] Y. Ting, Y. S. Chen, and H. C. J. J. o. R. S. Jar, "Modeling and control for a Gough-Stewart platform CNC machine," vol. 21, no. 11, pp. 609-623, 2004.

[28] H. S. Kim, Y. Shim, Y. M. Cho, and K.-I. J. K. i. j. Lee, "Robust nonlinear control of a 6 DOF parallel manipulator: Task space approach," vol. 16, no. 8, pp. 1053-1063, 2002.

[29] W. Khalil and S. J. I. T. o. R. Guegan, "Inverse and direct dynamic modeling of Gough-Stewart robots," vol. 20, no. 4, pp. 754-761, 2004.

[30] D. S. Negash and R. Mitra, "Integral sliding mode controller for trajectory tracking control of Stewart platform manipulator," in *2010 5th International Conference on Industrial and Information Systems*, 2010, pp. 650-654: IEEE.

[31] J. Merlet, "EDUCATION AND POST-DOCTORAL TRAINING," 2016.

[32] A. Ghobakhloo, M. Eghtesad, and M. Azadi, "Position control of a Stewart-Gough platform using inverse dynamics method with full dynamics," in *9th IEEE International Workshop on Advanced Motion Control, 2006.*, 2006, pp. 50-55: IEEE.

[33] I.-F. Chung, H.-H. Chang, and C.-T. Lin, "Fuzzy control of a six-degree motion platform with stability analysis," in *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 99CH37028)*, 1999, vol. 1, pp. 325-330: IEEE.

[34] D. Li and S. Salcudean, "Modeling, simulation, and control of a hydraulic Stewart platform," in *Proceedings of International Conference on Robotics and Automation*, 1997, vol. 4, pp. 3360-3366: IEEE.

[35] C. N. Riviere, J. Gangloff, and M. J. P. o. t. I. De Mathelin, "Robotic compensation of biological motion to enhance surgical accuracy," vol. 94, no. 9, pp. 1705-1716, 2006.

[36] H.-P. Hong *et al.*, "A design of auto-tuning PID controller using fuzzy logic," in *Proceedings of the 1992 International Conference on Industrial Electronics, Control, Instrumentation, and Automation*, 1992, pp. 971-976: IEEE.

[37] H. Abdellatif, B. Heimann, and C. J. I. P. V. Holz, "Applying efficient computation of the mass matrix for decoupling control of complex parallel manipulators," vol. 38, no. 1, pp. 493-498, 2005.

[38] L. Beji, A. Abichou, and M. Pascal, "Tracking control of a parallel robot in the task space," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, 1998, vol. 3, pp. 2309-2314: IEEE.

[39] S.-H. Chen and L.-C. J. I. T. o. c. s. t. Fu, "Output feedback sliding mode control for a Stewart platform with a nonlinear observer-based forward kinematics solution," vol. 21, no. 1, pp. 176-185, 2011.

[40] W.-H. Chen, D. J. Ballance, P. J. Gawthrop, and J. J. I. T. o. i. E. O'Reilly, "A nonlinear disturbance observer for robotic manipulators," vol. 47, no. 4, pp. 932-938, 2000.

[41] A. Omran, M. Bayoumi, A. Kassem, G. J. J. J. o. M. El-Bayoumi, and I. Engineering, "Optimal forward kinematics modeling of Stewart manipulator using genetic algorithms," vol. 3, no. 4, pp. 280-293, 2009.

[42] P. Nanua and K. J. Waldron, "Direct kinematic solution of a Stewart platform," in *Proceedings, 1989 International Conference on Robotics and Automation*, 1989, pp. 431-437: IEEE.

[43] R. Ortega, A. J. Van Der Schaft, I. Mareels, and B. J. I. C. S. M. Maschke, "Putting energy back in control," vol. 21, no. 2, pp. 18-33, 2001.

[44] J. Y. Hung, W. Gao, and J. C. J. I. t. o. i. e. Hung, "Variable structure control: A survey," vol. 40, no. 1, pp. 2-22, 1993.

[45] S. Wang and D. Yu, "Neural Network-Based Integral Sliding Mode Control for Nonlinear Uncertain Systems," in *Advances in Industrial Engineering and Operations Research*: Springer, 2008, pp. 245-257.

[46] T.-L. Chern and Y.-C. Wu, "Design of integral variable structure controller and application to electrohydraulic velocity servosystems," in *IEE Proceedings D (Control Theory and Applications)*, 1991, vol. 138, no. 5, pp. 439-444: IET.

