Analytical Workspace, Kinematics, and Foot Force Based Stability of Hexapod Walking Robots

Mohammad Mahdi Agheli Hajiabadi

Worcester Polytechnic Institute

Follow this and additional works at: https://digitalcommons.wpi.edu/etd-dissertations

Repository Citation

This dissertation is brought to you for free and open access by Digital WPI. It has been accepted for inclusion in Doctoral Dissertations (All Dissertations, All Years) by an authorized administrator of Digital WPI. For more information, please contact wpi-etd@wpi.edu.
Analytical Workspace, Kinematics, and Foot Force Based Stability of Hexapod Walking Robots

by

MOHAMMAD MAHDI AGHELI HAJIABADI

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Doctor of Philosophy

in

Mechanical Engineering

May 2013

APPROVED:

Professor Stephen S. Nestinger, Thesis Advisor

Professor Gregory S. Fischer, Graduate Committee Member

Professor Cagdas Onal, Graduate Committee Member

Professor Taskin Padir, Graduate Committee Member

Professor Simon W. Evans, Graduate Committee Representative
Abstract

Many environments are inaccessible or hazardous for humans. Remaining debris after earthquake and fire, ship hulls, bridge installations, and oil rigs are some examples. For these environments, major effort is being placed into replacing humans with robots for manipulation purposes such as search and rescue, inspection, repair, and maintenance. Mobility, manipulability, and stability are the basic needs for a robot to traverse, maneuver, and manipulate in such irregular and highly obstructed terrain. Hexapod walking robots are as a salient solution because of their extra degrees of mobility, compared to mobile wheeled robots. However, it is essential for any multi-legged walking robot to maintain its stability over the terrain or under external stimuli. For manipulation purposes, the robot must also have a sufficient workspace to satisfy the required manipulability. Therefore, analysis of both workspace and stability becomes very important.

An accurate and concise inverse kinematic solution for multi-legged robots is developed and validated. The closed-form solution of lateral and spatial reachable workspace of axially symmetric hexapod walking robots are derived and validated through simulation which aid in the design and optimization of the robot parameters and workspace. To control the stability of the robot, a novel stability margin based on the normal contact forces of the robot is developed and then modified to account for the geometrical and physical attributes of the robot. The margin and its modified version are validated by comparison with a widely known stability criterion through simulated and physical experiments. A control scheme is developed to integrate the workspace and stability of multi-legged walking robots resulting in a bio-inspired reactive control strategy which is validated experimentally.
Contents

1 Introduction .............................................. 1
  1.1 Thesis Contribution ...................................... 4
  1.2 Thesis Organization ...................................... 5

2 Modeling and Hardware ................................. 6
  2.1 Modeling .................................................. 6
  2.2 Robot Hardware Architecture ......................... 8
  2.3 Sensors ................................................... 9

3 Kinematics ................................................. 11
  3.1 Background ............................................... 11
  3.2 Inverse Kinematic Solution ............................... 13
    3.2.1 Hip joint angle calculation ........................... 14
    3.2.2 Knee joint vector calculation ......................... 16
    3.2.3 Knee leg vector calculation ............................ 16
    3.2.4 Knee and Ankle Angles Calculation .................... 17
    3.2.5 Analytical solution .................................... 18
    3.2.6 Inverse Kinematic Algorithm .......................... 18
  3.3 Validation ............................................... 19
    3.3.1 Walking ............................................. 19
    3.3.2 Object Manipulation ................................... 32

4 Workspace ................................................ 34
  4.1 Background .............................................. 34
  4.2 Lateral Workspace ....................................... 36
    4.2.1 Closed-form Solution for the Workspace Envelope ....... 38
    4.2.2 Discussion ........................................... 51
    4.2.3 Validation .......................................... 54
    4.2.4 Application .......................................... 55
  4.3 Spatial Workspace ....................................... 56
    4.3.1 Closed-form Solution for the Workspace Boundary ...... 57
    4.3.2 Optimization and Design .............................. 74
  4.4 Stable Workspace ....................................... 87
5 Stability and Control 89
  5.1 Background ......................................................... 89
  5.2 Foot Force Based Stability ........................................ 93
    5.2.1 Foot Force Stability Margin ................................... 93
    5.2.2 Modified Foot Force Stability Margin ......................... 98
  5.3 Validation .......................................................... 105
    5.3.1 Numerical Validation of FFSM and MFFSM ....................... 106
    5.3.2 Experimental Validation of FFSM and MFFSM .................... 117
  5.4 Bio-Inspired Reactive Stability Control using FFSM ............... 122
    5.4.1 Control Strategy ................................................. 124
    5.4.2 Implementation ................................................... 125

6 Conclusion and Future Work 128
  6.1 Conclusion .......................................................... 128
  6.2 Future Work ........................................................ 130
List of Figures

1.1 Welding inside of a narrow cylinder [1, 2]......................... 2
1.2 A conceptual pipe welding and repair application using hexapod robots. 3

2.1 Leg parameters definitions........................................ 7
2.2 Angle definitions for a leg of a HWR.................................. 7
2.3 The axially symmetric HWR used throughout the research............. 8
2.4 The electronic board of the hexapod robot under study.................. 9
2.5 The sensor housing, a) CAD model design and b) fabricated............. 10
2.6 The foot force sensor calibration...................................... 10

3.1 Coordinate definitions of the platform................................ 14
3.2 The kinematic closures of the robot.................................. 15
3.3 An arbitrary orientation of the platform................................ 18
3.4 The support and transfer phases in one walking cycle................. 21
3.5 Gait parameter definitions for a walking robot.......................... 21
3.6 The input and output of the foot trajectory planning algorithm........ 23
3.7 A schematic of a HWR with leg trajectories from top view.............. 24
3.8 The gait generation of a HWR with duty factor of $\beta = 0.75$............ 25
3.9 The desired leg trajectory for each leg during the transfer phase........ 25
3.10 Coordinates definition for defining D-H parameters presented in Ta- 27
   ble 3.1
3.11 The leg trajectories for 5 seconds of the walking process............. 28
3.12 Top view of the robot with three supporting legs and three transferring legs based on the tripod gait......................... 29
3.13 A snap shot of the robot in the wave gait motion in the simulation environment and the corresponding experiment.................. 30
3.14 The snap shots of the robot in the tripod motion in the simulation environment and the corresponding experiment................. 31
3.15 The manipulator tip has to be perpendicular to S and E.................. 32
3.16 The simulation result corresponding to Figure 3.15..................... 33
3.17 The experimental result corresponding to Figure 3.15..................... 33

4.1 An illustration of COW [3]........................................... 35
4.2 The lateral RW of a mobile machining system........................... 36
4.3 HWR performs as a planar 2-RPR parallel mechanism in the lateral plane. ............................................................ 37
4.4 A general 2-RPR planar parallel mechanism a) Parameters definition b) The RW. .......................................................... 39
4.5 RW of a 2-RPR planar parallel mechanism based on the structural parameters listed in Table 4.1. ................................. 40
4.6 The decomposition of the RW of a 2-RPR mechanism. ............... 41
4.7 The boundary points and curves of the RW. ................................. 43
4.8 superimposition of Figure 4.7-(a-d). ................................. 44
4.9 The cardinal and auxiliary points and curves of the workspace boundary. 44
4.10 The different 2-RPR configurations that generate the RW envelope. 46
4.11 The 2-RPR mechanism is modeled as a PRRR mechanism when $\theta_2$ is fixed at zero. ............................................................ 48
4.12 Four specific workspace examples. ........................................ 52
4.13 Different workspace cases of the 2-RPR planar parallel mechanism. 53
4.14 The results from the validation of the closed-form methodology against a numerical methods approach. ................................. 54
4.15 The maximum permissible rectangular area prescribed within the workspace. ............................................................ 55
4.16 The articulated leg replaced with prismatic joint (left) and HWR as three 2-RPR mechanisms (right). ................................. 58
4.17 The general 2-RPR mechanism (left) and 2-RPR mechanism with a general COW (right). ............................................................ 60
4.18 The COW is constrained by multiple circles. ................................. 61
4.19 The 2-RPR mechanism with symmetric COW. ................................. 62
4.20 The 3D workspace of a 2-RPR mechanism a) 2D and b) top view 3D c) isometric view. ............................................................ 66
4.21 A general axially symmetric hexapod robot from its top view. ......... 67
4.22 A general axially symmetric hexapod robot with its originator workspaces from its top view. ............................................................ 68
4.23 The summation of the 3D 2-RPR mechanisms before and after Boolean operation. ............................................................ 70
4.24 The symmetric workspace of hexapod robot. ............................................................ 70
4.25 The lateral COW of the hexapod robot covering the desired workspace. 80
4.26 Maximizing the workspace of hexapod robot. ................................. 83
4.27 The optimization setup for the design and workspace of a 2-RPR mechanism constrained within a desired rectangular region. .......... 84
4.28 The final optimized workspace with all cardinal points. ............... 86
4.29 The static stable workspace of HWRs from the top view. ............... 87

5.1 An illustration of SM presented by McGhee [4]. ................................. 90
5.2 An illustration of FASM for 2D case [5]. ........................................ 91
5.3 The general control architecture of a multi-legged robot [6]. ............... 93
5.4 A robot with \( n \) supporting legs over irregular terrain. 94
5.5 Multiple planar robot configurations with the same Foot Force Stability Margin. 99
5.6 The schematic of a general spatial \( n \)-legged robot traversing an irregular terrain. 101
5.7 A planar robot on irregular terrain. 104
5.8 A leg link with forces and moment acting on it. 107
5.9 A schematic of a HWR on an uneven terrain with external force and moment applying on the body. 109
5.10 The actual Lynxmotion hexapod under study. 111
5.11 Four motion phases described in the simulation setup. 113
5.12 The results of stability simulation comparing the FFSM, MFFSM and FASM. 113
5.13 The motion scenario for dynamic stability simulation. 114
5.14 The results of the numerical dynamic stability simulation for both the loaded and non-loaded scenarios. 114
5.15 The time snapshots at the experimental scenario. 118
5.16 The experimental validation of FFSM and MFFSM compared to FASM. 119
5.17 The stability scenario of a robot over irregular terrain for both simulation and experiment. 121
5.18 The simulated and experimental stability results for FFSM and FASM over uneven terrain. 121
5.19 The reactive control schematic for hexapod walking and manipulation. 126
5.20 The experimental implementation of the reactive stability. 127
6.1 The designed scalable hexapod walking robot for prototype. 131
# List of Tables

