

Design, analysis and development of MaPaMan

A Project Report

submitted by

R ARUN SRIVATSAN

*in partial fulfilment of the requirements
for the award of the degree of*

MASTER OF TECHNOLOGY



**DEPARTMENT OF Engineering Design
INDIAN INSTITUTE OF TECHNOLOGY MADRAS**

May 2012

THESIS CERTIFICATE

This is to certify that the thesis titled "**Design, analysis and development of MaPaMan**", submitted by **R Arun Srivatsan(ED07B003)**, to the Indian Institute of Technology Madras, for the award of **Dual Degree**, is a bona fide record of the research work done by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Sandipan Bandyopadhyay
Research Guide
Department of Engineering Design
IIT Madras

Dr. Nilesh Vasa
Head of the Department
Department of Engineering Design
IIT Madras

Place: Chennai

Date: 31st May, 2012

ACKNOWLEDGEMENTS

I would like to thank my guide Dr. Sandipan Bandyopadhyay whose guidance has helped in raising my understanding in robotics to a much higher level. His ideal of achieving excellence trained me hard to stand in difficult times throughout my stay at IIT.

I would like to thank my parents and my sister, who provided me strength and inspiration to reach this juncture of my life. Without their help it would not have been possible.

ABSTRACT

KEYWORDS: Parallel manipulators; Degrees of freedom,

Parallel manipulators with lower mobility have been a matter of cynosure to researchers for the last decade or so, due to their wide scope for application and lower power consumption as opposed to Stewart platform based six degrees-of-freedom (DoF) solutions. However there are two major concerns with existing designs of lower mobility manipulators; namely slow motion due to the presence of prismatic actuators or sliding joints and inability to be used for distinct tasks requiring different combinations of DoF. Thus a novel 3-DoF patial parallel manipulator, Madras Parallel Manipulator MaPaMan) was conceived of, to overcome these two problems to the extent possible. MaPaMan was designed as an improvement over the existing 3-RPS manipulator. It has no prismatic actuators or joints and uses three actuators of rotary type to generate its 3-DoF. An additional feature of MaPaMan is that it can be mechanically reconfigured to produce roll, pitch and heave motions or roll, pitch and yaw motions; which makes it versatile in applications. An indepth kinematic analysis of both the configurations of MaPaMan has been carried out and the claims w.r.t. the partitioning of DoF has been established. A gain-type singularity analysis of MaPaMan-I has also been carried out to find out the regions in the workspace where the manipulator loses controllability. It was attempted to find out a singular manifold in the task-space, that can have implications in path planning; however the expression of the manifold turned out to be very large to manipulate and study, hence the manifold was visualised numerically with the help of contour plots. The contour plots were then used to find convex regions for singularity free path planning.

Inorder to design a prototype of MaPaMan, the effect of its numerous design parameters have to be studied. As there exist no unique performance metrics

to compare two parallel manipulators, certain new metrics were developed. The variation of metric properties with change in the design parameters were studied and the overall behaviour of the metrics themselves were visualised with the help of a dynamic visualiser that was developed for this purpose. A genetic algorithm based optimization framework was developed to find the best set of dimensions for a desired range of motion of the platform, taking into account various constraints like singularities, joint limits, joint intersections etc. Based on this, two prototypes of MaPaMan-I were designed and fabricated. The first prototype was a skeletal model, built to exhibit the kinematic relations, while the second prototype was provided with a number of sensors to detect possible errors due to backlash of motor-gear, manufacturing inaccuracies etc. The three major design challenges were the choice of design for rotary joints, the design of spherical joints and the design of the unit housing the sensors and coupling it with the links. The prototype-II was developed to validate the control schemes developed for trajectory tracking of MaPaMan-I, in concurrent studies by Mehta (2012). The prototype was then developed into a motion simulation platform, wherein the end-effector would track any trajectory that is given to it using a dual loop control scheme. The platform was also controlled realtime using a joystick to demonstrate the capabilities of the manipulator and illustrate the immense scope for its application in various fields.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	i
ABSTRACT	ii
LIST OF TABLES	vii
LIST OF FIGURES	ix
NOTATION	x
1 Background	1
1.1 Literature review	1
1.1.1 Parallel manipulators	1
1.1.2 Applications of parallel manipulators	4
1.1.3 Demerits of some of the existing 3-DoF parallel manipulators	4
1.2 Motivation	5
1.3 Objectives	5
1.4 Organisation of the thesis	6
2 Conceptual design	8
2.1 Introduction	8
2.2 Design evolution	8
2.3 Detailed architecture	9
2.4 Design features	12
2.5 Conclusion	12
3 Kinematics	13
3.1 Introduction	13
3.2 Zeroth-order kinematics of MaPaMan-I	13

3.2.1	Forward kinematics	13
3.2.2	Inverse kinematics	22
3.3	Geometrical implication of the architecture of MaPaMan-I	25
3.4	First-order kinematics of MaPaMan-I	28
3.4.1	Forward kinematics	28
3.4.2	Characterising the degree-of-freedom of MaPaMan-I	29
3.4.3	Inverse kinematics	33
3.5	Zeroth-order kinematics of MaPaMan-II	33
3.5.1	Inverse kinematics	36
3.6	First-order kinematics of MaPaMan-II	38
3.6.1	Forward kinematics	38
3.6.2	Characterising the degrees-of-freedom of MaPaMan-II	39
3.7	Incremental approach to zeroth-order kinematics	40
3.8	Conclusion	41
4	Singularity analysis of MaPaMan-I	43
4.1	Introduction	43
4.2	Conditions for gain-type singularity	43
4.2.1	Singular manifold	45
4.2.2	Numerical studies of the gain-type singularities	47
4.3	Conclusion	52
5	Dynamics	53
5.1	Introduction	53
5.2	Development of the mathematical model in the actuator space	53
5.3	Conclusion	54
6	Parametric study of the manipulator architecture	56
6.1	Introduction	56
6.2	Description of performance metrics	56
6.2.1	Variation of stiffness	57
6.2.2	Variation of payload to self-weight inertia	64

6.2.3	Variation of mechanical advantage	68
6.3	Dynamic visualisation of behaviours of performance indices	72
6.4	Optimization for desired range of motion	73
6.5	Conclusion	77
7	Prototype	79
7.1	Introduction	79
7.2	Prototype-I	79
7.2.1	Design details	80
7.3	Prototype II	88
7.3.1	Design details	88
7.4	Details of the encoders	91
7.5	Details of the IMU	91
7.6	Details of the software used	92
7.6.1	Details of the electronic components	93
7.7	Conclusion	93
8	Developing MaPaMan-I for motion simulation applications	95
8.1	Introduction	95
8.2	Interfacing	95
9	Conclusion and future work	100
9.1	Overview	100
9.2	Future work	103
A	Appendix	104
A.1	Dimensions of MaPaMan prototype-I	104
B	Appendix	105
B.1	Dimensions of MaPaMan prototpe-II	105
C	Appendix	106
C.1	Inertia values of MaPaMan prototpe-II	106
C.2	CAD drawings of the mechanical components	107

LIST OF TABLES

6.1	Variable ranges used in the optimization study	76
6.2	Values of parameters used in the optimization study	76
6.3	Optimization results	77
7.1	Software details	92
7.2	Devices used in experiments	93
A.1	MaPaMan prototype-I dimensions	104
B.1	MaPaMan prototype-II dimensions	105
C.1	MaPaMan prototype-II mass values	106
C.2	MaPaMan prototype-II mass moment of inertia values about centroidal axis	106

LIST OF FIGURES

2.1 Evolution of MaPaMan from 3-RPS	8
2.2 The two reconfigurable architectures of MaPaMan	10
2.3 Reconfigurable joint between the strut and the coupler	11
3.1 Kinematic diagram of a leg of MaPaMan-I	14
3.2 The two branches of solutions in a four-bar	16
3.3 The pose of MaPaMan-I corresponding to the $\boldsymbol{\theta} = (0.8, 1.4, 1.1)^T$	21
3.4 Eight inverse kinematic configurations of MaPaMan-I	24
3.5 Kinematic representation of a leg of MaPaMan-II	34
3.6 The pose of MaPaMan-II corresponding to the $\boldsymbol{\theta} = (0.8, 1.4, 1.1)^T$	36
4.1 A gain-type singular pose of MaPaMan-I	44
4.2 Singularity countour showing 3-way symmetry	48
4.3 Singularity contours in terms of φ and ς at varying z_c	49
4.4 Singularity contours in roll and pitch at varying z_c	51
6.1 Kinematically equivalent 1-DoF mechanism representing a leg of MaPaMan-I for pure heave motion	57
6.2 FBD of MaPaMan-I	58
6.3 Variation of stiffness with change in input angle	61
6.4 Effect of design parameters on stiffness	63
6.5 Variation of <i>IR</i> with change in input angles	66
6.6 Effect of design parameters on <i>IR</i>	67
6.7 Variation of mechanical advantage with change in input angle .	68
6.8 Effect of design parameters on on mechanical advantage	70
6.9 Variation of torque in motor w.r.t. crank angle	71
6.10 Dynamic visualiser to visualise behaviour of performance metrics	72
6.11 Geometrical condition for maximum heave in MaPaMan-I	74

6.12 Geometrical condition for angle subtended by spherical joint at the end-effector in MaPaMan-I	75
7.1 Candidate designs for the revolute joints of MaPaMan	81
7.2 Typical designs of spherical joints	82
7.3 Typical designs of spherical joints	84
7.4 The crank-motor assembly	84
7.5 The design of strut	85
7.6 The design of base	86
7.7 The assembly of prototype-I	86
7.8 The base position of prototype-I	87
7.9 Accommodating the encoder at the rocker	89
7.10 Modification in the design of the coupler link for increased range of motion	90
7.11 The assembly of prototype-II	90
C.1 CAD drawing of strut in prototype-I	108
C.2 CAD drawing of crank	109
C.3 CAD drawing of rocker in prototype-I	110
C.4 CAD drawing of rocker-stand in prototype-I	111
C.5 CAD drawing of rocker-stand in prototype-II	112
C.6 CAD drawing of top part of socket in ball-socket joint	113
C.7 CAD drawing of bottom part of socket in ball-socket joint	114
C.8 CAD drawing of the end-effector	115
C.9 CAD drawing of the base plate	116

NOTATION

l_0	Length of the ground link of a leg of MaPaMan.
l_1	Length of the crank of a leg of MaPaMan.
l_2	Length of the coupler of a leg of MaPaMan.
l_3	Length of the rocker of a leg of MaPaMan.
r	Length of the strut of a leg of MaPaMan.
d_t	Length of each side of triangular end-effector.
r_t	Circumradius of top of MaPaMan.
x_1	Circumradius of base of MaPaMan.
m_1	Mass of the crank of a leg of MaPaMan.
m_2	Mass of the coupler of a leg of MaPaMan.
m_3	Mass of the rocker of a leg of MaPaMan.
m_4	Mass of the strut of a leg of MaPaMan.
m_5	Mass of the end-effector.
$\boldsymbol{\theta}$	Vector of all the active joint variables.
$\boldsymbol{\gamma}$	Vector of all the strut angles.
$\boldsymbol{\phi}$	Vector of all the coupler angles.
$\boldsymbol{\psi}$	Vector of all the rocker angles.
\boldsymbol{q}	Vector of all the active and passive joint variables.
\mathbf{p}_c	Pointer of the centre of the end-effector.
$\mathbf{J}_{\boldsymbol{\eta}\boldsymbol{\theta}}$	Jacobian matrix, resulting from the partial derivative of $\boldsymbol{\eta}$ with respect to $\boldsymbol{\theta}$.
$\mathbf{J}_{\boldsymbol{\eta}\boldsymbol{\gamma}}$	Jacobian matrix, resulting from the partial derivative of $\boldsymbol{\eta}$ with respect to $\boldsymbol{\gamma}$.
$\mathbf{J}_{\boldsymbol{\eta}\boldsymbol{q}}$	Jacobian matrix, resulting from the partial derivative of $\boldsymbol{\eta}$ with respect to \boldsymbol{q} .
$\mathbf{J}_{\mathbf{p}_c\boldsymbol{\theta}}$	Jacobian matrix, resulting from the partial derivative of \mathbf{p}_c with respect to $\boldsymbol{\theta}$.
$\mathbf{J}_{\mathbf{p}_c\boldsymbol{\gamma}}$	Jacobian matrix, resulting from the partial derivative of \mathbf{p}_c with respect to $\boldsymbol{\gamma}$.
\mathbf{J}_v	Velocity Jacobian matrix.
$\mathbf{M}_{\boldsymbol{\theta}}$	Mass matrix in active variable space.
$\mathbf{C}_{\boldsymbol{\theta}}$	Matrix representing centripetal and Coriolis terms.
$\mathbf{G}_{\boldsymbol{\theta}}$	Vector representing gravity induced forces.
$S(\boldsymbol{\gamma})$	Singularity function.

CHAPTER 1

Background

1.1 Literature review

A mechanism that enables a rigid body under study,i.e. an “end-effector”, to move in space in a desired fashion w.r.t. a fixed base can be called a manipulator. With appropriate controls, a manipulator can be made into a robot, as suggested by Hunt (1978) for the first time. A rigid body in space can have a maximum of six degrees-of-freedom: three rotations and three translations. A robot controls all DoF of the end-effector hence enabling it to achieve a desired position and orientation. Robotic manipulators are broadly classified into three types: serial, parallel and hybrid (Merlet, 2001). Serial manipulators consist of a series of links connected in succession by one-DoF actuated joints. While serial robots have the advantage of high dexterity and large works-paces for their size, they suffer from drawbacks such as low precision and load carrying capacity, which are critical requirements in many applications. This is primarily due to a predominant cantilevered architecture resembling a human arm. Parallel manipulators, on the other hand, offer the advantages of high precision coupled with high load-carrying capacity, albeit at the cost of smaller workspace volume and complex kinematic formulations and singularities within the workspace. This is because a the end-effector is connected to the base by a number of legs in a parallel fashion, sharing the load at all points of time. The hybrid manipulators are compositions of serial and parallel components.

1.1.1 Parallel manipulators

A parallel robot is made up of an end-effector with n -DoF, and of a fixed base, linked together by at least two independent chains. Actuation takes place through

n actuators. They hence differ strikingly from serial robots. Parallel manipulators due to the nature of their architecture, are highly precise and have greater stiffness than their serial counterparts of similar dimensions. Also they have higher payload to self-weight ratio which means they can carry greater payload as opposed to their serial counterparts. However due to the presence of closed loops in the geometry, they have lower workspace and encounter singularities which make it difficult to control at certain instances even when within the workspace.

Therefore in applications requiring high precision and weight carrying capabilities, parallel manipulators can fit in as better solutions. However when large workspace is a requirement or very quick motion is needed at the output, serial robots are generally the more preferred solution. Parallel manipulators are generally not very fast as most of the designs contain prismatic actuators or passive sliding joints; which are typically responsible for a lot of friction in the system. Manipulators having only rotary, universal and(or) spherical joints in their mechanism usually overcome this problem to a great deal. For example, the Delta robot (Clavel, 1988) inspite of its parallel architecture is faster than serial robots. The problem of workspace or singularities is inherent to parallel manipulators and cannot be overcome trivially. Typically a hybrid manipulator built using a parallel manipulator at base followed by a serial arm is used to offer a solution that can act as a compromise (Shahinpoor, 1992).

Lower mobility parallel manipulators

Parallel manipulators can have a maximum of 6-DoF and several manipulators have been developed with 6-DoF, such as the famous Stewart platform manipulator (Gough, 1940). However, in applications that require less than 6-DoF all six actuators still need to be actuated and this results in higher operational cost and complexity as well as higher cost of the manipulator (Li and Bone, 2001). To overcome this, manipulators with lower mobility have received a lot of attention from the research community in the past few decades.

Planar motions, translational motions, cylindrical motions and spherical motions are important examples of motion tasks that require less than 6-DoF and are of-

ten necessary in industrial applications. A classic example of a parallel robot with lower mobility is the DELTA robot (Clavel, 1988). It is arguably the fastest 3-DoF robot in the world. It has three translational DoF and is used widely in industries for pick and place operations.

Another classic and widely known 3-DoF manipulator is the 3-RPS parallel platform which was adopted as a micromanipulator (Lee and Shah, 1988). This manipulator has cylindrical motions and has been used for precision operations like bracing and clamping (Johnson and Lee, 1989).

Agile Eye is a popular 3-DoF manipulator with spherical motions (Gosselin and Hamel, 1994). It mimics motion of the eyes and has very simple closed-form kinematic solutions and also a large singularity free workspace. The shape of the links however are a little too complicated to be machined to precision and hence this manipulator never found much practical acceptance in general.

The CaPaMan is a 3-DoF parallel manipulator with cylindrical motions (Ceccarelli, 1997). It has the advantage of actuation using rotary actuators placed at the bottom and simple closed-form forward kinematics. However it suffers from a serious draw-back due to the presence of passive sliding joints in the design. These joints increase the friction in the system and also the cantilevered position of the link containing the sliding joint reduces the load carrying capacity of the manipulator.

The 3-UPU manipulator is one of the most well known parallel manipulator with provision for reconfigurability(Gregorio, 2003)(Tsai, 1996). It can be mechanically rearranged to obtain all three translation DoF or all three rotation DoF. Inspite of this advantage of reconfigurability in the mechanism, this manipulator never gained prominence as the configuration with all three rotations, requires a mechanical arrangement with a number of axis of joints intersecting at a point. This kind of arrangement is very difficult to achieve in reality and it has been shown that any error in rearrangement causes the manipulator to exist in a singular state always with infinitesimal motion at all points.

1.1.2 Applications of parallel manipulators

- Parallel manipulators find application in flight simulators and automobile simulators for example Inmotion simulations, SEG motion technologies etc. are commercial ventures developing simulation solutions that are based on the Stewart platform manipulator (SPM).
- Parallel manipulators due to the high precision that they possess are excellent choice for photonics/optical fiber alignment.
- Even though the workspace is not very large, the inertia of moving parts is low and precision is high in parallel manipulators. Hence they find use in applications like surgical tool holders, antennae alignment systems and for handling and assembly operations.
- Due to their fast motion capability and high payload to self-weight ratio coupled with stiffness, they are used in high speed and high-precision milling machines (Johnson and Lee, 1989).
- High stiffness and precision make parallel manipulator appropriate for application in rehabilitation; example an ankle rehabilitation device developed at Italian Institute of Technology use a redundant 3-RPS manipulator to provide roll and pitch motion for ankle rehabilitation (Saglia Jody A. and G., 2008).

1.1.3 Demerits of some of the existing 3-DoF parallel manipulators

The existing 3-DoF manipulator have two main demerits:

1. **Limited to application-specific design:** The lower mobility platform manipulators cannot be used for distinct tasks which require different combinations of DoF. For instance, the DELTA is designed for only three translations along the X, Y, and Z axes (also known as surge, sway and heave respectively) , while the 3-RPS can produce only rotations about two horizontal axes (i.e., roll and pitch) and an independent translation about the vertical axis (i.e., heave). Neither of these manipulators can be used where three independent rotations (i.e., roll, pitch and yaw) are required. A manipulator such as Agile Eye, on the other hand, provides these three rotations; however, it cannot generate any translation.
2. **Use prismatic/linear actuators:** Most of these manipulators (e.g,3-RPS) inherit the prismatic actuators from the Stewart-Gough platform or introduce passive sliding joints in their structure (e.g, CaPaMan) which has several drawbacks:
 - Greater friction and design complexity inherent to passive sliding joints.

- Bulkier, slower linear actuators (compared with similarly priced rotary actuators).
- Lower payload-to-self-inertia (a measure of what it carries to what it inherently weighs) ratio, as most often the linear actuators form the legs of the manipulator, and thus need to be moved as the manipulator moves the payload.

1.2 Motivation

Parallel manipulators have several advantages over their serial counterparts, yet they are not widely used primarily because of the complexities in their design, kinematics and controls. On top of this, most lower-mobility parallel manipulators have prismatic actuators or passive sliding joints in them, which have the drawbacks discussed above. Also most of the existing designs are application-specific, which means they cannot readily be used for distinct tasks with different combinations of DoF.

Hence an attempt has been made to develop a novel design of a 3-DoF spatial parallel manipulator that has rotary actuators, and joints of rotary and spherical types only. In addition the manipulator can be mechanically reconfigured to two distinct configurations having different combinations of 3-DoF, which make the manipulator versatile and increase the scope for applications in various fields.

1.3 Objectives

The end objective is to design a novel 3-DoF manipulator, analyse and compare various features of this manipulator with existing manipulators and then develop a physical prototype to demonstrate some of these features. A number of intermediate objectives that are achieved to realise the final objective are:

- To develop kinematic formulations for the manipulator and establish the claims regarding DoF and reconfigurability.
- To perform singularity analysis in order to find out workspace limitations and singular regions.

- To study effect of variation of various design parameters on the performance indices of the manipulator.
- To design and fabricate a prototype of the manipulator based on certain design requirements.

1.4 Organisation of the thesis

This project has been organised into nine chapters. We present the conceptualisation of the design in Chapter 2. It is discussed as to how the MaPaMan was conceived of as an improvement of the 3-RPS manipulator. The various features of the new manipulator are highlighted and the terminologies associated with the new architecture are introduced here. In Chapter 3, we present the kinematic formulations for both the reconfigurable architectures of MaPaMan. A complete analytical analysis has been carried out to find out zeroth-order relationships between the active joint inputs and end-effector pose. The DoF of both MaPaMan-I and MaPaMan-II have been characterised and the partitioning into rotations and translations is established. Armed with the DoF information, a first-order kinematic analysis is carried out. In addition to this a new approach to faster forward kinematics is presented. This approach is numerical in nature but has implications in realtime control of the manipulator. We also present numerical examples to illustrate the mathematical procedures followed all throughout the Chapter. Chapter 4 discusses the singularities of MaPaMan-I. Contour plots are presented to help visualise the gain-type singularities in the manipulator. Dynamic formulation is briefly presented in Chapter 5¹. The effect of variation of design parameters on the performance of the manipulator is discussed in detail in Chapter 6. As there exists no “unique” metrics in literature as per our knowledge; certain new metrics are conceived of and presented in detail. Using a number of plots, the effect of design variables is studied on these metrics. A dynamic visualiser is developed to findout the overall nature of the variation in the metrics w.r.t. input angles upon varying the design parameters. A genetic algorithm based optimization frame-

¹Dynamic formulation is discussed in detail in a concurrent study carried out by Mehta (refthesis)

work was developed to find out the best set of dimensions required to obtain a manipulator with desired range of motion, when considering various constraints like, joint intersections, singularities etc.

In Chapter 7 we present the design and fabrication of two prototypes of MaPaMan-I. Design and manufacturing details of each component is presented in this Chapter. The details of the electronic components and software used is also presented. Chapter 8 deals with the details of interfacing the prototype with a joystick and an inertial measurement unit (IMU) to simulate offline trajectories as well as mimic realtime motions to demonstrate the accuracy and motion capabilities of the manipulator. In the final Chapter we conclude the research work and discuss the scope for furthering this work.

CHAPTER 2

Conceptual design

2.1 Introduction

The Madras Parallel Manipulator (MaPaMan) is a novel 3-DoF parallel manipulator that has been designed as an improvement over the widely studied 3-RPS parallel manipulator. It is not as close to the CaPaMan in design as much as its name is; however it draws some of its design inspirations from the same. MaPaMan utilises rotary actuators, and joints of rotary and spherical types only, without the need for a linear actuator or a passive prismatic joint. Rotary actuators are fixed at the stationary base of the manipulator, for easy replacement and also so as not to add their weight to the moving mass of the manipulator. It is readily reconfigurable to produce either roll-pitch-heave or roll-pitch-yaw by means of a simple change (in less than one minute) in its link arrangements.

2.2 Design evolution

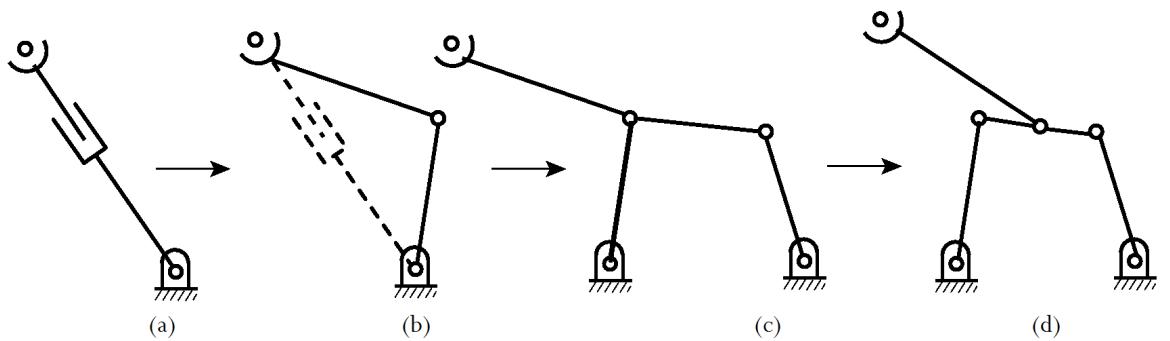


Figure 2.1: Evolution of MaPaMan from 3-RPS

The 3-RPS is a well-known parallel manipulator with 3-DoF and was reported by Lee and Shah in 1988. The degrees-of-freedom it possesses are roll, pitch

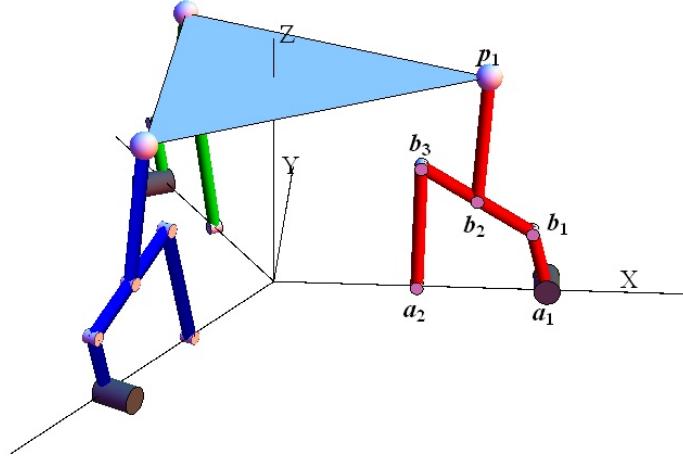
and heave. MaPaMan was designed as a design improvement over the 3-RPS manipulator. The design evolved from the 3-RPS manipulator in a stage by stage process as shown in Fig. 2.1. The stages of evolution are described as follows:

- Each leg of the 3-RPS manipulator has a revolute joint at the base, followed by a prismatic actuator and a spherical joint ending up at the end-effector as shown in Fig. 2.1(a).
- Prismatic actuators are typically slow and expensive. In order to avoid their presence in the design and still obtain the same kind of motion, a new mechanism was conceived of, wherein the prismatic actuators in each leg were replaced by two links connected by a revolute joint (see Fig. 2.1(b)). One revolute joint at the base in each leg were then actuated, to result in a mechanism with the same DoF as the 3-RPS nevertheless without the prismatic actuators or joints.
- The ground link was then replaced with a four-bar mechanism as shown in Fig. 2.1(c) . This was done because, a small force given at the crank gets magnified to a larger force at the rocker (e.g. Norton, 1999), hence improving the mechanical advantage of the mechanism.
- The link connected to the spherical joint at one end was then connected to a point on the coupler of the four-bar instead of the rocker at the other end. This link shall be henceforth referred to as the “strut ”. This was done so that the load carried by the strut could be distributed over to the crank and the rocker as opposed to the rocker taking all the load. The arrangement as shown in Fig. 2.1(d) offers greater stiffness to the manipulator in the predominant loading direction, i.e., the vertical.
- It was observed that if the strut is moved perpendicular to the plane of the four-bar, the 3-DoF change from roll, pitch and heave to roll, pitch and yaw. This reconfigurability is a novel feature not present in most parallel manipulators.

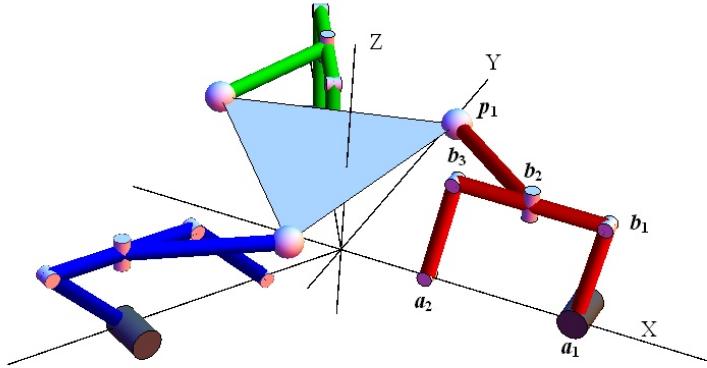
Thus MaPaMan was conceptualised as a manipulator with reduced friction, higher stiffness and reconfigurability.

2.3 Detailed architecture

The MaPaMan has three legs connected to an end-effector, which is in the form of a rigid triangle. Each leg mechanism of the manipulator consists of a four-bar attached to the fixed base (see Fig. 2.2), whose coupler (l_2) carries a revolute



(a) MaPaMan-I

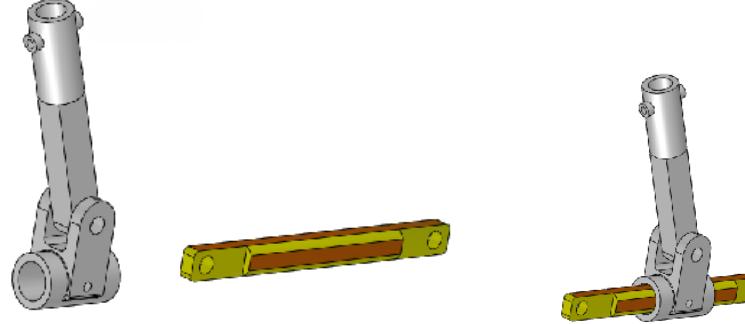


(b) MaPaMan-II

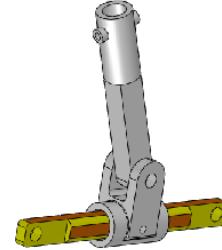
Figure 2.2: The two reconfigurable architectures of MaPaMan

joint (b_2), a strut (r) and a spherical joint (p), which connects the leg with the end-effector. The four-bars in each leg in turn are actuated by a motor located at a_1 . The axis of the revolute joint (b_2) is on a plane perpendicular to the plane of the four-bar. The three leg mechanisms are arranged such that the angle between any two is 120° , and the vertical planes containing them intersect at the Z-axis passing through the origin of the base plate.

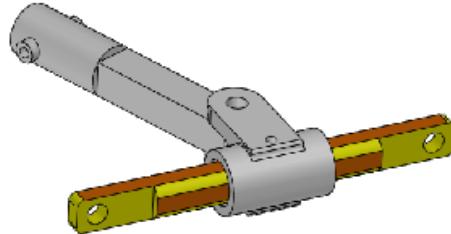
The orientation of the axis of rotation of the strut w.r.t. the coupler can be varied and locked at two discrete positions (see Fig. 2.3). If the axis of the revolute joint is perpendicular to the plane of the four-bar and parallel to the



(a) Geometry of the strut and coupler



(b) MaPaMan-I configuration



(c) MaPaMan-II configuration

Figure 2.3: Reconfigurable joint between the strut and the coupler

axis of the crank (l_1) as shown in Fig. 2.2(b), then the MaPaMan-I configuration with roll-pitch-heave DoF is obtained. If, on the other hand, the four-bar is a parallelogram linkage and the axis of revolute joint is perpendicular to the plane of the parallelogram and perpendicular to the axis of the crank (see Fig.2.2(b)), then the roll-pitch-yaw capable MaPaMan-II is obtained.

MaPaMan has 3 spherical joints, 12 rotary joints and 3 rotary actuators that by their geometrical construction ensure that the end-effector achieves different combinations of roll, pitch, yaw and heave (a total of three at a time). In addition, the manipulator has 7 design variables (length of crank, rocker, coupler, base, strut, top plate and position of the strut on the coupler) that can be varied suitably, thus broadening the scope of finding the most suitable manipulator for a given task, without changing the basic architecture.