[47] T.-L. Chern and Y.-C. Wu, "Integral variable structure control approach for robot manipulators," in *IEE Proceedings D (Control Theory and Applications)*, 1992, vol. 139, no. 2, pp. 161-166: IET.

[48] H. H. J. I. T. o. A. C. Choi, "LMI-based sliding surface design for integral sliding mode control of mismatched uncertain systems," vol. 52, no. 4, pp. 736-742, 2007.

[49] A. J. I. j. o. C. Levant, "Higher-order sliding modes, differentiation and output-feedback control," vol. 76, no. 9-10, pp. 924-941, 2003.

[50] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, 1995, vol. 4, pp. 1942-1948: IEEE.

[51] M. J. L. Clerc, UK, "Particle Swarm Optimization; ISTE Ltd," 2006.

[52] A. Lazinica, *Particle swarm optimization*. BoD–Books on Demand, 2009.

APPENDICES

Appendix A. Detailed Simulink representation of figure 4.2

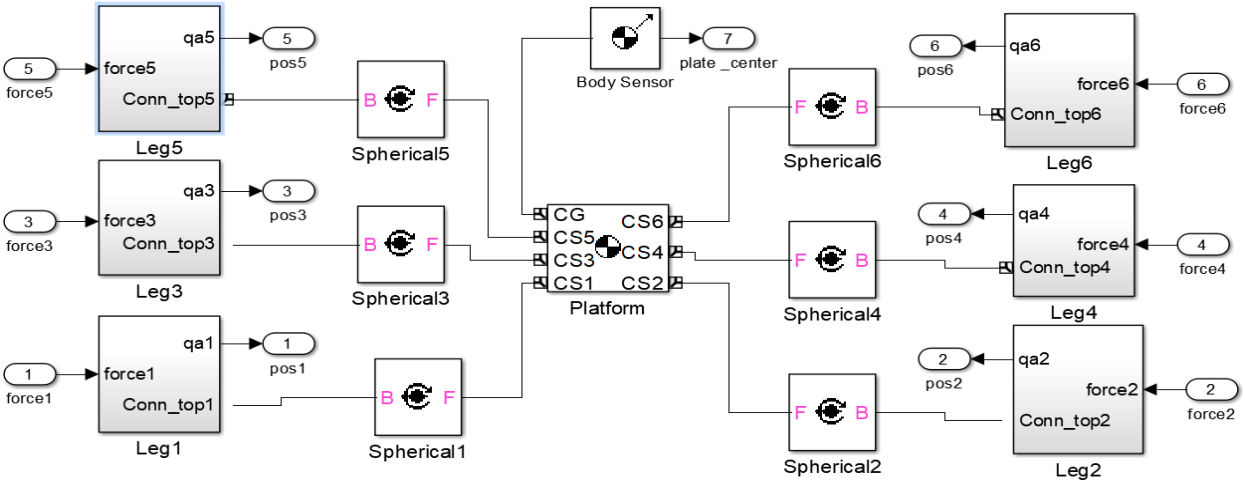


Figure A.1. Stewart platform model in Simmechanics

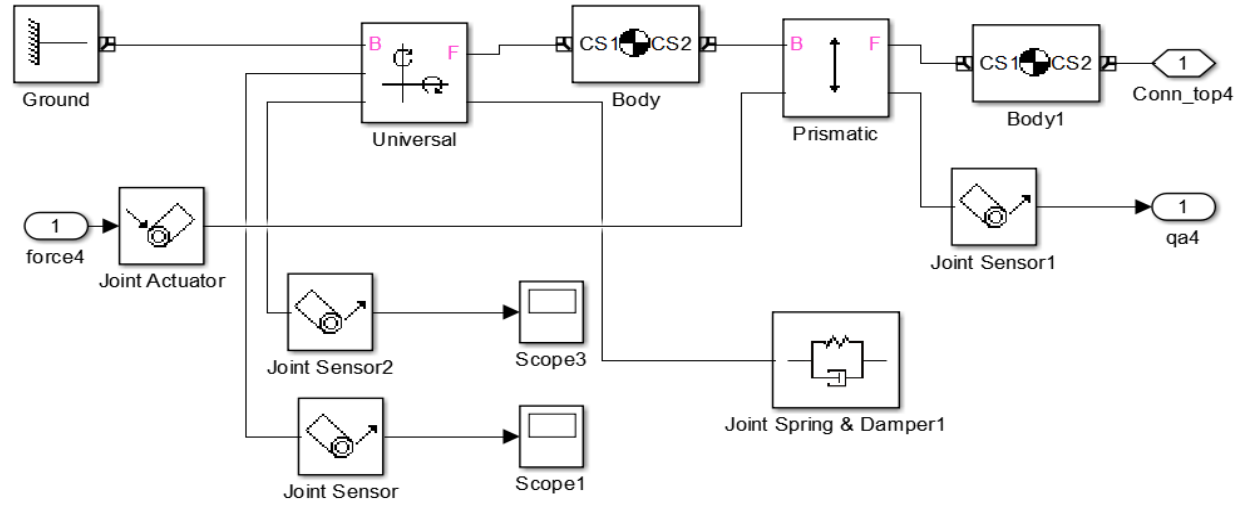


Figure A.2. Model of a leg in Simmechanics

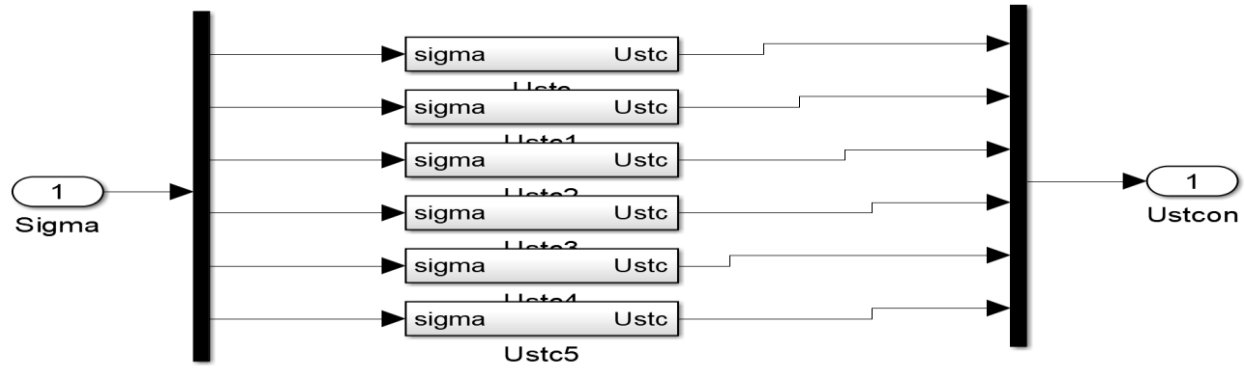