2.1 Definitions and specifications of i\textsuperscript{th} leg ........................................ 8

3.1 D-H parameters of a leg of hexapod robot with 3-DOF. .......................... 26

4.1 An example set of structural parameters for a 2-\textit{RPR} planar parallel mechanism ................................................................. 40

4.2 All possible structural parameter configurations for generating the sub-workspaces of the RW ......................................................... 42

4.3 All possible structural parameter configurations for generating the sub-workspaces of the RW ................................................................. 51

4.4 Location of the cardinal points for the presented example. ...................... 54

4.5 Equations of the final workspace solution for final upper surfaces. .......... 72

4.6 Equations of the final workspace solution for final lower surfaces. .......... 72

4.7 Equations of the final workspace solution for final upper curves. .......... 72

4.8 Equations of the final workspace solution for final lower curves. .......... 73

4.9 Equations of the final workspace solution for boundary side curves. ........ 73

4.10 Equations of the final workspace solution for final boundary points. .... 73

4.11 Equations of the final workspace solution for final upper surfaces. ....... 75

4.12 Equations of the final workspace solution for final lower surfaces. ....... 75

4.13 Equations of the final workspace solution for final upper curves. ....... 75

4.14 Equations of the final workspace solution for final lower curves. ....... 75

4.15 Equations of the final workspace solution for boundary side curves. ...... 76

4.16 Equations of the final workspace solution for final boundary points in \textit{x} and \textit{y} axes. ................................................................. 76

4.17 Equations of the final workspace solution for final boundary points in \textit{z} axis. ................................................................. 77

4.18 Equations of the final workspace solution for final upper surfaces. ....... 77

4.19 Equations of the final workspace solution for final lower surfaces. ....... 78

4.20 Equations of the final workspace solution for final upper curves. ....... 78

4.21 Equations of the final workspace solution for final lower curves. ....... 78

4.22 Equations of the final workspace solution for boundary side curves. ...... 79

4.23 Equations of the final workspace solution for final boundary points. .... 79
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CG</td>
<td>Center of Gravity</td>
</tr>
<tr>
<td>COW</td>
<td>Constant-Orientation Workspace</td>
</tr>
<tr>
<td>D-H</td>
<td>Denavit Hartenberg</td>
</tr>
<tr>
<td>DOF</td>
<td>Degrees of Freedom</td>
</tr>
<tr>
<td>FASM</td>
<td>Force Angle Stability Margin</td>
</tr>
<tr>
<td>FFD</td>
<td>Foot Force Distribution</td>
</tr>
<tr>
<td>FFSM</td>
<td>Foot Force Stability Margin</td>
</tr>
<tr>
<td>HWR</td>
<td>Hexapod Walking Robot</td>
</tr>
<tr>
<td>MFFSM</td>
<td>Modified Foot Force Stability Margin</td>
</tr>
<tr>
<td>RW</td>
<td>Reachable Workspace</td>
</tr>
<tr>
<td>SM</td>
<td>Stability Margin</td>
</tr>
<tr>
<td>SP</td>
<td>Stewart Platform</td>
</tr>
</tbody>
</table>
Acknowledgement

I would like to express the deepest appreciation to my supervisor Professor Stephen Nestinger who has the attitude and the substance of a genius: he continually and convincingly conveyed a spirit of adventure in regards to the research. Without his guidance and persistent help this dissertation would not have been possible.

I would like to thank my committee members, Professor Gregory Fischer, Professor Cagdas Onal, Professor Taskin Padir, and Professor Simon Evans for their comments and suggestions regarding my dissertation.

I am grateful to Professor Robert Norton, who provided me with much appreciated feedback regarding a large part of the analyses in this thesis.

I would like to thank the Mechanical Engineering department for the supporting my education, and the Mechanical Engineering staff especially Barbara Edilberti, Barbara Furhman, and Statia Canning for their huge amount of support and help.

I would like to thank all of the friends I have made at WPI.

A special thanks to my wife, Niloufar Zaremanshadi, soon to be born daughter, Ava Agheli, and my parents, Ali Agheli and Safa Amini. They are my reason for making it this far!
Chapter 1

Introduction

Many environments are inaccessible or hazardous for humans in which some sort of manipulation needs to be done. Examples of such environments include remaining debris after earthquake and fire, ship hulls, bridge installations, and oil rigs are. Generally, these locations suffer from lack of appropriate visual, respiratory, and safety conditions for human workers attempting search and rescue, inspection, repair, and maintenance. Welding inside of a long narrow cylinder is an example as shown in Figure 1.1. Therefore, it is more appropriate to use robots instead of humans in these environments. The requirements for a robot to be used in such environments are mobility, manipulability, and stability.

Multiple robotic solutions exist that may be employed within highly constrained environments including hull blasting [7], bridge inspection [8], pipe inspection [9], tank inspection [10], and sewer inspection [11]. There have been some attempts to automate structural maintenance and repair in Europe [12, 13]. However, these systems are still limited, requiring more DOF and unable to handle highly irregular terrain. Some automated tasks use rails for robot movement. Hence, human workers must first navigate through the treacherous and dark environment to lay down the
railing for the robot. Therefore, this type of technology cannot be used for highly constrictive, inaccessible, and rough environments. To truly automate manipulating in remote and constrained areas, mobile systems capable of traversing over irregular terrain while providing 6-DOF for repair are required.

Legged robots have emerged as a salient solution for traversing irregular terrain and maneuvering through highly obstructed passages. There are different types of legged robots such as bipeds, tripods, quadrupeds, hexapods, octopods, etc. Hexapod walking robots (HWRs) with various configurations and leg designs, which are widely used in practice due to their simplicity and innate static balance, can be potentially used for the purposes such as maintenance and operations. In terms of mobility, when comparing to mobile wheeled robots, HWRs are superior and more practical for uneven or irregular terrain with possible obstacles and gaps [14–18]. The enhanced mobility makes hexapod robots appealing for search and rescue [19, 20], planetary exploration [21], and wall climbing [22]. Hexapod robots can be used for applications seeking a system with capability of mobility, manipulating a large workspace, configuration flexibility, traversing irregular terrain, and working in constrained environments.
Figure 1.2: A conceptual pipe welding and repair application using hexapod robots.

Although hexapod robots have many appealing characteristics, current designs are limited with regards to the reachable workspace especially when considering stability and motion. During a manipulation process, the orientation of the robot may need to be maintained throughout a spatial motion as shown in a conceptual application depicted in Figure 1.2. Also, integrating the workspace and stability of the robot for control purposes is very important. When the robot needs to manipulate, it is important to know where the end-effector can reach and, at the same time, the stability of the robot needs to be guaranteed. Hence, a new methodology is required to integrate stability and workspace while maintaining the designed degrees of freedom.
1.1 Thesis Contribution

The proposed research develops a methodology for kinematics, workspace, and stability control of multi-legged robots for in-situ repair and maintenance of constrained and hazardous environments. An artistic rendition of an example scenario using HWRs was shown in Figure 1.2. The contributions of this research include the following:

- A hardware architecture for an existing hexapod robot is assembled and integrated for the whole system including mechanical and mechatronic hardware for experimental test purposes.

- An accurate and concise analytical inverse kinematic solution for multi-legged robots is developed and validated.

- The analytical solution for the workspace of axially symmetric hexapod robots is developed and validated for both lateral and spatial cases.

- A foot force stability margin for legged and wheeled robots is developed based on the normal foot forces of the robot to be used for reactive stability and validated using both simulation and experiment.

- A modified version of the foot force stability margin was developed and validated to take into consideration the effect of geometry and top-heaviness.

- The concept of Stable Workspace is developed for control of the robot when manipulating.

- A bio-inspired reactive stability control algorithm is developed and validated experimentally to help legged robots remain stable and not tipping over against external stimuli.
• A scalable hexapod walking robot for in-situ repair and maintenance in con-
strained and hazardous environments is proposed and prototyped with extend-
able size and workspace, and ability to walk with different steps and speeds.

1.2 Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 presents the physical
robot, hardware, and model used for analysis and experimental work through out
the thesis. Chapter 3 analyzes the kinematics of multi-legged robots and presents an
accurate and concise inverse kinematic solution for arbitrary position and orientation
of the robot. The presented solution is validated through both simulation and
experimental work. Chapter 4 provides an analytical solution for both the lateral
and the spatial workspace boundary of the axially symmetric HWRs. The workspace
solutions were validated through simulation and used in a design and optimization
of the robot parameters and workspace. Chapter 5 investigates the stability of
multi-legged walking robots and provides a new foot force based stability margin
which is compared with a widely known stability criterion through simulated and
physical experiments for validation. A control scheme was developed to integrate
the analytical workspace and the novel stability margin which resulted in a bio-
inspired reactive control strategy for hexapod walking robots. The developed bio-
inspired reactive control architecture uses the presented stability margin for reaction
of the robot under unpredicted external stimuli. The reactive control is validated
experimentally. Chapter 6 concludes the thesis and provides some suggestions for
future work.
Chapter 2

Modeling and Hardware

In what follows, the model of the robot and the hardware used throughout the thesis is presented.

2.1 Modeling

In most HWRs, the hip joints are either distributed axially symmetric around the platform or distributed evenly along a rectangular body. HWRs can be broken down into four types based on the DOF of the legs [17]: leg DOF of two, three, four, and six. Hexapods with 3-DOF legs are the most common because they have the simplest design in terms of minimum required DOF per each leg while retaining required walking ability and good flexibility for handling unstructured terrain and different obstacles [14–18]. The articulated hexapod legs consist of three joints providing each leg with 3-DOF. Each leg includes three separate segments which are connected together by revolute joints. The names, magnitudes, and limitations of the leg segments and joints are listed in Table 2.1. Figure 2.1 and Figure 2.2 provide a visual representation of the leg segments and joints. All of the legs are connected to the main body (platform) of the robot through a hip joint. To simplify
the representation of the inverse kinematic equations, the legs on the left side of the robot are designated as odd numbered and the right side legs as even numbered.