2.4 Design features

The following are the key features of this manipulator:

- The manipulator has 3 rotary actuators, all located on the fixed base. This gives us the flexibility to change the actuators as per requirement without affecting the rest of the manipulator.
- As the motors remain stationary on the ground at all points of time, its inertial properties do not feature in the dynamics of the mechanism. Thus the ratio of moving inertia of the payload to that of the manipulator is high. An analysis has been carried out to verify this claim and is presented in Section 6.2.2.
- The manipulator can be easily reconfigured to get two different types of 3-DoF motions at the end-effector, by changing the axis of rotation of the rotary joint at the strut. Reconfigurability by just mechanically changing the orientation of a link is a huge advantage for the end-user as different types of motions can be obtained from the same device which would be very difficult to achieve otherwise. This versatility of the manipulator widens its scope for various applications.
- The four-bars at the base of the manipulator increase the stiffness of the overall mechanism and provide a mechanical advantage. They also provide us with the option of playing with a number of design variables (such as option of varying crank length, coupler length, rocker length etc), which are not available in the conventional 3-RPS manipulator. This broadens the scope of specialization for a variety of applications. A detailed study of effect of variation of different design variables on the properties of the manipulator have been illustrated in Section 6

2.5 Conclusion

In this Chapter we have presented the details of the conceptual design of a new 3-DoF spatial parallel manipulator, MaPaMan. The evolution of the manipulator from the design of an existing manipulator has been explained. The various terminologies associated with the new manipulator that would be used throughout this project has been introduced here. The various design features of this manipulator have been presented. These features are analysed in detail and many of the claims w.r.t. DoF and reconfigurability have been established in subsequent Chapters.

CHAPTER 3

Kinematics

3.1 Introduction

In this Chapter first a zeroth-order kinematic analysis is carried out for MaPaMan-I. Zeroth-order kinematics gives the relationship between the values of the active variables and the pose of the end-effector. This is studied under two main categories: the forward and the inverse kinematics. Following that a characterisation of DoF is carried out to establish that roll, pitch and heave are the degrees-of-freedom of the MaPaMan-I. Finally a first-order kinematic analysis is carried out. First-order kinematics gives the relationship between active joint rates and velocity of the end-effector. The entire exercise is then repeated for MaPaMan-II.

3.2 Zeroth-order kinematics of MaPaMan-I

This section deals with the zeroth-order kinematics of MaPaMan-I. Firstly forward kinematic formulations are developed, followed by an example to illustrate the scheme. Following this, inverse kinematic formulations are presented along with an example showing the various inverse kinematic configurations possible. Finally a geometric implication of MaPaMan-I is presented which acts as a precursor to the discussions presented in Section 3.4.2.

3.2.1 Forward kinematics

Given the values of the active variables, the process of finding the pose of the end-effector is called forward kinematics. Forward kinematics of parallel manipulators are generally more complicated to analyse as opposed to serial manipulators due

to the presence of closed loops in the architecture of the former. In the case of manipulators like Agile Eye, the geometry explicitly gives us the pose of the end-effector given the active joint positions Gosselin and Hamel (1994); however, in the case of more complicated manipulators, this approach can be very difficult or even impossible. In such cases, the forward kinematics problem is solved by constructing the loop-closure equations and solving them to finally obtain the pose of the end-effector (Ghosal, 2006). This is the approach adopted for MaPaMan-I. The variables defining the kinematics of MaPaMan-I are shown in Fig. 3.1. These

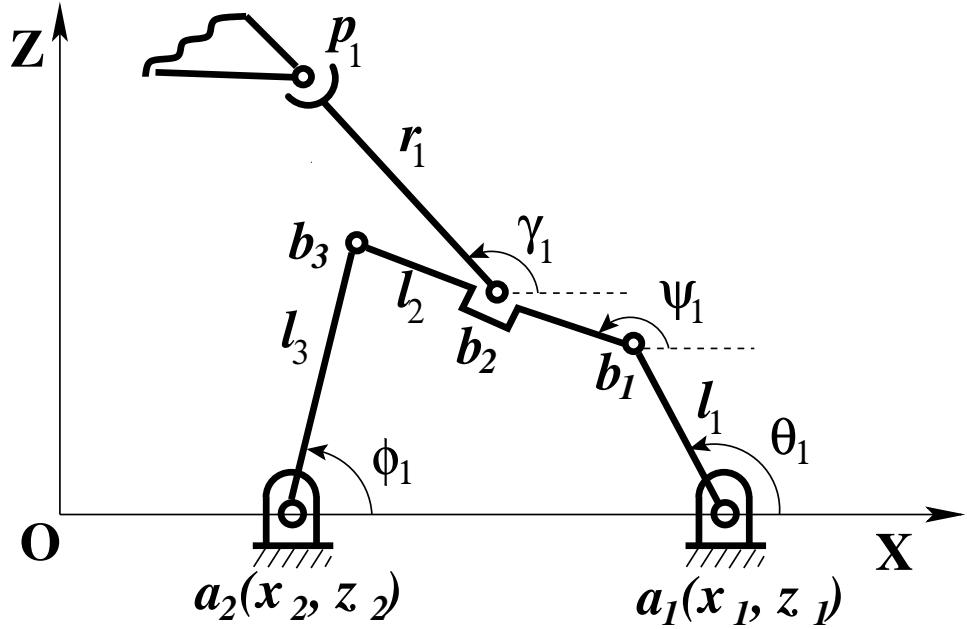


Figure 3.1: Kinematic diagram of a leg of MaPaMan-I

are grouped in two sets: the active variables attached to the actuated links given by

$$\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)^T, \quad (3.2.1)$$

and the non-actuated or passive variables given by

$$\boldsymbol{\phi} = (\phi_1, \phi_2, \phi_3)^T, \quad (3.2.2)$$

$$\boldsymbol{\psi} = (\psi_1, \psi_2, \psi_3)^T, \quad (3.2.3)$$

$$\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \gamma_3)^T. \quad (3.2.4)$$

The global reference frame is placed such that the XZ plane coincides with the plane of first four-bar as shown in Fig. 3.1. The strut is placed at point \mathbf{b}_2 on the coupler such that $\|\mathbf{b}_1 - \mathbf{b}_2\| = nl_2$.

The coordinates of \mathbf{p}_1 on the first leg are found in terms of the active variable θ_1 and passive variables ϕ_1 , ψ_1 and γ_1 :

$$\mathbf{p}_1 = \begin{pmatrix} l_1 \cos \theta_1 + nl_2 \cos \psi_1 + r \cos \gamma_1 + x_1 \\ 0 \\ l_1 \sin \theta_1 + nl_2 \sin \psi_1 + r \sin \gamma_1 + z_1 \end{pmatrix} \quad (3.2.5)$$

Making use of the symmetry in the geometry of the architecture, the coordinates of \mathbf{p}_2 are found by rotating the plane of the corresponding four-bar by 120° CCW about Z axis (by using $\mathbf{R}_z(\phi)^1$), so that it now coincides with the XZ plane; and the coordinates are found by a procedure similar to that used for finding \mathbf{p}_1 and then rotated back by 120° about Z axis.

$$\begin{aligned} \mathbf{p}_2 &= \mathbf{R}_z\left(\frac{2\pi}{3}\right) \begin{pmatrix} l_1 \cos \theta_2 + nl_2 \cos \psi_2 + r \cos \gamma_2 + x_1 \\ 0 \\ l_1 \sin \theta_2 + nl_2 \sin \psi_2 + r \sin \gamma_2 + z_1 \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{2}(-l_1 \cos \theta_2 - nl_2 \cos \psi_2 - r \cos \gamma_2 - x_1) \\ \frac{\sqrt{3}}{2}(l_1 \cos \theta_2 + nl_2 \cos \psi_2 + r \cos \gamma_2 + x_1) \\ l_1 \sin \theta_2 + nl_2 \sin \psi_2 + r \sin \gamma_2 + z_1 \end{pmatrix} \end{aligned} \quad (3.2.6)$$

¹ $\mathbf{R}_z(\phi) = \begin{pmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix}$ is the rotation matrix for CCW rotation about Z axis through an angle ϕ

The same process is repeated for finding \mathbf{p}_3 ; leading to:

$$\begin{aligned}\mathbf{p}_3 &= \mathbf{R}_z \left(\frac{4\pi}{3} \right) \begin{pmatrix} l_1 \cos \theta_3 + nl_2 \cos \psi_3 + r \cos \gamma_3 + x_1 \\ 0 \\ l_1 \sin \theta_3 + nl_2 \sin \psi_3 + r \sin \gamma_3 + z_1 \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{2}(-l_1 \cos \theta_3 - nl_2 \cos \psi_3 - r \cos \gamma_3 - x_1) \\ -\frac{\sqrt{3}}{2}(l_1 \cos \theta_3 + nl_2 \cos \psi_3 + r \cos \gamma_3 + x_1) \\ l_1 \sin \theta_3 + nl_2 \sin \psi_3 + r \sin \gamma_3 + z_1 \end{pmatrix} \quad (3.2.7)\end{aligned}$$

The points \mathbf{p}_1 , \mathbf{p}_2 and \mathbf{p}_3 form the three vertices of an equilateral triangle of side d and hence, they satisfy the geometric constraints:

$$\eta_1 \stackrel{\triangle}{=} (\mathbf{p}_1 - \mathbf{p}_2) \cdot (\mathbf{p}_1 - \mathbf{p}_2) - d_t^2 = 0 \quad (3.2.8)$$

$$\eta_2 \stackrel{\triangle}{=} (\mathbf{p}_2 - \mathbf{p}_3) \cdot (\mathbf{p}_2 - \mathbf{p}_3) - d_t^2 = 0 \quad (3.2.9)$$

$$\eta_3 \stackrel{\triangle}{=} (\mathbf{p}_3 - \mathbf{p}_1) \cdot (\mathbf{p}_3 - \mathbf{p}_1) - d_t^2 = 0 \quad (3.2.10)$$

These equations are also called the loop-closure equations, denoted by:

$$\boldsymbol{\eta} = (\eta_1, \eta_2, \eta_3)^T = \mathbf{0}. \quad (3.2.11)$$

The kinematics of each four-bar is independent of the motion in the rest of the

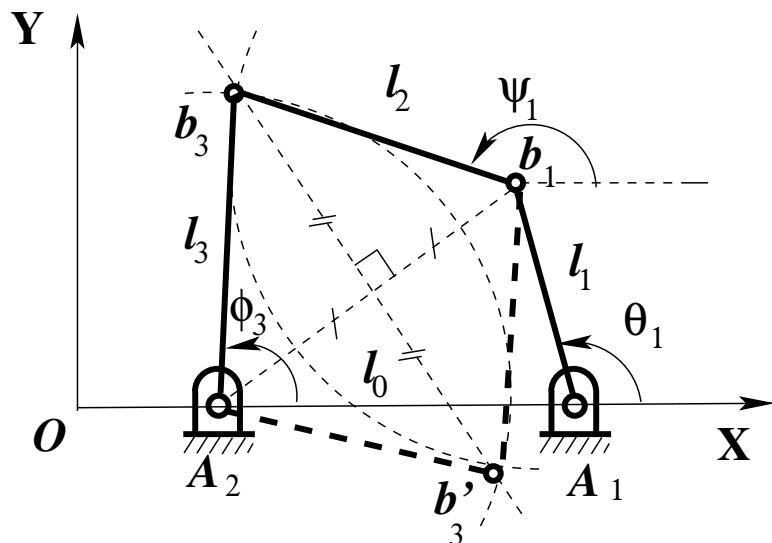


Figure 3.2: The two branches of solutions in a four-bar

mechanism and this property is utilised by solving the kinematics of the four-bar first, and then solving for the position of the strut and finally the end-effector. In the four-bars, the passive links have two distinct configurations for any given input as shown in Fig. 3.2. From geometry, we find two possible values for ψ_1 and ϕ_1 in terms of θ_1 . While both the solutions we obtain for the four-bar are valid, we choose the elbow up configuration due to physical constraints and obtain ψ_i and ϕ_i as:

$$\psi_i = \arccos\left(\frac{l_2^2 + l_{4i}^2 - l_3^2}{2l_2l_{4i}}\right) + \arccos\left(\frac{l_1^2 + l_{4i}^2 - l_0^2}{2l_1l_{4i}}\right) + \theta_i - \pi \quad (3.2.12)$$

$$\phi_i = \pi - \arccos\left(\frac{l_0^2 + l_{4i}^2 - l_1^2}{2l_0l_{4i}}\right) - \arccos\left(\frac{l_3^2 + l_{4i}^2 - l_2^2}{2l_3l_{4i}}\right), \text{ where} \quad (3.2.13)$$

$$l_{4i} = \sqrt{l_0^2 + l_1^2 - 2l_1l_0 \cos \theta_i}, \quad i = 1, 2, 3. \quad (3.2.14)$$

All the passive variables have been explicitly obtained in terms of active variables but for γ . Two approaches can be adopted to find γ :

1. Substitute all the numerical values of the known variables and then obtain solutions for the unknown variables using some numerical solver (e.g. G. Aguirre and Ottaviano, 2003).
2. Continue with the symbolic expressions and systematically eliminate unknown variables to finally end up in a univariate expression which can then be solved either analytically or numerically depending on the case (e.g. Husty, 1994).

The major drawback of the former approach is that it is not always guaranteed to yield solutions in all the cases. Sometimes the equations need to be simplified further or a few variables need to be eliminated before the numerical scheme becomes feasible. Also, numerical solvers typically introduce numerical inaccuracies in the solutions. The second approach, however, provides more accurate results and in some cases exact analytical solutions (e.g. Ghosal, 2006). Most of the times we end up in an analytical expression of a univariate polynomial, which is then solved to obtain the roots. Even if numerical root finding is applied at the final stage, the solutions are better than the first approach, as the coefficients are computed analytically and all the solutions are obtained. Sometimes the second method results in more solutions than expected as the elimination process can

introduce spurious roots (Cox *et al.*, 1991). These need to be taken care of on a case by case basis after all the roots have been obtained. A simple way to do this is to check all the solutions against the original loop-closure constraints, i.e, Eq.3.2.11.

An elimination scheme is developed to solve for γ from η as depicted in the following schematic:

$$\left. \begin{array}{l} \eta_1(\theta_1, \theta_2, \gamma_1, \gamma_2) = 0 \\ \eta_3(\theta_3, \theta_1, \gamma_3, \gamma_1) = 0 \\ \eta_2(\theta_2, \theta_3, \gamma_2, \gamma_3) = 0 \end{array} \right\} \xrightarrow{\times \gamma_1} \eta_4(\boldsymbol{\theta}, \gamma_2, \gamma_3) = 0 \Rightarrow \eta_6(\boldsymbol{\theta}, t_2, t_3) = 0 \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \xrightarrow{\times t_2} \eta_7(\boldsymbol{\theta}, t_3) = 0 \quad (3.2.15)$$

The functions η_1 and η_3 are both found to be linear in $\cos \gamma_1$ and $\sin \gamma_1$. Eq. (3.2.8), is of the form $\eta_1 = A_1 \cos \gamma_1 + B_1 \sin \gamma_1 + C_1 = 0$ and Eq. (3.2.10) is of the form $\eta_2 = A_2 \cos \gamma_1 + B_2 \sin \gamma_1 + C_2 = 0$. Solving for the sine and cosine of γ_1 linearly from these equations, we get

$$\begin{aligned} \cos \gamma_1 &= \frac{|BC|}{|AB|}, \quad \sin \gamma_1 = \frac{|CA|}{|AB|}, \text{ where} \\ |BC| &= (B_1 C_2 - B_2 C_1), \\ |AB| &= (A_1 B_2 - A_2 B_1), \\ |CA| &= (C_1 A_2 - C_2 A_1). \end{aligned} \quad (3.2.16)$$

Assuming $|AB| \neq 0$,

$$\gamma_1 = \text{atan2}(\sin \gamma_1, \cos \gamma_1) \quad (3.2.17)$$

where $\text{atan2}(\sin(\cdot), \cos(\cdot))$ represents the two-argument arc-tangent function.

Using the identity $\cos^2 \gamma_1 + \sin^2 \gamma_1 - 1 = 0$ and Eq. (3.2.16)), we eliminate γ_1 to obtain a new equation $\eta_4(\boldsymbol{\theta}, \gamma_2, \gamma_3) = 0$. The equations $\eta_2 = 0$ and $\eta_4 = 0$ are both trigonometric in γ_2 and γ_3 . At first an elimination strategy was followed that involved elimination of $\sin \gamma_2$ and $\cos \gamma_2$ simultaneously from η_2 and η_4 and

then finally obtaining a univariate equation in γ_3 . Upon converting into algebraic equations by substituting the trigonometric terms, $\cos \gamma_3$ and $\sin \gamma_3$ as follows (see, Ghosal (2006)):

$$\cos \gamma_3 = \frac{1 - t_3^2}{1 + t_3^2}, \quad \sin \gamma_3 = \frac{2t_3}{1 + t_3^2}, \quad t_3 = \tan\left(\frac{\gamma_3}{2}\right), \quad (3.2.18)$$

we obtain a 20 degree polynomial in t_3 after rationalisation. However numerical computations reveal that out of the 20 roots, 4 roots never lead to any physically meaningful solution. These roots arise due to the spurious factors that creep in at some stages of the elimination of the variables (Cox *et al.*, 1991). Hence an alternate strategy was developed to obtain a lower-degree univariate polynomial. The functions η_2 and η_4 were converted to algebraic functions η_5 and η_6 by using $t_2 = \tan \frac{\gamma_2}{2}$. The function η_5 is of the form $A_1 t_2^2 + B_1 t_2 + C_1$ and η_6 is of the form $A_2 t_2^4 + B_2 t_2^3 + C_2 t_2^2 + D_2 t_2 + E_2$.

A typical approach that could be followed at this stage was to compute the resultant of the two polynomials w.r.t. t_2 . However as we know the nature of the polynomials *a priori*, a faster way of solving the equations is employed. Treating η_6 and η_5 as polynomials in t_2 and dividing η_6 by η_5 yields a remainder that is at the most linear in t_2 as η_5 is quadratic in t_2 . Further the remainder must vanish when $\eta_5 = 0$ and $\eta_6 = 0$. Thus,

$$\text{Remainder}(\eta_6, \eta_5, t_2) = A_3 t_2 + B_3 = 0 \Rightarrow t_2 = -\frac{B_3}{A_3} \quad (3.2.19)$$

$$\Rightarrow \gamma_2 = 2 \operatorname{atan2}(-A_3, B_3). \quad (3.2.20)$$

Substituting t_2 obtained from Eq. (3.2.19) in η_5 , we eliminate t_2 and obtain a new equation unknown only in γ_3 :

$$\eta_7 \stackrel{\Delta}{=} A_1 A_3^2 + B_1 A_3 B_3 + C_1 B_3^2 = 0 \quad (3.2.21)$$

Converting to the algebraic form using $t_3 = \tan \frac{\gamma_3}{2}$. It is found that this equation is a polynomial in t_3 of order 16. The polynomial is 50MB in size², but upon

²Size refers to the number of bytes used internally by Mathematica. Note that 34KB is approximately equal to the amount of text in one A4 size page

simplification we obtain an expression 2MB in size. Note that all computations were done symbolically upto this point using the symbolic computation software **Mathematica**. As analytical solution cannot be found for general polynomials of degree greater than 4, roots of this polynomial equation are obtained numerically at this point. The real solutions are chosen and the others are discarded. After having found t_3 , γ_3 is found as

$$\gamma_3 = 2 \arctan(t_3). \quad (3.2.22)$$

Having computed γ_3 , γ_2 is obtained from Eq. (3.2.20) and γ_1 is obtained from Eq. (3.2.17).

The formulation is illustrated with the help of a numerical example³. The values of inputs ($\theta_1 = 0.8$, $\theta_2 = 1.4$, $\theta_3 = 1.1$), and all link length parameters ($l_0=100$, $l_1=75$, $l_2=100$, $l_3=75$, $n=0.5$, $r=100$, $d_t=160\sqrt{3}$) are substituted. At the base, coordinates of \mathbf{a}_1 and \mathbf{a}_2 are $(150, 0, 0)^T$ and $(50, 0, 0)^T$ respectively. Substitution of all the numbers in Eq. 3.2.21 leads to:

$$\begin{aligned} & 6.434t_3^{16} - 3.073t_3^{15} - 175.644t_3^{14} + 3.572t_3^{13} - 613.29t_3^{12} + 87.625t_3^{11} + 1682.99t_3^{10} + 71.051t_3^9 \\ & + 4652.29t_3^8 - 410.957t_3^7 - 6825.14t_3^6 - 319.268t_3^5 - 10128.1t_3^4 + 620.877t_3^3 \\ & + 11828.1t_3^2 + 153.542t_3 + 1561.43 = 0 \end{aligned} \quad (3.2.23)$$

Solving Eq. 3.2.23 for t_3 using “NSolve” in **Mathematica**, we get the following values of t_3

$$\begin{aligned} & -5.292, -1.285, -1.261, -1.224, -0.015 \pm 1.558i, -0.003 \pm 0.347i, \\ & 0.005 \pm 1.588i, 0.011 \pm 1.617i, 1.212, 1.255, 1.377, 5.702 \end{aligned}$$

Complex values of t_3 are naturally ignored and so are the negative real values as they do not represent physically meaningful configurations. The feasible values of t_3 are 1.212, 1.255, 1.377, 5.702, for which the corresponding γ_3 obtained are 1.762, 1.796, 1.885, 2.794 respectively.

³All values of angles are in radians and all lengths are in mm in this report unless mentioned otherwise explicitly

Following this, Eq. (3.2.20) and Eq. (3.2.17) are used to obtain γ_2 and γ_1 . It is observed that except for $\gamma_3 = 1.762$, γ_2 and γ_1 values obtained for other values of γ_3 are either negative or complex. Hence $\gamma_3 = 1.762$ is chosen and the corresponding values of $\gamma_2 = 1.605$ and $\gamma_1 = 2.004$ obtained.

Solving for ϕ_i and ψ_i ($i = 1, 2$ and 3) from Eq. (3.2.12) and Eq. (3.2.13) yields $\phi_1 = 0.8$, $\phi_2 = 1.4$, $\phi_3 = 1.1$, $\psi_1 = 3.141$, $\psi_2 = 3.141$, $\psi_3 = 3.141$.

A validation of the numerical solution is carried out by substituting the values of passive variables into the Eq. 3.2.11 . On substituting the values of $\boldsymbol{\gamma}$, $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ into $\boldsymbol{\eta}$ we get the residues as:

$$\boldsymbol{\eta} = (-4.667, -0.354, -4.470)^T \times 10^{-11} \quad (3.2.24)$$

Substituting all the passive variables to obtain \mathbf{p}_i ($i = 1, 2$ and 3) results in:

$$\mathbf{p}_1 = \begin{pmatrix} 108.178 \\ 0 \\ 149.103 \end{pmatrix}, \quad \mathbf{p}_2 = \begin{pmatrix} -54.562 \\ 94.504 \\ 178.846 \end{pmatrix}, \quad \mathbf{p}_3 = \begin{pmatrix} -55.296 \\ -95.775 \\ 169.193 \end{pmatrix} \quad (3.2.25)$$

This completes the forward position analysis of MaPaMan-I.

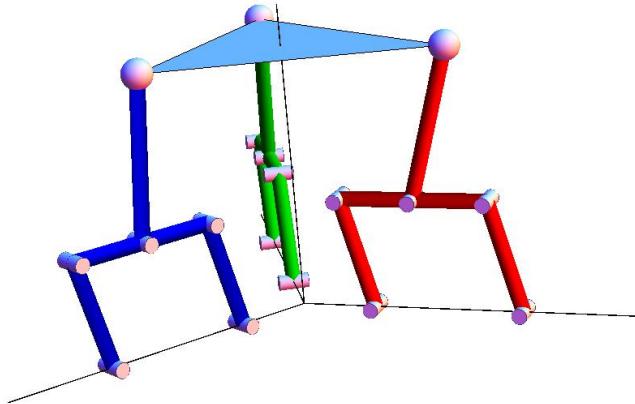


Figure 3.3: The pose of MaPaMan-I corresponding to the $\boldsymbol{\theta} = (0.8, 1.4, 1.1)^T$

3.2.2 Inverse kinematics

In the inverse kinematic problem, the orientation and position of the end-effector are known and the actuator angles that produce the specified pose of the end-effector are computed. The coordinates of \mathbf{p}_1 , \mathbf{p}_2 and \mathbf{p}_3 are known and θ_1 needs to be computed. From Fig.3.1,

$$\mathbf{b}_2 = \begin{pmatrix} l_1 \cos \theta_1 + nl_2 \cos \psi_1 + x_1 \\ 0 \\ l_1 \sin \theta_1 + nl_2 \sin \psi_1 + z_1 \end{pmatrix} \quad (3.2.26)$$

The distance between \mathbf{p}_1 and \mathbf{b}_2 is r , i.e.,

$$f_1 \triangleq (\mathbf{p}_1 - \mathbf{b}_2) \cdot (\mathbf{p}_1 - \mathbf{b}_2) - r^2 = 0 \quad (3.2.27)$$

Upon simplification, we obtain Eq.3.2.27 to be linear in terms of $\cos \psi_1$ and $\sin \psi_1$. Eq. (3.2.10) that was obtained in Section.3.2.1 is also an equation linear in $\cos \psi_1$ and $\sin \psi_1$. Eq. (3.2.10) and Eq. (3.2.27) are used simultaneously to eliminate ψ_1 and to obtain an equation $f_2 = 0$ purely in terms of θ_1 .

As the function f_2 is trigonometric in θ_1 , it is converted into an algebraic equation as shown in Eq. (3.2.18) to obtain $f_3 = 0$ as a polynomial equation in terms of variable t_1 , where

$$t_1 = \tan\left(\frac{\theta_1}{2}\right). \quad (3.2.28)$$

The function f_3 turns out to be a univariate polynomial in t_1 of degree 8. This polynomial is then solved numerically using the software **Mathematica** to obtain 8 roots for t_1 . The real solutions are chosen and one feasible branch of the solution is selected.

The point \mathbf{p}_2 is rotated by 120° CCW about Z axis to bring it to the XZ plane. After that the same procedure as that used for finding θ_1 , is adopted to find θ_2 . Similarly θ_3 is computed from \mathbf{p}_3 .

As an example, the same link lengths and position of \mathbf{p}_i ($i=1,2$ and 3) are taken as in Section 3.2.1.

$$\mathbf{p}_1 = \begin{pmatrix} 108.178 \\ 0 \\ 149.103 \end{pmatrix}, \quad \mathbf{p}_2 = \begin{pmatrix} -54.562 \\ 94.504 \\ 178.846 \end{pmatrix}, \quad \mathbf{p}_3 = \begin{pmatrix} -55.296 \\ -95.775 \\ 169.193 \end{pmatrix} \quad (3.2.29)$$

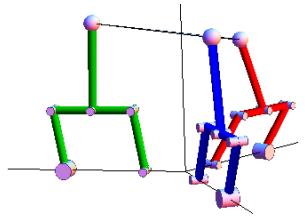
On solving for the actuator angles as described above, the following values are obtained.

$$\theta_1 = 0.8, 2.232 \quad (3.2.30)$$

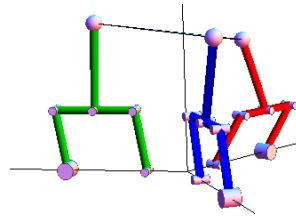
$$\theta_2 = 1.4, 1.639 \quad (3.2.31)$$

$$\theta_3 = 1.1, 1.916 \quad (3.2.32)$$

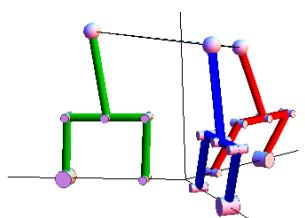
A total of 8 different combinations of input angles can result in the same pose as shown in Fig. 3.4. The inputs that were chosen in the section on forward kinematics was $\theta_1 = 0.8$, $\theta_2 = 1.4$ and $\theta_3 = 1.1$, which is clearly a subset of what is obtained (see Fig. 3.4(a)).



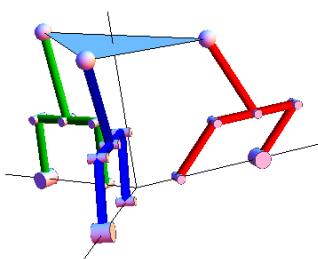
(a) The pose of MaPaMan-I corre-
sponding to $\theta = (0.8, 1.4, 1.1)^T$



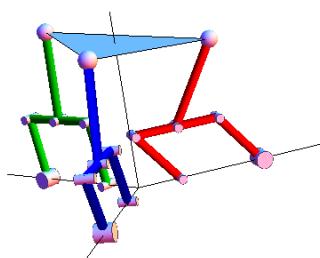
(b) The pose of MaPaMan-I corre-
sponding to $\theta = (0.8, 1.4, 1.916)^T$



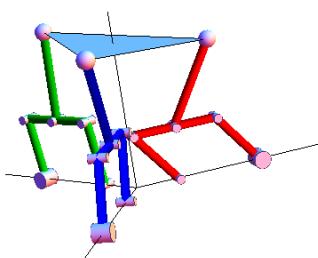
(c) The pose of MaPaMan-I corre-
sponding to $\theta = (0.8, 1.639, 1.1)^T$



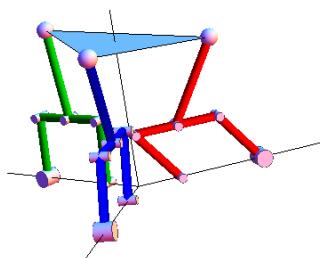
(d) The pose of MaPaMan-I corre-
sponding to $\theta = (0.8, 1.639, 1.916)^T$



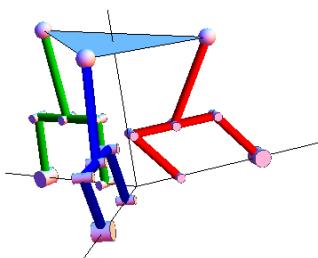
(e) The pose of MaPaMan-I corre-
sponding to $\theta = (2.232, 1.4, 1.1)^T$



(f) The pose of MaPaMan-I corre-
sponding to $\theta = (2.232, 1.4, 1.916)^T$



(g) The pose of MaPaMan-I corre-
sponding to $\theta = (2.232, 1.639, 1.916)^T$



(h) The pose of MaPaMan-I corre-
sponding to $\theta = (2.232, 1.639, 1.1)^T$

Figure 3.4: Eight inverse kinematic configurations of MaPaMan-I

3.3 Geometrical implication of the architecture of MaPaMan-I

An interesting observation is made at this point, which arises purely from the geometry of the architecture. We represent the orientation of the end-effector in terms of Rodrigue's parameters c_1, c_2, c_3 (e.g. Selig, 1996) and represent the position of its centroid \mathbf{p}_c using (x_c, y_c, z_c) . Rodrigue's parameters were used as they are algebraic in nature as opposed to the trigonometric Euler angles. By definition, Rodrigue's parameters $(c_1, c_2, c_3) = (k_x \tan \frac{\phi}{2}, k_y \tan \frac{\phi}{2}, k_z \tan \frac{\phi}{2},)$; where (k_x, k_y, k_z) is the unit vector representing the instantaneous axis of rotation and ϕ is the instantaneous tilt about that axis.