Figure A.3. Simulink model of super twisting controller

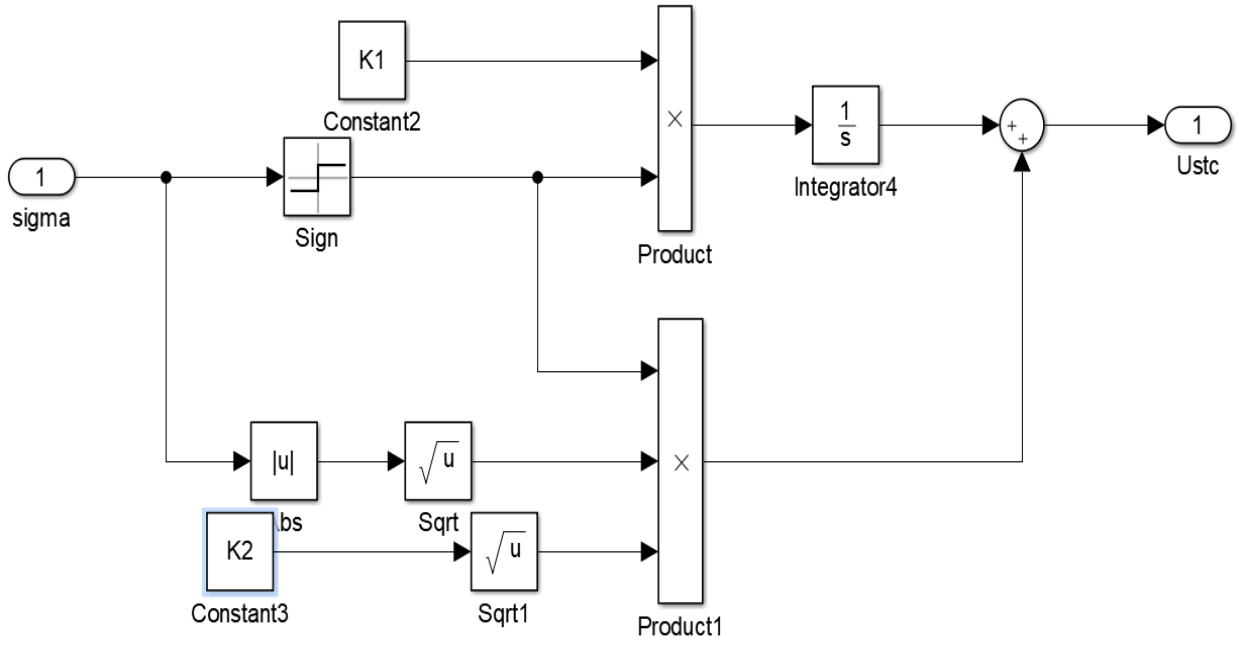


Figure A.4. Simulink model inside super twisting controller block

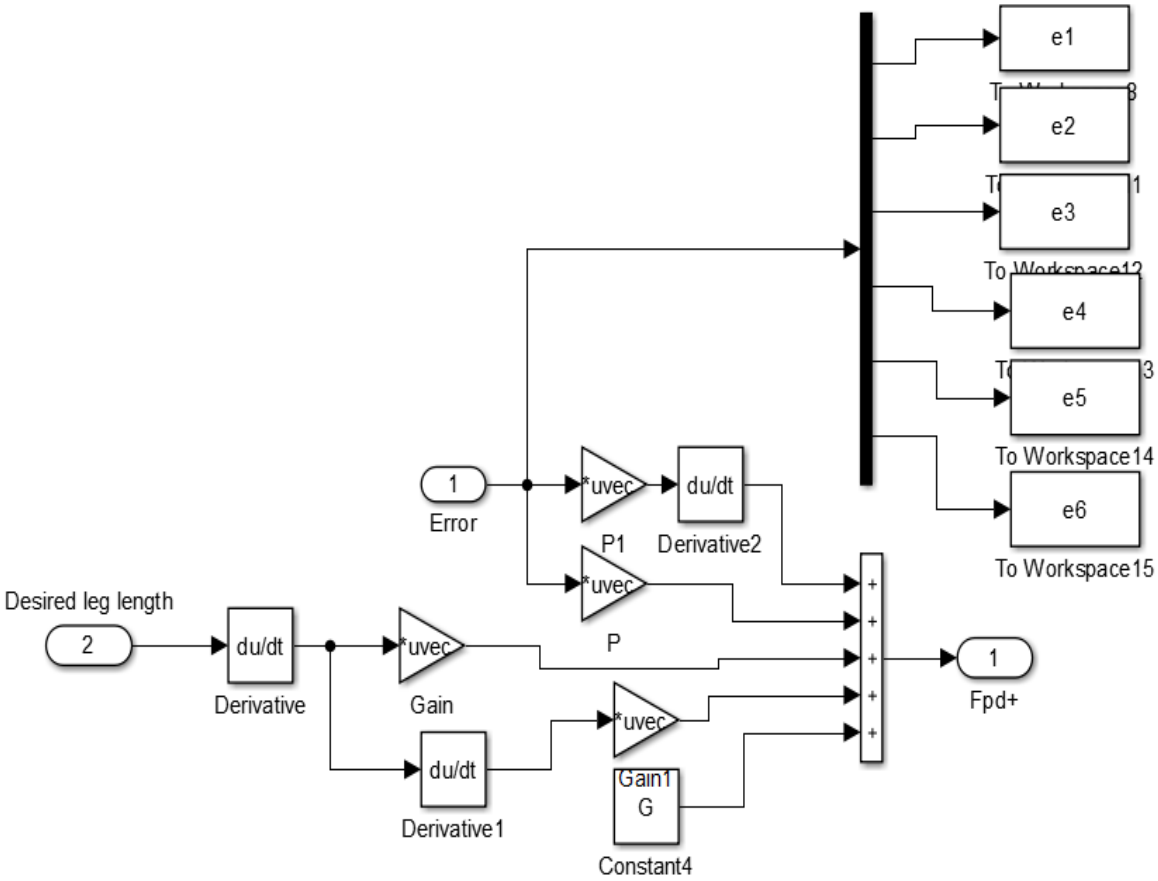


Figure A.5. Simulink model inside PD+ controller block

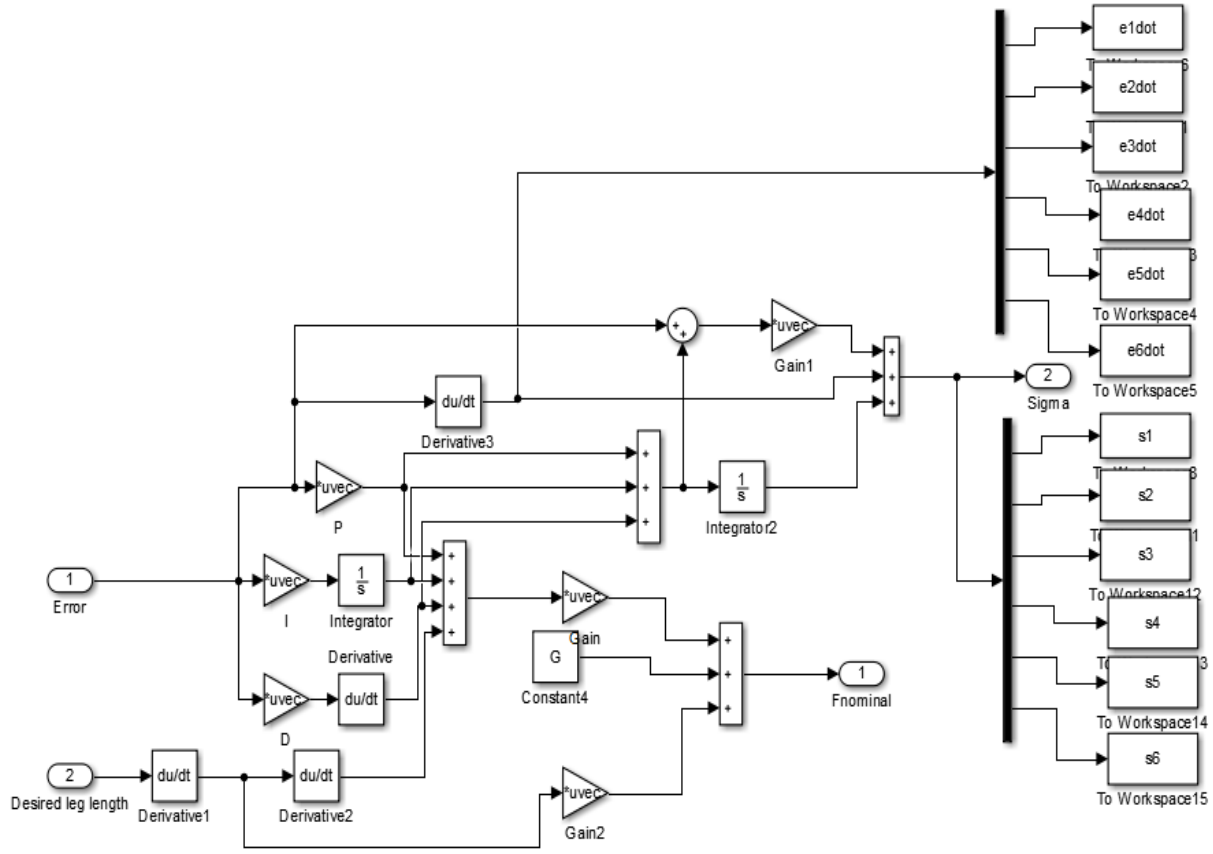


Figure A.6. Simulink model inside sliding surface and nominal controller block

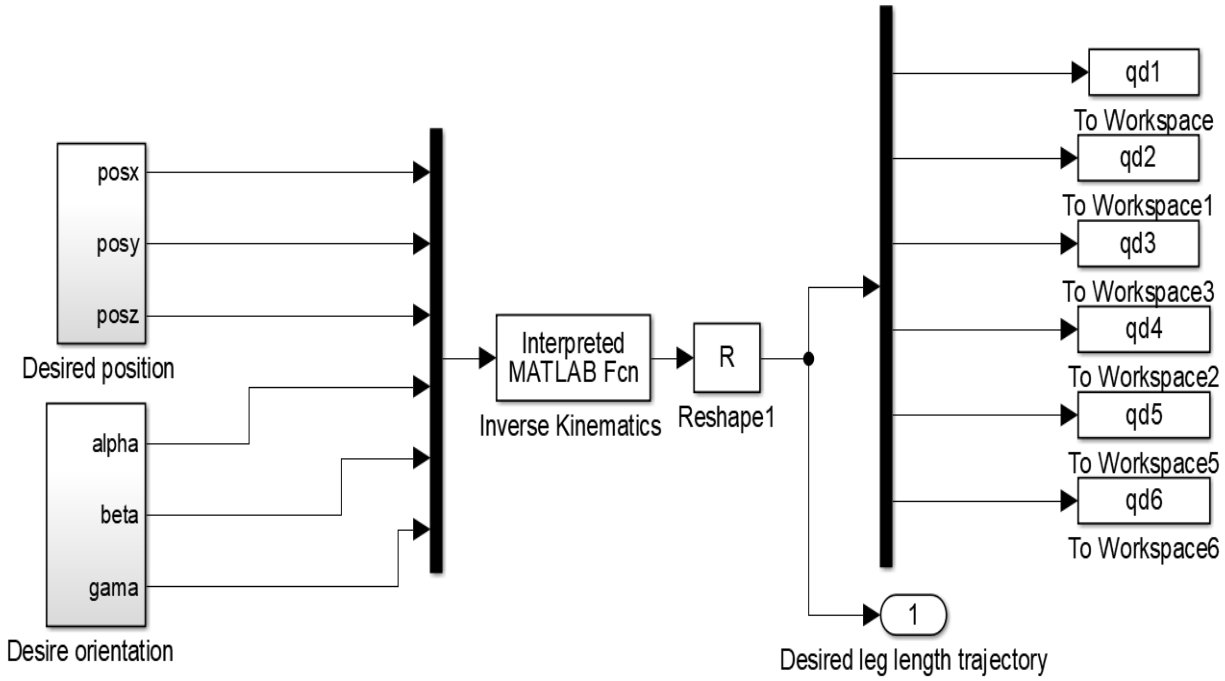


Figure A.7. Simulink model of leg length trajectory subsystem

Appendix B. MATLAB Source Code of Inverse Kinematics, Jacobian, Matrix, Singularity Analysis