Therefore, the selected robot for this research is an axially symmetric HWR which has a round-shape platform which is provided 6-DOF by legs where each leg by itself has 3-DOF. Therefore, the whole system has $6(3) + 6 = 24$-DOF. Figure 2.3 shows an axially symmetric HWR in isometric and top view which is used for this research. The diameter of the platform is 300 $mm$. 
Table 2.1: Definitions and specifications of $i$th leg

<table>
<thead>
<tr>
<th>Segment No.</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment Name</td>
<td>Coxa</td>
<td>Femur</td>
<td>Tibia</td>
</tr>
<tr>
<td>Segment Length</td>
<td>$l_{1i} = 20mm$</td>
<td>$l_{2i} = 70mm$</td>
<td>$l_{3i} = 100mm$</td>
</tr>
<tr>
<td>Joint Name</td>
<td>Hip</td>
<td>Knee</td>
<td>Ankle</td>
</tr>
<tr>
<td>Joint Angle</td>
<td>$-\frac{\pi}{2} \leq \alpha_i \leq \frac{\pi}{2}$</td>
<td>$-\frac{\pi}{6} \leq \beta_i \leq \frac{\pi}{2}$</td>
<td>$0 \leq \gamma_i \leq \frac{2\pi}{3}$</td>
</tr>
</tbody>
</table>

Figure 2.3: The axially symmetric HWR used throughout the research.

2.2 Robot Hardware Architecture

The hexapod robot, shown in Figure 2.4, consists of a Lynxmotion hexapod robot kit [23] and a Gumstix Verdex Pro XM4-BT COM tiny computer [24]. The hexapod robot has 18 HS-485HB servos controlled by a SSC-32 sequencer. There is a built-in proportional controller for each servo. Each leg consists of three servos and three leg segments. The Gumstix tiny computer acts as the high-level controller of the robot and communicates with the SCC-32 using UART via the Robostix expansion board. The Gumstix runs Linux 2.6 and is connected to the Netpro-VX expansion board for wireless connectivity. The housing for the Gumstix and expansion boards was fabricated using a rapid prototype machine.
2.3 Sensors

To physically measure the normal foot forces, the Lynxmotion hexapod robot, shown in Figure 2.4, was equipped with Force-Sensitive Resistors (FSR-402) similar to [25]. The sensors were calibrated after they were embedded into rapid prototyped housings, as shown in Figure 2.5. The calibration results, shown in Figure 2.6, were used to fit a curve for each sensor allowing a direct correlation between the output voltage and the applied force.
Figure 2.5: The sensor housing, a) CAD model design and b) fabricated.

Figure 2.6: The foot force sensor calibration.
Chapter 3

Kinematics

To control the motion of the robot and enable it to walk and manipulate, inverse and forward kinematics solutions of both the robot and the legs are necessary. Inverse kinematics of hexapod robots is finding the geometry parameters required to achieve a given position and attitude of the end-effector of the robot and the forward kinematics is finding the position and attitude of the end-effector of the robot while the geometry parameters of the robot are given. Within this chapter, when talking about Center of Gravity (CG) for the “kinematics,” the center of the robot (center of the platform) is the point of interest.

3.1 Background

Inverse kinematics of parallel legged robots can be studied in two aspects: inverse kinematics of the parallel robot and the inverse kinematics of each single leg. As with the inverse kinematics, forward kinematics can be also studied in two aspects: forward kinematics of the parallel robot by itself and that of each single leg.

The inverse kinematic solution of parallel legged robots for any arbitrary orientation is necessary to control the motion of the robot, enable it to walk over uneven
terrain, and manipulate objects. The kinematics for hexapod robots have been previously studied but have led to solutions which are either not analytical or do not deal with complicated leg configurations and have high calculation cost [26–30]. Some studies either did not present the solution or only solved the problem partially [31–36]. Yanto Go et al. [14], Duan et al. [15], and Netto et al. [16] each presented a HWR and discussed the kinematics of the robot. The legs of their robot have 3-DOF. They developed a mathematical model for kinematics of the hexapod robot. However, their solutions have the same assumption. They assumed that the body of the robot is horizontal at all times in order to simplify the solution. Regardless of their method, their solutions do not encompass the motion of the body in different orientations and cannot be generalized for different orientations. Arai et al. [18, 37–40] discuss the mechanical design and basic control of a hexapod robot. They have investigated two kinds of leg designs. One type [37–40] uses a six-bar linkage as the legs. The other type [18] has 4-DOF leg motion. It was indicated that they had solved the inverse kinematics of the robot. However, no solution to the inverse kinematics was presented. Regardless of their solution, they limited the solution by considering one degree of freedom for the inclination of the platform while the robot can have up to three different angles about three different axes ($x, y, z$) for complicated terrains.

Inverse kinematics of serial legs of hexapod robots is similar to any other serial robot and can be calculated using the geometry of the leg by using the law of cosines. The inverse kinematic solution for the 3-DOF legs of a hexapod robot is presented in the next section.

A general analytical solution for the forward kinematics of hexapod robots does not exist. Hence, numerical solutions are used instead. The forward kinematics of serial legs of parallel legged robots can be found using the Denavit-Hartenberg
(D-H) parameters. However, the forward kinematics of parallel hexapod robots is beyond the scope of this research and is not dealt with in this work.

3.2 Inverse Kinematic Solution

In some scenarios, it is necessary for a robot to keep its body in a specific arbitrary orientation while walking or to change the orientation of its body within the workspace while manipulating. For instance, assume a situation in which the robot has to walk with a specific orientation and inclination for its body such as walking on a surface with a complicated slope where it needs to maintain a horizontal body configuration. The complicated slope may have inclinations with respect to multiple axes simultaneously. As another example, to manipulate a spherical surface, the robot may need to move its body while maintaining a complex orientation of the end-effector normal to the surface. Therefore, it is imperative to have an inverse kinematic solution that takes into account any arbitrary position and orientation of the hexapod robot body. To the knowledge of the author, no concise inverse kinematic solution for arbitrary orientation of HWRs with 3-DOF leg motion has been previously presented. This section provides such a solution whereas the other studies generally assume that the robot body remains horizontal with the ground.

To solve the inverse kinematics of the robot when walking, a general ground coordinate system, \(O\), is defined \((X,Y,Z)\) as well as a local coordinate system, \(P\), fixed to the center of the platform \((x,y,z)\). From the top view of the robot in its initial configuration, the \(x\) and \(y\) axes are collinear with the \(X\) and \(Y\) axes, respectfully, as shown in Figure 2.2 and Figure 3.1. Six local coordinate systems \((x_i,y_i,z_i)\) are defined in the platform frame, one per leg, where the platform is connected to the legs by hip joints, as shown in Figure 3.1. Since all of the legs
Looking at Figure 5 (a), the first loop closure includes the position vector \( \overrightarrow{OP} \), foot point vector \( \overrightarrow{ui} \), hip joint vector \( \overrightarrow{s_{i1}} \), and hip leg vector \( \overrightarrow{l_{i}} \). The first loop closure equation is used to determine the hip leg vector and the hip joint angle. From Figure 5 (a), the first loop closure equation is given as

\[
\overrightarrow{l_{i}} = \overrightarrow{O} + R\overrightarrow{s_{i1}} - \overrightarrow{ui} \tag{3.1}
\]

where \( \overrightarrow{O} \) is the position vector from the origin of the general coordinate system to that of the local coordinate system of the platform, \( \overrightarrow{s_{i1}} \) represents the position of \( i \)th hip joint in the platform local coordinate system, \( \{P\} \), \( \overrightarrow{ui} \) is the ground contact point of the \( i \)th leg in the general coordinate system \( \{O\} \), \( \overrightarrow{l_{i}} \) is the \( i \)th hip leg vector and \( R \in so(3) \) is the rotational matrix of the platform of the robot with respect to the general coordinate system on the ground to take into consideration the roll.
Finally, the analytical IKS of the hexapod walking robot is given by

\[
\begin{align*}
\pi \gamma & \quad \alpha \\
\beta & \quad \rho \\
\alpha & \quad \beta
\end{align*}
\]

(8)

The IKS presented in the previous section can be applied to any kind of parallel legged robot (tripods, quadrupeds, hexapods, octopods, and so forth) with 3-DOF leg motion. The IKS algorithm for parallel legged robots can be summarized by the following procedures.

1- Calculate rotation matrix \( \mathbf{R} \) based on the desired orientation of the main body of the robot.
2- Calculate all of the leg vectors based on the desired position \((x,y,z)\) and calculated rotation matrix, \( \mathbf{R} \), according to the Equation (1).
3- Calculate the hip angle \( \alpha \) according to the Equation (2).
4- Calculate the new leg vectors by applying the inverse kinematic problem again using the Equations (3) and (4).
5- Calculate the intermediate angles using Equations (6) and (7).
6- Calculate the knee \( \beta \) and ankle \( \gamma \) angles using Equation (8).

Two case studies are provided to validate the present inverse kinematics solution. The first case study focuses on walking with an arbitrary body orientation. The second study looks at a manipulation application were specific body orientations are necessary.

Figure 3.2: The kinematic closures of the robot.
pitch, and yaw angles of the platform with respect to the general coordinate system. After solving Equation 3.1 for \( \vec{l}_i \), the required angle for the first revolute joint can be calculated from the hip leg vector because all of the leg segments are coplanar. The angle \( \alpha_i \) is derived from

\[
\alpha_i = \arctan \left( \frac{l_{i,Y}}{l_{i,X}} \right)
\]

(3.2)

where \( l_{i,X} \) and \( l_{i,Y} \) are the magnitudes of projection of the leg vector \( \vec{l}_i \) on the \( X \) and \( Y \) axes in the general coordinate system, respectively.

### 3.2.2 Knee joint vector calculation

As shown in Figure 3.2-b, the second loop closure equation includes the hip joint vector \( \vec{s}_{i1} \), coxa vector \( \vec{l}_{1i} \), and knee joint vector \( \vec{s}_{i2} \). The second loop closure is used to find the knee joint vector \( \vec{s}_{i2} \) which is in the platform local frame, \( \{P\} \). The local coordinate system of \( i \)th leg is used to calculate the new vector for each leg using

\[
\vec{s}_{i2} = \begin{bmatrix}
  s_{i1x} + (-1)^i \cdot l_{i1} \cdot \cos(\alpha_i) \\
  s_{i1y} + (-1)^i \cdot l_{i1} \cdot \sin(\alpha_i) \\
  s_{i1z}
\end{bmatrix}
\]

(3.3)

where \( s_{i1x}, s_{i1y}, \) and \( s_{i1z} \) are the magnitude of the projection of the knee joint vector \( \vec{s}_{i2} \) on the \( x \), \( y \), and \( z \) axes in the platform local coordinate system, respectively.