MaPaMan-I is a 3-DoF manipulator and hence we should require only three parameters at any point to describe the pose of the end-effector. Thus we attempt to find the relationship between $c_1, c_2, c_3, x_c, y_c, z_c$. The coordinates of the end-effector is found in a frame fixed to itself with the origin at its centroid, normal oriented with the Z-axis and the X-axis oriented along \mathbf{p}_1 :

$$\mathbf{p}_1 = \begin{pmatrix} r_t \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{p}_2 = \mathbf{R}_z \left(\frac{2\pi}{3} \right) \mathbf{p}_1 = \begin{pmatrix} -\frac{r_t}{2} \\ \frac{\sqrt{3}r_t}{2} \\ 0 \end{pmatrix}, \quad \mathbf{p}_3 = \mathbf{R}_z \left(\frac{4\pi}{3} \right) \mathbf{p}_1 = \begin{pmatrix} -\frac{r_t}{2} \\ -\frac{\sqrt{3}r_t}{2} \\ 0 \end{pmatrix} \quad (3.3.1)$$

The transformed coordinates of the end-effector can be written as:

$$\mathbf{p}_{it} = \mathbf{R}\mathbf{p}_i + \mathbf{p}_c, \quad (3.3.2)$$

where, \mathbf{R} is the rotation matrix in terms of c_1, c_2, c_3 defined as

$$\mathbf{R} = \frac{1}{1 + c_1^2 + c_2^2 + c_3^2} \begin{pmatrix} c_1^2 - c_2^2 - c_3^2 + 1 & 2c_1c_2 - 2c_3 & 2(c_2 + c_1c_3) \\ 2(c_1c_2 + c_3) & 1 - c_1^2 + c_2^2 - c_3^2 & -2(c_1 - c_2c_3) \\ 2(c_1c_3 - c_2) & 2(c_1 + c_2c_3) & 1 - c_1^2 - c_2^2 + c_3^2 \end{pmatrix} \quad (3.3.3)$$

The transformed coordinates obtained are,

$$\boldsymbol{p}_{1t} = \frac{1}{c_\Delta} \begin{pmatrix} (c_1^2 - c_2^2 - c_3^2 + 1) r_t + (1 + c_1^2 + c_2^2 + c_3^2) x_c \\ 2(c_1 c_2 + c_3) r_t + (1 + c_1^2 + c_2^2 + c_3^2) y_c \\ 2(c_1 c_3 - c_2) r_t + (1 + c_1^2 + c_2^2 + c_3^2) z_c \end{pmatrix} \quad (3.3.4)$$

$$\boldsymbol{p}_{2t} = \frac{1}{c_\Delta} \begin{pmatrix} \frac{r_t}{2} (\sqrt{3}(2c_1 c_2 - 2c_3) - (c_1^2 - c_2^2 - c_3^2 + 1)) + (1 + c_1^2 + c_2^2 + c_3^2) x_c \\ \frac{r_t}{2} (-2(c_1 c_2 + c_3) + \sqrt{3}(-c_1^2 + c_2^2 - c_3^2 + 1)) + (1 + c_1^2 + c_2^2 + c_3^2) y_c \\ r_t ((c_1 c_3 - c_2) + \sqrt{3}(c_1 + c_2 c_3)) + (1 + c_1^2 + c_2^2 + c_3^2) z_c \end{pmatrix} \quad (3.3.5)$$

$$\boldsymbol{p}_{3t} = \frac{1}{c_\Delta} \begin{pmatrix} -\frac{r_t}{2} (\sqrt{3}(2c_1 c_2 - 2c_3) + (c_1^2 - c_2^2 - c_3^2 + 1)) + (1 + c_1^2 + c_2^2 + c_3^2) x_c \\ -\frac{r_t}{2} (2(c_1 c_2 + c_3) + \sqrt{3}(-c_1^2 + c_2^2 - c_3^2 + 1)) + (1 + c_1^2 + c_2^2 + c_3^2) y_c \\ -r_t ((c_1 c_3 - c_2) + \sqrt{3}(c_1 + c_2 c_3)) + (1 + c_1^2 + c_2^2 + c_3^2) z_c \end{pmatrix}, \quad (3.3.6)$$

where $c_\Delta = (1 + c_1^2 + c_2^2 + c_3^2)$.

From the geometry of the manipulator, \boldsymbol{p}_1 is constrained to move in the XZ plane (see Fig. 3.1), i.e., its y-coordinate must be zero. Hence

$$f_1 \triangleq (\boldsymbol{p}_{1t} \cdot \boldsymbol{e}_y) = 0, \quad \text{where } \boldsymbol{e}_y = (0, 1, 0)^T. \quad (3.3.7)$$

Similarly \boldsymbol{p}_2 and \boldsymbol{p}_3 are constrained to move in their respective planes, hence

$$f_2 \triangleq \left(\boldsymbol{R}_z \left(\frac{-2\pi}{3} \right) \boldsymbol{p}_{2t} \right) \cdot \boldsymbol{e}_y = 0, \quad (3.3.8)$$

$$f_3 \triangleq \left(\boldsymbol{R}_z \left(\frac{-4\pi}{3} \right) \boldsymbol{p}_{3t} \right) \cdot \boldsymbol{e}_y = 0, \quad (3.3.9)$$

Upon simplification we obtain,

$$f_1 \triangleq 2c_2c_1r_t + 2c_3r_t + c_1^2y_c + c_2^2y_c + c_3^2y_c + y_c = 0 \quad (3.3.10)$$

$$\begin{aligned} f_2 &\triangleq \sqrt{3}c_1^2r_t - 2c_2c_1r_t - \sqrt{3}c_2^2r_t + 4c_3r_t - \sqrt{3}c_1^2x_c \\ &- \sqrt{3}c_2^2x_c - \sqrt{3}c_3^2x_c - \sqrt{3}x_c - c_1^2y_c - c_2^2y_c - c_3^2y_c - y_c = 0 \end{aligned} \quad (3.3.11)$$

$$\begin{aligned} f_3 &\triangleq -\sqrt{3}c_1^2r_t - 2c_2c_1r_t + \sqrt{3}c_2^2r_t + 4c_3r_t + \sqrt{3}c_1^2x_c \\ &+ \sqrt{3}c_2^2x_c + \sqrt{3}c_3^2x_c + \sqrt{3}x_c - c_1^2y_c - c_2^2y_c - c_3^2y_c - y_c = 0 \end{aligned} \quad (3.3.12)$$

The function f_1 is linear in y_c and hence we obtain from Eq.3.3.7 $y_c = -\frac{2(c_1c_2r_t+c_3r_t)}{1+c_1^2+c_2^2+c_3^2}$.

Note that the denominator of y_c is always non-zero. The value of y_c obtained is substituted in Eq.3.3.8 and Eq.3.3.9 to obtain $f_4 = 0$ and $f_5 = 0$. Upon performing $f_4 + f_5$, we obtain $12c_3r_t = 0$. This implies $c_3 = 0$, which further leads to:

$$y_c = -\frac{2c_1c_2r_t}{1 + c_1^2 + c_2^2}. \quad (3.3.13)$$

Substituting $c_3 = 0$ in $f_4 = 0$ and solving for x_c , we obtain:

$$x_c = r_t \frac{c_1^2 - c_2^2}{1 + c_1^2 + c_2^2}. \quad (3.3.14)$$

We observe that at all instances $c_3 = 0$ implying $k_z = 0$. Therefore at any moment, the instantaneous axis of rotation cannot have a component along the Z axis; thus it cannot exhibit yaw motion. Position coordinates x_c and y_c have been obtained explicitly in terms of c_1 and c_2 , but z_c is still independent. Hence using the geometry of the manipulator, we have been able to explicitly find out a relationship between x_c, y_c and c_1, c_2 . Also we were able to establish that $c_3 = 0$ and z_c is independent of all other parameters. This result is quite fascinating as the geometry directly hints at the fact that MaPaMan-I has two rotation DoF and one translation DoF about Z-axis(A claim that is validated in Section.3.4.2). Thus this analysis shows that the 3-dimensional task-space of MaPaMan-I can be explicitly parameterised in terms of $\mathbf{r}_{ts} = (c_1, c_2, z_c)^T$.

3.4 First-order kinematics of MaPaMan-I

First-order kinematics deals with finding the relationships between active joint rates and the end-effector velocities. Given the active joint rates finding out the end-effector velocity is called the first-order forward kinematics. Computing the active joint rates given the end-effector velocity is called first-order inverse kinematics.

3.4.1 Forward kinematics

The end-effector has linear as well as angular velocities given by \mathbf{v}_c and $\boldsymbol{\omega}$. The linear velocity is computed as follows:

$$\frac{d\boldsymbol{\eta}(\mathbf{q})}{dt} = \mathbf{0}, \mathbf{v}_c = \frac{d\mathbf{p}_c}{dt} = \frac{\partial \mathbf{p}_c}{\partial \boldsymbol{\theta}} \dot{\boldsymbol{\theta}} \quad (3.4.1)$$

$$= \mathbf{J}_v \dot{\boldsymbol{\theta}}, \text{ where } \mathbf{J}_v = \frac{\partial \mathbf{p}_c}{\partial \boldsymbol{\theta}}. \quad (3.4.2)$$

The coordinates of the centroid of the end-effector $\mathbf{p}_c = \frac{1}{3}(\mathbf{p}_1 + \mathbf{p}_2 + \mathbf{p}_3)$. The points $\mathbf{p}_i, i=1, 2, 3$ are obtained in Eq.3.2.5, Eq.3.2.6 and Eq.3.2.7.

In order to compute the angular velocity, we first find out the rotation matrix of the end-effector in a space fixed reference frame:

$$\mathbf{R} = (\mathbf{X} \quad | \quad \mathbf{Y} \quad | \quad \mathbf{Z}) \quad (3.4.3)$$

where, $\mathbf{X} = \frac{\mathbf{p}_1 - \mathbf{p}_2}{\|\mathbf{p}_1 - \mathbf{p}_2\|}$, $\mathbf{Z} = \frac{(\mathbf{p}_1 - \mathbf{p}_2) \times (\mathbf{p}_1 - \mathbf{p}_3)}{\|(\mathbf{p}_1 - \mathbf{p}_2) \times (\mathbf{p}_1 - \mathbf{p}_3)\|}$ and $\mathbf{Y} = \mathbf{Z} \times \mathbf{X}$. The angular velocity of the end-effector $\boldsymbol{\omega}$ is obtained as follows:

$$\boldsymbol{\Omega} = \dot{\mathbf{R}} \mathbf{R}^T, \quad (3.4.4)$$

$$\boldsymbol{\Omega} = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \quad (3.4.5)$$

The matrix $\boldsymbol{\Omega}$ is skew-symmetric and the corresponding vector obtained from it is $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)^T$ (Ghosal, 2006). Once we obtain $\boldsymbol{\omega}$, we extract \mathbf{J}_ω from it such

that:

$$\boldsymbol{\omega} = \mathbf{J}_{\boldsymbol{\omega}} \cdot \dot{\boldsymbol{\theta}} \quad (3.4.6)$$

The Jacobian matrix $\mathbf{J}_{\boldsymbol{\omega}}$ is called the angular velocity Jacobian of the end-effector. Thus given the rates of motion of the actuators, we can compute the linear and angular velocities of the end-effector.

3.4.2 Characterising the degree-of-freedom of MaPaMan-I

In this section, we validate the fact that MaPaMan-I has roll, pitch and heave DoF. Bandyopadhyay (2009) presented a partitioning of DoF based on the first-order properties of motion. The same approach is followed here to establish the nature of DoF of MaPaMan-I. A rigid body in space typically has 6-DoF; of which 3 are rotations (roll, pitch and yaw) and 3 are translations (surge, sway and heave). When a manipulator has lower mobility, for example 3-DoF, it could be a combination of 2 rotations and 1 translation (2R-1T), or 1 rotation and 2 translations (1R-2T) or all 3 rotations (3R-0T) or all 3 translations(0R-3T). To find out the partitions of the DoF in a manipulator, we need to find out the ranks of two matrices \mathbf{g} and \mathbf{g}_V . These matrices themselves are defined as follows (Bandyopadhyay, 2009):

$$\begin{aligned}\mathbf{g} &= \mathbf{J}_{\boldsymbol{\omega}}^T \mathbf{J}_{\boldsymbol{\omega}} \\ \mathbf{g}_V &= \mathbf{J}_V^T \mathbf{J}_V, \quad \mathbf{J}_V = \mathbf{J}_v \mathbf{X},\end{aligned}$$

where

$$\mathbf{X} \in Null(\mathbf{g}).$$

Note that \mathbf{J}_v and $\mathbf{J}_{\boldsymbol{\omega}}$ are velocity Jacobian matrices defined in Section 3.4.1. For the sake of illustration, we take the configuration of MaPaMan-I given in Table

B.1. For $\boldsymbol{\theta} = (0.8, 1.4, 1.1)^T$, we obtain the following:

$$\begin{aligned}\mathbf{J}_v &= \begin{pmatrix} 0.042 & -0.002 & -0.021 \\ 0.014 & -0.008 & 0.017 \\ 0.275 & 0.049 & 0.156 \end{pmatrix}, \\ \mathbf{J}_\omega &= \begin{pmatrix} 0.510 & -0.050 & -0.163 \\ 0.001 & 0.079 & -0.256 \\ -0.013 & -0.005 & 0.024 \end{pmatrix}, \\ \mathbf{g} &= \begin{pmatrix} 0.261 & -0.025 & -0.084 \\ -0.025 & 0.009 & -0.012 \\ -0.084 & -0.012 & 0.093 \end{pmatrix}, \\ \text{Null}(\mathbf{g}) &= \begin{pmatrix} 0.186 \\ 0.938 \\ 0.291 \end{pmatrix} \\ \mathbf{g}_v &= 0.02\end{aligned}$$

The rank of \mathbf{g} turns out to be 2 and the rank of \mathbf{g}_v is 1, hence validating the 2R-1T nature of MaPaMan-I.

Though this method tells us that MaPaMan-I has 2R-1T DoF, it still does not provide any information as to which among roll, pitch and yaw form a part of the 2R DoF and which among surge, sway and heave form the 1T DoF. However the geometrical analysis carried out in Section 3.3 established that no yaw motion is possible, hence the 2R DoF have to be roll and pitch. We also observed that z_c was independent of c_1 and c_2 as well as x_c and y_c and hence the 1T DoF has to be about Z axis(heave). Hence MaPaMan-I has roll, pitch and heave as its 3-DoF. As neither \mathbf{J}_v nor \mathbf{J}_ω have full-rank, we cannot obtain $\dot{\boldsymbol{\theta}}$ given \mathbf{v}_c or $\boldsymbol{\omega}$; which would be required in the first-order inverse kinematics problem. Therefore, we find a composite Jacobian in the task-space which is of full-rank except at singularities. This requires us to find the velocity of the end-effector in its task space (\mathbf{r}_{ts}).

$$\mathbf{r}_{ts} = (c_1, c_2, z_c)^T, \quad (3.4.7)$$

We find the coordinates of the end-effector in terms of c_1 , c_2 and z_c as shown in Section 3.3. We know the coupler point \mathbf{b}_2 in each leg i.e. \mathbf{b}_{21} , \mathbf{b}_{22} and \mathbf{b}_{23} respectively in terms of $\boldsymbol{\theta}_i$ and $\boldsymbol{\psi}_i$, $i = 1, 2, 3$:

$$\mathbf{b}_{21} = \begin{pmatrix} x_1 + l_1 \cos \theta_1 + nl_2 \cos \psi_1 \\ 0 \\ l_1 \sin \theta_1 + nl_2 \sin \psi_1 \end{pmatrix} \quad (3.4.8)$$

$$\mathbf{b}_{22} = \begin{pmatrix} \frac{1}{2}(-l_1 \cos \theta_2 - n\frac{l_2}{2} \cos \psi_2 - x_1) \\ \frac{\sqrt{3}}{2}(l_1 \cos \theta_2 + n\frac{l_2}{2} \cos \psi_2 + x_1) \\ l_1 \sin \theta_2 + nl_2 \sin \psi_2 + z_1 \end{pmatrix} \quad (3.4.9)$$

$$\mathbf{b}_{23} = \begin{pmatrix} \frac{1}{2}(-l_1 \cos \theta_3 - n\frac{l_2}{2} \cos \psi_3 - x_1) \\ -\frac{\sqrt{3}}{2}(l_1 \cos \theta_3 + n\frac{l_2}{2} \cos \psi_3 + x_1) \\ l_1 \sin \theta_3 + nl_2 \sin \psi_3 + z_1 \end{pmatrix}. \quad (3.4.10)$$

Noting that the length of strut is r we frame three constraint equations (ref fig):

$$f_1 \triangleq (\mathbf{b}_{21} - \mathbf{p}_1) \cdot (\mathbf{b}_{21} - \mathbf{p}_1) - r^2 = 0 \quad (3.4.11)$$

$$f_2 \triangleq (\mathbf{b}_{22} - \mathbf{p}_2) \cdot (\mathbf{b}_{22} - \mathbf{p}_2) - r^2 = 0 \quad (3.4.12)$$

$$f_3 \triangleq (\mathbf{b}_{23} - \mathbf{p}_3) \cdot (\mathbf{b}_{23} - \mathbf{p}_3) - r^2 = 0 \quad (3.4.13)$$

and

$$\mathbf{f}(\mathbf{r}_{ts}, \boldsymbol{\theta}, \boldsymbol{\psi}) = (f_1, f_2, f_3)^T = \mathbf{0} \quad (3.4.14)$$

Differentiating Eq.3.4.14 w.r.t. time we obtain:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{r}_{ts}} \dot{\mathbf{r}}_{ts} + \frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}} \dot{\boldsymbol{\theta}} + \frac{\partial \mathbf{f}}{\partial \boldsymbol{\psi}} \dot{\boldsymbol{\psi}} = \mathbf{0} \quad (3.4.15)$$

Rate of motion in the task space is thus obtained as:

$$\dot{\mathbf{r}}_{ts} = (\mathbf{J}_{f\mathbf{r}_{ts}})^{-1} (\mathbf{J}_{f\boldsymbol{\theta}} \dot{\boldsymbol{\theta}} + \mathbf{J}_{f\boldsymbol{\psi}} \dot{\boldsymbol{\psi}}), \quad (3.4.16)$$

where,

$$\mathbf{J}_{f\theta} = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}}, \quad \mathbf{J}_{f\psi} = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\psi}}, \quad \mathbf{J}_{fr_{ts}} = \frac{\partial \mathbf{f}}{\partial \mathbf{r}_{ts}}. \quad (3.4.17)$$

The Eq. (3.4.16) gives us the relationship of the velocity of the end-effector in the task-space with the joint rates. However, given the active joint rates, we need to perform the intermediate step of calculating the passive joint rates $\dot{\boldsymbol{\psi}}$ to get the end-effector velocity. We obtain $\dot{\boldsymbol{\psi}}$ in terms of $\boldsymbol{\theta}$ in Eq. (3.4.14). Differentiating $\boldsymbol{\psi}$ with respect to time to get:

$$\dot{\boldsymbol{\psi}} = \mathbf{J}_{\psi\theta}\dot{\boldsymbol{\theta}}, \quad \mathbf{J}_{\psi\theta} = \frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{\theta}} \quad (3.4.18)$$

Substituting $\dot{\boldsymbol{\psi}}$ from Eq. (3.4.18) in Eq. (3.4.16) we get:

$$\dot{\mathbf{r}}_{ts} = \begin{pmatrix} \dot{c}_1 \\ \dot{c}_2 \\ \dot{z}_c \end{pmatrix} = (\mathbf{J}_{fr_{ts}})^{-1}(\mathbf{J}_{f\theta} + \mathbf{J}_{f\psi}\mathbf{J}_{\psi\theta})\dot{\boldsymbol{\theta}} = \mathbf{J}_{r_{ts}\theta}\dot{\boldsymbol{\theta}}, \quad (3.4.19)$$

(3.4.20)

where $\mathbf{J}_{r_{ts}\theta} = (\mathbf{J}_{fr_{ts}})^{-1}(\mathbf{J}_{f\theta} + \mathbf{J}_{f\psi}\mathbf{J}_{\psi\theta})$, is a composite velocity Jacobian matrix. Given $\dot{\boldsymbol{\theta}}$, we can now obtain \dot{c}_1 , \dot{c}_2 , \dot{z}_c . All the Jacobian matrices appearing above can be computed at any configuration after the position kinematics problem has been solved.

Sample results of the first-order forward kinematics

Example 1: Consider the configuration of the manipulator given by: $\boldsymbol{\theta} = (0.8, 1.4, 1.1)^T$. From forward kinematics (3.4.1),

$$\boldsymbol{\psi} = \begin{pmatrix} \pi \\ \pi \\ \pi \end{pmatrix}, \quad \boldsymbol{\phi} = \begin{pmatrix} 0.8 \\ 1.4 \\ 1.1 \end{pmatrix}, \quad \boldsymbol{\gamma} = \begin{pmatrix} 2.004 \\ 1.605 \\ 1.796 \end{pmatrix} \quad (3.4.21)$$

With this configuration, if the active links are driven with the initial velocity, $\dot{\boldsymbol{\theta}} = (0.100, -0.200, -0.100)^T$, using Eq. (3.4.19) we calculate the velocity of the end-effector in the task space:

$$\begin{pmatrix} \dot{c}_1 \\ \dot{c}_2 \\ \dot{z}_c \end{pmatrix} = \begin{pmatrix} 0.002 \\ -0.049 \\ -0.003 \end{pmatrix}$$

3.4.3 Inverse kinematics

A composite Jacobian was formulated in Section 3.4.2 that relates the velocity of the end-effector in its task-space with the active joint rates. This Jacobian is of full-rank except at singularities. The active joint rates are computed from the velocity of the end-effector in the task-space as follows:

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_{\mathbf{r}_{ts}\boldsymbol{\theta}}^{-1} \dot{\mathbf{r}}_{ts}. \quad (3.4.22)$$

Results of first-order inverse kinematics

Given the same inputs as in the example of section (3.4.2) and $\dot{\mathbf{r}}_{ts} = (0.002, -0.049, -0.003)^T$, the active links angular velocity can be calculated using Eq. (3.4.22)

$$\dot{\boldsymbol{\theta}} = (0.09, -0.20, -0.09)^T.$$

The $\dot{\boldsymbol{\theta}}$ obtained matches with the values used in the Section 3.4.2, hence validating the first-order kinematics.

3.5 Zeroth-order kinematics of MaPaMan-II

In this section, forward kinematic formulations of MaPaMan-II are presented followed by inverse kinematic formulations. An example is also provided to illustrate both the formulations.

Forward kinematics

The forward kinematic formulation of MaPaMan-II is very similar to that of MaPaMan-I. Only the expressions for \mathbf{p}_i ($i=1, 2, 3$) are different due to the difference in the geometry of the manipulator. The coordinates of the point \mathbf{p}_1 on

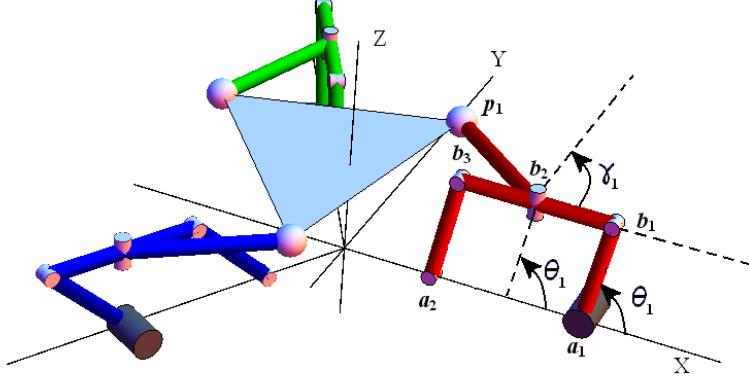


Figure 3.5: Kinematic representation of a leg of MaPaMan-II

the first leg are found in terms of the active variable θ_1 and passive variables ϕ_1 , ψ_1 and γ_1 . The global reference frame is placed such that the XZ plane coincides with the plane of the corresponding four-bar (see Fig.3.5).

As the four-bar in the case of MaPaMan-II is a parallelogram, $\phi = \theta$ and $\psi = (\pi, \pi, \pi)^T$. Hence γ are the only unknown passive variables to be computed. The point \mathbf{p}_1 is obtained as follows from geometry of the manipulator:

$$\mathbf{p}_1 = \begin{pmatrix} l_1 \cos \theta_1 + r \cos \gamma_1 + \frac{1}{2}(x_1 + x_2) \\ r \sin \gamma_1 \\ l_1 \sin \theta_1 \end{pmatrix}. \quad (3.5.1)$$

Using the symmetry in the architecture, we find points \mathbf{p}_2 and \mathbf{p}_3 as described in Section 3.2.1.

$$\mathbf{p}_2 = \begin{pmatrix} \frac{1}{2} \left(-l_1 \cos \theta_2 - r \cos \gamma_2 - \frac{1}{2}(x_1 + x_2) \right) - \frac{1}{2}\sqrt{3}r \sin \gamma_2 \\ \frac{1}{2}\sqrt{3} \left(l_1 \cos \theta_2 + r \cos \gamma_2 + \frac{1}{2}(x_1 + x_2) \right) - \frac{1}{2}r \sin \gamma_2 \\ l_1 \sin \theta_2 \end{pmatrix}, \quad (3.5.2)$$

$$\mathbf{p}_3 = \begin{pmatrix} \frac{1}{2} \left(-l_1 \cos \theta_3 - r \cos \gamma_3 - \frac{1}{2}(x_1 + x_2) \right) + \frac{1}{2}\sqrt{3}r \sin \gamma_3 \\ -\frac{1}{2}\sqrt{3} \left(l_1 \cos \theta_3 + r \cos \gamma_3 + \frac{1}{2}(x_1 + x_2) \right) - \frac{1}{2}r \sin \gamma_3 \\ l_1 \sin \theta_3 \end{pmatrix}. \quad (3.5.3)$$

After finding \mathbf{p}_1 , \mathbf{p}_2 and \mathbf{p}_3 , the same procedure as that adapted for MaPaMan-I in Section 3.2.1) is followed and all the passive variables are computed. After computing all the passive variables, \mathbf{p}_1 , \mathbf{p}_2 and \mathbf{p}_3 are obtained by substituting the computed values into corresponding expressions. As an example, the values of inputs ($\theta_1 = 0.8$, $\theta_2 = 1.4$, $\theta_3 = 1.1$) and all link length parameters ($l_0 = 100$, $l_1 = 50$, $l_2 = 100$, $l_3 = 50$, $n = 0.5$, $r = 100$, $d = 160\sqrt{3}$) are substituted. At the base, coordinates of \mathbf{a}_1 and \mathbf{a}_2 are $(210, 0, 0)^T$ and $(110, 0, 0)^T$ respectively. We obtain:

$$\begin{aligned} & -2.180t_3^{16} - 5.142t_3^{15} - 15.487t_3^{14} - 19.081t_3^{13} - 18.386t_3^{12} - 8.676t_3^{11} + 113.819t_3^{10} \\ & - 10.575t_3^9 + 518.475t_3^8 - 131.344t_3^7 + 1139.61t_3^6 - 682.798t_3^5 + 1386.1t_3^4 - 1048.97t_3^3 \\ & + 838.827t_3^2 - 481.681t_3 + 192.76 = 0 \end{aligned} \quad (3.5.4)$$

Solving Eq. 3.5.4 for t_3 yields the following values:

$$\begin{aligned} & -2.325, -1.211 \pm 1.678i, -0.843 \pm 1.283i, \\ & -0.003 \pm 1.829i, \pm i, 0.501 \pm 0.309i, 0.698 \pm 1.202i, 1.681 \end{aligned} \quad (3.5.5)$$

Complex values of t_3 are naturally ignored and so are negative real ones as they do not represent physically meaningful solutions. The only feasible value of t_3 is 1.681, for which the corresponding γ_3 obtained is 2.068. From here, the routine described earlier is followed to obtain γ_2 and γ_1 . Upon computation, we obtain $\gamma_2 = 2.121$ and $\gamma_1 = 2.435$.

A validation of numerical results is carried out by substituting the values of

passive variables into the Eq. 3.2.11. On substituting the values of γ_i and θ_i ($i = 1, 2, 3$) into $\boldsymbol{\eta}$, we get:

$$\boldsymbol{\eta} = (6.217, 8.659, 1.110)^T \times 10^{-15} \quad (3.5.6)$$

Substituting all the passive variables to obtain \mathbf{p}_i ; $i = 1, 2$ and 3 , gives

$$\mathbf{p}_1 = \begin{pmatrix} 72.351 \\ 68.123 \\ 53.802 \end{pmatrix}, \quad \mathbf{p}_2 = \begin{pmatrix} -106.401 \\ 5.326 \\ 73.908 \end{pmatrix}, \quad \mathbf{p}_3 = \begin{pmatrix} 37.954 \\ -118.818 \\ 66.840 \end{pmatrix}. \quad (3.5.7)$$

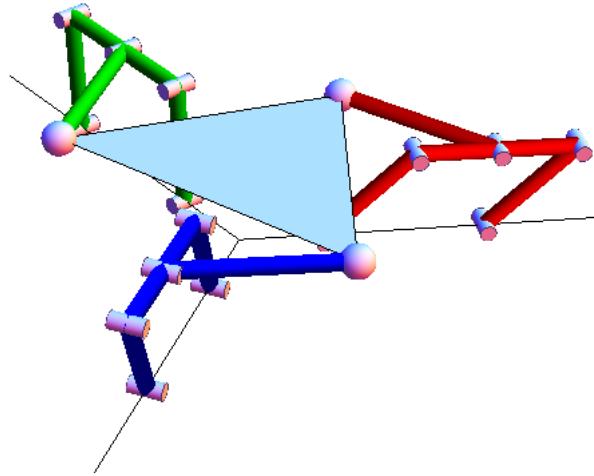


Figure 3.6: The pose of MaPaMan-II corresponding to the $\boldsymbol{\theta} = (0.8, 1.4, 1.1)^T$

3.5.1 Inverse kinematics

The coordinates of \mathbf{p}_1 , \mathbf{p}_2 and \mathbf{p}_3 are known and $\boldsymbol{\theta}$ need to be computed. From Fig. 3.5, Z coordinate of \mathbf{p}_1 is $l_1 \sin \phi_1$ which gives:

$$\phi_1 = \arcsin \left(\frac{\mathbf{p}_1 \cdot \mathbf{e}_z}{l_1} \right), \text{ where } \mathbf{e}_z = (0, 0, 1)^T. \quad (3.5.8)$$

Note that due to the geometry of the manipulator, the inverse kinematics problem can be solved in closed form for MaPaMan-II as opposed to MaPaMan-I where we obtain 8 solutions for a given pose of end-effector. As the four-bar is a parallelogram linkage in MaPaMan-II,

$$\theta_1 = \phi_1 \quad (3.5.9)$$

To find out θ_2 , symmetry of the architecture is used. The plane of the second four-bar is rotated by 120° CCW about Z axis, so that it now coincides with the XZ plane, and the transformed coordinates of \mathbf{p}_2 are found. Then θ_2 is computed in the same fashion θ_1 was computed. In a similar fashion θ_3 is computed.

As an example, the same link lengths and the positions of \mathbf{p}_i ($i = 1, 2$ and 3) are taken as in the Section 3.6.1:

$$\mathbf{p}_1 = \begin{pmatrix} 72.351 \\ 68.123 \\ 53.802 \end{pmatrix}, \quad \mathbf{p}_2 = \begin{pmatrix} -106.401 \\ 5.326 \\ 73.908 \end{pmatrix}, \quad \mathbf{p}_3 = \begin{pmatrix} 37.954 \\ -118.818 \\ 66.840 \end{pmatrix}. \quad (3.5.10)$$

On solving for the actuator angles as described above, the following values are obtained.

$$\theta_1 = 0.799 \quad (3.5.11)$$

$$\theta_2 = 1.399 \quad (3.5.12)$$

$$\theta_3 = 1.098 \quad (3.5.13)$$

The inputs that were chosen in the Section 3.5 were $\theta_1 = 0.8$, $\theta_2 = 1.4$ and $\theta_3 = 1.1$, which is equal to what is obtained from inverse kinematics computation, hence validating it.

3.6 First-order kinematics of MaPaMan-II

First-order kinematic formulations of MaPaMan-II are developed on the lines of the formulations for MaPaMan-II. The characterising of DoF is also carried out to establish the partition in the DoF in MaPaMan-II. Numerical examples are provided at the end of each formulation.

3.6.1 Forward kinematics

The same approach as that used for MaPaMan-I is followed for finding the Jacobians \mathbf{J}_v and \mathbf{J}_ω . The velocities of end-effector \mathbf{v}_c and $\boldsymbol{\omega}$ are computed from \mathbf{J}_v and \mathbf{J}_ω and $\dot{\boldsymbol{\theta}}$ using Eq. 3.4.1 and Eq. 3.4.6. All the Jacobian matrices appearing above can be computed at any configuration after the position kinematics problem has been solved. Therefore, the linear velocity of any point of interest can be obtained by the above process if the joint velocity vector, $\dot{\boldsymbol{\theta}}$, is known and so is the angular velocity of the end-effector.

Sample results of the first-order forward kinematics

Example 1: Consider the configuration of the manipulator given by: $\boldsymbol{\theta} = (0.8, 1.4, 1.1)^T$. From forward kinematics (3.5), $\boldsymbol{\psi} = (\pi, \pi, \pi)^T$, $\boldsymbol{\phi} = (0.8, 1.4, 1.1)^T$, $\boldsymbol{\gamma} = (2.435, 2.121, 2.068)^T$. With this configuration, if the active links are driven with the initial velocity, $\dot{\boldsymbol{\theta}} = (0.1000, -0.2000, -0.1000)^T$, using Eq. (3.4.19) we calculate the velocity of the end-effector:

$$\begin{aligned}(\dot{c}_1, \dot{c}_2, \dot{c}_3)^T &= (0.050, 0.006, 0.015)^T \\(\dot{x}_c, \dot{y}_c, \dot{z}_c)^T &= (-0.011, 0.078, -0.002)^T.\end{aligned}$$

3.6.2 Characterising the degrees-of-freedom of MaPaMan-II

For the sake of illustration, we take the same configuration of MaPaMan-II as that used in section 3.6.1.

$$\begin{aligned}\mathbf{J}_\omega &= \begin{pmatrix} 0.313 & -0.042 & -0.108 \\ -0.014 & 0.058 & -0.190 \\ -0.144 & -0.094 & -0.106 \end{pmatrix} \\ \Rightarrow \mathbf{g} &= \begin{pmatrix} 0.119 & -0.0005 & -0.016 \\ -0.0005 & 0.014 & 0.003 \\ -0.016 & 0.003 & 0.059 \end{pmatrix}\end{aligned}$$

The rank of \mathbf{g} turns out to be 3 unlike in MaPaMan-I. As \mathbf{g} is of full-rank, its nullspace is empty. Hence \mathbf{g}_v cannot be found, validating 3R and 0T nature of MaPaMan-II.