B.1. Source Code of Inverse Kinematics

```
function leg_length=inverse_kinematics(x,y,z,alpha,beta,gama)
  rb=0.8;%base radius in meter
  rp=0.5;%platform radius in meter
  h=1.5; %initial height
ba=[pi/6 pi/2 5*pi/6 7*pi/6 3*pi/2 11*pi/6];
pa=[pi/12 7*pi/12 3*pi/4 5*pi/4 17*pi/12 23*pi/12];
for i=1:6
base_point(i,:) = rb* [cos(ba(:,i)), sin(ba(:,i)), 0];
platform_point(i,:) =[rp*cos(pa(:,i)), rp*sin(pa(:,i)), 0];
end
   xp=x;
   yp=y;
   zp=z;
   ap=alpha;
   bp=beta;
   gp=gama;
   Rpz=[cos(gp) -sin(gp) 0;sin(gp) cos(gp) 0; 0 0 1];
   Rpy=[cos(bp) 0 sin(bp);0 1 0;-sin(bp) 0 cos(bp)];
   Rpx=[1 0 0;0 cos(ap) -sin(ap);0 sin(ap) cos(ap)];
   Rp=Rpz*Rpy*Rpx;% rotation matrix of the platrom
   p=platform_point;
   b=base_point;
%calculation of vector of leg length and unit leg vectors
for i=1:6
      pp(i,:)=p(i,:);
      po(i,:)=Rp*pp(i,:)'+[xp yp zp+h]';
      a=(p(i,:)-b(i,:))/norm(po(i,:)-b(i,:));
      Jp(i,:)=[a cross(Rp*pp(i,:)'+[xp yp zp+h]',a)];
     l(i)=norm(po(i,:)-b(i,:));
end
```

```
    leg_length=l;
end
```