### 3.2.3 Knee leg vector calculation

The third loop closure equation includes the position vector \( \vec{O} \), foot point vector \( \vec{u}_i \), knee joint vector \( \vec{s}_{i2} \), and knee leg vector \( \vec{l}'_i \) as shown in Figure 3.2. The third loop closure is used to find the knee leg vector \( \vec{l}'_i \) and thereby its length, \( l'_i \). Applying
Equation 3.1-c again for the third loop closure, the knee leg vector, $\vec{l}_i'$, can be calculated according to

$$\vec{l}_i' = \vec{O} + R\vec{e}_{i2} - \vec{u}_i$$  \hspace{1cm} (3.4)

### 3.2.4 Knee and Ankle Angles Calculation

As shown in Figure 3.2-d, the fourth loop closure includes the knee leg length $l'_i$, femur length $l_{2i}$, and tibia length $l_{3i}$, and is used to find the knee and ankle angles $(\beta_i, \gamma_i)$ using the law of cosines which are calculated from

$$\cos(\lambda_i) = \frac{l_i'^2 + l_{3i}^2 - l_{2i}^2}{2l_i' l_{3i}}$$

$$\cos(\pi - \gamma_i) = \frac{l_{2i}^2 + l_{3i}^2 - l_i'^2}{2l_{2i} l_{3i}}$$

$$\cos(\rho_i + \phi_i + \beta_i) = \frac{l_{2i}^2 + l_i'^2 - l_{3i}^2}{2l_{2i} l'_i}$$  \hspace{1cm} (3.5)

where $\lambda_i$, $\gamma_i$, and $\rho_i$ are intermediate angles shown in Figure 3.3.

Looking at Figure 3.3, knowing $\vec{l}_i'$ and $\vec{l}_i$ provides $h_i'$ and $h_i$ since $h_i' = l'_i, z$ and $h_i = l_i, z$. The angle $\rho_i$ is then calculated using

$$\rho_i = \tan^{-1}\left[ \frac{h_i'}{\sqrt{l_i'^2, x + l_i'^2, y}} \right]$$  \hspace{1cm} (3.6)

and the angle $\phi_i$ is derived from

$$\phi_i = \sin^{-1}\left[ \frac{h_i' - h_i}{l_{1i}} \right]$$  \hspace{1cm} (3.7)
3.2.5 Analytical solution

The analytical inverse kinematic solution of the HWR is given by

\[
\begin{align*}
\alpha_i &= \arctan \left( \frac{l_{i,y}}{l_{i,x}} \right) \\
\beta_i &= \cos^{-1} \left( \frac{l_{i2}^2 + l_{i3}'^2 - l_{i3}^2}{2l_{i2}l_{i3}'} \right) - (\phi_i + \phi_i) \\
\gamma_i &= \pi - \cos^{-1} \left( \frac{l_{i2}^2 + l_{i3}^2 - l_{i3}'^2}{2l_{i2}l_{i3}} \right)
\end{align*}
\] (3.8)

As illustrated in Figure 3.2 and shown in the corresponding equations, by dividing the leg model into four individual loop closures and using given position and orientation (rotation matrix), hip, knee, and ankle angles of all of the legs can be calculated using Equation 3.8 to satisfy the required position and orientation of the platform. The solution is applicable to any parallel legged robot with 3-DOF of leg motion no matter how many legs the robot has on the ground.

3.2.6 Inverse Kinematic Algorithm

The inverse kinematic algorithm for parallel legged robots can be summarized by the following procedures.
1. Calculate rotation matrix R based on the desired orientation (roll, pitch, yaw) of the robot platform.

2. Calculate all of the leg vectors based on the desired position of the platform \((x, y, z)\) and calculated rotation matrix, \(R\), according to Equation 3.1.

3. Calculate the hip angle \((\alpha_i)\) according to Equation 3.2.

4. Calculate the knee leg vectors by applying the inverse kinematic problem again using Equation 3.3 and Equation 3.4.

5. Calculate the intermediate angles \(\gamma_i\) and \(\rho_i\) using Equation 3.6 and Equation 3.7.

6. Calculate the knee \((\beta_i)\) and ankle \((\gamma_i)\) angles using Equation 3.8.

### 3.3 Validation

Two case studies are provided to validate the present inverse kinematic solution. The first case study focuses on walking with an arbitrary body orientation. The second study looks at a manipulation application were specific body orientations are necessary.

#### 3.3.1 Walking

A gait is the order or manner of the landing and lifting of legs of a multi-legged robot to provide a walking or running procedure. For every leg of the robot, there are two phases: the support (stance) phase and the transfer (swing) phase. When walking, the order of changing between support and transfer phase will define the gait of the robot and a trajectory should be followed in transfer phase for each leg.
For a multi legged robot to walk, both gait analysis and foot trajectory planning should be studied.

**Gait Analysis**

There is a substantial amount of literature about walking robot gaits with a wide range of definitions and applications [4, 20, 41–66]. In general, there are two types of walking gaits: free gait (non-periodic gait) and periodic gait.

**Free Gait (Non-Periodic Gait)**

In a free gait, the feet do not follow a periodic behaviour [4, 67, 68]. Any gait which is not periodic is a free gait or non-periodic gait. Free gait is useful for a walking robot which is considered for walking in unknown environments over uneven terrain.

**Periodic Gait**

In a periodic gait, each leg does the same behavior periodically. The wave gait, crawl gait, crab-walking, turning gait, creeping gait, and tripod and tetrapod gaits are different types of periodic gaits [69–75]. The most known periodic gait for walking robots is the wave gait in which the legs have a wavy motion from the rear of the robot to the front or vice versa. It has been shown that the wave gait is the most optimally stable gait among periodic gaits [41, 47].

Considering Figure 3.4 and Figure 3.5, some important definitions for better understanding of periodic gaits are as follows.

- **Regular gait** is a periodic gait in which all the legs have the same duty factor.
- **Duty factor** ($\beta$) is the ratio of the supporting interval to the cycle time.
- **Tripod gait** is a wave gait with duty factor of 0.5 ($\beta = 0.5$).
There are two general types of walking gaits:

• Free Gaits (Non-Periodic Gaits)
• Periodic Gaits:

In a periodic gait, each leg does the same behavior periodically. Regular gait is of periodic gaits in which all the legs have the same duty factor. If the motion of the legs of any right-left pair is exactly half of cycle out of phase, the gait is said to be symmetric. The wave gait is a symmetric forward gait in which leg lifting occurs from rear of the robot to front as a wave, and each adjacent pair of legs (right or left) has \((1 - \beta)\) phase difference with each other. It means, when a leg touch the support surface, the adjacent leg lifts up at the same time. It has been shown that the wave gaits are the most optimally stable gaits among periodic gaits.

Figure 3.4: The support and transfer phases in one walking cycle.

Figure 3.5: Gait parameter definitions for a walking robot.
• **Cycle time** is the time period of one walking step.

• **Stride length** $L$ is the body translated displacement in one cycle time.

• **Kinematic Cycle Phase** ($\phi$) is the translated distance of the body since the last placement of leg 1 normalized by stride length. It can also be defined as the time normalized by one cycle time.

• **Relative Phase of leg $i$** ($\varphi_i$) is the touch-down instance normalized by one cycle time. In other words, it is the value of kinematic phase when the leg touches down. It is assumed that $\varphi_1 = 0$.

• **Symmetric gait** is a gait in which the motion of the legs of any right-left pair is exactly half of cycle out of phase.

• **Forward wave gait** is a wave gait in which the leg lifting occurs from rear of the robot to front as a wave, and each adjacent pair of legs (right or left) has $(1 - \beta)$ phase difference with each other. Hence, when a leg touches the support surface, the adjacent leg lifts up at the same time.

• **Symmetric forward wave gait** is the forward wave gait in which the relative phase of legs 1, 3, and 5 differs from that of legs 2, 4, and 6 by 0.5.

• **Crab angle** is the angle between robots heading (assumed to be leg 1) and direction of the CG velocity ($\alpha(t) = (2n - 1) \pi/6$).

An example of gait generation will be presented in the next section, to show how one can determine the gait and foot trajectory planning of the robot for a desired walking process.
Foot Trajectory Planning

Foot trajectory planning is the process of determining the position and velocity of each foot during its transfer phase. As shown in Figure 3.6, the input to the foot trajectory planning algorithm are the desired gait parameters such as kinematic cycle phase, duty factor, and relative phase as well as desired foot positions and speeds [76–79].

Other inputs are defined as follows.

- **Leg Phase Variable of leg i** ($\phi_{Li}$) is the difference between kinematic phase of the leg and its relative phase. If the leg is in the support phase, $0 \leq \phi_{Li} \leq \beta$, and if it is in the transfer phase, then $\beta \leq \phi_{Li} \leq 1$.

- **Transfer Phase Variable of leg i** ($\phi_{Ti}$) is defined as $\phi_{Ti} = \frac{\phi_{Li} - \beta}{1 - \beta}$.

Once all these desired parameters are given, the trajectory planning algorithm will solve for the required positions and speeds of the joints of the leg according to the kinematics of the leg. The leg in transfer phase can follow any arbitrary trajectory. During the transfer phase, the following equation is valid.

$$u(t) = \frac{\beta(t)}{1 - \beta(t)} v(t) \rightarrow \bar{x}_{f/g} = \frac{1}{\tau_T} \int_{t=0}^{\tau_T} \bar{x}_{f/g}(t) dt = \frac{\beta}{1 - \beta} \bar{x}_{b/g}$$  \hspace{1cm} (3.9)

where $\bar{x}_{f/g}$ is the average velocity of the leg during the transfer phase with respect to the ground coordinate system and $\bar{x}_{b/g}$ is the average velocity of the center of
the gravity of the robot body with respect to the ground coordinate system. In this analysis, there are three different coordinate systems: \{b\} which is the body fixed frame, \{g\} which is the ground (inertial) coordinate system, and \{f\} which is the foot coordinate system.

**Implementation**

According to Figure 3.7, the crab angle is \( \alpha = \pi/6 \). Let’s select the duty factor to be \( \beta = 0.75 \). Then, Figure 3.8 shows the gait planning of the given system. For each leg, the grey area shows the support phase and the white area shows the transfer phase. For the robot to walk properly, inverse kinematics of the parallel robot by itself should be used for support phase as described before and foot trajectory planning should be used for transfer phase as follows.