Inverse kinematics

As \mathbf{J}_ω matrix is of full rank when not at a singularity of loss-type (Zlatanov, Bonev, and Gosselin, 2002), we follow the same formulation in Section 3.6.1, to calculate $\dot{\boldsymbol{\theta}}$ for given values of $\boldsymbol{\omega}$. Using $\boldsymbol{\omega}$ and \mathbf{J}_ω , we calculate:

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_\omega^{-1} \dot{\boldsymbol{\omega}}. \quad (3.6.1)$$

Given the angular velocity of the end-effector, active joint rates can be calculated using Eq. (3.6.1).

Results of first-order inverse kinematics

Given the same inputs as in the example of Section 3.6.1 and $\dot{\mathbf{r}}_{ts} = \begin{pmatrix} 0.050 \\ 0.006 \\ 0.015 \end{pmatrix}$,

the active links angular velocity can be calculated using Eq. (3.6.1)

$$\dot{\boldsymbol{\theta}} = \begin{pmatrix} 0.099 \\ -0.199 \\ -0.099 \end{pmatrix}$$

The $\dot{\boldsymbol{\theta}}$ obtained matches with the values used in the Section 3.6.1, hence validating the first-order kinematics.

3.7 Incremental approach to zeroth-order kinematics

Realtime applications like computed torque control ((e.g. Ghosal, 2006)) require forward kinematic computations at each actuation cycle. However it is observed that the computation time for each set of γ_t values is 0.156s using **Mathematica**. Hence the present approach to forward kinematics would not help in real-time applications and so an alternate technique to approximately find out the passive variables at any time for a given set of inputs has been developed.

In this method, we make use of the first-order properties of motion and estimate the value of the passive variables at an instant given its value in the previous instant and a sampling time. The error in the estimation depends on the choice of sampling time and the speed of motion involved.

Differentiating Eq. 3.2.11 with respect to time, we obtain

$$\frac{\partial \boldsymbol{\eta}}{\partial t} = \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\theta}} \dot{\boldsymbol{\theta}} + \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\psi}} \dot{\boldsymbol{\psi}} + \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\gamma}} \dot{\boldsymbol{\gamma}} = \mathbf{0}. \quad (3.7.1)$$

Or,

$$\left(\frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\theta}} + \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\psi}} \mathbf{J}_{\psi\theta} \right) \dot{\boldsymbol{\theta}} + \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\gamma}} \dot{\boldsymbol{\gamma}} = \mathbf{0} \quad (3.7.2)$$

$$\Rightarrow (\mathbf{J}_{\eta\theta} + \mathbf{J}_{\eta\psi} \mathbf{J}_{\psi\theta}) \dot{\boldsymbol{\theta}} + \mathbf{J}_{\eta\gamma} \dot{\boldsymbol{\gamma}} = \mathbf{0}. \quad (3.7.3)$$

Hence we obtain

$$\dot{\gamma} = -\mathbf{J}_{\eta\gamma}^{-1}(\mathbf{J}_{\eta\theta} + \mathbf{J}_{\eta\psi}\mathbf{J}_{\psi\theta})\dot{\theta} = \mathbf{J}_{\gamma\theta}\dot{\theta}. \quad (3.7.4)$$

If we know the θ_t and γ_t at any instant t , then for a given $\dot{\theta}$, we choose a sampling rate $\Delta\theta$ such that;

$$\theta(t + \Delta t) = \theta + \Delta\theta. \quad (3.7.5)$$

$$\text{Thus, } \gamma(t + \Delta t) = \gamma(t) + \mathbf{J}_{\gamma\theta}\Delta\theta. \quad (3.7.6)$$

A truncation error keeps building up with time due to the approximation involved. This error can be rectified once in a while depending on the chosen sampling rate, by solving the Eq.3.2.11. It is observed that the computation time for each set of γ_t values is 0.016s as opposed to 0.156s using the classical method on **Mathematica**.

3.8 Conclusion

In this Chapter, we have presented the forward and inverse kinematic relationship between the end-effector and the active joints in zeroth-order and first-order. The forward kinematics problem of MaPaMan-I and MaPaMan-II resulted in a 16-degree univariate polynomial whose coefficients are obtained in closed-form. The inverse kinematics yielded an 8-degree polynomial for MaPaMan-I and closed-form solutions for MaPaMan-II. We also presented the results and their validations for these formulations.

A new incremental approach to forward kinematics was presented that is computationally much faster than the classical method. The merits and demerits of this approach are discussed. With the forward and inverse kinematic relationships presented in this Chapter, we can obtain the end-effector position and orientation anywhere within the workspace except at singular points. At the singular points, the relationship between the end-effector and the active joints degenerates. In the next Chapter we study a class of these singular cases, specifically the gain-type

singularities associated MaPaMan-I.

CHAPTER 4

Singularity analysis of MaPaMan-I

4.1 Introduction

Parallel manipulators typically have three types of singularities: loss-type, gain-type and a combination of these two (Zlatanov *et al.*, 2002). During a loss-type singularity, there exist some non zero active joint rates that result in zero velocity of the end-effector. During a gain-type singularity, the end-effector has an infinitesimal instantaneous motion when all active joint rates are zero. Hence from the perspective of controlling the manipulator loss-type singularities give a sense of boundary of the workspace while gain-type singularities do lead to loss of controllability even inside the workspace. Hence from the perspective of controlling the manipulator, gain-type singularities are of prime importance. In this chapter, we present the conditions for gain-type singularities of MaPaMan-I. An attempt to obtain a manifold in the task-space representing the singular regions in the workspace has been presented. Finally the singularities are visualised in a numerical sense using contour plots at the end of the chapter.

4.2 Conditions for gain-type singularity

It has already been discussed in Section 3.2.1, that the kinematics of the four-bars at the base are independent of the rest of the mechanism. Hence their singularities are studied separately from the rest of the mechanism. For the sake of simplicity let us assume that the four-bar is not allowed to reach singularity. This can be assured by ensuring that the four-bars adhere to Grashof conditions. This assumption ensures that $\mathbf{J}_{\psi\theta}$ exists. From Eq. 3.7.1 we have $(\mathbf{J}_{\eta\theta} + \mathbf{J}_{\eta\psi}\mathbf{J}_{\psi\theta})\dot{\theta} + \mathbf{J}_{\eta\gamma}\dot{\gamma} = \mathbf{0}$. If $\mathbf{J}_{\eta\gamma}$ becomes singular, then for any $\dot{\gamma}$, $\dot{\theta}$ will remain zero. In other

words the motors get locked even when the end-effector has a non-zero velocity (Bandyopadhyay and Ghosal, 2006). Hence the singularity condition for gain-type of singularity is

$$\det(\mathbf{J}_{\eta\gamma}) = 0. \quad (4.2.1)$$

Upon simplification, we always obtain r^3 as a factor of Eq. 4.2.1. As r is a design variable, a factor of r^3 can have an effect on the value of Eq. 4.2.1 depending on the unit of measurement or scaling of the overall system of dimensions. Hence we remove this factor from the expression for singularity and obtain a new singularity condition

$$S(\boldsymbol{\gamma}) \triangleq \frac{\det(\mathbf{J}_{\eta\gamma})}{r^3} = 0, \quad (4.2.2)$$

This condition however has certain link dimensions appearing in it and since no further factorisation was possible, the link dimensions are first normalised. Without any loss of generality, we normalise the link lengths with respect to x_1 .ie., set $x_1=1$.

Fig. 4.1 shows MaPaMan-I in a singular state for the sake of better visualization

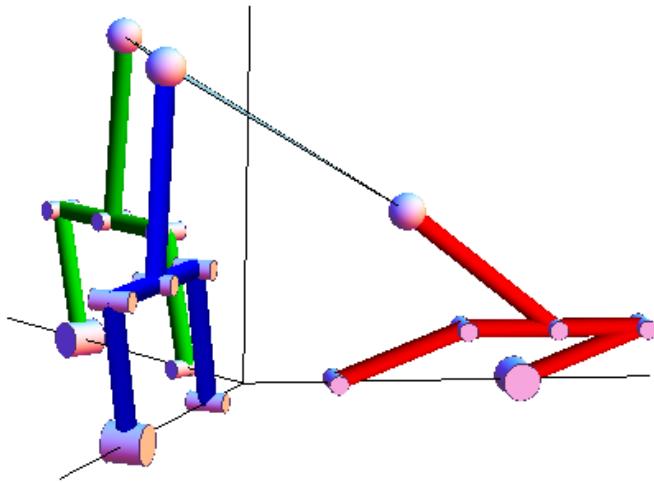


Figure 4.1: A gain-type singular pose of MaPaMan-I

of a gain-type singularity; for the link dimensions given in Table B.1. The values of the actuator angles $\boldsymbol{\theta}$ are $(0.25, 1, 1)^T$ and the corresponding values of passive

variables are $\phi = (0.33, 1, 1)^T$

$\psi = (3.14, 3.14, 3.14)^T$

$\gamma = (2.546, 1.865, 1.865)^T$. On substituting all these values into $S(\gamma)$, we obtain

$$S(\gamma) = -3.76 \times 10^{-6} \quad (4.2.3)$$

, which is 0; hence validating that it is at gain-type singularity. Note how the strut appears in the same plane as the end-effector. While this example demonstrates gain-type singularity, it should be noted that this need not be the only physical condition where gain-type singularities might be encountered as suggested by Basu and Ghosal (1997) in the case of 3-RPS manipulator.

4.2.1 Singular manifold

Singular manifold is the locus of the end-effector in the task-space, such that it is always satisfies the conditions for gain-type singularity. Such a manifold is generated to be free of all active and passive variables and solely obtained as a set of one or more surfaces in a three dimensional space $\mathbf{r}_{ts} = (c_1, c_2, z_c)^T$. Such a surface has a lot of implications in path planning, where the path is algorithmically chosen such that it never intersects the singularity manifold and hence avoids any encounters with gain-type singularities in the workspace. Trajectory tracking it becomes very helpful if a singularity-free convex region can be found in the workspace so that any path in that region could be tracked without loss of control. In the field of parallel manipulators it is well-known that obtaining closed-form expression for singularity manifolds is not always possible due to highly non-linear nature of the constituent equations from which the expression for the manifold is derived. However, due to the enormity of the impact that the expression for such a manifold can have on problems of path-planning etc., it is attempted to find an expression for the same in this report. In order to obtain the singularity manifold, equations that relate the active variables to the task-space variables are first formulated. The active variables are then eliminated to finally obtain an equation devoid of all variables except the task-space variables.

As shown in Section 3.4.2, the coordinates of end-effector are first found in terms of \mathbf{r}_{ts} . The coordinates of \mathbf{b}_{2i} ($i=1, 2, 3$) are known from Eq.3.4.8. The following constraint equations can be framed as the length of the strut is constant.

$$f_i(\boldsymbol{\theta}, \mathbf{r}_{ts}) \triangleq (\mathbf{p}_i - \mathbf{b}_{2i}) \cdot (\mathbf{p}_i - \mathbf{b}_{2i}) - r^2, \quad i = 1, 2, 3. \quad (4.2.4)$$

$$\mathbf{f} = (f_1, f_2, f_3)^T = \mathbf{0} \quad (4.2.5)$$

As the constraints are invariant w.r.t. time, differentiating Eq.4.2.5 gives:

$$\frac{d\mathbf{f}}{dt} = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}} \dot{\boldsymbol{\theta}} + \frac{\partial \mathbf{f}}{\partial \dot{\mathbf{r}}_{ts}} \dot{\mathbf{r}}_{ts} = \mathbf{0} \quad (4.2.6)$$

$$\Rightarrow \mathbf{J}_{f\boldsymbol{\theta}} \dot{\boldsymbol{\theta}} + \mathbf{J}_{f\mathbf{r}_{ts}} \dot{\mathbf{r}}_{ts} = \mathbf{0} \quad (4.2.7)$$

$$(4.2.8)$$

Hence the condition for gain-type singularity is

$$f_4(\boldsymbol{\theta}, \mathbf{r}_{ts}) \triangleq \det(\mathbf{J}_{f\mathbf{r}_{ts}}) = 0 \quad (4.2.9)$$

$$(4.2.10)$$

Upon solving the four equations ($f_i = 0, \quad i = 1, 2, 3, 4$), we can eliminate $\boldsymbol{\theta}$ to obtain a manifold solely in \mathbf{r}_{ts} . The $\boldsymbol{\theta}$ in $f_i(i = 1, 2, 3, 4)$ are all converted to algebraic expressions (3.2.18) to obtain $g_1(t_1, \mathbf{r}_{ts})$, $g_2(t_2, \mathbf{r}_{ts})$, $g_3(t_3, \mathbf{r}_{ts})$ and $g_4(t_1, t_2, t_3, \mathbf{r}_{ts})$ respectively. The following is the scheme used to eliminate $\boldsymbol{\theta}$ in a step by step fashion:

$$\left. \begin{array}{l} g_1(t_1, \mathbf{r}_{ts}) \\ g_4(t_1, t_2, t_3, \mathbf{r}_{ts}) \\ g_2(t_2, \mathbf{r}_{ts}) = 0 \end{array} \right\} \xrightarrow{\times t_1} g_5(t_2, t_3, \mathbf{r}_{ts}) = 0 \quad \left. \begin{array}{l} g_5(t_2, t_3, \mathbf{r}_{ts}) = 0 \\ g_3(t_3, \mathbf{r}_{ts}) = 0 \end{array} \right\} \xrightarrow{\times t_2} g_6(t_3, \mathbf{r}_{ts}) = 0 \quad \left. \begin{array}{l} g_6(t_3, \mathbf{r}_{ts}) = 0 \\ g_7(\mathbf{r}_{ts}) = 0 \end{array} \right\} \xrightarrow{\times t_3} g_7(\mathbf{r}_{ts}) = 0 \quad (4.2.11)$$

The functions g_1 and g_4 are quadratic in t_1 . The variable t_1 is eliminated from $g_1 = 0$ and $g_4 = 0$ to obtain $g_5(t_2, t_3, \mathbf{r}_{ts}) = 0$ which is quartic in both t_2 and

t_3 . As the size of the expression is large $\approx 4.3\text{MB}$, the coefficients are replaced with “dummy-coefficients”. Replacing the coefficients with dummies reduces the size of the expression while retaining the structure of the equations. However any simplification or cancelation of terms that may occur at later stages due to the actual nature of the coefficients is compromised for by replacing them with dummy coefficients.

The function g_5 is quartic in t_2 while g_2 is quadratic in t_2 , hence t_2 is eliminated from these two to obtain $g_6(t_3, \mathbf{r}_{ts}) = 0$, where g_6 is an eight degree polynomial in t_3 .

The coefficients of t_3 in g_6 are once again replaced with dummy coefficients due to their large size. The function g_3 is quadratic in t_3 and hence g_6 is divided by g_3 with respect to t_3 , to return a remainder that would be no more than linear in t_3 . Solving the linear equation in t_3 and substituting back in $g_3 = 0$, we obtain the required singularity manifold in terms of \mathbf{r}_{ts} free of $\boldsymbol{\theta}$.

Upon substituting back the values of the dummy coefficients into the final expression, we obtain an extremely large expression ($\approx 89\text{GB}$ in size) whose coefficients are obtained in closed-form in terms of length parameters and \mathbf{r}_{ts} . An expression of such a large size is very difficult to manipulate or study and hence it is abandoned at this.

4.2.2 Numerical studies of the gain-type singularities

As a closed-form derivation of the singularity manifold was fruitless, a numerical approach is followed to understand the singularities better. For clarity in visualising the singular regions contour plots are generated that show the singularity regions for a given z_c , as well as the distance of any point in the task-space from singular manifold.

Visualisation of the singular manifold in the task-space

For the dimensions presented in Table B.1, a region is spanned in the (c_1, c_2) space at a given z_c and contour curves are plotted for $S(\boldsymbol{\gamma})$. As $S(\boldsymbol{\gamma})$ is a function of $\boldsymbol{\theta}$

and γ , for each value of (c_1, c_2) at a given z_c , inverse kinematic calculations are performed to obtain θ and γ , which are then used to find the value of $S(\gamma)$.

The brief description of the physical meaning of Rodrigue's parameters are given in Section 3.3. In case of MaPaMan-I, $c_1 = k_x \tan \frac{\phi}{2}$, $c_2 = k_y \tan \frac{\phi}{2}$, $c_3 = 0$; where $k_x^2 + k_y^2 = 1$. Hence we choose $k_x = \cos \varsigma$ and $k_y = \sin \varsigma$. In the contour plot, the X-axis represents the angle ς and Y-axis represents φ . The parameters φ and ς are chosen instead of c_1 and c_2 as they have a periodic boundary limit of $[0, 2\pi]$ as opposed to c_1, c_2 that can take any value in \mathbb{R} . Moreover it is easier to visualise the task-space in terms of φ and ς instead of c_1 and c_2 . The value of ς indicates the orientation of the instantaneous axis of rotation in the XY plane while φ indicates the tilt of the end-effector about that axis. The curves on the contour correspond

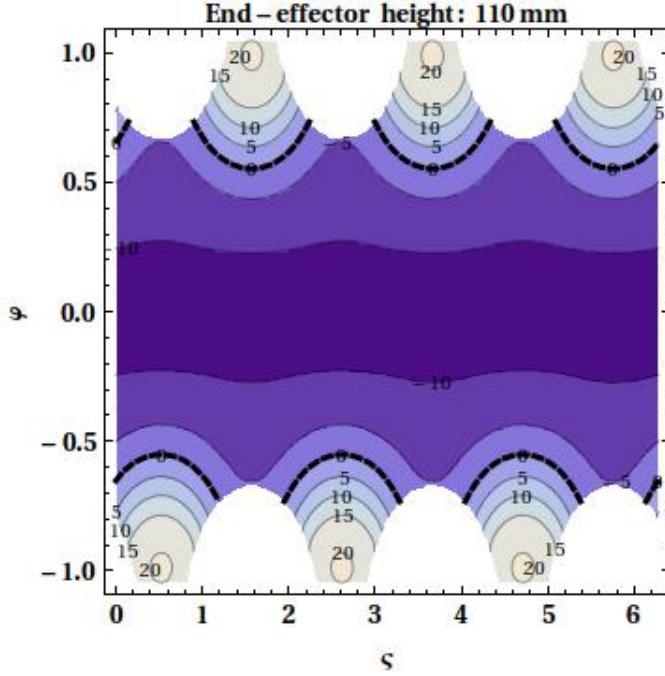
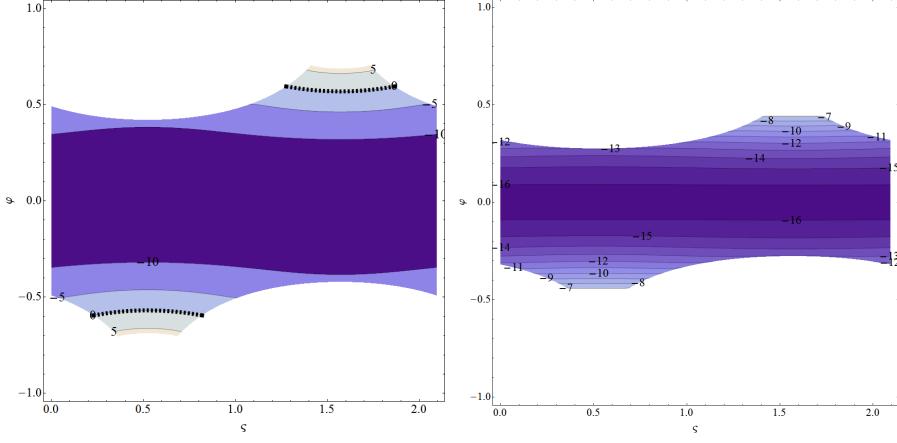


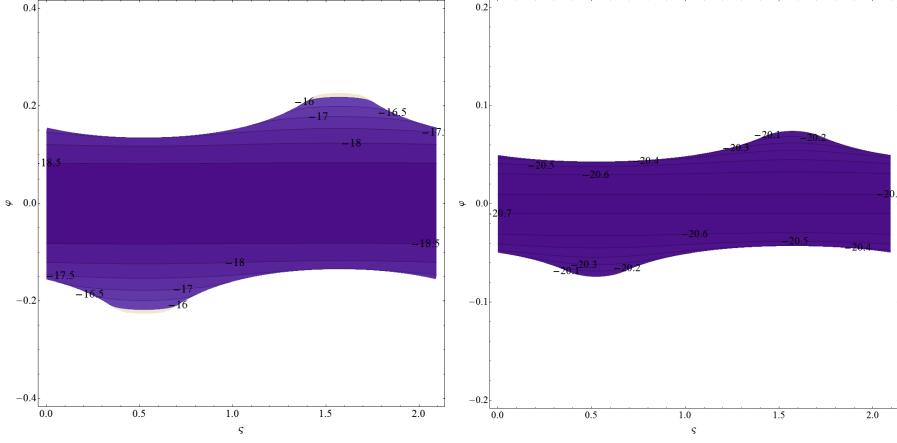
Figure 4.2: Singularity countour showing 3-way symmetry

to the magnitude of $S(\gamma)$ which represents the ‘distance’ from the singularity. In Fig. 4.2 the zero valued curve (shown by dotted line) on the contour represents the singular curve in the workspace, while the white space indicates infeasible points.

The Fig. 4.2 shows the singularity contour for $z_c = 110$. Note the 3-way symmetry appearing in the plot owing to the symmetry in the architecture of the manipulator. For the sake of clarity, the contour is plotted next for one-third of the range of ς . The Fig. 4.3(a-d) show how the contours change with variation



(a) Contour plot of singular manifold in r_{ts} space at $z_c = 135$ (b) Contour plot of singular manifold in r_{ts} space at $z_c = 150$



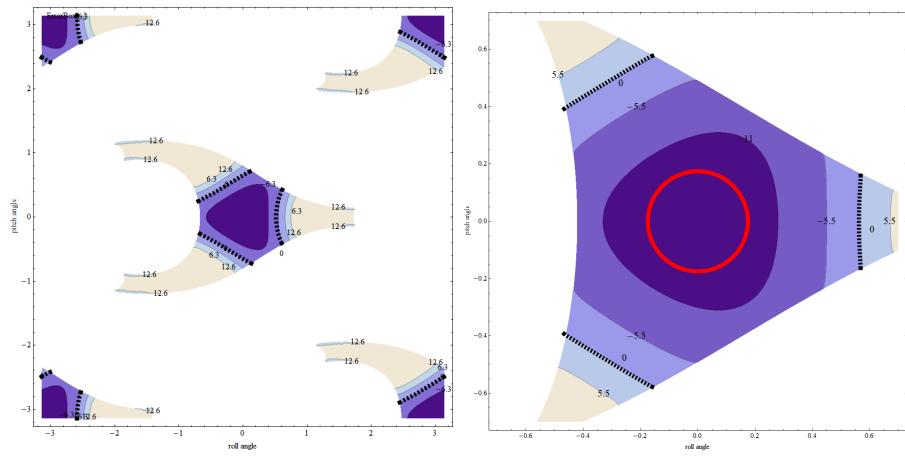
(c) Contour plot of singular manifold in r_{ts} space at $z_c = 165$ (d) Contour plot of singular manifold in r_{ts} space at $z_c = 175$

Figure 4.3: Singularity contours in terms of φ and ς at varying z_c

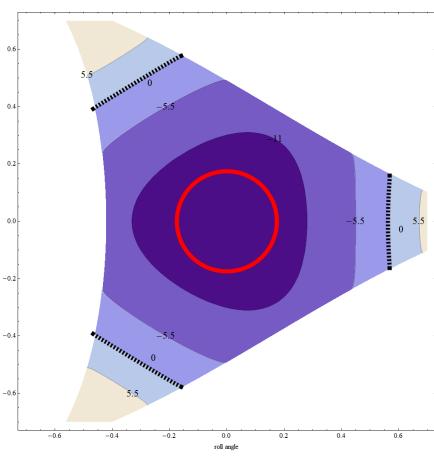
in z_c . It is observed that above $z_c = 150$, zero-valued curve on the contour plot disappears (see Fig. 4.3(a)). This shows that above this value of heave, gain-type singularity is not possible. However, the predominant white space indicates that the singularity free workspace decreases with increasing heave as shown in Fig. 4.3(b).

Visualisation of the singular manifold in terms of roll, pitch and heave

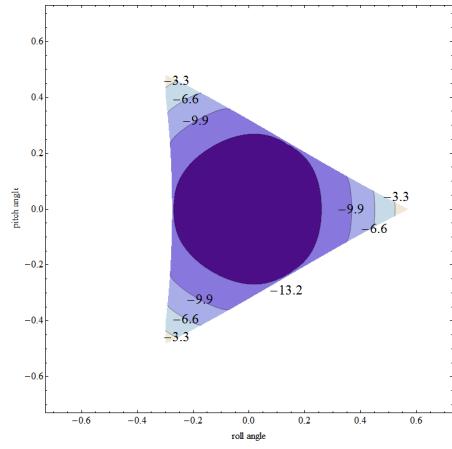
In several practical applications, orientation is better visualised in terms of roll and pitch instead of Rodrigue's parameters or ς, φ that are derived from them. Hence the contour plots are also plotted in a more intuitive roll, pitch, heave framework. Similar to the earlier plots, for various roll and pitch values, inverse kinematic calculations are performed and θ and γ values are computed to obtain the value of $S(\gamma)$ at those points (see Fig. 4.4). Again note that for $z_c > 150mm$, we don't observe any gain-type singularity. The advantage of plotting in roll, pitch and heave framework is that it is easier to visualise and extremely useful in planning paths in the workspace. For example, during path-planning it becomes very helpful if a singularity-free convex region can be found in the workspace so that any path in that region could be tracked without losing control en-route. This can be realised by fitting a circle in the contour plot to obtain a singularity free region as shown in Fig. 4.4(b).



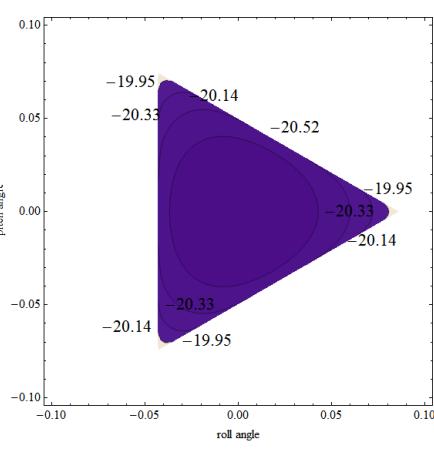
(a) Contour plot of singular manifold at $z_c = 110$



(b) Contour plot of singular manifold at $z_c = 135$



(c) Contour plot of singular manifold at $z_c = 150$



(d) Contour plot of singular manifold at $z_c = 175$

Figure 4.4: Singularity contours in roll and pitch at varying z_c

4.3 Conclusion

In this Chapter we have presented the gain-type singularities associated with MaPaMan-I. A closed-form expression for the singular manifold has been derived but due to its very large size ($89GB$), it is impossible to manipulate and study such an expression and study; hence it was abandoned. A numerical study of the singular manifold through contour plots has been presented. The plots reveal 3-way symmetry due to the symmetry inherent in the geometry of the manipulator. The plots show that after a certain value of heave, gain-type singularities do not appear, but workspace becomes very limited in size. This property becomes useful in practical applications as the manipulator can be controlled without encountering any gain-type singularity after a certain value of heave.

CHAPTER 5

Dynamics

5.1 Introduction

Kinematics provides information about the position of the end-effector and its time derivatives, and their relationship with the actuator displacement and velocities. However, to find out the effect of forces and torques acting on the system a dynamic analysis is necessary. The dynamic analysis of MaPaMan forms a part of a concurrent study reported in Mehta (2012).

While this report does not deal with the dynamic formulations of MaPaMan, yet we require certain elements of the same to understand the effect of the design parameters on variation of torque which is presented in Section 6.2.3. While all the computations required are already a part of Mehta (2012), a brief section has been included in this report for the sake of logical continuity and completeness of the contents.

5.2 Development of the mathematical model in the actuator space

A mathematical model was developed in the actuator space to study the dynamics of the system. The advantage of working in the actuator space is that the model becomes equivalent to that of a serial manipulator and we can directly obtain the actuator torques. The following is the general form of the dynamic model in actuator space $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)^T$:

$$\boldsymbol{M}_{\boldsymbol{\theta}} \ddot{\boldsymbol{\theta}} + \boldsymbol{C}_{\boldsymbol{\theta}} \dot{\boldsymbol{\theta}} + \boldsymbol{G}_{\boldsymbol{\theta}} = \boldsymbol{\tau}_{\boldsymbol{\theta}}. \quad (5.2.1)$$

This formulation is valid as long as the manipulator encounters no gain-type singularity in its trajectory. Given a desired trajectory we can perform the inverse kinematics (refer to Section 3.2.2 and 3.4.3) to calculate the corresponding desired values of $\boldsymbol{\theta}$, ϕ , ψ , γ , $\dot{\theta}$, $\dot{\psi}$, $\dot{\phi}$ and $\dot{\gamma}$. Using formulations in Ghosal (2006), we can calculate \mathbf{M}_θ , \mathbf{C}_θ and \mathbf{G}_θ . Hence for a non-singular trajectory we obtain the torques on the actuator $\boldsymbol{\tau}_\theta$ at any instant. For the analysis carried out in this project, we require the torques on the actuator in a static scenario. In statics, the rates of all joints are zero. Thus

$$\boldsymbol{\tau}_\theta = \mathbf{G}_\theta \quad (5.2.2)$$

where,

$$\mathbf{G}_\theta = \mathbf{J}_{q\theta}^T \mathbf{G}, \quad \mathbf{J}_{q\theta} = \frac{\partial \mathbf{q}}{\partial \boldsymbol{\theta}}. \quad (5.2.3)$$

The vector \mathbf{G} is obtained from a Lagrangian formulation of the dynamic model in full configuration space, $\mathbf{q} = (\boldsymbol{\theta}, \boldsymbol{\gamma})^T$:

$$V = g \sum_{i=1}^n m_i p_{c_i} \quad (5.2.4)$$

$$\mathbf{G} = \frac{\partial V}{\partial \mathbf{q}} \quad (5.2.5)$$

where,

p_{c_i} : height of center of gravity of i^{th} link from reference plane z=0. Thus we obtain the torques in each motor for a static scenario.

5.3 Conclusion

A mathematical model was developed in the actuator space. We need to compute the torques required at motors to maintain a static configuration, which we obtain from Eq. 5.2.2. This would be used in Section 6.2.3 that deals with parametric

variations and their effect on the torques at the three actuators. The dynamic model also serves as the basis for choosing the right actuators for a required task. Section 7 deals with design details of two prototypes of MaPaMan-I. The dynamic model was used to validate the choice of motors used in it to satisfy certain target requirements.

CHAPTER 6

Parametric study of the manipulator architecture

6.1 Introduction

Parametric studies have been carried out to evaluate the impact of the design parameters in MaPaMan-I on various performance parameters of the manipulator. As there exists no formal basis of comparison for two parallel manipulators in general, a few basic performance metrics are studied such as: stiffness, payload inertia to self-weight inertia, mechanical advantage and range of motion. Parametric studies have been carried out that bring out the essential features of the manipulator for a given set of link dimensions. These studies aid in designing the most suitable manipulator for a given task. In the Chapter 7 these studies are utilised to come up with the best set of dimensions for a given target specification of a prototype of MaPaMan-I.

6.2 Description of performance metrics

Availability of a large number of design parameters is a feature of MaPaMan that makes it more adoptable to different tasks than many other existing manipulators such as 3-RPS, Agile Eye etc. For the sake of simplicity, we reduce the design space a little by assuming the four-bar to be a parallelogram. The design parameters under consideration include l_0, l_1, n, r, r_t for a given x_1 . Understanding the effect of variation of each of these parameters on overall performance of the manipulator is essential to decide on the link dimensions.