B.2. Commands to run simulation and mathematical model

```
clear all;
clc;
rb=0.8;%base radius in meter
rp=0.5;%platform radius in meter
h=1.5;%height in meter
he=[pi/6 pi/2 5*pi/6 7*pi/6 3*pi/2 11*pi/6];
me=[pi/12 7*pi/12 3*pi/4 5*pi/4 17*pi/12 23*pi/12];
deg2rad = pi/180;
ba=he;
pa=me;
t = 3;
x = 0.5*(1-exp(-pi*t))*cos(1.88*pi*t);
y =  0.5*(1-exp(-pi*t))*sin(1.88*pi*t);
z=3+(0.02/1+0.9*t)*sin(2*pi*t*(0.1+5.9*t)/(10.5))+pi/24;
roll = 0;
pitch = 0.5*(1-exp(-pi*t))*sin(0.86*pi*t);
yaw = 0.3*(1-exp(-pi*t))*sin(0.74*pi*t);
dotpitch=0.5*exp(-pi*t)*sin(0.86*pi*t)+0.43*pi*(1-exp(-pi*t))*cos(0.86*pi*t);
dotyaw=0.3*exp(-pi*t)*sin(0.74*pi*t)+0.222*pi*(1-exp(-pi*t))*cos(0.74*pi*t);
dotroll=0;
X=[roll,pitch,yaw,x,y,z];
ad=X(1,1);
alpha=ad*deg2rad;
bd=X(1,2);
beta=bd*deg2rad;
gd=X(1,3);
gama=gd*deg2rad;
x=X(1,4);
y=X(1,5);
z=X(1,6);
```

```
da=deg2rad*dotroll;
db=deg2rad*dotpitch;
dg=deg2rad*dotyaw;
for i=1:6
base_point(i,:) = rb* [cos(ba(:,i)), sin(ba(:,i)), 0];
platform_point(i,:) =[rp*cos(pa(:,i)), rp*sin(pa(:,i)), 0];
end
p=platform_point';
b=base_point';
j=jacobian(x,y,z,alpha,beta,gama,p,b);
%Mass and inertia of the Stewatr Platform
lower_leg_mass= 4;
lower_leg_inertia=[0.03,0,0;0,0.03,0;0,0,0.002];
upper_leg_mass=4;
upper_leg_inertia=[0.75,0,0;0,0.75,0;0,0,0.018];
platform_mass=32;
platform_inertia=[2,0,0;0,2,0;0,0,4];
%determination of inertia, coriolis/centifugal and gravity matrix of the platform
Ix=2;
Iy=2;
Iz=4;
mp=platform_mass;
M44=Ix*(cos(beta))^2*(cos(gama))^2+Iy*(cos(beta))^2*(sin(gama))^2+Iz*(sin(bete))^2;
M45=(Ix-Iy)*cos(beta)*cos(gama)*sin(gama);
M46=Iz*sin(beta);
M54=M45;
M55=Ix*(sin(gama))^2+Iy*(cos(gama))^2;
M64=M46;
M66=Iz;
Asb=[M44 M45 M46;M54 M55 0;M64 0 M66];
M=[mp*eye(3) 0*eye(3);0*eye(3) Asb];
G=[0;0;mp*9.8;0;0;0];
D1=cos(br)*sin(beta)*(Ix*(cos(gama))^2+Iy*(sin(geta))^2-Iz);
D2=(cos(beta))^2*cos(gama)*sin(gama)*(Ix-Iy);
```

```
D3=cos(gama)*sin(beta)*sin(gama)*(Ix-Iy);
D4=0.5*cos(beta)*(cos(gama)-sin(gama))*(cos(gama)+sin(gama))*(Ix-Iy);
D5=cos(gama)*sin(gama)*(Ix-Iy);
C22=[-D1*db-D2*dg  -D1*da-D3*db+D4*dg  -D2*da+D4*db;-D1*da+D4*dg  D5*dg
D4*da+D5*db;D2*da-D4*db -D4*da-D5*db 0];
C=[0*eye(3) 0*eye(3);0*eye(3) C22];
M=inv(j)'*M*inv(j);
C=inv(j)'*C*inv(j);
G=inv(j)'*G;
```

B.3. Jacobian Matrix

```
function j=jacobian(x,y,z,ar,br,gr,pp,bb)
Rz=[cos(gr) -sin(gr) 0;sin(gr) cos(gr) 0;0 0 1];
Ry=[cos(br) 0 sin(br);0 1 0;-sin(br) 0 cos(br)];
Rx=[1 0 0;0 cos(ar) -sin(ar);0 sin(ar) cos(ar)];
bRp=Rz*Ry*Rx;
bdp=[x,y,z]';
bDp=[bdp,bdp,bdp,bdp,bdp,bdp];
leg_vector=bRp*pp+bDp-bb;
j=[(bRp*pp+bDp-bb)',(cross(bRp*pp,bDp-bb))']/norm(leg_vector);
end
```