Considering the desired trajectory shown in Figure 3.9, the following equation
Figure 3.8: The gait generation of a HWR with duty factor of $\beta = 0.75$.

Figure 3.9: The desired leg trajectory for each leg during the transfer phase.

can be written:

$$x_{f/b}(t) = x_{f/g}(t) - x_{b/g}(t)$$

and thereby

$$v_{f/b}(t) = v_{f/g}(t) - v_{b/g}(t)$$

It is assumed that the robot velocity is in direction of $x$ axis and the leg trajectories and velocities are in the same direction. Therefore, the only components of
Table 3.1: D-H parameters of a leg of hexapod robot with 3-DOF.

<table>
<thead>
<tr>
<th></th>
<th>( \theta )</th>
<th>( d )</th>
<th>( a )</th>
<th>( \alpha )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \theta_1 )</td>
<td>0</td>
<td>( l_1 )</td>
<td>(-\pi/2)</td>
</tr>
<tr>
<td>2</td>
<td>( \theta_2 )</td>
<td>0</td>
<td>( l_2 )</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>( \theta_3 )</td>
<td>( l_3 )</td>
<td>0</td>
<td>(-\pi/2)</td>
</tr>
<tr>
<td>4</td>
<td>( \pi/2 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

the leg trajectory and velocity will be \( x \) and \( z \) and the following can be written for the position and velocity of the legs.

\[
\begin{bmatrix}
    x_{f/b}(t) \\
    z_{f/b}(t)
\end{bmatrix} =
\begin{bmatrix}
    x_{f/g}(t) \\
    z_{f/g}(t)
\end{bmatrix} -
\begin{bmatrix}
    x_{b/g}(t) \\
    z_{b/g}(t)
\end{bmatrix}
\tag{3.12}
\]

and

\[
\begin{bmatrix}
    \dot{x}_{f/b}(t) \\
    \dot{z}_{f/b}(t)
\end{bmatrix} =
\begin{bmatrix}
    \dot{x}_{f/g}(t) \\
    \dot{z}_{f/g}(t)
\end{bmatrix} -
\begin{bmatrix}
    \dot{x}_{b/g}(t) \\
    \dot{z}_{b/g}(t)
\end{bmatrix}
\tag{3.13}
\]

The right hand side of Equations 3.12 and 3.13, which are the components of position and velocity of the foot and body CG with respect to the ground, are given. Therefore, the left hand side of the equations can be calculated which are the components of the position and velocity of the foot with respect to the body of the robot during the transfer phase.

Joint positions and velocities are the only parameters understandable for the robot to accomplish the foot trajectories. The inverse kinematics of serial legs and inverse jacobian of the legs can be used to calculate for positions and velocities of the joints to satisfy the desired trajectory. Figure 3.10 and Table 3.1 show the coordinates definition of the leg and D-H parameters required for calculation of the position of the foot with respect to the body frame and thereby the jacobian matrix required for calculation of the joint velocities.
As mentioned, Figure 3.9 shows the position and velocity outputs from the foot trajectory planning to satisfy the motion. For calculating the joint positions, inverse kinematics of the serial leg is used, and for calculating the joint speeds, the jacobian matrix is used. Both tripod and wave gaits are validated through simulation and physical experiment. In the simulation and experiment, for the wave gait, the robot follows what is shown in Figure 3.8 and Figure 3.9 for the gait generation and leg trajectory, respectively, and for the tripod gait, \( \beta = 0.5 \) is used for the gait generation while the robot follows the leg trajectory shown in Figure 3.11.

For the tripod gait, during the motion, legs 1, 4 and 5 will create a tripod and legs 2, 3, and 6 will create another tripod. When a gait is active, it causes the platform to move, and all the joints are moving according to the angles provided by the inverse kinematic solution which are based on the next desirable position and orientation (inclination) of the platform. The legs of the active gait are in the support phase and the legs in the other gait are in the transfer phase.

After a gait is selected, suitable leg trajectories have to be determined to create the gait [44]. Figure 3.11 shows the leg trajectories for 5 seconds of the walking
Figure 3.11: The leg trajectories for 5 seconds of the walking process.

process based on the tripod gait. Legs 1, 4, and 5 correspond to the first gait and legs 2, 3, and 6 correspond to the second gait. Therefore, leg 4 is chosen from the first gait as shown in Figure 3.11-a and leg 3 is chosen from the second gait as shown in Figure 3.11-b. The transferring legs, which are not in the active zones, will lift up, rotate at the hip, and rotate back down. This process will interchange between the gaits periodically and enable the robot to walk.

When walking, it is important to make sure that the robot remains stable during all necessary motions [17,44]. Figure 3.12 shows the top view of the robot with three supporting legs and three transferring legs based on the tripod gait. The support polygon is the triangle created by three ground contact points $g_1$, $g_4$, and $g_5$. The stability margin is defined as the minimum distance the projected CG is from the support polygon.
This issue will be considered in the simulations and experiments such that the robot does not exceed a desired stability margin. From Figure 3.12, the minimum stability threshold in the home pose of the robot, before moving, is \( d = d_1 = d_2 = d_3 \).

**Walking simulation and Experiment**

The presented inverse kinematics was used for the robot to walk based on the wave gait and the tripod gait. As mentioned, for the wave gait, the robot follows the gait generation and leg trajectory shown in Figure 3.8 and Figure 3.9, respectively, and the tripod gait uses \( \beta = 0.5 \) for the gait generation while the robot follows the leg trajectory shown in Figure 3.11.

As an example to show that the presented kinematic solution is accurate and can be used for the wave gait, both simulation and experiment were completed. A snap shot of wave gait walking using the presented solution is shown in Figure 3.13 for both simulation and experiment. The Open Dynamics Engine (ODE) [80] was used to simulate the hexapod robot. It is assumed that the robot has to walk in \( X \) direction with 10 mm heading in each cycle such that it has to keep an arbitrary
height of 50 mm. The simulation and experiment were completed with a horizontal platform for the wave gait.

As another example to show the capability of the presented inverse kinematic solution, an arbitrary position and orientation of the platform and heading direction of the robot were selected. With the $X - Y - Z$ direction from Figure 3.2, it is assumed that the robot has to walk in a straight line at an angle of 25 degrees relative to the $X$ direction. The robot advances 10 mm with each step and must maintain an arbitrary height of 23 mm and a platform orientation of 10°, 10°, and 5° about the $X$, $Y$, and $Z$ axes, respectively.

The left side of Figure 3.14-(a-g) provide snap shots of the robot in motion in the simulation environment and the corresponding experiment. Figure 3.14-a shows the home pose of the robot. Figure 3.14-b shows that the robot has satisfied the orientation (inclination) of the robot platform. When the robot satisfies the orientation of the platform, it should walk with that orientation to satisfy the position and heading direction. The legs of the first gait lift up, according to Figure 3.11-a, after the time 0 ms. At the same time, the other legs in the second gait moves exactly based on what the inverse kinematic solution provides which corresponds to
6.1.2 Walking Experiment

To validate the simulation results, the same inverse kinematics driven walking strategy was executed on a physical robot. The snapshots of the physical robot in motion are shown in Figure 10 and were taken to coincide with the simulation snapshots shown in Figure 9.

6.2 Object Manipulation

This case study demonstrates the capability of the robot to satisfy any arbitrary position and orientation of the robot tip for manipulation purpose. The best example which can show the capability of the presented solution to manipulate is to manipulate a spherical surface. Figure 11 shows two complicated sample points on a sphere, S and E, to which the manipulator tip of the robot has to be perpendicular. One of the points (S) is picked up as a complicated point on the surface of the sphere.

Figure 14: The snap shots of the robot in the tripod motion in the simulation environment and the corresponding experiment.

the same period of time in Figure 3.11-b. Figure 3.14-c shows this period, before the legs of the first gait come back down on the ground. After that, the legs related to the second gait lift up and the same thing as what is already discussed happens for this gait. This is what Figure 3.14-d represents. Figure 3.14-(e-g) are the next sequential steps of walking process to show how the robot walks forward.

To validate the simulation results, the same inverse kinematics driven walking strategy was executed on a physical robot. The snapshots of the physical robot in motion are shown in the right hand side of Figure 3.14 and were taken to coincide with the simulation snapshots shown in the left side of Figure 3.14.
3.3.2 Object Manipulation

This case study demonstrates the capability of the robot to satisfy any arbitrary position and orientation of the robot’s end-effector for manipulation purposes. The best example which can show the capability of the presented solution to manipulate is to manipulate a spherical surface. Figure 3.15 shows two sample points on a sphere, S and E, to which the manipulator tip of the robot has to be perpendicular. One of the points (S) is picked up for validation. Figure 3.16 shows how the presented solution enables the robot to do so. Since the tool has to be perpendicular to the surface at this point, the robot platform has to go to a specific position and orientation maintaining locations $x, y, z$ and angles $a, b, c$ simultaneously. Angles $a, b, c$ are the rotation angles of the platform, and thereby the frame $\{P\}$, with respect to the frame $\{O\}$ around $X, Y, Z$ respectively. Figure 3.17 shows the experiment of the case and it corresponds to Figure 3.16 with all details. This example shows that the presented solution makes the robot able to be in any position and orientation as long as it is within the workspace of the robot in which the robot is stable.
Figure 3.16: The simulation result corresponding to Figure 3.15.

Figure 3.17: The experimental result corresponding to Figure 3.15.
Chapter 4

Workspace

When the HWR is not walking but standing for manipulation purposes, the footholds do not change and the combination of all points reachable by the manipulator end-effector attached to the robot defines the workspace of the robot.

4.1 Background

The workspace of hexapod robots can be divided into two categories: the constant orientation workspace (COW) [3,81–84] and the orientation workspace [85–88]. Orientation workspace can be further categorized into reachable (maximal), inclusive, total orientation, and dexterous workspace. Among all the types of orientation workspace, the Reachable Workspace (RW) is very important for all types of hexapod robots because it surrounds all other types of workspace and provides information on the size of a space in which the robot can safely work. While under COW, the orientation of the robot platform is kept constant for a given angle throughout the motion of the hexapod robot, the orientation workspace relaxes the fixed orientation constraint of the platform. COW represents all spatial points which can be reached by the end-effector of the robot’s platform while maintaining a fixed given
A number of researchers have studied the COW of 6-DOF parallel manipulators [3, 81–85, 89–93]. While some of their methods are based on parametric equations, there is no closed-form solution for the final workspace boundary. Also there is an analogous symmetrical theorem of workspace for spatial parallel manipulators with identical kinematic chain presented in [94]. However, their method cannot be considered for mechanisms with non-symmetric and non-identical kinematic chains.