6.2.1 Variation of stiffness

One basis of comparison between two manipulators could be the stiffness that they offer in certain chosen direction. The stiffness would vary with change in configuration and would be the least when near singularity. We assume that a pay-load (mass= m) is placed at the centroid of the end-effector. A lot of effort would be required to move it up ad down as opposed to just tilting the end-effector as the potential energy of the payload changes much more while lifting it rather than while tilting. Therefore, it is necessary for the mechanism to be stiff in the vertical direction while lifting the payload. Hence, to obtain an overall feel of stiffness we subject the manipulator to produce a pure heave motion while carrying a payload (see Fig. 6.1). The deflection of the manipulator in the vertical

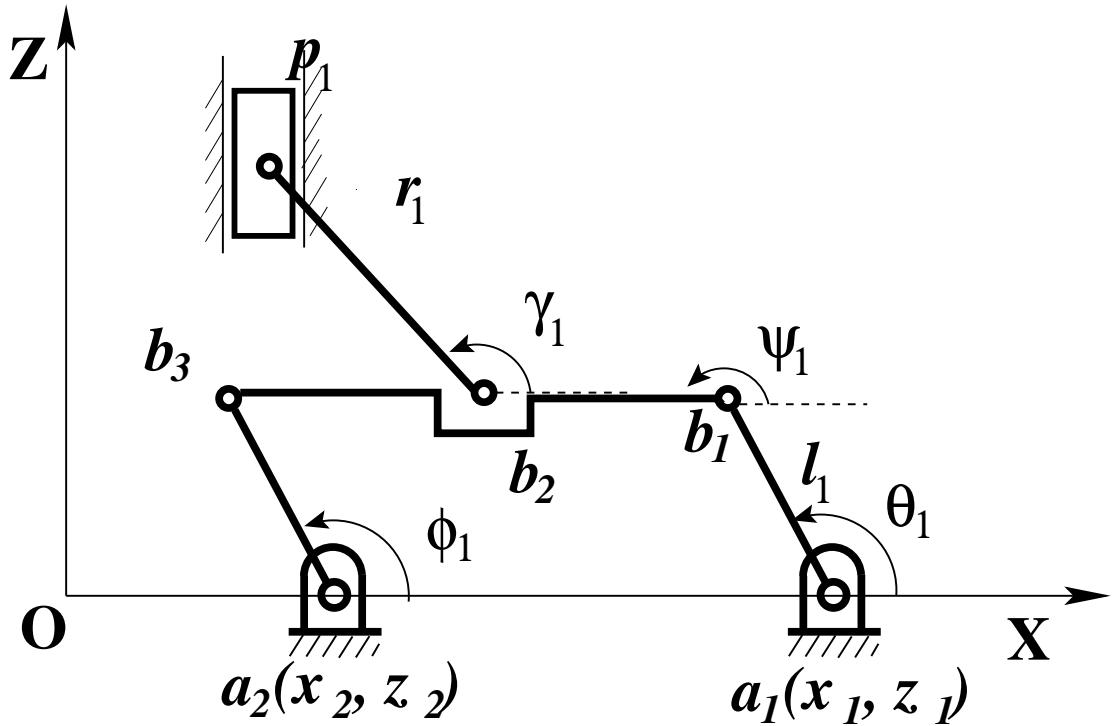


Figure 6.1: Kinematically equivalent 1-DoF mechanism representing a leg of MaPaMan-I for pure heave motion

direction is calculated assuming static conditions at each instance and the stiffness is computed as the force required to produce unit deflection along the direction of its application. This way instead of looking at the entire stiffness matrix of the mechanism, we look at one specific dominating term of the matrix, namely the stiffness in the vertical direction; which essentially provides us with a scalar

representative measure at each instant to quantitatively study the stiffness. Fig.

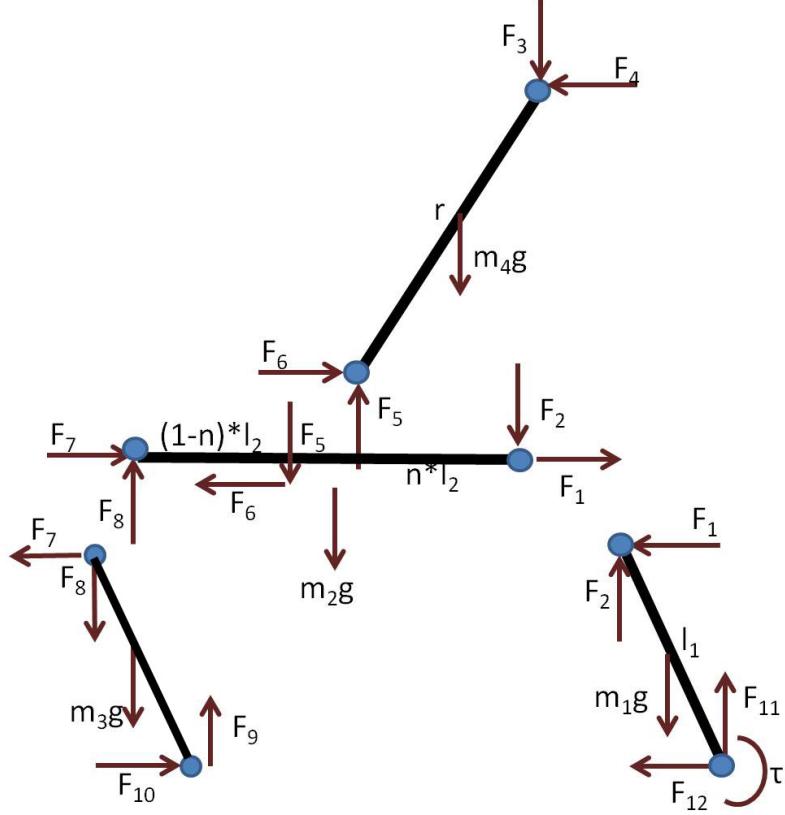


Figure 6.2: FBD of MaPaMan-I

6.2(a) shows a kinematically equivalent 1-DoF mechanism representing the first leg when the top platform undergoes pure heave motion. A free-body diagram(FBD) is constructed as shown in Fig. 6.2(b). Due to the symmetry in loading, $F_3 = \frac{mg}{3}$. From the FBD of each leg, it is clear that there are 12 unknowns; F_i ($i = 1, \dots, 12, i \neq 3$) and τ . These are obtained by solving the following representative

static equilibrium of the individual components.

$$F_4 - F_6 = 0 \quad (6.2.1)$$

$$F_3 + m_4 g - F_5 = 0 \quad (6.2.2)$$

$$F_3 \cos \gamma_1 - F_4 \sin \gamma_1 + \frac{m_4}{2} g \cos \gamma_1 = 0 \quad (6.2.3)$$

$$m_2 g + F_5 - F_8 + F_2 = 0 \quad (6.2.4)$$

$$F_6 - F_1 - F_7 = 0 \quad (6.2.5)$$

$$\begin{aligned} & F_8(1-n) \cos \psi_1 - F_7(1-n) \sin \psi_1 - \\ & m_2 g(1/2-n) \cos \psi_1 + F_2 n \cos \psi_1 + F_1 \sin \psi_1 = 0 \end{aligned} \quad (6.2.6)$$

$$\tau + F_1 l_1 \sin \theta_1 + F_2 l_1 \cos \theta_1 - \frac{m_1 g l_1}{2} \cos \theta_1 = 0 \quad (6.2.7)$$

$$F_1 + F_{12} = 0 \quad (6.2.8)$$

$$F_2 + F_{11} - m_1 g = 0 \quad (6.2.9)$$

$$F_7 - F_{10} = 0 \quad (6.2.10)$$

$$F_8 - F_9 + m_3 g = 0 \quad (6.2.11)$$

$$F_7 l_3 \sin \phi_1 - F_8 l_3 \cos \phi_1 - \frac{m_3 g l_3}{2} \cos \phi_1 = 0 \quad (6.2.12)$$

Upon solving the Eq.6.2.1-Eq.6.2.12 for the 12 variables, we obtain all the forces and the torque. The deflections in each member due to the action of forces are found. Some generic assumptions are made to simplify the calculations:

- All members are assumed to behave in a linear elastic manner, i.e., Hooke's law is applicable.
- All links are taken to be straight and of uniform rectangular cross-section.
- The deflection due to self-weight of the links are neglected w.r.t. the deflection due to the forces acting.
- The top moving plate is assumed to be rigid.

The deflection in the strut is due to axial loading. Assuming the area of cross-section to be A and Young's modulus to be E , the deflection in axial direction is computed as:

$$\delta_{strut_a} = \frac{\sqrt{F_3^2 + F_4^2} r}{AE} \quad (6.2.13)$$

The deflection in the coupler is due to axial loading as well as bending. The deflection due to axial loading is computed as:

$$\delta_{coupler_a} = \frac{F_6 l_0}{AE}, \quad (6.2.14)$$

The deflection due to bending is approximated as in the bending of a pinned-pinned beam. The deflection in this member is computed as shown in Timoshenko (1930):

$$\delta_{coupler_b} = \frac{F_5 l_0^3}{48.7EI}, \quad (6.2.15)$$

where I is the area moment of inertia of the cross-section about its neutral axis. The deflection in the rocker is due to the axial loading alone, which is computed as:

$$\delta_{rocker_a} = \frac{\sqrt{F_7^2 + F_8^2} l_1}{AE}, \quad (6.2.16)$$

The deflection in the crank is due to axial loading as well as bending. The axial deflection is computed as:

$$\delta_{crank_a} = \frac{(-F_2 \sin \theta_1 - F_1 \cos \theta_1)l_1}{EA}, \quad (6.2.17)$$

The deflection due to bending is approximated as a cantilever bending and is computed as:

$$\delta_{crank_b} = \frac{(F_2 \cos \theta_1 + F_1 \sin \theta_1)l_1^3}{3EI} \quad (6.2.18)$$

The ratio deflection in the crank due to bending and due to axial loading is decided by the ratio of the length of the crank, to its thickness.

$$\frac{\delta_{crank_b}}{\delta_{crank_a}} = \left(\frac{F_b l^3}{EI} \right) / \left(\frac{F_a l}{AE} \right) = \frac{Al^2}{I} = \frac{b^2 l^2}{b^4/12} = 12 \frac{l^2}{b^2}, \quad b \ll l \quad (6.2.19)$$

as $F_b \simeq F_a$ (same order)

As the thickness is very small compared to its length, the deflection due to axial loading comes out to be very small compared to that due to bending. Hence δ_{crank_a} is neglected w.r.t. δ_{crank_b} . The overall deflection in the vertical direction is the sum of all the individual components in that direction:

$$\delta_z = \delta_{strut_a} \sin \gamma_1 + \delta_{coupler_b} + \frac{\delta_{rocker_a} \sin \theta_1 + (\delta_{crank_b} \cos \theta_1)}{2} \quad (6.2.20)$$

The overall deflection in the Z direction is found to be δ_z . Hence the component of stiffness in the vertical direction is:

$$K = \frac{mg}{\delta_z} \quad (6.2.21)$$

The variation of the stiffness is calculated for different input angles for the design parameters given in Table B.1. We observe that the stiffness is higher at higher input angles. The stiffness value is high and varies very little over the entire range of motion (see Fig. 6.3). The design parameters are varied and the variation in stiffness is noted.

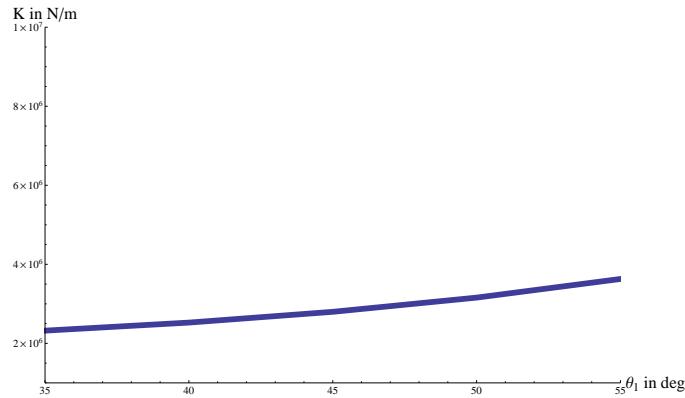
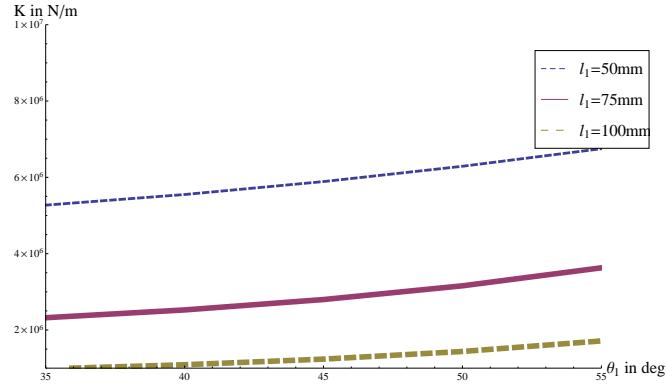


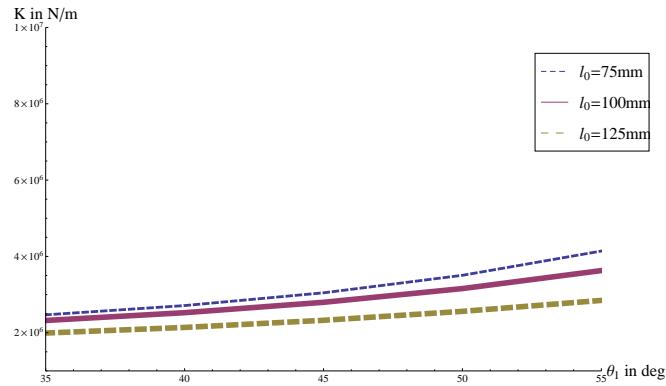
Figure 6.3: Variation of stiffness with change in input angle

- Lower values of l_0 as well as l_1 provide higher stiffness to the system. The effect of l_1 is however more profound as seen in Fig. 6.4(a) and Fig. 6.4(b). This is because lower the l_1 value, lesser will be the deflection in it and hence greater the stiffness of the system. On the other hand, l_0 does not feature directly in the calculations of deflection, but affects the kinematics of the manipulator which in turn affects the stiffness. Hence its effect is not as significant as that of l_1 .

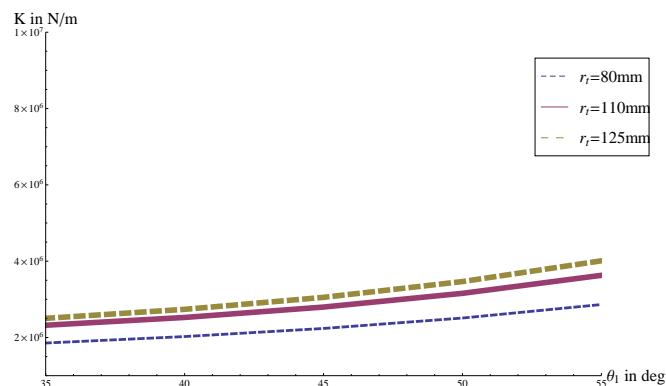
- A lower value of r_t imparts lesser stiffness to the system than a higher value of r_t (see Fig. 6.4(c)). This is expected and can be explained by the fact that the end-effector comes in the same plane as the strut at smaller displacements of the crank when the r_t value is lower, than at higher values of r_t . When this happens, the manipulator loses stiffness along the direction normal to the end-effector.
- Variation of other parameters show very little effect on the stiffness of the manipulator as compared to l_1, l_0 and r_t .



(a) Effect of l_1 on stiffness



(b) Effect of l_0 on stiffness



(c) Effect of r_t on stiffness

Figure 6.4: Effect of design parameters on stiffness

6.2.2 Variation of payload to self-weight inertia

Payload to self-weight ratio plays an important role when comparing the load carrying abilities of two manipulators. The ratio of mass of payload to mass of system says very little about its behaviour in motion, hence a new metric based on equivalent inertia in motion is used to find out the ratio of moving inertia of payload to the system. While inertia of the payload is a simple quantity to measure, inertia of the manipulator as a whole is too complicated to be quantified by a single scalar. Hence we subject the manipulator to a pure heave motion, as in the case of quantifying stiffness so that all the active joint rates are the same and then we find the equivalent inertia of the system. Also the payload to self-weight ratio becomes a critical factor when lifting a payload vertically up, hence choice of a pure heave motion is a justifiable decision.

When the manipulator executes a positive heave motion, the kinetic energy of the entire system is computed and an equivalent inertia of the entire system (Fig 6.1) is found. As the active joint rates are all equal, the kinetic energy of each link in one leg will be the same as the corresponding ones in the others.

$$T_{crank} = \frac{1}{2} \left(\frac{m_1 l_1^2}{3} \right) \dot{\theta}_1^2 \quad (6.2.22)$$

$$T_{coupler} = \frac{1}{2} m_2 l_1^2 \dot{\theta}_1^2 \quad (6.2.23)$$

$$T_{rocker} = \frac{1}{2} \left(\frac{m_3 l_1^2}{3} \right) \dot{\theta}_1^2 \quad (6.2.24)$$

$$T_{strut} = \frac{1}{2} (m_4 \mathbf{v} \cdot \mathbf{v}) + \frac{1}{2} \left(\frac{m_4 r^2}{12} \right) \dot{\gamma}_1^2, \quad (6.2.25)$$

where,

$$\mathbf{v} = \begin{pmatrix} -l_1 \sin \theta_1 \dot{\theta}_1 - \frac{r}{2} \sin \gamma_1 \dot{\gamma}_1 \\ 0 \\ l_1 \cos \theta_1 \dot{\theta}_1 + \frac{r}{2} \cos \gamma_1 \dot{\gamma}_1 \end{pmatrix} \quad (6.2.26)$$

As the motion under consideration is pure heave, γ is known explicitly in terms of θ from forward kinematics(see Section 3.2.1).

$$\gamma_1 = \arccos \left(\frac{1}{r} (-l_1 \cos \theta_1 + l_2 n + r_t - x_1) \right) \quad (6.2.27)$$

and,

$$\dot{\gamma} = \mathbf{J}_{\gamma\theta} \dot{\theta} \quad (6.2.28)$$

$$\Rightarrow \dot{\gamma}_1 = -\frac{l_1 (2l_1 \sin \theta_1 \cos \theta_1 + \sin \theta_1 (l_1 \cos \theta_1 - 3l_2 n + 3r \cos \gamma_1 + 3x_1))}{r (\sin \gamma_1 (3l_1 \cos \theta_1 - 3l_2 n + r \cos \gamma_1 + 3x_1) + 2r \sin \gamma_1 \cos \gamma_1)} \dot{\theta}_1. \quad (6.2.29)$$

Similarly kinetic energy of top-plate and payload are computed.

$$T_{top-plate} = \frac{1}{2} m_5 \dot{z}_c^2 \quad (6.2.30)$$

$$T_{payload} = \frac{1}{2} m \dot{z}_c^2 \quad (6.2.31)$$

where z_c is known explicitly in terms of θ_1 .

$$z_c = l_1 \sin \theta_1 + r \sin \gamma_1 \quad (6.2.32)$$

$$\Rightarrow \dot{z}_c = l_1 \cos \theta_1 \dot{\theta}_1 + r \cos \gamma_1 \dot{\gamma}_1. \quad (6.2.33)$$

, where $\dot{\gamma}_1$ is known in terms of $\dot{\theta}$ from Eq.6.2.29. Hence total kinetic energy:

$$T_{system} = 3(T_{crank} + T_{coupler} + T_{rocker} + T_{strut}) + T_{top-plate} + T_{payload} \quad (6.2.34)$$

Having found the total kinetic energy of the system, we find the equivalent inertia of all three legs as reflected at the actuators; denoted by I_{system} . Similarly we find out the equivalent inertia of the payload as reflected at the actuator; denoted by

$I_{payload}$.

$$T_{system} = \frac{1}{2} I_{system} \dot{\theta}_1^2 \quad (6.2.35)$$

$$T_{payload} = \frac{1}{2} I_{payload} \dot{\theta}_1^2 \quad (6.2.36)$$

$$\Rightarrow IR \triangleq \frac{I_{payload}}{I_{system}} = \frac{T_{payload}}{T_{system}} \quad (6.2.37)$$

$$(6.2.38)$$

The variation of IR is plotted against change in input angles and its variation with respect to change in design parameters is noted.

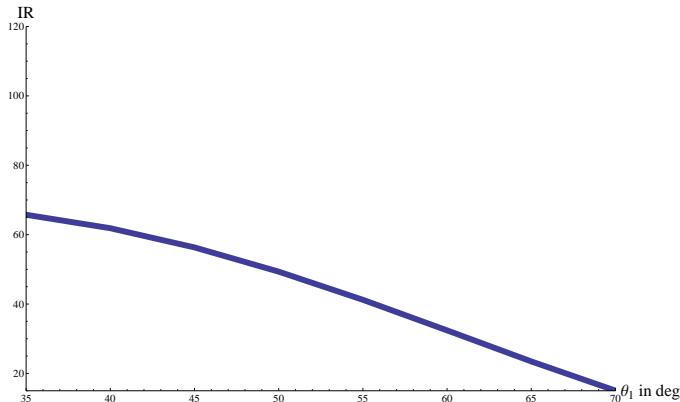
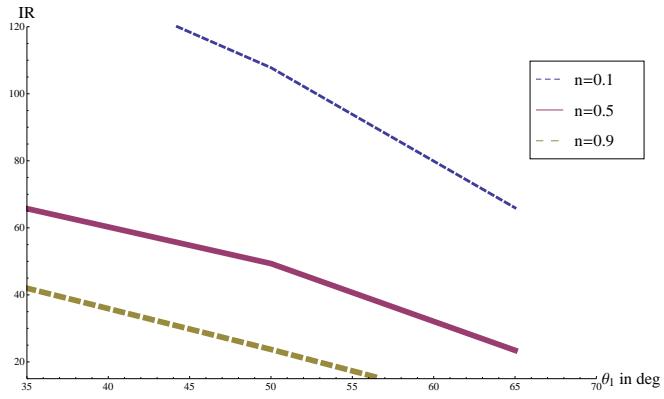
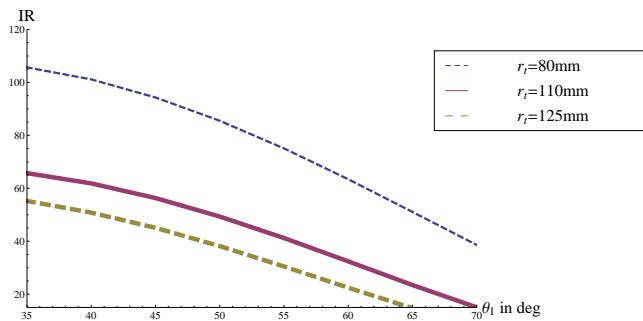


Figure 6.5: Variation of IR with change in input angles

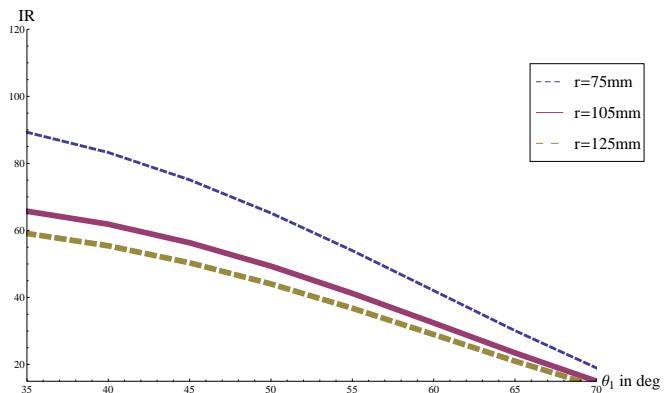
- The payload to self-weight ratio decreases with increase in input angles as shown in Fig. 6.5. This is because when the strut is perpendicular to the end-effector it is at loss-type singularity. With greater crank displacement, the manipulator nears loss-type singularity where the velocity of the end-effector drops to zero. Hence with increase in displacement of the crank, IR keeps decreasing.
- We observe that if the strut is placed closer to the rocker then IR decreases (see Fig. 6.6(a)). Similarly lower values of r_t produce high IR values as shown in Fig. 6.6(b). This can also be explained by the same argument used for the previous case. The manipulator reaches loss-type singularity at lower displacements of crank when the strut is closer to rocker as well as when the r_t value is high. Hence a low value of r_t provides lower IR and by the same argument IR decreases as the strut is moved away from the crank.
- Lower values of r as well as l_1 provide higher payload to self-weight ratio (see Fig. 6.6(c) and Fig. 6.6(d)). This behavior is expected as lower link lengths implies lower moment of inertia when in motion.



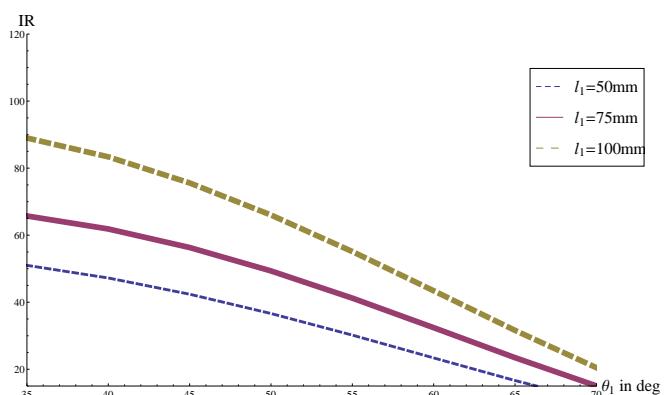
(a) Effect of n on IR



(b) Effect of r_t on IR



(c) Effect of r on IR



(d) Effect of l_1 on IR

Figure 6.6: Effect of design parameters on IR

6.2.3 Variation of mechanical advantage

For a pure heave motion, the variation of torque with the change in input angles is easily obtained from the dynamic model (see Section 5.2). The ratio of force applied by the payload to the torque required to equilibrate it gives an indication of the mechanical advantage of the system. If the active joint positions are all different, then we obtain different torques in each motor which makes it difficult to analyse the change in torque at different input angles. Hence we assume a scenario where all the motors have the same joint positions at each time so that the variation of torque with change in active joint positions can be studied easily. In a static scenario, joint rates are all zero, hence

$$\tau_\theta = G_\theta \quad (6.2.39)$$

From Section 5.2, for the above case, we find:

$$\begin{aligned} \tau_1 = & \frac{gl_1}{2} \left(-m_4 \cot \gamma_1 \sin \theta_1 - \frac{2}{3} (m + m_5) \cot \gamma_1 \sin \theta_1 + m_1 \cos \theta_1 \right. \\ & \left. + 2m_2 \cos \theta_1 + m_3 \cos \theta_1 + 2m_4 \cos \theta_1 + \frac{2}{3} (m + m_5) \cos \theta_1 \right) \end{aligned} \quad (6.2.40)$$

This need not necessarily depict the worst case scenario for torques required at the actuator for the given actuator angle; however, it gives us an indication of the variation of torques in general with change in input angles. The change in the nature of this plot with change in design parameters is noted.

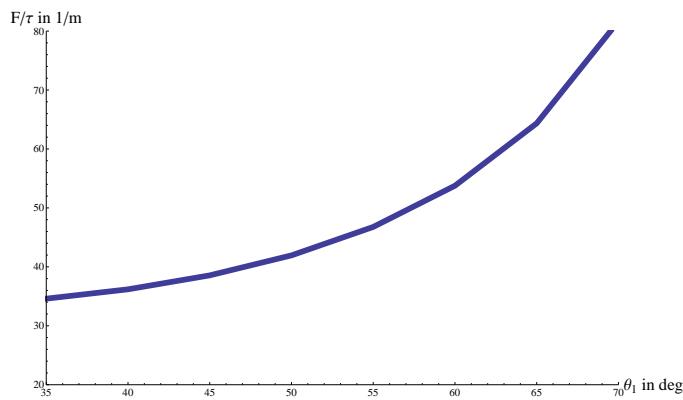
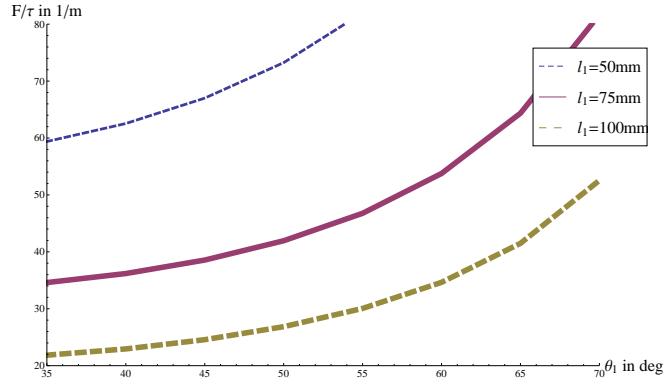


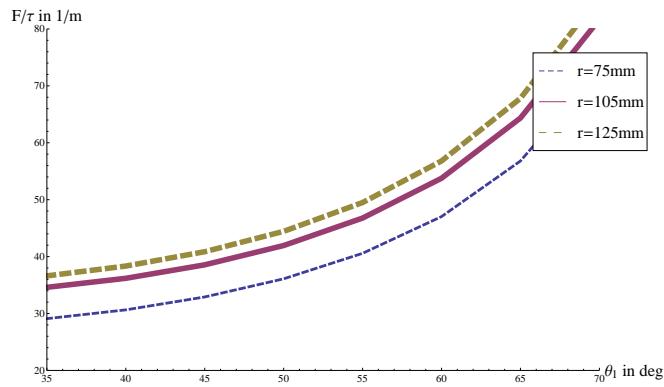
Figure 6.7: Variation of mechanical advantage with change in input angle

- The mechanical advantage is higher when the strut is closer to the rocker (see Fig. 6.8(a)).
- We observe that higher values of strut length increases the mechanical advantage(Fig. 6.8(b)).
- Decreasing the value of l_1 greatly increases the mechanical efficiency while r_t and l_0 exhibit opposite behaviour(Fig. 6.8(c), Fig. 6.8(d), Fig. 6.8(e), Fig. 6.8(f)).

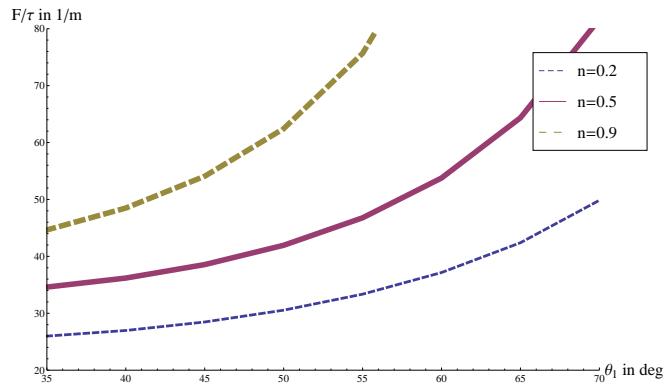
To further substantiate the study, by fixing the position of two of the motor angles, the third angle alone is varied and the torque variation is studied. Keeping θ_2 and θ_3 fixed at 35° and 45° respectively, θ_1 is varied. The torque required to maintain the configuration is higher at lower input angles and decreases with increase in input angles (see Fig. 6.9).