B.4. Source code of singularity analysis for rotary Stewart platform

```
function ratios=sing(p1,p1s,p1e,p1inc,p2,p2s,p2e,p2inc,x,y,z,alpha,beta,gama)
% Angular coords of base and platform attachment points [rad]
he=[pi/6 pi/2 5*pi/6 7*pi/6 3*pi/2 11*pi/6];
me=[pi/12 7*pi/12 3*pi/4 5*pi/4 17*pi/12 23*pi/12];
beta_b=he;
beta_p=me;
% building Parameters
Rb = 0.8;                % Base radius [m]
Rp = 0.5;                % Platform radius [m]
a = 0.2;                 % servo operating arm [m]
s = 1.49;                % servo operating leg [m]
```

```
% ith position of servo arm on base
b = [Rb*cos(beta_b(1)), Rb*sin(beta_b(1)), 0;
     Rb*cos(beta_b(2)), Rb*sin(beta_b(2)), 0;
     Rb*cos(beta_b(3)), Rb*sin(beta_b(3)), 0;
     Rb*cos(beta_b(4)), Rb*sin(beta_b(4)), 0;
     Rb*cos(beta_b(5)), Rb*sin(beta_b(5)), 0;
     Rb*cos(beta_b(6)), Rb*sin(beta_b(6)), 0];
bb = b';
% ith position of fixed leg(s)attachment point on platform
p = [Rp*cos(beta_p(1)), Rp*sin(beta_p(1)), 0;
     Rp*cos(beta_p(2)), Rp*sin(beta_p(2)), 0;
     Rp*cos(beta_p(3)), Rp*sin(beta_p(3)), 0;
     Rp*cos(beta_p(4)), Rp*sin(beta_p(4)), 0;
     Rp*cos(beta_p(5)), Rp*sin(beta_p(5)), 0;
     Rp*cos(beta_p(6)), Rp*sin(beta_p(6)), 0];
pp = p';
if (p1==4)|(p1==5)|(p1==6)
    p1s=p1s*pi/180;
    p1e=p1e*pi/180;
    p1inc=p1inc*pi/180;
else
    x=0;
end
%for parameter2
if (p2==4)|(p2==5)|(p2==6)
    p2s=p2s*pi/180;
    p2e=p2e*pi/180;
    p2inc=p2inc*pi/180;
else
    y=0;
end
% allocate the space for 2D array of singular value ratios
ratios=ones((p1e-p1s)/p1inc,(p2e-p2s)/p2inc);
% for each element in the array
```

```
l=1;
for p1temp=p1s:p1inc:p1e
   m=1;
   for p2temp=p2s:p2inc:p2e
      % set the approprate parameter
      if p1==1
         x=p1temp;
      elseif p1==2
         y=p1temp;
      elseif p1==3
         z=p1temp;
      elseif p1==4
         alpha=p1temp;
      elseif p1==5
         beta=p1temp;
      elseif p1==6
         gama=p1temp;
      end
      if p2==1
         x=p2temp;
      elseif p2==2
         y=p2temp;
      elseif p2==3
         z=p2temp;
      elseif p2==4
         alpha=p2temp;
      elseif p2==5
         beta=p2temp;
      elseif p2==6
         gama=p2temp;
      end
      j=jacobian(x,y,z,alpha,beta,gama,pp,bb);
      J=rank(j)
      %singular value decomposition
```

```
        svs=svd(j);
        % save the ratio of macx to min singular value ratios
       w= min(svs);
       ww=max(svs);
       dd= ww/w;
        ratios(l,m)=dd;
        % limit max ratios b/c they get really big
        if ratios(l,m)>250
            ratios(l,m)=250;
        end
        m=m+1;
    end
    l=l+1;
end
% set axis label, names and units
if p1==1
    name1='x (m)';
elseif p1==2
    name1='y(m)';
elseif p1==3
    name1='z (m)';
elseif p1==4
    name1='roll angle (degree)';
elseif p1==5
    name1='pitch angle (degree)';
elseif p1==6
    name1='yaw angle (degree)';
end
if (p1>=4)&&(p1<=6)
    p1e=p1e*180/pi;
    p1s=p1s*180/pi;
    units1='degree';
else
    units1='m';
```

```
end
if p2==1
   name2='x (m)';
elseif p2==2
   name2='y (m)';
elseif p2==3
   name2='z (m)';
elseif p2==4
   name2='roll angle (degree)';
elseif p2==5
   name2='pitch angle (degree)';
elseif p2==6
   name2='yaw angle (degree)';
end
 if (p2>=4)&&(p2<=6)
   p2e=p2e*180/pi;
   p2s=p2s*180/pi;
   units2='degree';
else
   units2='m';
 end
clf;
mesh(ratios,'linewidth',2);
title('Condition number of the platform jacobian versus yaw and pitch angle for x=0.024m,y=0.032m,z=0.086m,roll=0 deg');
axis equal;
text(0.78,0.075,name1,'sc');
text(0.03,0.09,name2,'sc');
text(0.02,0.9,[num2str(p1s),'<',name1,'<',num2str(p1e),units1],'sc')
text(0.02,0.85,[num2str(p2s),'<',name2,'<',num2str(p2e),units2],'sc');
text(0.02,0.8,[num2str(min(min(ratios))),'< ratios <',num2str(max(max(ratios)))],'sc');   end
```