A number of researchers have studied the orientation workspace of different types of planar and spatial parallel manipulators [86, 93, 95–99]. However, their work either did not study the RW or presented a numerical solution for the RW.

None of the aforementioned research presented a distinct closed-form solution for the final boundary of COW and/or RW providing the workspace boundary equations
based on structural parameters. Hence, previously described methods cannot be used for different robots unless each new robot has to go through specific numerical algorithm steps to find the final workspace and its boundary. Also, these methods are not readily usable for the optimum design of the structure and workspace, and are not general enough to be used for both COW and RW simultaneously. The following sections study the workspace of the robot from two aspects: the lateral (planar or 2D) and the spatial (3D) workspace, both in a closed-form manner and for COW and RW.

4.2 Lateral Workspace

The lateral tooling workspace, depicted in Figure 4.2, represents the planar workspace of the robot. Therefore, the lateral translation of a hexapod robot can be reduced to a planar mechanism where each of the six articulated legs of the HWR can be virtually replaced with a prismatic joint. Considering this replacement, in a view perpendicular to the lateral plane shown in Figure 4.2, the HWR performs as a planar 2-$RPR$ parallel mechanism, as shown in Figure 4.3.

The workspace of planar parallel manipulators has been investigated by a number
of researchers. The workspace of the common 5R planar parallel mechanism has been investigated but either focus on the singularity problem [100, 101] or only provide a numerical solution [102, 103]. The workspace of the 3-\textit{RPR} has been previously investigated but they either only solve for the numerical solution for the RW [85] or provide a COW [104, 105]. The 3-\textit{PRRR} and 3-DOF mechanisms are other types of planar parallel mechanisms which have been previously studied [106–109], but their workspace solution is specific to their mechanisms and cannot be generalized to other cases. The workspace solutions of other notable parallel mechanisms also provide numerical solutions to the workspace of their mechanism [99,110–115].

However, as with hexapod robots, there has been no analytical or mathematical closed-form solution to the RW of planar parallel mechanisms where the workspace spans a surface. A direct or closed-form solution is always more favorable due to conciseness and direct calculation. This section analyzes and provides a closed-form solution to the RW of the planar 2-\textit{RPR} parallel mechanism. Since the solution is closed-form, it provides an exact solution to the RW of the mechanism. The provided solution can be generalized for solving any other types of workspace (constant and orientation). Furthermore, it can be employed to solve the workspace of similar mechanisms in a closed-form manner including spatial multi-legged parallel
mechanisms such as hexapod robots even if they are not axially symmetric and have non-identical kinematic chain.

4.2.1 Closed-form Solution for the Workspace Envelope

As mentioned, solving the closed-form solution for the lateral RW of an axially symmetric hexapod robot is equivalent to find the solution for the RW of a planar 2-\textit{RPR} parallel mechanism.

\textbf{2-\textit{RPR} Planar Parallel Mechanism Model}

A general 2-\textit{RPR} planar parallel mechanism, shown in Figure 4.4-a, is constructed from two prismatic links that are grounded at one end and connected to a moving platform (triangle $efb$) at the other end. Each specific configuration of the 2-\textit{RPR} mechanism is defined by a tuple of structural parameters, $C(a, b, c, d, e, \beta, \theta_1, \theta_2)$ where $a$ is the length of the first or left prismatic link, $b$ is the length of the second or body link, $c$ is the length of the third or right prismatic link, $d$ is the distance between the grounded links, $e$ is the length of the follower, $\beta$ is the follower angle, $\theta_1$ is the angle between link $d$ and the ground plane, and $\theta_2$ is the input angle or the angle between link $a$ and the ground plane. The angles $\theta_3$ and $\theta_4$ are dependent variables. When links $a$ and/or $c$ vary, the configuration varies accordingly and moves the reference point $p$ (tool center point) within the RW of the mechanism. The 2-\textit{RPR} mechanism is similar to a four-bar mechanism with two additional prismatic joints and has driven 3-DOF ($\theta_2$, $a$, and $c$) that control the reference point of the mechanism.

The RW of the 2-\textit{RPR} mechanism can be interpreted as the successive combination of the coupler curves formed by varying the prismatic lengths. With constant prismatic lengths, the 2-\textit{RPR} becomes a four-bar mechanism which has been thor-
Figure 4.4: A general 2-RPR planar parallel mechanism a) Parameters definition b) The RW.

oughly studied with regards to trajectory and path generation [116,117]. The complete geometrical (numerical) solution of the four-bar coupler curve can be found in [116]. Due to the varying grounded link lengths \( a \) and \( c \), the workspace of the 2-RPR mechanism is a coupler surface, as shown in Figure 4.4-b.

The RW of the 2-RPR mechanism can be found numerically by looping through the possible crank angles, \( \theta_2 \), while varying lengths \( a \) and \( c \). Figure 4.5 shows the RW of a 2-RPR planar parallel mechanism based on the structural parameters listed in Table 4.1. The numerical solution assumed \( e = b/2 \) and \( \beta = 0 \). As the angle describing the ground plane, the effects of \( \theta_1 \) on the RW solution reduces to a planar transformation about the origin of the \( x - y \) coordinate system. Hence, the contribution of \( \theta_1 \) can be applied after the trivial, \( \theta_1 = 0 \), RW solution has been found. The link angles, \( \theta_2 \) and \( \theta_4 \) are bound between 0 and \( \pi \) since the left and right grounded prismatic links are assumed to be physically constrained by the ground at those boundary angles.

From Figure 4.5, the RW of the 2-RPR is symmetric about the \( A \)-axis, utilizing the same minimum and maximum constraints for lengths \( a \) and \( c \). Constraining
Figure 4.5: RW of a 2-RPR planar parallel mechanism based on the structural parameters listed in Table 4.1.

Table 4.1: An example set of structural parameters for a 2-RPR planar parallel mechanism.

<table>
<thead>
<tr>
<th></th>
<th>(a) (mm)</th>
<th>(b) (mm)</th>
<th>(c) (mm)</th>
<th>(d) (mm)</th>
<th>(\theta_1) (rad)</th>
<th>(\theta_2) (rad)</th>
<th>(\theta_3) (rad)</th>
<th>(\theta_4) (rad)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>260</td>
<td>270</td>
<td>260</td>
<td>480</td>
<td>0</td>
<td>0</td>
<td>(-\pi)</td>
<td>0</td>
</tr>
<tr>
<td>Max</td>
<td>360</td>
<td>270</td>
<td>360</td>
<td>480</td>
<td>0</td>
<td>(\pi)</td>
<td>(\pi)</td>
<td>(\pi)</td>
</tr>
</tbody>
</table>

the problem to symmetrical prismatic extensions reduces the analysis to half of the workspace. Hence, once the solution to either the left or right half of the RW has been found, it can then be mirrored about the \(A\)-axis in order to obtain the full RW.

For ease of discussion and presentation, the rest of the paper assumes \(e = b/2\) and \(\theta = 0\). However, the presented solution methodology is applicable to the general case.

RW Decomposition

The RW of the 2-RPR mechanism, shown in Figure 4.5, is decomposed into eight sub-workspaces, \(W_i\) for \(i = 1, \ldots, 8\), as shown in Figure 4.6, to facilitate finding the closed-form solution of the boundary of the RW. Each sub-workspace
In Fig. 3, $\gamma$ is a coupler curve (bold line) if $\mu$ is odd, and is a coupler surface if $\mu$ is even. All coupler surfaces $\gamma(\mu):$ even are bounded between two coupler curves, $\gamma^{-1}$ and $\gamma^{+1}$. Generally, the RW envelope is a combination of the boundary of the coupler surfaces. The boundary of a coupler surface is created under critical (singular) configurations that occur either when the link hits the ground ($\theta_2 = 0$) or when $\varphi$ and $\omega$ are aligned, $\theta_2 = \theta_7 - \mu, \pi \mu = 0, 1$. Also, all coupler curves, as part of the boundary of the coupler surfaces, are created when both prismatic links are in their maximum or minimum extension. Focusing on the right half side of the workspace, the boundary of the RW is created from several separate cardinal curves, $\mathcal{C}_{\mu}\sigma$, which are bounded by cardinal points $\gamma_{\mu}$ and $\gamma_{\lambda}$. In general, there are eight cardinal points $\gamma_{\mu}, \mu = 0…7$, and nine cardinal curves including $\mathcal{C}_{\omega1}, \mathcal{C}_{\omega2}, \mathcal{C}_{\omega3}, \mathcal{C}_{\omega4}, \mathcal{C}_{\omega5}, \mathcal{C}_{\omega6}$, and part of $\gamma_{\mu}$ and $\gamma_{\lambda}$. The cardinal points and curves, which are shown in Fig. 4, exist in all possible shapes of the RW created by different sets of structural parameters. Changes to the structural parameters will affect $\gamma_{\mu}$ and $\gamma_{\lambda}$.

**Figure 4.6:** The decomposition of the RW of a 2-\textit{RPR} mechanism.

is based on changes in the structural parameters as given in Table 4.2 where $W_i$ is the $i$th sub-workspace, and $l_{\min}$ and $l_{\max}$ are the minimum and maximum leg lengths, respectively, related to the grounded links $a$ and $c$. When $i$ is odd, the sub-workspace $W_i$ collapses into a coupler curve, $C_i$. When $i$ is even, the sub-workspace forms a coupler surface, $S_i$, bounded between two coupler curves, $C_{i-1}$ and $C_{i+1}$. The coupler curves and surfaces are shown in Figure 4.7.

Generally, the RW envelope is a combination of the boundaries of the coupler surfaces which are created under critical (singularity) configurations that oc-
Table 4.2: All possible structural parameter configurations for generating the sub-workspaces of the RW.