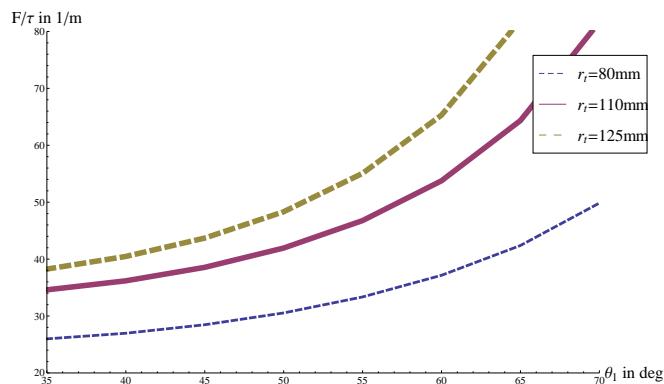
(a) Effect of l_1 on mechanical advantage



(b) Effect of r on mechanical advantage



(c) Effect of n on mechanical advantage



(d) Effect of r_t on mechanical advantage

Figure 6.8: Effect of design parameters on mechanical advantage

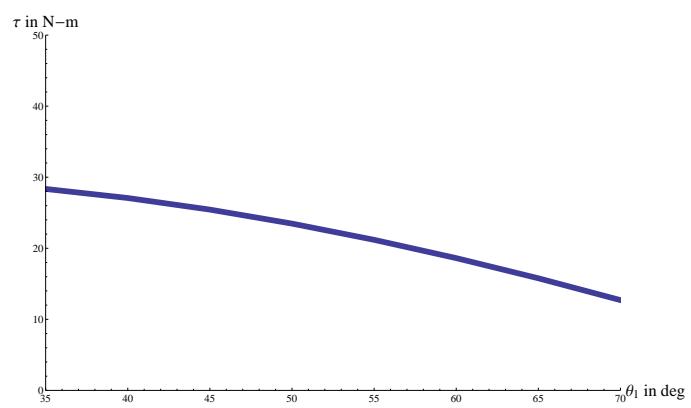


Figure 6.9: Variation of torque in motor w.r.t. crank angle

6.3 Dynamic visualisation of behaviour of performance indices

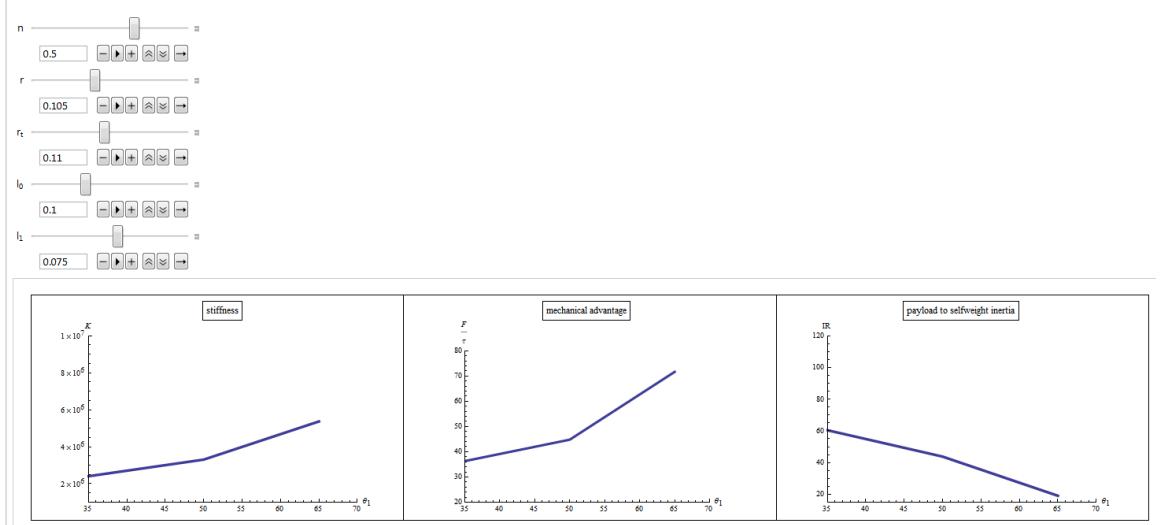


Figure 6.10: Dynamic visualiser to visualise behaviour of performance metrics

A dynamic visualiser was developed using **Mathematica** to understand behaviour of all the performance-metrics under study simultaneously. The motivation behind doing this was to see if there was any definitive trend that could be observed so that some of the metrics can be clubbed together and studied. The dynamic visualiser (see Fig. 6.10) has a provision to vary the values of design parameters, so that at each chosen value of design parameter, the variation of each performance-metric w.r.t. crank angle can be observed. It is observed that stiffness and mechanical advantage go hand in hand while IR behaves in an opposite fashion. The advantage of this study is that, when designing MaPaMan for any requirement, the performance metrics with similar behaviour can be clubbed together. The values of the design variables offering the best compromise between the metrics having opposing nature can then be chosen instead of studying the effect of each variable on each metric; which can reduce the efforts spent in the design process. The dynamic visualiser also helps us visualise as to which of the design parameters cause greater change in nature of each performance metric w.r.t. the others. This can aid in choosing the most critical design variables for any given requirement such that varying its value causes greater influence on one

of the metrics more than the others. For example, n has a greater impact on the mechanical advantage than the other two metrics. So increasing the value of n profoundly influences the mechanical advantage, while the stiffness doesn't change much.

6.4 Optimization for desired range of motion

The contour plots in Section 4.2 give us an idea about the theoretical limits of tilt possible in the manipulator. However, practical limitations such as link interference, joint angle limits etc, can further reduce the range of motion. In order to find out the practical range of motion for a given set of link lengths, a genetic algorithm(GA) based optimization was carried out.

Genetic algorithms have been used widely for solving global optimisation problems as they typically explore the search domain better than gradient-based methods that converge at local minima closest to the initial guess (e.g. Deb, 1995). In the present optimization scenario we have two objectives and nineteen constraints, and a large design space. In concurrent studies carried out by Badduri (2012) it has been found that GA-based optimization tools like Non Sorted Dominating Genetic Algorithm (NSGA-II) (Deb, Agrawal, Pratap, and Meyarivan, 2002) can be very effective in solving such problems. The optimization problem at hand is treated as follows:

$$\begin{aligned} & \text{Minimize} && f_i(x) \\ & \text{subject to} && g_i(x) > 0 \end{aligned}$$

with variable bounds

$$a_i < x_i < b_i, \quad i = 1, \dots, n$$

The first objective is to maximise the tilt of the end-effector as expressed in terms of Rodrigues parameters c_1 and c_2 . Following the approach demonstrated in Section 4.2.2, $c_1 = \cos \varsigma \tan(\frac{\varphi}{2})$ and $c_2 = \sin \varsigma \tan(\frac{\varphi}{2})$. Thus the aim is to maximise tilt φ of the end-effector. The second objective is to maximise the range of heave motion which can be directly related to the lengths of the crank and the strut as shown

in Fig. 6.11.

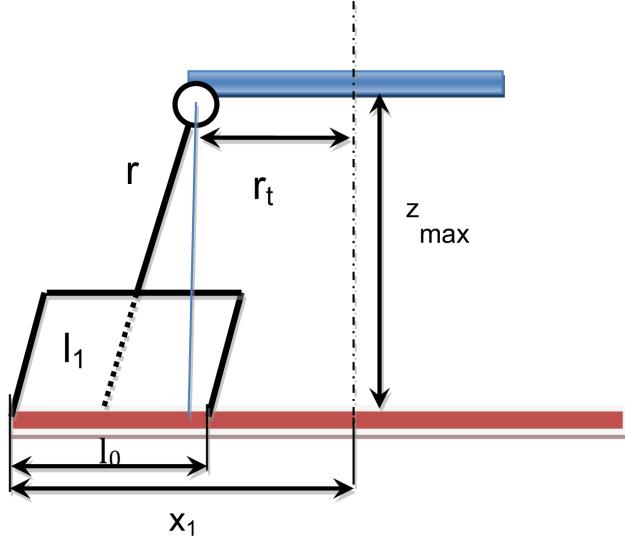


Figure 6.11: Geometrical condition for maximum heave in MaPaMan-I

$$f_1(x) = -\varphi$$

$$f_2(x) = -z_{max}, \text{ where } z_{max}^2 = \left((l_1 + r)^2 + \left(x_1 - \frac{l_0}{2} - r_t \right)^2 \right)$$

and the constraints $g(x)$ includes the singularity condition obtained in Chapter 4.2, joint limits for the crank, strut as well as the spherical joint. The following are the constraint conditions:

- Configuration should be non singular;

$$g_1(x) = S(\gamma)^2 - \epsilon, \quad (6.4.1)$$

ϵ is chosen to be 10^{-4} on the trial and error.

- There is a limitation on crank angles based on physical joint limits. The limits can be decided based on the physical design of the manipulator; however for the sake of illustration, we have chosen the following limits.

$$g_2(x) = \theta_1 - \frac{\pi}{6} \quad (6.4.2)$$

$$g_3(x) = \frac{\pi}{2} - \theta_1 \quad (6.4.3)$$

Similarly we find $g_4(x), g_5(x), g_6(x), g_7(x)$ corresponding to θ_2 and θ_3 .

- The strut angles are limited by physical joint limits. The range of these values can be decided based on the physical design, however we have chosen the following limits for the sake of illustration.

$$g_8(x) = \gamma_1 \quad (6.4.4)$$

$$g_9(x) = \frac{3\pi}{4} - \gamma_1 \quad (6.4.5)$$

Similarly we find $g_{10}(x), g_{11}(x), g_{12}(x), g_{13}(x)$ corresponding to γ_2 and γ_3 .

- Spherical-joint limits: The spherical joint has a restricted range of motion

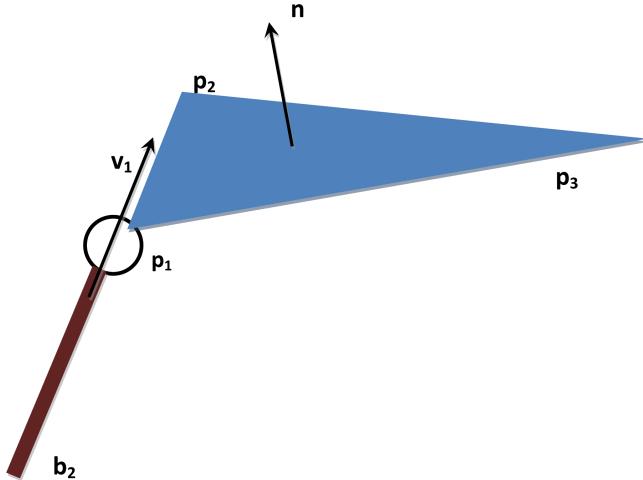


Figure 6.12: Geometrical condition for angle subtended by spherical joint at the end-effector in MaPaMan-I

due to its mechanical construction and hence the angle (α_i) subtended by the spherical joint with the end-effector adds constraints to the motion as shown in Fig. 6.12. This angle is computed by finding a unit vector along the direction of strut(v_1) and measuring the angle subtended by it with the unit normal vector (n) to the end-effector.

$$\mathbf{v}_1 = \frac{(\mathbf{b}_1 - \mathbf{p}_1)}{r} \quad (6.4.6)$$

$$\mathbf{n} = \frac{(\mathbf{p}_1 - \mathbf{p}_2) \times (\mathbf{p}_1 - \mathbf{p}_3)}{\sqrt{3}d_t^2/2} \quad (6.4.7)$$

$$\alpha_1 = \arccos(\mathbf{n} \cdot \mathbf{v}_1) \quad (6.4.8)$$

Hence the constraint functions are:

$$g_{14}(x) = \arccos \left(2(\mathbf{b}_1 - \mathbf{p}_1) \cdot \left(\frac{(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)}{\sqrt{3}rd_t^2} \right) \right) \quad (6.4.9)$$

$$g_{15}(x) = \frac{\pi}{3} - \arccos \left(2(\mathbf{b}_1 - \mathbf{p}_1) \cdot \left(\frac{(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)}{\sqrt{3}rd_t^2} \right) \right) \quad (6.4.10)$$

Similarly we frame constraints $g_{16}(x)$, $g_{17}(x)$, $g_{18}(x)$, $g_{19}(x)$ corresponding to \mathbf{p}_2 and \mathbf{p}_3 .

The design parameters are $\mathbf{x} = (l_0, l_1, r, r_t, \varsigma, \varphi, z_c)^T$. The variable bounds for the design variables are given in Table 6.1.

Table 6.1: Variable ranges used in the optimization study

Variable	Lower limit	Upper Limit
l_0	60	125
l_1	60	100
r	75	110
r_t	90	125
ς	-3.14	3.14
φ	-3.14	3.14
z_c	140	180

Based on the parametric study carried out by Badduri (2012), the optimal parameters chosen are for NSGA-II are shown in Table 6.2.

Table 6.2: Values of parameters used in the optimization study

Parameter	value
N_{pop}	400
N_{gen}	400
Probability of crossover	0.7
Probability of mutation	0.7
SBX parameter	14
Coefficient of mutation	35

We obtain about 100 solutions for this optimization problem. The Table 6.3 shows the results of a few of the 100 solutions obtained. Certain key observations of the optimization results are, maximum tilt= -1° and it is achieved at low heave values.

Table 6.3: Optimization results

	l_0	l_1	r	r_t	ς	z
	125	96.86	121.5	110	2.7	145
	125	96.56	103	110	2.8	147.2
	125	84.8	109.6	110	2.01	142.3

Lower values of r_t and high values of l_0 , r and l_1 produce good range of motion.

6.5 Conclusion

In this Chapter, we defined a few performance metrics and studied the variation of those metrics on changing the values of design parameters. A dynamic visualiser was created to study the parameter space systematically, to understand cooperative/competitive nature of pairs of metric variations. The metrics are sensitive to certain variables more than the other; these are identified to be able to study from a smaller set of variables. The impact of the design variables on various metrics are studied.

An optimiser was used to study the effect of design variables on the range of motion. It was used to find the best set of dimensions for a desired range of motion of the platform, taking into account geometric constraints like singularities, and design constraints like joint limits, joint intersections etc. The aim of creating the optimiser module was not to optimise the dimensions for a specific problem, but to provide a platform where constraints arising specifically due to the design can be plugged into the code and upon choosing the appropriate values of variables, characteristic to GA, the best link lengths can be obtained. As observed in the example, we obtain several optimal solutions, hence we can first truncate the values to desired position after decimal and then perform a local optimisation at these points to obtain the optimal solution. Alternatively more constraints can be added from dynamics or kinetostatics etc to reduce the solution space. Detailed study of this however falls beyond the scope of this report as the contribution of

this report is only towards creating a framework for optimising the link dimensions in order to achieve maximum range of motion,

CHAPTER 7

Prototype

7.1 Introduction

In the previous chapters detailed studies have been carried out to analyse the kinematics of the manipulator, singularities encountered, dynamics, effect of various design parameters on certain performance metrics of the manipulator etc. To validate most of these studies as well as adopt the control strategies developed in concurrent studies by Mehta (2012), a physical prototype was required. It was decided to first build a prototype of MaPaMan-I as most of the theoretical studies were first carried out on this configuration of MaPaMan-I to begin with. This Chapter discusses the design details of two versions of prototypes of MaPaMan-I that were designed and fabricated. The first prototype was built to mimic the kinematic model, while the second prototype has a number of sensors embedded in it for error correction and real-time control. The electronic and software interfaces that were used to get the manipulator running are also discussed in brief at the end of the Chapter.

7.2 Prototype-I

It was decided that a desktop prototype would be most ideal to fabricate and perform experiments on. Therefore, the dimensions of the prototype were decided assuming it would fit in a cube of side $1m$ when fully assembled. Based on past experience of working with fabrication of manipulators, it was decided that the prototype would be built completely out of metal; preferably steel because:

- If plastics are used for fabricating the links, then they would be compliant and introduce inaccuracies to the system.

- Aluminium could be an option for building the prototype, but since most of the components like bearings and fasteners are fabricated using steel, the compatibility between these two materials become a problem, aluminium being a softer metal.
- Steel on the other hand would make the links heavy, but it would have very high load carrying capability and it is easy to weld and can be machined to great accuracies to obtain desired dimensions.
- Also steel is readily available and relatively inexpensive.

7.2.1 Design details

The first design decision taken was the selection of actuators. Maxon-RE150 motors were chosen as the actuators for the prototype, because they were already available at the Robotics Lab, Engineering Design. Based on the torque capacity of the motor and the maximum space limitation, the following target specifications were decided upon.

- The target load to be carried = $5kg$
- range of motion in roll = 30°
- range of motion in pitch = 30°
- range of motion in heave = $8cm$

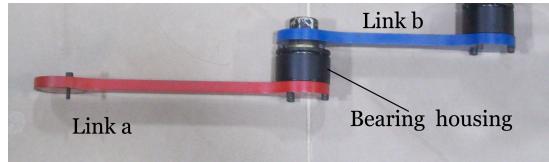
The target specifications were chosen to be able to compete with a scaled down version of an existing design of a commercial 3-DoF manipulator by Inmotion built to be used as a gaming chair. Based on the target specifications, dimensions of the links were chosen using the approach followed in Section 6.2. The dimensions shown in Table A.1, were the final dimensions selected such that the target load could be lifted when the motors works at 25% of its full torque capacity.

The design of MaPaMan-I has 12 passive rotary joints and three spherical joints. Hence the various options available to design these were first studied.

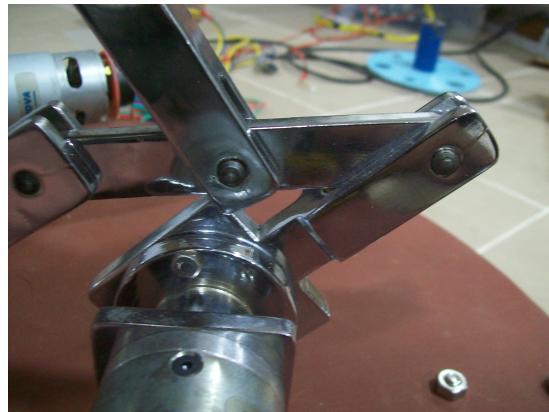
1. **Rotary joints:** Two typical designs for rotary joints are shown in Fig. 7.1 and Fig. 7.2. The following are some details about each:
 - The first design involves two links coupled by a unit that houses a bearing. The two links lie on parallel planes at all points of time and hence they never intersect each other(see Fig. 7.1(b)). This design

provides complete freedom for range of motion between the two links. The drawback of this design however is that the load applied on one link is taken in a cantilevered fashion by the pin joining the inner-race of the bearing and one of the links. This means there is possibility for deflection of one of the links w.r.t. the other at the pin. Also by virtue of their construction, bearings typically have a single row of balls which allows for a small misalignment, causing the links to have a small out of plane motion w.r.t. each other.

- The second design involves a fork arrangement, wherein one of the links is trapped inside the fork of the other. A bearing is placed on the inner link and a pin connecting the fork and the inner-race of the bearing completes the assembly as shown in Fig. 7.1(b). This design can take more loads than the previous design as the pin is not more cantilevered. Also the out of plane bending due to inaccuracies in the bearing is reduced in this design, as the inner link is trapped in the fork and can move out of plane only subject to the tolerances provided in while fabricating the slot in the fork. The demerit of this design is that the links lie on the same plane and hence are subjected to link-interferences. This reduces the relative motion of the links w.r.t. each other.



(a) Links in parallel planes connected by a bearing



(b) Fork type arrangement of links connected by a pin

Figure 7.1: Candidate designs for the revolute joints of MaPaMan

2. **Spherical joints:** There are two typical solutions that are generally employed when spherical joints are needed. These are shown in Fig. 7.2. The following are some details about each:

- The first solution is a rod-end bearing. It is typically used as an alternate to rotary joints to take care of misalignments in assemblies. The rod-end bearings are designed to provide full rotation in one axis and a slack of about 15° in the other two axes. As a result the range of motion is very low for these joints.
 - Commercial manufacturers like “Hephaist” fabricate precision spherical joints for parallel manipulators. These can undergo a full rotation about one axis and 60° of motion about the other two axes. The motion is smooth and the joint can take considerable loading. However the greatest disadvantage of this solution is that the joints are very expensive (Rs 40,000 a unit). Further more these need to be imported which adds on to the cost.

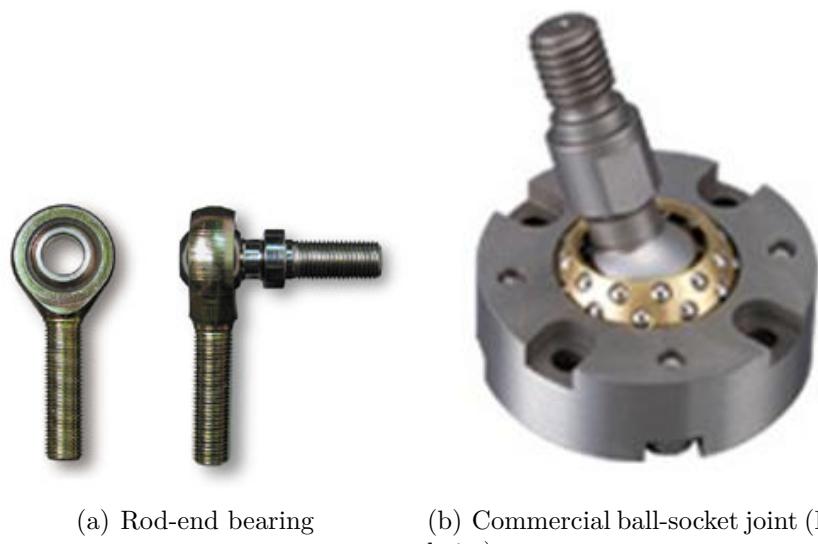


Figure 7.2: Typical designs of spherical joints

Based on an analysis of various options in hand, it was decided to use a fork-like arrangement for the rotary joints. This was because, for the dimensions of links chosen, the displacement of the crank required to produce the required range of motion is not very high. A suitable slot-length can be provided in the fork to enable the links to move the required range of motion, hence avoiding link interferences in the desired work-space. Fig. 7.1(b) shows the design at the coupler-rocker joint b_1 .

A new solution was conceived of for the spherical joint as rod-end bearings had a range of motion that was too low for the purpose in hand and the commercial joints were very expensive. Spherical joints were fabricated in-house to reduce cost

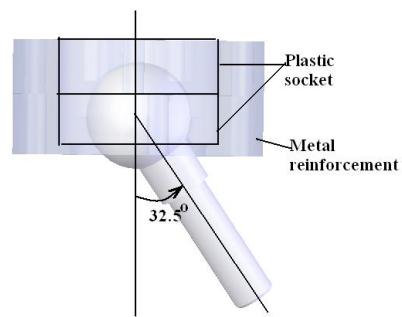
as well as obtain desired range of motion. The designed spherical joint has two parts, the socket and the ball. The socket needs to provide an accurate spherical groove to guide the ball properly. The socket was hence manufactured to very high precision ($\approx 0.1mm$) using ABS plastic on a Rapid Prototyping machine using Fused Deposition Molding technique. The socket was then reinforced with a casing made of mild-steel. This ensures that the socket provides the right contour for the ball to move while the metal reinforcements ensure that it withstands the loading. The ball that stays arrested in the socket was made out of a bearing ball. The ball is too hard to be machined and a hole was made in it using EDM. A rod was then inserted into it and fit tight. The rod is threaded at the other end to fix it onto any link and a flat portion for the spanner to tighten. The entire assembly is shown in Fig. 7.3. This solution is almost as inexpensive as the rod-end bearing and has a range of motion similar to the commercial spherical joint (65° in 2 axes and 360° in one axis).

Thus using the fork-type arrangement for rotary joints and the in-house fabricated spherical joints all the components were fabricated (refer to appendix() for the details of the CAD drawings of each link). The following are some nuances of the design details of individual parts:

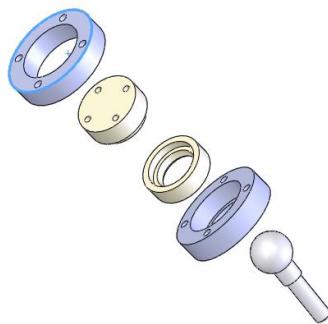
- Crank: The crank is designed as shown in Fig. 7.4. It is coupled to the motor through a coupler, which is fit to the motor shaft with a key and connected to the crank with nuts and bolts.
- Rocker: The rocker is similar in design to the crank. The only difference is that it is not actuated, hence it is arrested in the fork of the rocker-stand. It houses a bearing whose inner race is connected to the rocker-stand through a pin.
- Coupler: The coupler is the link that is arrested in the fork of the rocker, crank as well as the strut. It has three bearings tight-fit to it and then connected to the rocker, crank and strut through pins.
- Strut: The strut stalk is designed as the fork that meets the coupler at one end, and houses the rod of the ball-joint at the other end. It is provided with a bend as shown in Fig. 7.5 in order to increase the range of motion. This is because, the dimensions of the links are such that the strut is completely stretched out when the crank angle is 88°; which means, the motion of the crank on either side of 88° would produce an identical pose of the end-effector and hence the crank is chosen to move only on one direction. This implies that if the strut was a straight link, then the range of motion of the spherical



(a) Spherical joint designed in-house



(b) Range of motion



(c) Assembling the spherical joint

Figure 7.3: Typical designs of spherical joints

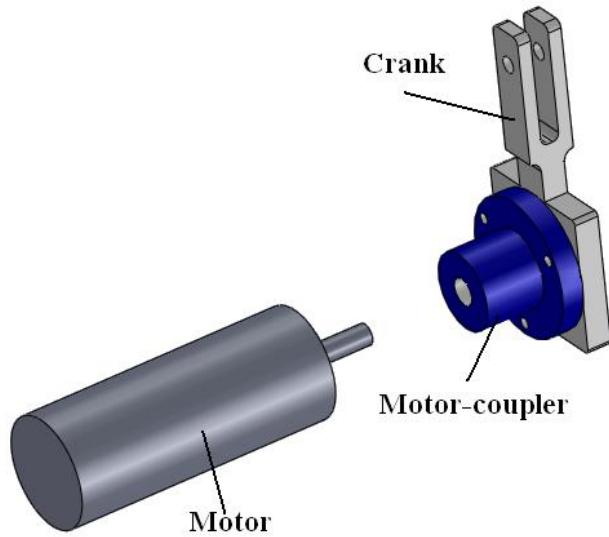


Figure 7.4: The crank-motor assembly

joint that would be utilised would be half of its total capacity. Thus it is bent as shown in Fig. 7.5 to utilise the entire range of motion provided

by the spherical joint. It has a blind threaded hole at one end, where the rod-end of the ball joint can be screwed in place.

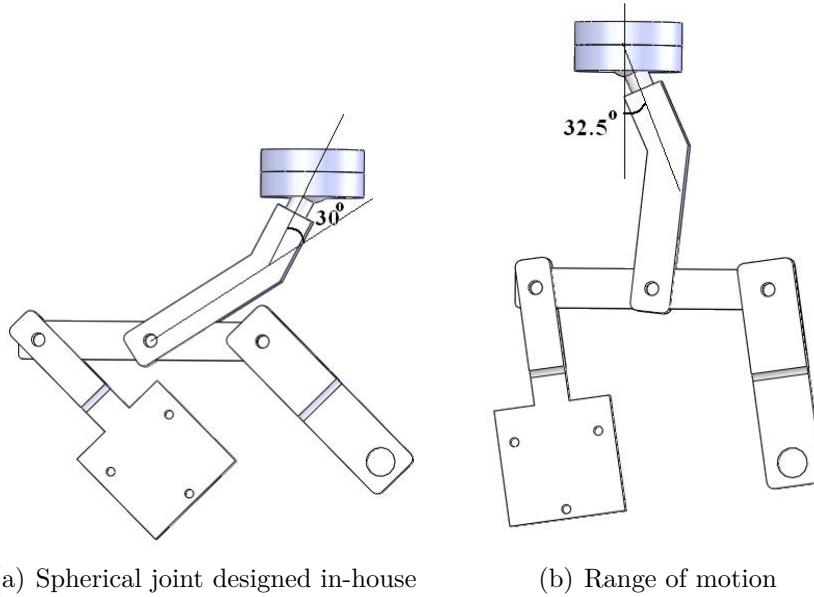


Figure 7.5: The design of strut

- The crank, rocker, coupler and the strut are all manufactured from mild-steel using CNC milling to obtain good accuracy of required dimensions.
- Due to the fork arrangement, metal surfaces may rub over each other due to manufacturing inaccuracies. To reduce friction that can arise due of this, the surfaces are chrome-plated. Chrome-plating makes the surfaces hard and gives them a smooth finish.
- The base is made heavy to withstand the reaction forces from all the legs and to keep the system from shaking when in motion. Three set screws are provided to adjust for its orientation with the ground to maintain parallelism. Wheels are attached at the base to easily move the setup as shown in Fig. 7.6.

The fabricated components were assembled as shown in Fig. 7.7. All the geometric and inertia parameters, used for experiments, are retrieved from the respective CAD models. It is to be noted that the encoders present in the motors are of incremental type and hence the actual value of motor angles when all the links are all in the base state(see Fig. 7.8) are obtained from the CAD model and are incremented here after using the sensor readings. An inertial measurement unit(IMU) is used to measure the values of the link angles in the actual prototype

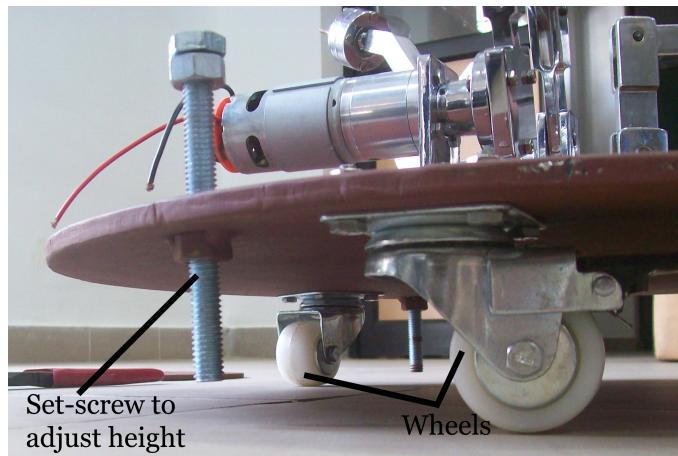


Figure 7.6: The design of base



Figure 7.7: The assembly of prototype-I

and cross-check them with the angles measured from the CAD model to compensate for any minor manufacturing inaccuracies.

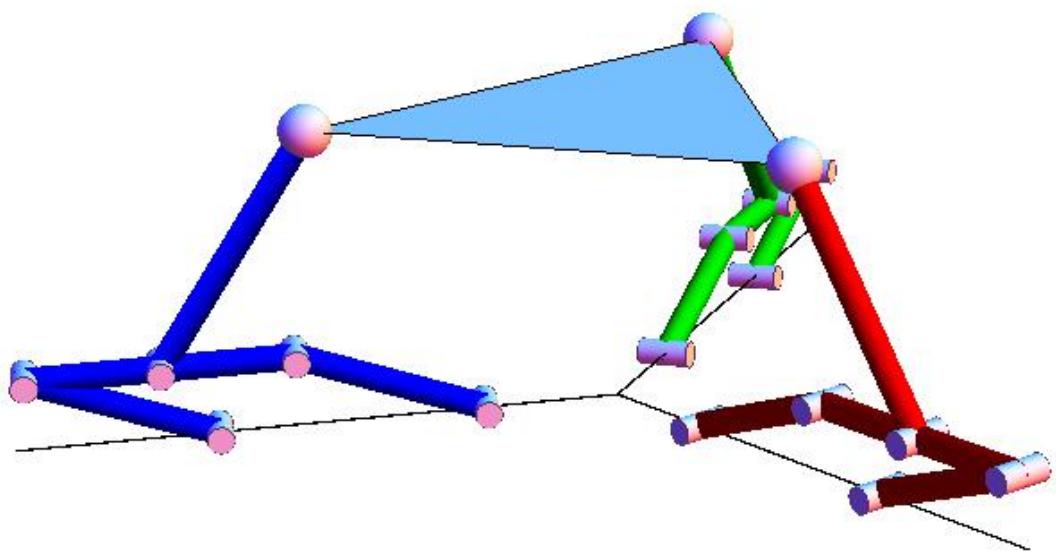


Figure 7.8: The base position of prototype-I

7.3 Prototype II

The first prototype had one serious drawback; that of backlash. The encoder present in the motors are placed such that they do not observe the backlash coming from the gear-box. This means that when the encoder reads a position, it is not necessarily the position of the crank. Thus to compensate for this, another encoder was used at the rocker end in prototype-II. This encoder would read the position of the rocker which would be the same as that for the crank (as the four-bar is a parallelogram in this case) and then correct for itself from the reading of the motor-encoder.

While performing a closed-loop control, it is required to perform a forward kinematic computation at each step as was pointed out in Section 3.7, this is a time consuming task and hence an incremental approach to forward kinematics was developed(see Section 3.7). However another approach to solving this problem was conceived, by adding a sensor at the joint between the coupler and the strut and directly sensing it rather than computing it (refer Mehta (2012)). Hence another sensor was added in each joint b_2 in prototype-II. Except for adding a total of 6 sensors to the set-up, the design of the links remain mostly the same in prototype-II as in prototype-I.

7.3.1 Design details

Most of the parts have the same design, while minor alterations are made to some of them to accommodate the sensors. The sensors used are incremental encoders of metallic type. They have a rotating shaft and a fixed threaded portion. The design alterations basically accommodate the sensors in such a way that the shaft is fixed to one of the links while the threaded portion is fixed to the other w.r.t. which the connected link moves. Following are some nuances of the design details of the components:

- An encoder is fixed at the rocker-stand joint to read the position of the rocker. The pin that is used in the earlier design remains fixed to the forks of the stand while the rocker moves w.r.t. the pin due to the bearing. Such a design would pose difficulties in accommodating the encoder, hence the

design is modified a bit. The bearing is removed from the rocker and two bearings are added to the forks of the rocker-stand. A square hole is made in the rocker through an EDM process and a pin is inserted. The pin now stays fixed w.r.t the rocker and moves on the bearings at the fork. The pin has a provision to accommodate the shaft of the encoder, where it lies inside, held by grub-screws. The threaded portion of the encoder is connected to the rocker stand through a connecting plate with the help of nuts and bolts as shown in Fig. 7.9. Thus when the rocker moves w.r.t. the rocker stand, the encoder shaft moved w.r.t. the threaded portion and the rotation is sensed.

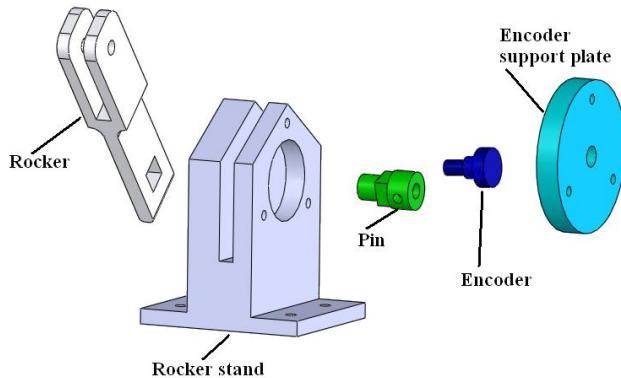


Figure 7.9: Accommodating the encoder at the rocker

- A similar philosophy is used for placing the encoder at the joint between the strut and the coupler. However the exact same implementation as that for the rocker cannot be done due to space constraints. There is lack of enough space on the forks of the strut to house two bearings, hence only one bearing is used while the other side is made smooth using chrome plating. The pin is also chrome plated so that these two surfaces move easily over each other. Adding a thin nylon washer was first contemplated, but later abandoned when it was realised that the two chrome plated surfaces worked as well as expected. The coupler has a square hole just like the rocker and a pin is inserted to connect the fork of the strut to the coupler. The encoder shaft is inserted and held using grub-screws while the threaded portion is connected to a plate that is then connected to the strut.
- Due to the modifications made to the rocker stand, it now becomes wider than earlier. This makes it difficult to place the three rocker-stands as close to each other as they were before. Hence they are moved out a little and the x_1 value now becomes 150mm from 135mm.
- Increasing the x_1 value decreases the range of motion a little. To compensate for it, the range of motion of the coupler w.r.t. the crank is increased. In the earlier design, there was link intersection at crank angle is 34.6° . By modifying the coupler as shown in Fig. 7.10, the link intersection now occurs at 21.49° circ crank angle. This more than makes up for the loss in range of

motion due to increase in x_1 . The final dimensions of the links are provided in Table B.1

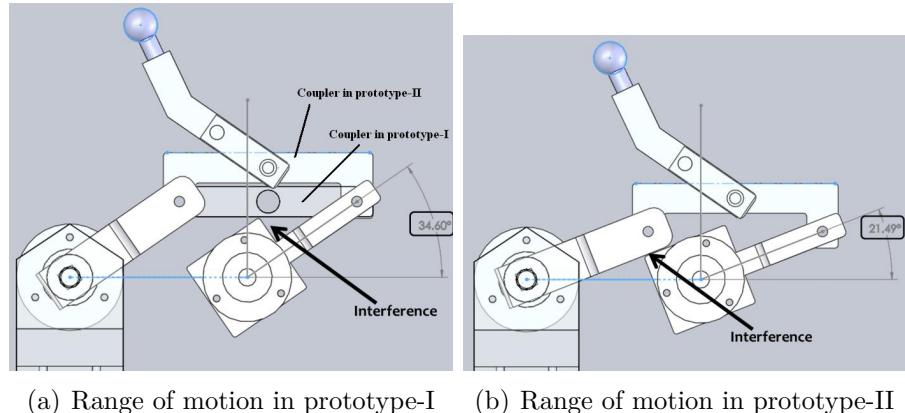


Figure 7.10: Modification in the design of the coupler link for increased range of motion

The components are assembled and the new set-up is shown in Fig. 7.11.



Figure 7.11: The assembly of prototype-II

7.4 Details of the encoders

For correcting the backlash, incremental encoders were placed at the rocker end. The choice of these encoders were made as follows:

- The error due to backlash was measured to be about 2° (refer Mehta (2012)). To correct this, we need the resolution of the encoder to be smaller than 2° .
- To make the encoder compatible with the rest of the prototype we needed it to be made of metal.
- It is difficult to obtain absolute-type encoders with high resolution and so incremental encoders were chosen. Among incremental encoders, quadrature-type encoders were chosen as they provide data about direction of motion which is important in our application.
- SE-5, of US-Digital matched our requirements as they are metallic, quadrature-type and offer 400 pulses per rotation. In a quadrature encoder, each clock pulse gives 4 readings, hence $400ppr = \frac{360}{400 \times 4}$ which tantamounts to 0.25° .
- Incremental encoders with higher resolution are also available, but due to cost-constraints, SE-5 was purchased.

Upon choosing the encoders, it is also important to decide how to read simultaneously from 6 of them and send that data to a CPU for processing and controls. A 1047-PhidgetEncoder HighSpeed 4-input IC, was used to perform this objective. The data is read parallelly from all the 6 encoders, packaged into a packet and then sent serially via the USB port to the computer. It has a high count rate of 2,50,000 counts per second, which means it can read from an encoder with 1600 counts per second data as long as the speed of the strut is less than 10,000rpm. As this suits our requirement completely we use phidget to read from the encoders.

7.5 Details of the IMU

An IMU was used to find out the actual inclinations of the links at base position to cross-check them with the values obtained from the CAD model. Also IMU was used to find out the horizontality of the base which could be adjusted with the help

of set-screws. Vector Nav was the IMU chosen for this purpose because it has an accuracy as high as 0.5° in roll and pitch in static sensing; and all the sensing we required was to be done in static scenario. Also it has a dynamic accuracy of $\leq 2^\circ$, along with a tunable Kalman filter; which would be useful in performing real-time control based on its readings. Though that work is beyond the scope of this report, the same IMU might be useful for such applications in future. Chapter 8 talks about an application of MaPaMan-I, which uses an IMU for dynamic sensing. The IMU has a 3-axis gyro, magnetometer and accelerometer. It measures rates of roll, pitch, yaw, surge, sway and heave and sends them in the form of serial data to the computer. It also performs an onboard integration and sends roll, pitch and yaw data also. However this data is subject to constantly building drift and it needs to be corrected once in a while through external means.

7.6 Details of the software used

The Table 7.1, provides the details of the software used for setting up the interface from computer to the controller and to drive the actuators:

Table 7.1: Software details

S No.	Software	Description	Purpose
1	Matlab	Version-7.0	Computing the desired trajectories etc.
2	Galil Smart Terminal	Galil controller software (revision-1.0 C)	Interface between Matlab and controller
3	Windows XP-SP3	Operating system (32-bit)	Runs the Matlab code, interfaces the input/output signal
4	Phidget Terminal	Encoder reader	Reads from multiple encoders, Interface between encoders and Matlab
5	Vector Nav	Inertial measurement unit	Measures roll, pitch, yaw and their rates

7.6.1 Details of the electronic components

The details of the actuators, controller, driver etc., used in experiments are given in table 7.2:

Table 7.2: Devices used in experiments

S No.	Part	Model	Specifications	Number
1	D.C. motor (Graphite brushes)	Maxon RE40,218009	150 Watts, Nominal torque - 0.177 Nm	3
2	Encoder (Incremental type)	Maxon L250,225787	1024 PPR	3
3	Gearbox (Planetary gearhead)	Maxon GP42C,203123	Gear ratio - 74:1	3
4	Controller	Galil DMC-1846	4-axis, PCI-based motion controller	1
5	Amplifier	Galil AMP-19540	4-axis, 500 Watts per axis	1
6	Communication cable	Galil Cable-100-4M	100 pin HD cable	1
7	Computer	Intel(R) core 2Duo	E8500, 3.16 GHz 2GB RAM	1
8	Power supply	VI microsystem Elpower	30V, 5A	1
9	Encoder (Incremental type)	US Digital S4	400 PPR	6
10	Encoder reader IC	Phidget 1047	Time resolution $1\mu s$	2
11	Inertian Measurement Unit	Vector Nav VN-100 rugged	Orientation accuracy 0.5° static condition 2° dynamic condition Sensor sample rate 120kHz	1

7.7 Conclusion

Two prototypes of MaPaMan-I were designed and fabricated based on certain target specifications. The prototypes were used to test control strategies developed in concurrent studies by Mehta (2012). The basis for designing was the dynamic analysis carried out by Mehta and the studies on parametric variations (see Chap-

ter 6). The actuators in the prototype were interfaced with a CPU through Galil controller, while the data from the encoders were read using a Phidget module. The prototype-II was built as an improvement over prototype-I with a number of sensors for error reduction and realtime control. Section 8 discusses the details of a couple of applications of prototype-II.

CHAPTER 8

Developing MaPaMan-I for motion simulation applications

8.1 Introduction

A prototype of MaPaMan-I was developed with features like backlash correction and sensor based realtime control. The details of the same can be found in Section 7.3. A number of control strategies were developed for trajectory tracking of MaPaMan-I by Mehta (2012) and were implemented on the prototype. Based on inputs from his work, the prototype was developed into a motion simulation platform for two purposes:

1. To simulate offline trajectories of different vehicles
2. To mimic motion of a joystick in realtime

8.2 Interfacing

A “dual-loop” control strategy has been developed to control MaPaMan-I. Details on this can be found in Mehta (2012). However a brief explanation of this scheme is presented here.

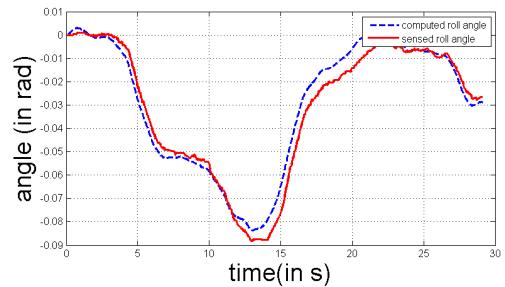
A desired trajectory is chosen in terms of roll, pitch and heave. Inverse kinematic computations are performed to compute the desired motor angles. These values are sent to the motor via the motor controller which internally has a PD control. The motor tries to attain its position but due to backlash in the gears, etc. the crank does not reach the desired position. Hence the reading from the rocker is noted along with the reading from the sensor at the strut-coupler joint. Using these readings, upon back computation, sensed values of roll-pitch and heave are

obtained. An integral control (dual loop) is applied to correct the error between the desired and sensed values of the task-space. The corrected values are then subjected to inverse kinematic computations to obtain new values for the motor angles thus completing a control loop.

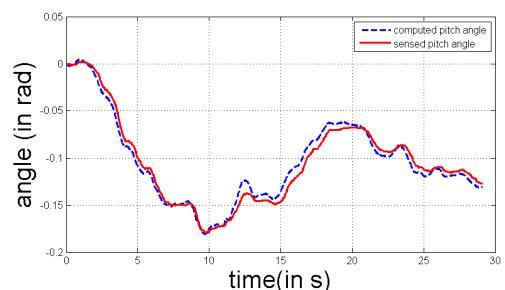
An analysis was carried out to compute the maximum theoretical error possible in the measurement of task-space coordinates given the resolution of the encoders used for error detection (Mehta, 2012). Based on the analysis, we obtain 0.5° to the max theoretical error in roll and pitch and $0.8mm$ as the error in heave. A number of experiments have been carried out to track various mathematically well-defined trajectories with the dual-loop scheme yielding satisfactory results. However it was decided to test the capability of the prototype to track trajectories that are not so well-defined; as most motions in real life can hardly be represented by simple mathematical functions. An IMU was used to dynamically measure the roll and pitch angles of a bicycle and a skateboard under motion and MaPaMan-I was made to follow that trajectory.

Simulate offline trajectories of different vehicles

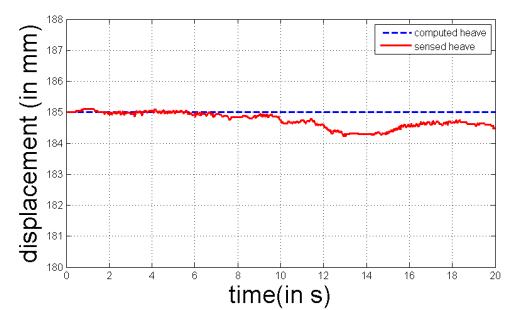
Roll angle and pitch angle data were collected from the IMU and stored offline. The noise in the data was filtered using a simple low-pass filter using Matlab. The data was then fed into the code for dual-loop control of MaPaMan-I. Data from a number of trajectories are obtained and simulated. The results for one motion each for bicycle and skateboard are illustrated here. Fig. 8.1 shows the experimental results obtained, when MaPaMan-I is made to track a trajectory in roll and pitch obtained from a bicycle. The rms errors in roll and pitch angles are 0.29° and 0.4° , while heave shows an rms deviation of $0.4mm$.



(a)



(b)



(c)

Figure 8.1:

Similarly Fig. 8.2 shows the simulation results for roll and pitch variations obtained by simulating the motion of a skateboard. Note that the skateboard only has one-DoF; but the roll-axis of the IMU is not aligned to the roll-axis of the skateboard to measure both roll and pitch at each instant. The rms error in tracking roll angle and pitch angle are 0.27° and 0.29° respectively. Though the heave input to the system was zero, the system was made to track the desired roll and pitch at a fixed heave = 185mm. The rms error in heave is 0.6mm. Note that all the rms errors obtained are below the theoretical accuracy of the system, hence validating that the system is fairly accurate.

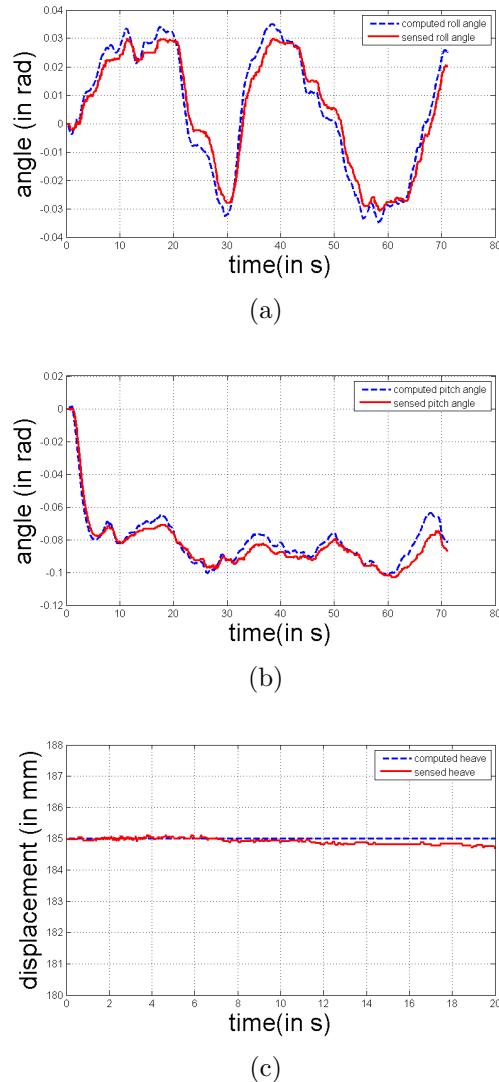


Figure 8.2:

Conclusion

The maximum errors obtained while simulating the non-periodic trajectories as obtained from a cycle and skateboard are typically of the order of 0.3° in roll and pitch and about $0.5mm$ in heave. Such low values of error are extremely good for accurate simulations of different types of motions. Hence this exercise consolidates the possibility of application of MaPaMan in the field of motion simulation and gaming chairs.

Mimic joy-stick motion in realtime

An Attack-3 joy-stick was interfaced with manipulator through a computer and its readings were read inside **Matlab**. The joystick has potentiometers in it and an on-board processor. Digital data about the absolute position is sent to the CPU via USB. The format of the packets sent from the joystick follow a standard protocol and can be directly accessed using the command `vrjoystick` in **Matlab**. When the stick is moved, we get values for roll and pitch that varies in $[-1, 1]$ in each case. This range is mapped to a desired range of roll and pitch. The joystick has a third axis that is used to obtain heave input. Upon mapping in a required fashion, the roll, pitch and heave data is subjected to a dual-loop control on the prototype. Thus MaPaMan-I is controlled realtime using a joystick. **Conclusion** Realtime trajectory tracking using joystick opens avenues for realtime trajectory tracking using various other input devices as well for various applications. This prototype can be used as a rehabilitation device, where the physiotherapist provides the required motion using a joystick and the patient's ankle which rests on the end-effector of MaPaMan-I mimics the motion and provides physiotherapy. If the joystick is made wireless and then interfaced, it could serve as a remote teleoperating surgical device. The high precision in the motion makes it suitable for fine orientation requirements in a telescope or a satellite antenna.

CHAPTER 9

Conclusion and future work

9.1 Overview

In this report, we have introduced a novel 3-DoF spatial parallel manipulator, MaPaMan, with reconfigurable degrees-of-freedom. MaPaMan has the advantage of producing swift motions unlike the 3-RPS from which it was designed, thanks to its linkage mechanism that contains no prismatic actuators or sliding joints. Also the design is such that the actuators are placed fixed on the ground, which means they can be replaced with actuators of any specification as per requirement without changing the entire mechanism. On top of all this, the mechanism can be mechanically reconfigured from one type of DoF to another type, which makes the design more versatile than most existing designs. A complete list of features of MaPaMan over existing manipulators is provided in Section 2.4.

In the next Chapter, kinematic formulations of MaPaMan-I is presented. An in-depth kinematic analysis of MaPaMan-I and MaPaMan-II; covering aspects of zeroth-order and first-order has been carried out. It is observed that we obtain a 16-degree univariate polynomial in closed form upon solving the forward kinematics problem of both MaPaMan-I and MaPaMan-II. The computations are all made symbolically till this stage to keep the formulation general throughout. Since any general 16th-degree polynomial cannot be solved analytically, we resort to numerical computations at this stage. The inverse kinematics problem of MaPaMan-I yields 8 different configurations for a given pose, however due to a more simplified geometry, we observe that the inverse kinematic problem of MaPaMan-II results in a single closed form solution. Adopting the approach used in Bandyopadhyay (2009), the partitioning in the DoF of MaPaMan-I and MaPaMan-II are found. As expected from geometry we obtain roll, pitch and heave as the 3-DoF in MaPaMan-I and roll, pitch and yaw as the 3-DoF in MaPaMan-II.

Chapter 4 discusses the gain-type singularities associated with MaPaMan-I. Gain-type singularities are of prime importance in parallel manipulator as they can lead

to loss of control of the end-effector even when in the workspace. The conditions for gain-type singularity were framed and an attempt was made to find the singular manifold in the task-space in closed form. The expression for the singular manifold was indeed obtained in closed form but it was very large in size to manipulate and study, and hence it was abandoned. An alternate approach was followed to visualise the singular manifold, by using contour curves. The contour curves are slices of the singular manifold taken at fixed heave values. The contour plots were generated in the r_{ts} space as well as in the roll, pitch and heave space. The singular manifold in the r_{ts} reveal 3 way symmetry which comes from the inherent symmetry of the manipulator. It easily helps us visualise the maximum theoretical tilt possible in the manipulator. The contour plot in the roll, pitch and heave space on the other hand give a more intuitive feel to the workspace. A convex region can be easily marked out in it to plan a path in which the manipulator never encounters singularity.

Parametric study of the manipulator architecture was carried out to find out the effect of variation of the design variables on different performance metrics of the manipulator. As there are no unique performance metrics for parallel manipulators in literature, three major new metrics were conceived of; namely stiffness, mechanical advantage and inertia ratio of payload to self-weight computed in a kineto-static sense. A dynamic visualiser was developed to understand the variation of the metric values with change in input angles, when dynamically changing the values for the design parameters. The dynamic visualiser helps us understand which metrics compliment each other in general and which dont. It is observed that stiffness and mechanical advantage compliment each other while *IR* behaves in an opposite manner. Thus studying just one of stiffness or mechanical advantage while designing the manipulator would be sufficient as it would result in us understanding both. Also we can find out which metrics are sensitive to which design factors, from the dynamic visualiser which again aids us in the design process. Finally a GA based optimization framework is developed to find the best set of link dimensions that can produce a desired range of motion taking into consideration design constraints like joint limits etc as well as geometric constraints like singularities. This framework can be extended to incorporate more constraints

based on the design of the manipulator. An example is illustrated to demonstrate the usefulness of this framework.

Based on the inputs from Chapter 6, two prototypes of MaPaMan were designed and fabricated. As most of the theoretical analysis were initially carried out for MaPaMan-I, it was decided to first build the prototype in the MaPaMan-I configuration. The first prototype was a demonstration of the kinematic model while the second prototype contains a number of sensors in it to correct for error arising from various potential sources and carry out dual-loop based, trajectory tracking control. The most important design challenge was to obtain a ball-socket joint with desired range of motion at an inexpensive price. A novel design for the ball-socket joint was conceived of and fabricated in house, that was cheap as well as met the design requirements in terms of range. The next challenge was to design all the rotary joints such that there are no out of plane deflections. A fork-type arrangement was chosen for the purpose as it not only reduced the possibility of out of plane motion, but also maintained parallelism between the links. The final important design challenge was to modify the design of the joints so that they can accommodate the rotary encoders. This problem was solved as shown in Fig. 7.9. The prototype-II was used to validate control schemes developed in concurrent studies by Mehta (2012). Prototype-II was made to track mathematically defined trajectories to great accuracies following a dual-loop based control scheme. In order to findout the capability of this prototype to track non-periodic trajectories, it was made to track trajectories obtained from real-life situations like riding of bicycle, skateboard etc. as shown in Chapter 8. The experimental results show that with dual-loop control, the required trajectories can be tracked to very high accuracies at par with theoretical accuracy of the system, that are computed based on precision of the sensors involved. The prototype was also interfaced to a joystick through a computer and the end-effector was made to mimic the motion of the joystick. The prototype successfully mimics the motion of the joystick, hence revealing its capability for usage as a rehabilitation device or teleoperating device etc.

9.2 Future work

Though this report deals with design, analysis and development of MaPaMan, most of the work presented is presented for MaPaMan-I. This is because, MaPaMan-I was given priority over MaPaMan-II as the former relates itself more to 3-RPS than the latter; 3-RPS being a well-known manipulator. Though the kinematics has been solved for both MaPaMan-I and MaPaMan-II, a geometrical implication directly suggesting the partition in DoF like in the case of MaPaMan-I (see Section 3.3) has not yet been obtained for MaPaMan-II. Such a relation coming solely out of geometry would have a lot of implications in path planning.

The physical interpretations of the singularity conditions can be obtained by following the approach presented in Basu and Ghosal (1997). One possible configuration of singularity in MaPaMan-I is presented in Fig. 4.1, where the strut is in the same plane as the end-effector. The singular manifold of MaPaMan-II on the lines of MaPaMan-I can be found. If it turns out to be very large, like in the case of MaPaMan-I, then it can be visualised numerically in the roll, pitch and yaw space or in (c_1, c_2, c_3) space. This would be a necessary step when a prototype of MaPaMan-II is built and is subjected to trajectory tracking.

The entire section on parametric variation studies can be carried out for MaPaMan-II. Instead of a pure-heave motion, the important motion under study could be a pure-yaw motion. Newer metrics might have to be defined for performance characterisation e.g. torsional stiffness could be found instead of stiffness in the vertical direction as in the case of MaPaMan-I.

Prototype of MaPaMan-II can be built following most of the designs as used in MaPaMan-I. The strut link would undergo a change in design and so would the coupler. A CAD model of a possible design-candidate for reconfiguring MaPaMan-I to MaPaMan-II has already been presented in Fig. 2.3. With modifications to the coupler and the strut, prototype of MaPaMan-II can be fabricated and assembled. Once the prototype is fabricated, the procedure followed for trajectory-tracking as used in MaPaMan-I can be directly adapted for MaPaMan-II.

APPENDIX A

Appendix

A.1 Dimensions of MaPaMan prototype-I

Table A.1: MaPaMan prototype-I dimensions

No.	Feature	Representation	Value (in mm)
1	Base length	l_0	100
2	Crank length	l_1	75
3	Rocker length	l_3	75
4	Coupler length	l_2	100
5	Strut length	r_1	105
6	Top plate radius	d_t	110
7	Base point 1	x_1	135
8	Base point 2	x_2	35
9	Coupler height	h_1	19

APPENDIX B

Appendix

B.1 Dimensions of MaPaMan prototpe-II

Table B.1: MaPaMan prototype-II dimensions

No.	Feature	Representation	Value (in mm)
1	Base length	l_0	100
2	Crank length	l_1	75
3	Rocker length	l_3	75
4	Coupler length	l_2	100
5	Strut length	r_1	105
6	Top plate radius	d_t	110
7	Base point 1	x_1	150
8	Base point 2	x_2	50
9	Coupler height	h_1	19

APPENDIX C

Appendix

C.1 Inertia values of MaPaMan prototpe-II

Table C.1: MaPaMan prototype-II mass values

No.	Feature	Representation	Value (in kg)
1	Mass of crank	m_1	0.242
2	Mass of coupler	m_2	0.120
3	Mass of rocker	m_3	0.139
4	Mass of strut and encoder assembly	m_4	0.281
5	Mass of top plate	m_5	2.250

Table C.2: MaPaMan prototype-II mass moment of inertia values about centroidal axis

No.	Feature	Representation	Value (in kg-m ²)
1	Inertia of crank	I_1	3.24×10^{-4}
2	Inertia of coupler	I_2	1.85×10^{-4}
3	Inertia of rocker	I_3	0.95×10^{-4}
4	Inertia of strut	I_4	2.02×10^{-4}
5	Inertia of top plate	I_5	$(57.3, 57.3, 114.5) \times 10^{-4}$