<table>
<thead>
<tr>
<th></th>
<th>(W_1)</th>
<th>(W_2)</th>
<th>(W_3)</th>
<th>(W_4)</th>
<th>(W_5)</th>
<th>(W_6)</th>
<th>(W_7)</th>
<th>(W_8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>(l_{\max})</td>
<td>(l_{\max})</td>
<td>(l_{\max})</td>
<td>(l_{\max} \ldots l_{\min})</td>
<td>(l_{\min})</td>
<td>(l_{\min})</td>
<td>(l_{\min})</td>
<td>(l_{\min} \ldots l_{\max})</td>
</tr>
<tr>
<td>c</td>
<td>(l_{\max})</td>
<td>(l_{\max} \ldots l_{\min})</td>
<td>(l_{\min})</td>
<td>(l_{\min})</td>
<td>(l_{\min} \ldots l_{\max})</td>
<td>(l_{\max})</td>
<td>(l_{\max})</td>
<td></td>
</tr>
</tbody>
</table>

cur either when the link hits the ground \((\theta_2 = 0)\), when \(a\) and \(b\) are aligned, \(\theta_2 = \theta_3 \pm i\pi, i = 0, 1,\) or when both prismatic links are in their maximum or minimum extension.

Focusing on the right half side of the workspace as shown in Figure 4.7, the boundary of the RW is created from several curves, \(C_{ij}\), which connects points \(p_i\) and \(p_j\), and the coupler curves \(C_i\). The cardinal curves and cardinal points are the curves and points that exist in all possible reachable workspaces generated by any set of structural parameters. In general, as shown in Figure 4.7, there are up to eight cardinal points \(p_i, i = 0, \ldots, 7,\) and nine cardinal curves \(p_{ij}\) including \(C_{01}, C_{12}, C_{23}, C_{34}, C_{35}, C_{56}, C_{46}, C_{47},\) and \(C_7\). Changes to the structural parameters will change the placement of these cardinal points and curves and, accordingly, vary the shape of the workspace envelope. In some configurations, the cardinal curves may intersect, defining the position of auxiliary points. Auxiliary points trim cardinal curves where the remaining curves are considered auxiliary curves. Therefore, once the cardinal points and curves have been calculated, the auxiliary points and curves can be found accordingly. Figure 4.8 superimposes Figure 4.7-(a-d), showing all cardinal curves and points.

Looking at Figure 4.8 and Figure 4.9, for the structural parameters in Table 4.1, there are two auxiliary points \(p_8\) and \(p_9\) as a result of intersecting the cardinal curves \(C_{35} - C_{46}\) and \(C_{47} - C_7,\) respectively. As a result of these intersections, a part of the boundary of the RW will be created by the auxiliary curves \(C_{48}, C_{58}, C_{69},\) and \(C_{79}\).
cardinal curves and cardinal points are the curve and points that exist in all possible reachable workspaces generated by any set of structural parameters. In general, as shown in Fig. 4, there are up to eight cardinal points $p_0 \ldots p_7$, and nine cardinal curves including $C_{01}, C_{35}, C_{7}, C_{47}$, respectively, bounded between the points, $p_4 - p_8$, $p_5 - p_8$, $p_6 - p_9$, and $p_7 - p_9$, respectively. Therefore, the shape of the RW strongly depends on where the cardinal points and curves are located and where the curves intersect and create auxiliary points and curves.

As mentioned before, it is very important to know the RW envelope because every point inside the envelope would be reachable for the end-effector of the robot. In other words, the end-effector of the robot cannot go beyond the RW envelope

Figure 4.7: The boundary points and curves of the RW.
As shown in Fig. 6, for the given structural parameters in Table 2, there are two auxiliary points $p_6$ and $p_7$ as a result of intersecting the cardinal curves $C_{56}$ and $C_{46}$, respectively. As a result of these intersections, a part of the RW envelope will be created by the auxiliary curves $C_{56}$, $C_{46}$, $C_{47}$, and $C_{35}$ which are part of the cardinal curves $C_{56}$, $C_{46}$, $C_{47}$, and $C_{35}$, respectively, bounded between the points $p_5$, $p_6$, $p_7$, and $p_8$, respectively. Therefore, the shape of the RW strongly depends on where the cardinal points and curves are located and where the curves intersect and create auxiliary points and curves.

Figure 4.8: Superimposition of Figure 4.7-(a-d).

Figure 4.9: The cardinal and auxiliary points and curves of the workspace boundary.
at all. In what follows, all cardinal and auxiliary curves, $C_{ij}$, and corresponding bounding points, $p_i$ and $p_j$, are solved in a closed-form manner. If the closed-form solution of all cardinal and auxiliary points and curves are given, the closed-form solution to the entire RW envelope will be defined.

**Closed-form Solution for Cardinal Points and Curves**

The first point, $p_0$ in Figure 4.10-a, represents the center of link $b$ corresponding to $W_1$ when $\theta_3 = 0$. Using the format $p_i = \langle p_{ix}, p_{iy} \rangle$, the calculation for point $p_0$ is given by

$$p_0 = \left\langle \frac{d}{2}, l_{\text{max}} \sin \left( \theta_0^0 \right) \right\rangle$$

(4.1)

where $\theta_0^0 = \cos^{-1} \left[ \frac{d-b}{2l_{\text{max}}} \right]$. The angle $\theta_i^j$ is the angle $\theta_j$ when the center of the link $b$ is at point $p_i$.

Point $p_0$ is the start point of the curve $C_{01}$ which is part of the original coupler curve $C_1$. The closed-form solution for the coupler curve $C_1$ is given in [117–119]. The cardinal curve $C_{01}$ is bounded between points $p_0$ and $p_1$. This curve and the related points are shown in Figure 4.10-a. The solution for $p_1$ is given by

$$p_1 = \left\langle (l_{\text{max}} + \frac{b}{2}) \cos \left( \theta_1^1 \right), (l_{\text{max}} + \frac{b}{2}) \sin \left( \theta_1^1 \right) \right\rangle$$

(4.2)

where $\theta_1^1 = \cos^{-1} \left( \frac{(l_{\text{max}}+b)^2+d^2-l_{\text{max}}^2}{2(l_{\text{max}}+b)d} \right)$.

The cardinal curve $C_{12}$ is part of the boundary of the coupler surface $S_2$ which is generated when the left prismatic link, $a$, and the platform link, $b$, are aligned, $\theta_2 = \theta_3$. During the generation of the cardinal curve $C_{12}$, $a = l_{\text{max}}$ and $l_{\text{max}} \geq c \geq l_{\text{min}}$. The curve $C_{12}$ is part of a circle whose center is at the origin $(0, 0)$ with a radius equal to the sum of the left link plus the half of the platform link, $l_{\text{max}} + \frac{b}{2}$, bounded between points $p_1$ and $p_2$. The initial and final configurations creating curve $C_{12}$ and
Figure 4.10: The different 2-RPR configurations that generate the RW envelope.
the corresponding points are shown in Figure 4.10-b. The equation for the cardinal curve $C_{12}$, bounded between $p_1$ and $p_2$, is given by

$$x^2 + y^2 - \left(l_{\text{max}} + \frac{b}{2}\right)^2 = 0 \quad (4.3)$$

and the solution for point $p_2$ is given by

$$p_2 = \left\langle (l_{\text{max}} + \frac{b}{2})\cos(\theta_2^2), (l_{\text{max}} + \frac{b}{2})\sin(\theta_2^2) \right\rangle \quad (4.4)$$

where $\theta_2^2 = \cos^{-1}\left(\frac{(l_{\text{max}}+b)^2+d^2-l_{\text{min}}^2}{2(l_{\text{max}}+b)d_{\text{min}}}\right)$.

As with the cardinal curve $C_{01}$, the cardinal curve $C_{23}$ is part of the coupler curve $C_3$ and generated when $a = l_{\text{max}}$ and $c = l_{\text{min}}$. As shown in Figure 4.10-c, this curve is bounded between points $p_2$ and $p_3$. The solution for the curve $C_{23}$ is given in [119] and the solution for $p_3$ is given by

$$p_3 = \left\langle l_{\text{max}} + \frac{b}{2}\cos(\theta_3^3), \frac{b}{2}\sin(\theta_3^3) \right\rangle \quad (4.5)$$

where $\theta_3^3 = \cos^{-1}\left(\frac{b^2+(d-l_{\text{max}})^2-l_{\text{min}}^2}{2b(d-l_{\text{max}})-l_{\text{min}}}\right)$.

As shown in Figure 4.10-d, the cardinal curve $C_{34}$ is part of the boundary of the coupler surface $S_4$. The curve $C_{34}$ is created when $\theta_2 = 0$, $c = l_{\text{min}}$, and $l_{\text{max}} \geq a \geq l_{\text{min}}$. Therefore, the 2-RPR mechanism can be modeled as a PRRR mechanism since $\theta_2$ is fixed at zero as shown in Figure 4.11. In Figure 4.11, the center of the platform, $p(x, y)$, follows the path $C_{34}$ which is part of a circle with a moving center. The equation of the motion for $p(x, y)$ is given by

$$(x - a)^2 + y^2 - \left(\frac{b}{2}\right)^2 = 0 \quad (4.6)$$
Finally, the boundary of the workspace will be created by connecting all curves. 3.1. Determining the Bounding Curves

In this curve, geometrically, always \( a = l_{\text{min}} \cdots l_{\text{max}} \). This curve is bounded between points \( \theta_3 \) and \( \theta_4 \). Curve 2 \( C_{\theta_2} \), as with the other curves, the path traversed by the couplar point follows the boundary of the workspace. Each bounding curve can be correlated to the prismatic link lengths as provided in Table 2.

Using the angle relation \( \cos(\theta_3) = 2(x - a)/b \) derived from Figure 4.11 and the law of cosines, the following can be derived.

\[
c^2 = b^2 + (d - a)^2 - 4(d - a)(x - a) \tag{4.7}
\]

Solving Equation 4.7 for \( a \) as a function of \( x \) gives

\[
a = \frac{1}{3}d + \frac{2}{3}x \pm \frac{1}{3}\sqrt{4d^2 - 8xd + 4x^2 - 3c^2 + 3b^2} \tag{4.8}
\]

Figure 4.11: The 2-RPR mechanism is modeled as a PRRR mechanism when \( \theta_2 \) is fixed at zero.