C.2 CAD drawings of the mechanical components

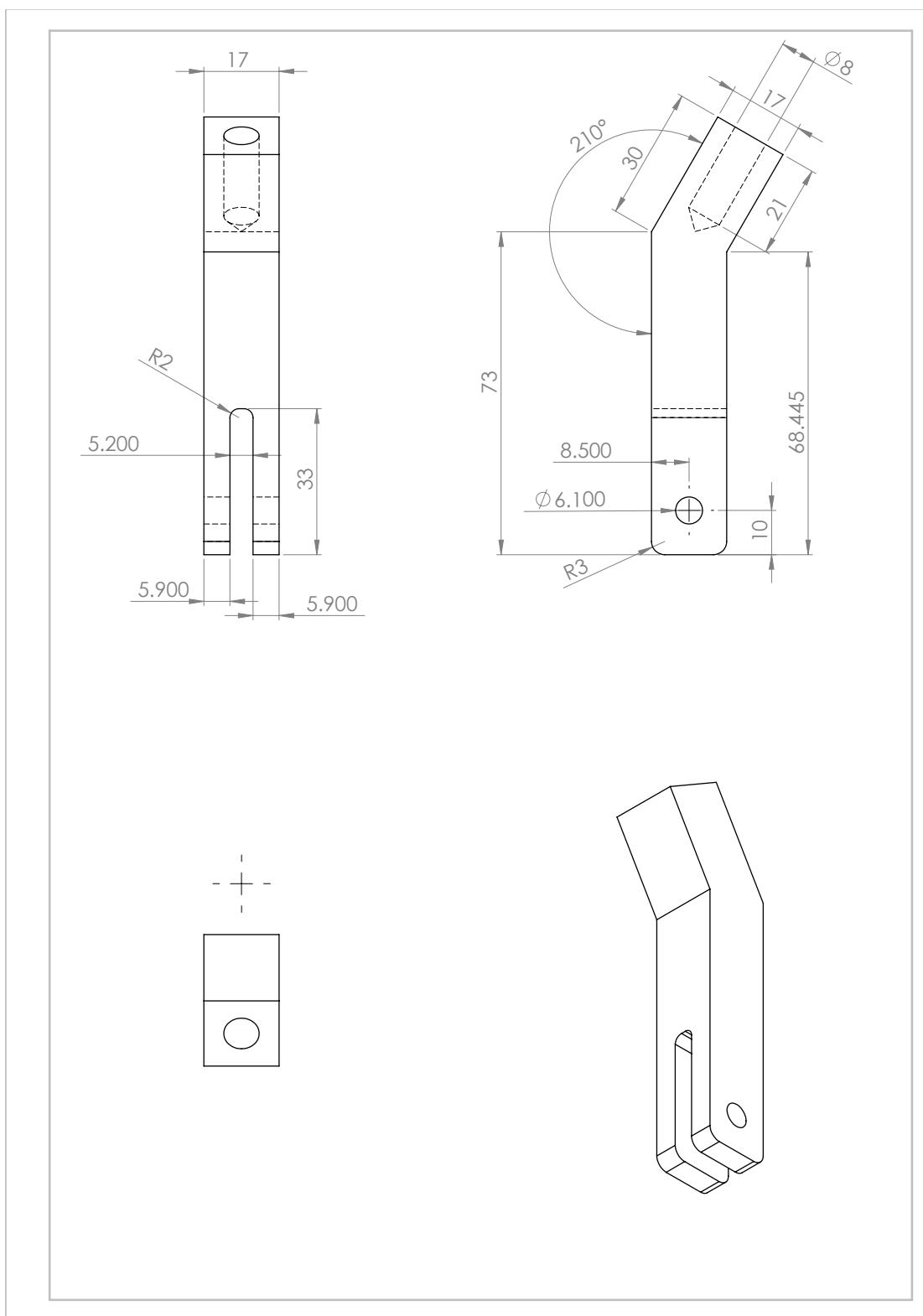


Figure C.1: CAD drawing of strut in prototype-I

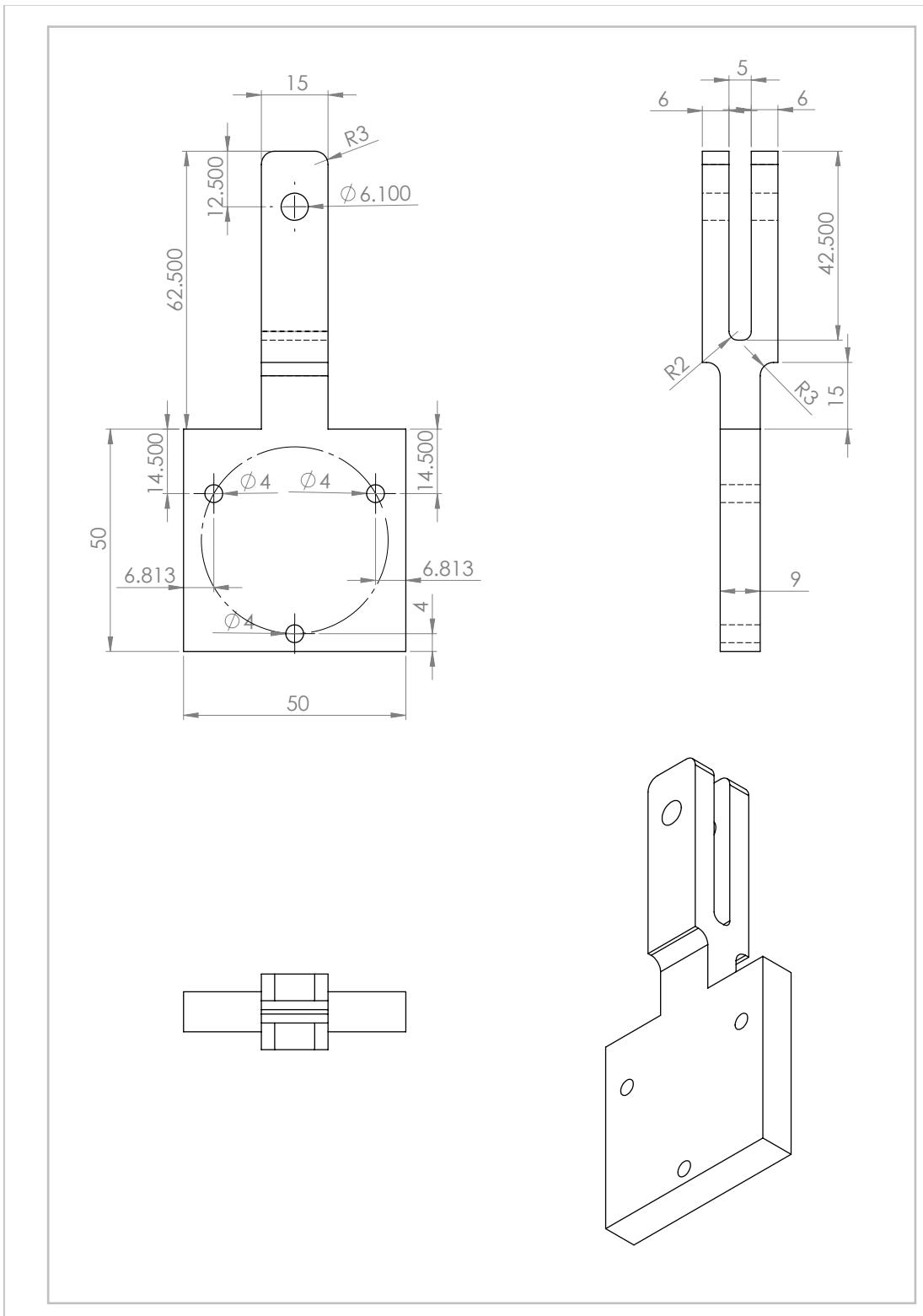


Figure C.2: CAD drawing of crank

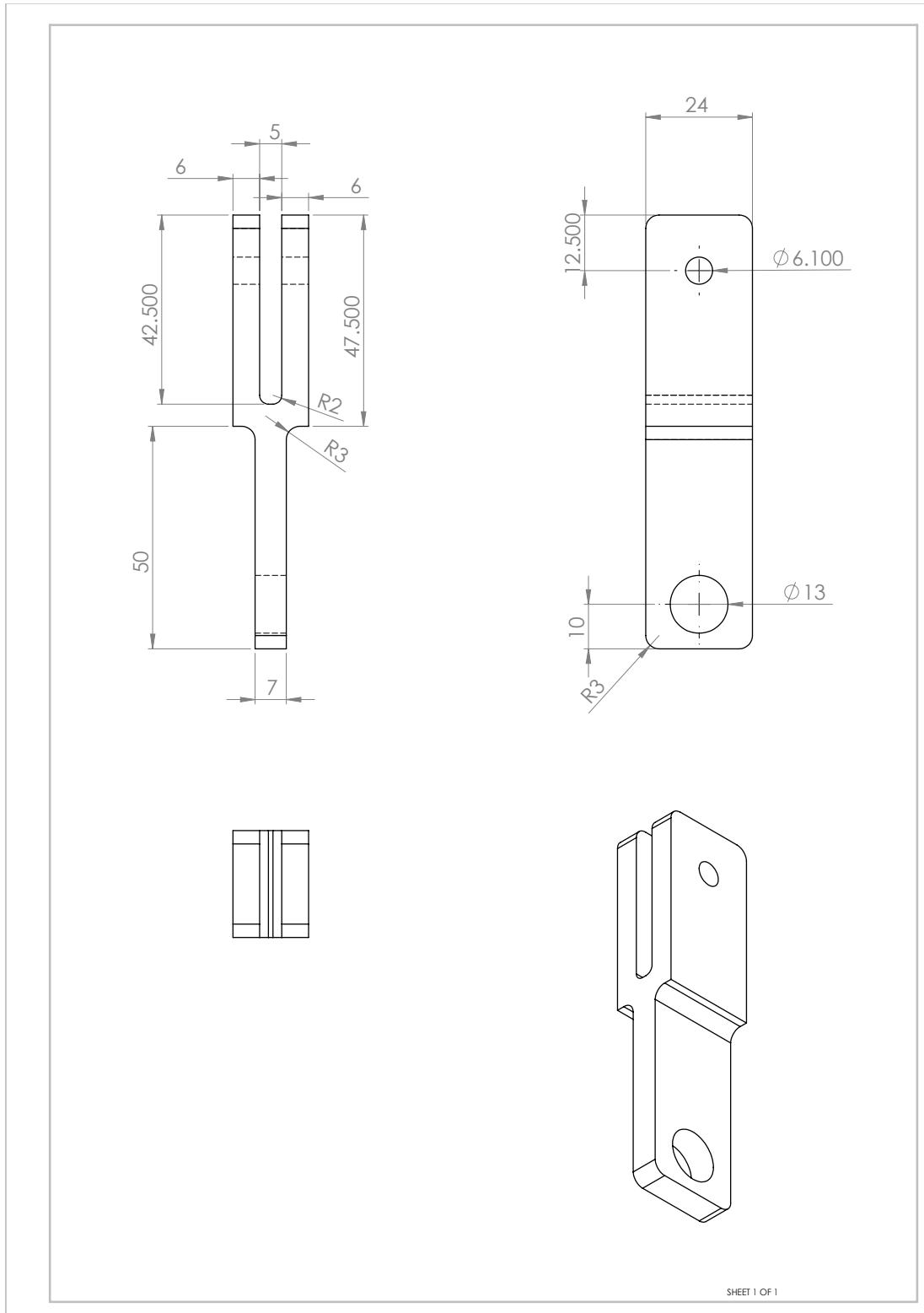


Figure C.3: CAD drawing of rocker in prototype-I

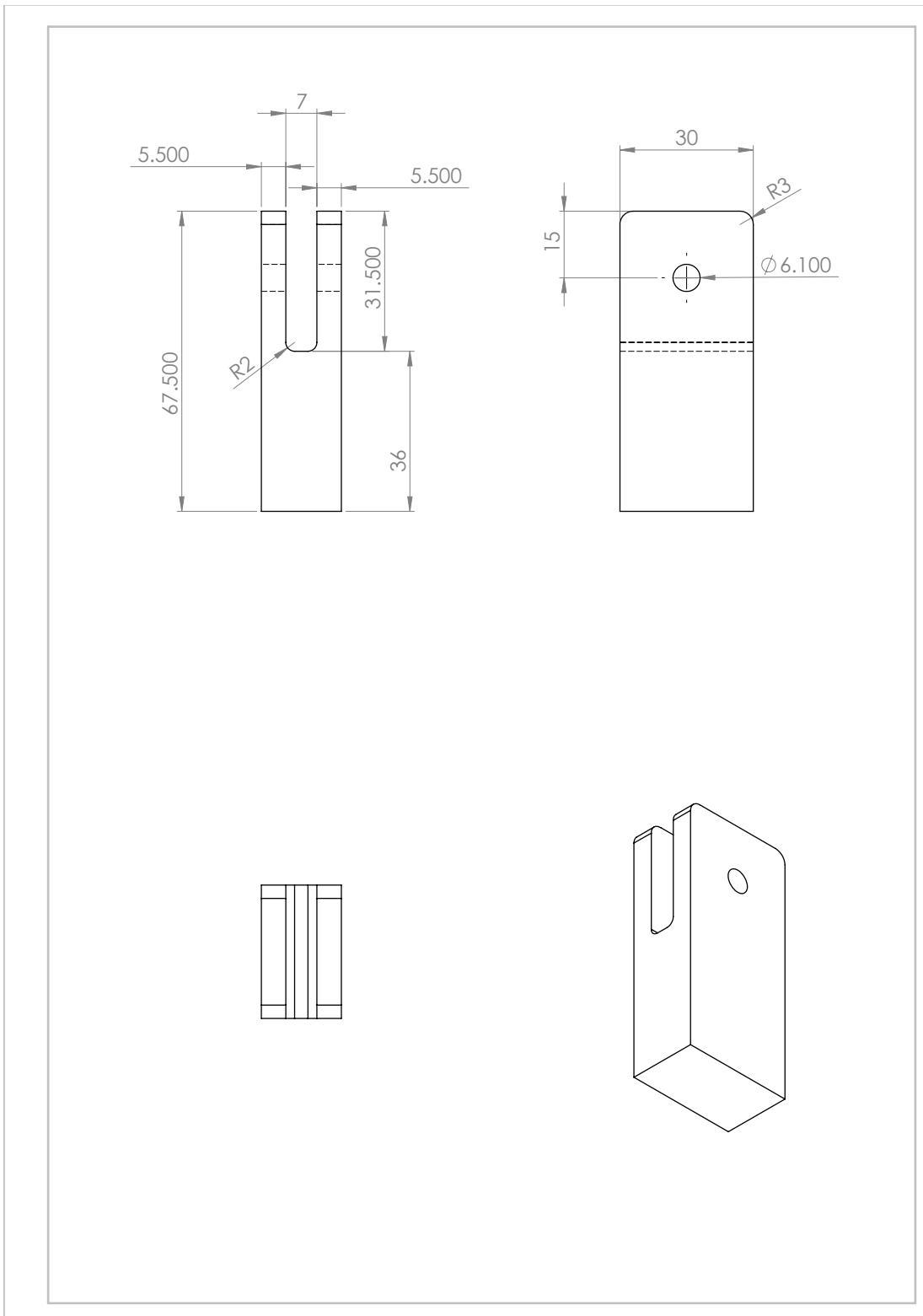


Figure C.4: CAD drawing of rocker-stand in prototype-I

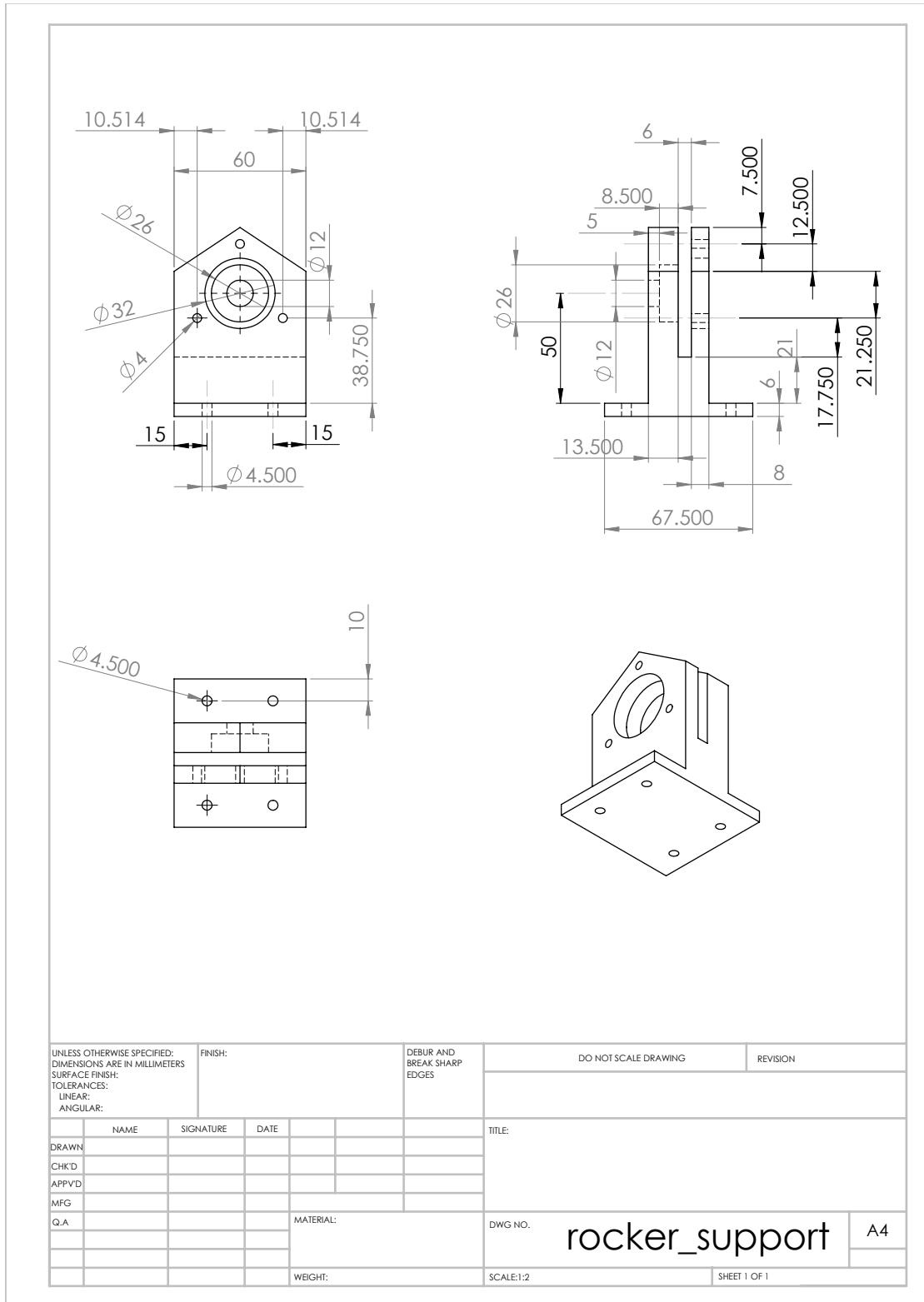


Figure C.5: CAD drawing of rocker-stand in prototype-II

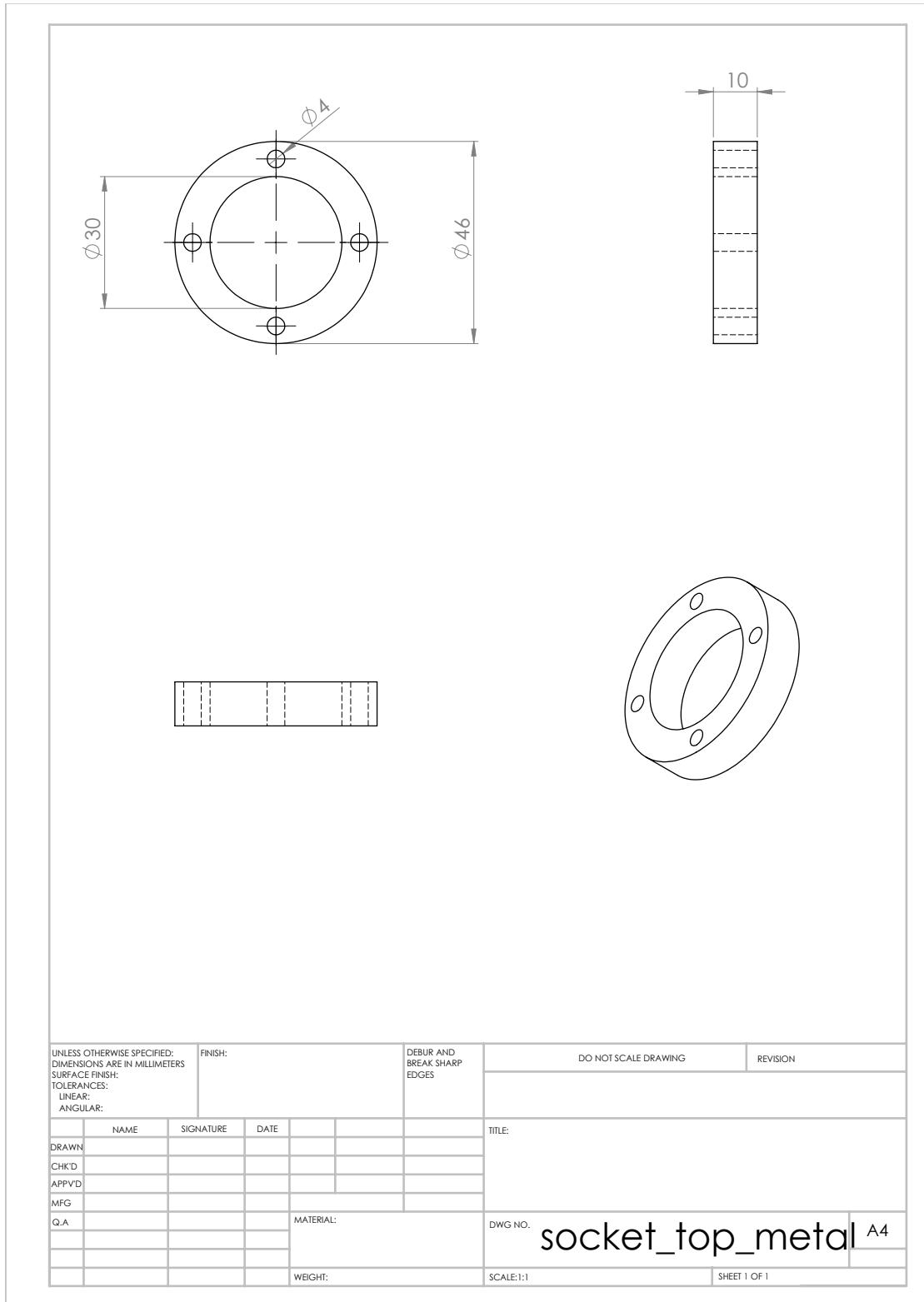


Figure C.6: CAD drawing of top part of socket in ball-socket joint

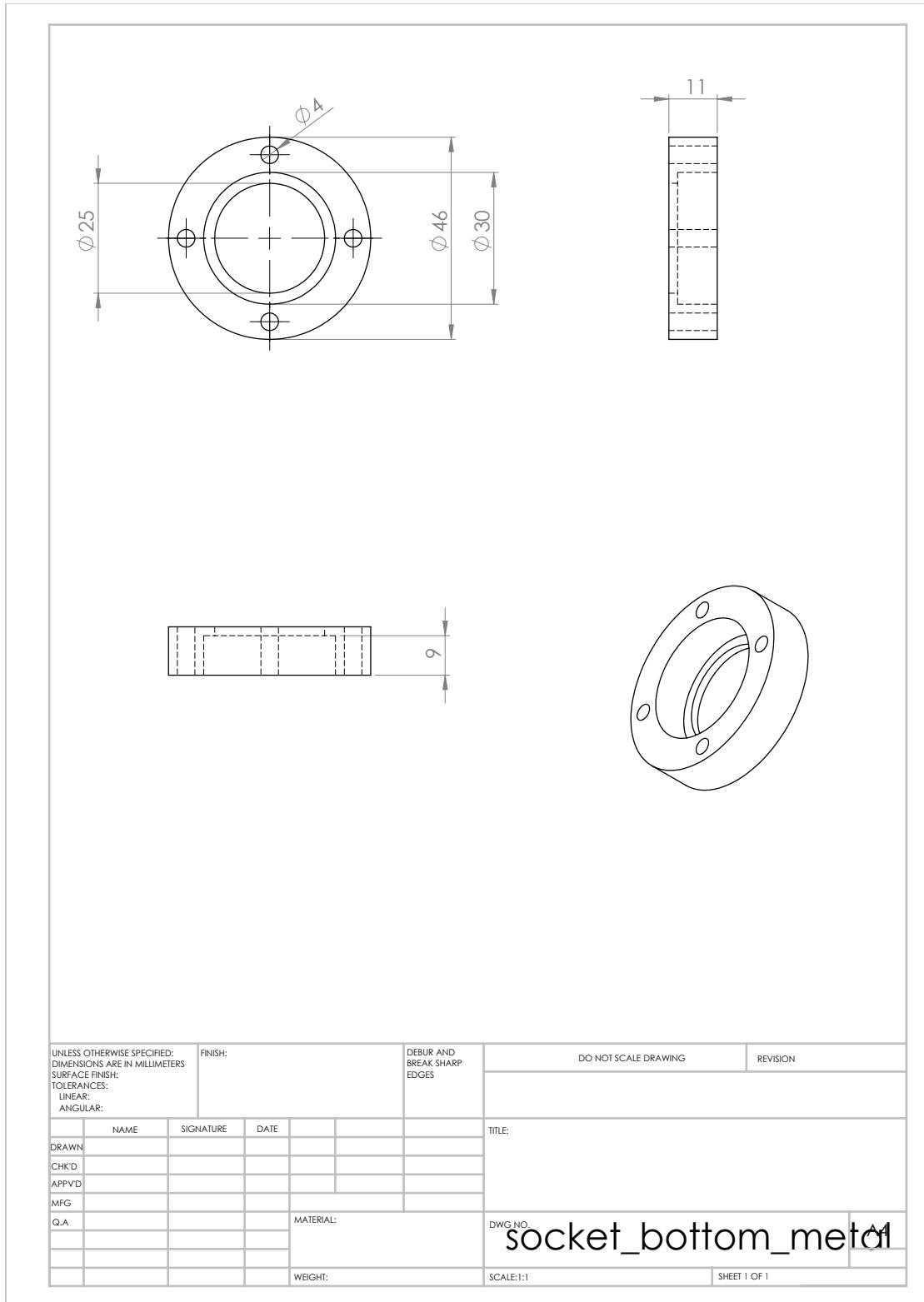


Figure C.7: CAD drawing of bottom part of socket in ball-socket joint

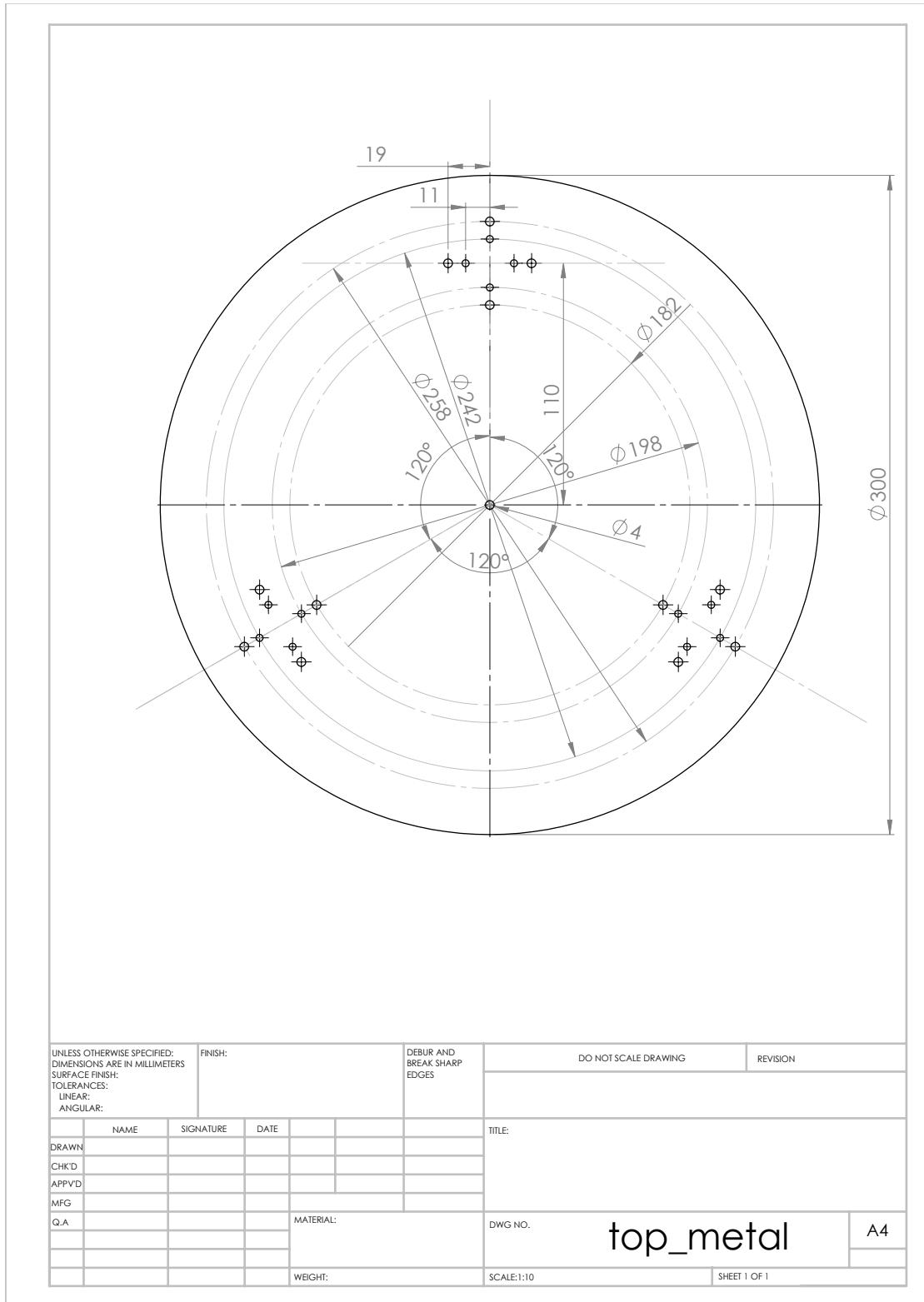


Figure C.8: CAD drawing of the end-effector

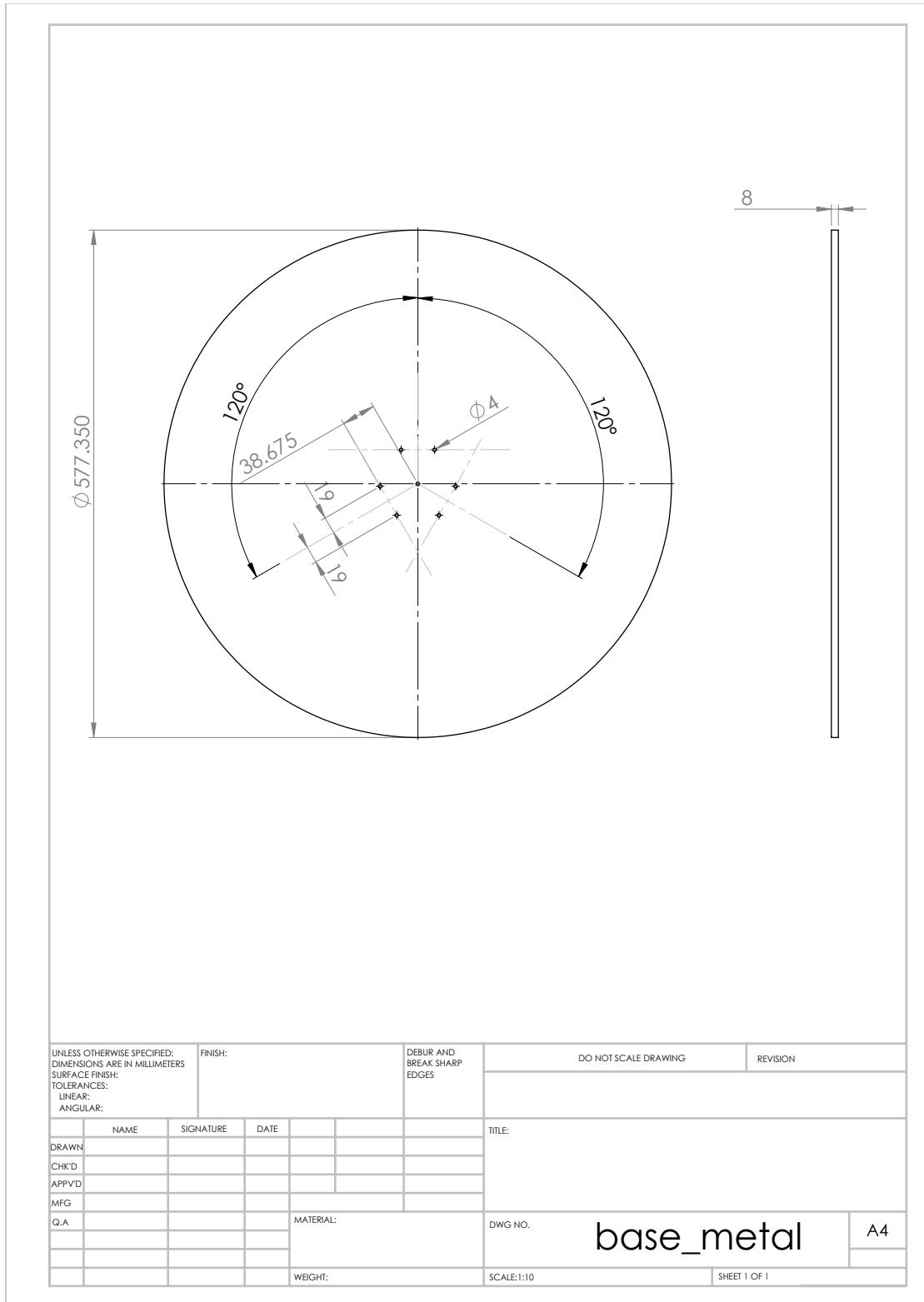


Figure C.9: CAD drawing of the base plate

REFERENCES

1. **Badduri, J.** (2012). *Development of a framework for optimal synthesis of mechanisms*. M. Tech. report, Indian Institute of Technology Madras, Chennai, India.
2. **Bandyopadhyay, S.** (2009). A novel characterization of spatial manipulators based on their degrees-of-freedom. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*.
3. **Bandyopadhyay, S.** and **A. Ghosal** (2006). Geometric characterization and parametric representation of the singularity manifold of a 6-6 Stewart platform manipulator. *Mechanism and Machine Theory*, **41**(11), 1377–1400.
4. **Basu, D.** and **A. Ghosal** (1997). Singularity analysis of platform-type multi-loop spatial mechanisms. *Mechanism and Machine Theory*, **32**, 375–389.
5. **Ceccarelli, M.** (1997). A new 3 D.O.F spatial parallel mechanism. *Mech. Mach. Theory Vol 32, No.8, pp.895-902*.
6. **Clavel, R.** (1988). Delta: A fast robot with parallel geometry. *Proceedings of the International symposium on Industrial Robots*.
7. **Cox, D., J. Little, and D. O’Shea**, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer-Verlag, New York, 1991.
8. **Deb, K.**, *Optimization for Engineering Design*. Prentice-HAll, India, 1995.
9. **Deb, K., S. Agrawal, A. Pratap, and T. Meyarivan** (2002). A fast and elitist multi-objective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, **6**(2), 182–197.
10. **G. Aguirre, G. C., M. Acevedo and E. Ottaviano** (2003). Kinematic and dynamic analyses of a 3 dof parallel manipulator by symbolic formulations. *Thematic Conference on Advances in Computational Multibody Dynamics, ECCOMAS*.

11. **Ghosal, A.**, *Robotics: Fundamental Concepts and Analysis*. Oxford University Press, New Delhi, 2006.
12. **Gosselin, M.** and **J.-F. Hamel** (1994). The agile eye: a high-performance three-degree-of-freedom camera-orienting device. *Proceedings of the IEEE International Conference on Robotics and Automation*.
13. **Gough, V.**, Contribution to discussion of papers on research in automobile stability, control and tyre performance. *In Proceedings of the Institution of Mech. Engineers Auto Division (1956/1957)*. 1940.
14. **Gregorio, R. D.** (2003). Kinematics of the 3-UPU wrist. *Mechanism and Machine Theory*, **38**, –.
15. **Hunt, K. H.**, *Kinematic Geometry of Mechanisms*. Clarendon Press, Oxford, 1978.
16. **Husty, M. L.** (1994). An algorithm for solving the direct kinematic of stewart-gough-type platforms. *Mech. Mach. Theory Vol 31, No.4*, pp.365-380,.
17. **Johnson, R.** and **K.-M. Lee** (1989). Static characteristics of an in-parallel actuated manipulator for clamping and bracing applications. *Proceedings of the IEEE International Conference on Robotics and Automation*.
18. **Lee, K. M.** and **D. Shah** (1988). Kinematic analysis of a three-degrees-of-freedom in-parallel actuated manipulator. *IEEE Journal of Robotics and Automation*, **4**(3), 354–360.
19. **Li, Y.** and **G. Bone** (2001). Are parallel manipulators more energy efficient? *IEEE Int. Symp. on Computational Intelligence in Robotics and Automation*.
20. **Mehta, T.** (2012). *Dynamics, Trajectory-tracking and Testing of MaPaMan*. M. Tech. report, Indian Institute of Technology Madras, Chennai, India.
21. **Merlet, J.-P.**, *Parallel Robots*. Kluwer Academic Press, Dordrecht, 2001.
22. **Norton, R. L.**, *Design of machinery: An introduction to the synthesis and analysis of mechanisms and machines*. Tata McGraw-Hill, United States of America, 1999, 2 edition.

23. **Saglia Jody A., D. J. S. and C. D. G.** (2008). Geometry and kinematic analysis of a redundantly actuated parallel mechanism that eliminates singularities and improves dexterity. *Journal of mechanical design, vol. 130, no12, [Note(s): 124501.1-124501.5]*.
24. **Selig, J. M.**, *Geometric fundamentals of robotics*. Springer, New York, 1996, 2 edition.
25. **Shahinpoor, M.** (1992). Kinematics of a parallel-serial (hybrid) manipulator. *Journal of Robotic Systems, doi: 10.1002/rob.4620090103.*
26. **Timoshenko, S.**, *Strength of materials*. D. Van. Nostrand company, New York, 1930.
27. **Tsai, L.-W.** (1996). Kinematics of a three-dof platform manipulator with three extensi- ble limbs. *Recent Advances in Robot Kinematics, pp. 401–410, Kluwer Academic Publishers.*,
28. **Zlatanov, D., I. A. Bonev, and C. M. Gosselin**, Constraint singularities of parallel mechanisms. *In Proceedings of the 2002 IEEE International Conference on Robotics and Automation*. Washington, DC, 2002.