Finally, substituting Equation 4.8 into Equation 4.6 and solving for \( y \) as a function of \( x \), the closed-form solution for the PRRR mechanism trajectory, shown in Figure 4.11, bounded between points \( p_3 \) and \( p_4 \), is given by

\[
y = \pm \frac{1}{6}\sqrt{E} \tag{4.9}
\]

where \( E = \pm 8\sqrt{4(x - d)^2 - 3c^2 + 3b^2(x - d) - 20(x - d)^2 + 12c^2 - 3b^2} \). The solution for \( p_4 \) is given by

\[
p_4 = \left\langle \frac{l_{\text{min}}}{2} \cos(\theta_3^4), \frac{b}{2} \sin(\theta_3^4) \right\rangle \tag{4.10}
\]
where $\theta_3^i = \cos^{-1}\left(\frac{b^2 + (d-l_{\text{min}})^2 - l_{\text{max}}^2}{2b(d-l_{\text{min}})}\right)$.

The cardinal curve $C_{35}$, which is part of the boundary of the coupler surface $S_2$, is created when $\theta_2 = 0$, $a = l_{\text{max}}$, and $l_{\text{min}} \leq c \leq l_{\text{max}}$. Geometrically, $C_{35}$ is part of a circle centered at $(l_{\text{max}}, 0)$ with a radius of $\frac{b}{2}$ and bounded between points $p_3$ and $p_5$, as shown in Figure 4.10-e. The equation for the curve $C_{35}$ is given by Equation 4.11.

\[
(x - l_{\text{max}})^2 + y^2 - \left(\frac{b}{2}\right)^2 = 0 \tag{4.11}
\]

The solution for point $p_5$ is given by

\[
p_5 = \left\langle l_{\text{max}} + \frac{b}{2}\cos\left(\theta_3^5\right), \frac{b}{2}\sin\left(\theta_3^5\right) \right\rangle \tag{4.12}
\]

where $\theta_3^5 = \cos^{-1}\left(\frac{b^2 + (d-l_{\text{max}})^2 - l_{\text{max}}^2}{2b(d-l_{\text{min}})}\right)$.

The cardinal curve $C_{46}$, which is part of the boundary of the coupler surface $S_6$, is generated when $\theta_2 = 0$, $a = l_{\text{min}}$, and $l_{\text{min}} \leq c \leq l_{\text{max}}$. Geometrically, $C_{46}$ is part of a circle centered at $(l_{\text{min}}, 0)$ with a radius of $\frac{b}{2}$, but bounded between points $p_4$ and $p_6$. This can be seen in Figure 4.10-f. The equation for $C_{46}$ is given as

\[
(x - l_{\text{min}})^2 + y^2 - \left(\frac{b}{2}\right)^2 = 0 \tag{4.13}
\]

As with the cardinal curve $C_{34}$, the cardinal curve $C_{56}$ is created when $\theta_2 = 0$, $c = l_{\text{max}}$, and $l_{\text{max}} \geq a \geq l_{\text{min}}$. Therefore, the 2-RPR mechanism can be modeled as a $PRRR$ mechanism because $\theta_2$ is fixed to zero. The only difference between $C_{56}$ and $C_{34}$ is that in $C_{56}$, $c = l_{\text{max}}$ while in $C_{34}$, $c = l_{\text{min}}$, and $C_{56}$ is bounded between points $p_5$ and $p_6$ as shown in Figure 4.10-g. The solution for $p_6$ is given by

\[
p_6 = \left\langle l_{\text{min}} + \frac{b}{2}\cos\left(\theta_3^6\right), \frac{b}{2}\sin\left(\theta_3^6\right) \right\rangle \tag{4.14}
\]
where $\theta_3^6 = \cos^{-1}\left(\frac{b^2+(d-l_{min})^2-l_{max}^2}{2b(d-l_{min})}\right)$.

The coupler curve $C_7$ is generated when $a = l_{min}$ and $c = l_{max}$. The coupler curve $C_5$, which forms the cardinal curve $C_{47}$, is generated when $a = l_{min}$ and $c = l_{min}$. The solutions to the curves are given in [119]. The solution to the point $p_7$, shown in Figure 4.10-h, is given by

$$p_7 = \left\langle \frac{d}{2}, l_{min}\sin\left(\frac{\theta_7^2}{2}\right) \right\rangle$$ (4.15)

where $\theta_2^7 = \cos^{-1}\left[\frac{d-b}{2l_{min}}\right]$. The point $p_7$ represents the center of the link $b$ corresponding to $C_5$ when $\theta_3 = 0$.

Closed-form Solution for Auxiliary Points and Curves

From Figure 4.9, the auxiliary points $p_8$ and $p_9$ can be found using the following:

$$C_{46}(x, y) - C_{35}(x, y) = 0 \rightarrow p_8$$

$$C_7(x, y) - C_5(x, y) = 0 \rightarrow p_9$$ (4.16)

The expressions for the curves $C_{35}$ and $C_{46}$ are given in Equation 4.11 and Equation 4.13, respectively, and the equations of $C_7$ and $C_5$ are given in [119].

Summary of the Methodology

For any given set of structural parameters, the direct solution for all eight cardinal points can be found from Equations 4.1, 4.2, 4.4, 4.5, 4.10, 4.12, 4.14, and 4.15 which are summarized in Table 4.3. Also, all nine cardinal curves can be found in a closed-form manner from Equations 4.3, 4.9, 4.11, and 4.13. Wherever the coupler curve equation is needed, the solution presented in [119] is used. The auxiliary
points and curves can be calculated using the cardinal points and curves.

### 4.2.2 Discussion

**Boundary shape:** Points $p_8$ and $p_9$ are auxiliary points which are created as a result of intersecting curves $C_{35}$ – $C_{46}$ and $C_7$ – $C_5$, respectively. Auxiliary points in a workspace may not exist in another workspace. The existence of auxiliary points and their location within the workspace completely depend on the structural parameters. A specific example is when $C_{35}$ intersects $C_{34}$ instead of intersecting $C_{46}$, as shown in Figure 4.12-a. Under this circumstance, the point $p_8$ is no longer on curve $C_{46}$ but on curve $C_{34}$ (almost at top of $p_4$ in this specific figure) and point $p_4$ is no longer a part of the boundary. Another specific type of configurations is when $C_{35}$ and $C_{56}$ are completely enclosed within the workspace and are no longer a part of the boundary as shown in Figure 4.12-b. In Figure 4.12-b, the point $p_9$ does not exist anymore. Instead, a new auxiliary point is created as a result of the intersection between $C_{46}$ – $C_{47}$. The coupler curves $C_7$ and $C_5$ do not intersect. In another specific
exists when point \( p^k \) is located above \( p^{k'} \) (larger \( \kappa \)) as shown in Fig. 9b. Therefore, complete information on the outer and inner boundaries can be achieved once the cardinal points and curves have been derived. However, detailed singularity analysis of the mechanism is another point of interest which is out of the scope of this paper.

In a more general case, the relationship between the terms \( \alpha \) and \( \beta \) can be used to gain an initial understanding of the shape of the workspace. If \( \beta \leq \alpha \), the surface area of the workspace will be bigger than when \( \alpha \geq \beta \). When \( \beta \geq \alpha \), no singularities exist within the workspace. When \( \alpha = \beta \), the bottom of the workspace converges into a single point. All three cases are shown in Fig. 10.

Figure 4.12: Four specific workspace examples.

configuration, the point \( p_6 \) is located in the left-half of the workspace region and the symmetry axis, \( A \)-axis, intersects the boundary of the workspace. This creates two new auxiliary points for the boundary of the workspace as shown in Figure 4.12-c. That portion of the \( A \)-axis between these two points is located within the workspace and is not part of the boundary. Figure 4.12-d shows an example where all of \( C_5 \) and \( C_7 \) lie inside of the workspace.

**Singularity:** The boundary of the workspace is created as a result of the singular configurations of the mechanism. This includes both outer and inner boundaries. There may be inner singular spaces within the workspace as shown in Figure 4.12-c.
One of the advantages of the presented closed-form solution is that once all the cardinal points and curves have been computed based on the structural parameters and constraints, the auxiliary points and curves can be found which define any inner boundaries within the workspace as shown in Figure 4.12-c. In general, if $C_5$ and $C_7$ intersect on the left side of point $p_7$, no inner singularity nor boundary will exist. The same result exists when point $p_6$ is located above $p_7$ (larger $y$) as shown in Figure 4.12-b. Therefore, complete information on the outer and inner boundaries can be achieved once the cardinal points and curves have been derived. However, detailed singularity analysis of the mechanism is another point of interest which is out of the scope of this research.

In a more general case, the relationship between the terms $\frac{d-b}{2}$ and $l_{min}$ can be used to gain an initial understanding of the shape of the workspace. If $l_{min} < \frac{d-b}{2}$, the surface area of the workspace will be bigger than when $\frac{d-b}{2} < l_{min}$. When $l_{min} < \frac{d-b}{2}$, no singularities exist within the workspace. When $\frac{d-b}{2} < l_{min}$, singularities will exist within the workspace. When $\frac{d-b}{2} = l_{min}$, the bottom of the workspace converges into a single point. All three cases are shown in Figure 4.13.
4.2.3 Validation

An example is used to validate the proposed closed-form solution against the numerical solution presented in [116]. Consider a 2-RPR mechanism with the half workspace shown in Figure 4.14. The mechanism has the following structural parameters: $l_{\text{min}} = 100$, $l_{\text{max}} = 200$, $b = 400$, and $d = 600$. Figure 4.14 shows the RW found through numerical simulation (gray area) and closed-form solution (dash line) calculated from Table 4.3 for the points, and Equations 4.3, 4.9, 4.11, and 4.13. From Figure 4.14, the closed-form solution gives an exact solution to the boundary of the RW of the mechanism. Table 4.4 lists the cardinal points for example mechanism.

<table>
<thead>
<tr>
<th>$i$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{ix}(mm)$</td>
<td>300</td>
<td>376</td>
<td>394.4</td>
<td>393.7</td>
<td>300</td>
<td>375</td>
<td>285</td>
<td>300</td>
</tr>
<tr>
<td>$p_{iy}(mm)$</td>
<td>173.2</td>
<td>136.4</td>
<td>66.4</td>
<td>49.60</td>
<td>0</td>
<td>96.8</td>
<td>75.9</td>
<td>0</td>
</tr>
</tbody>
</table>
As shown in Figure 4.14, the cardinal points of the RW are $p_0$, $p_1$, $p_2$, $p_3$, and $p_4$. The intersection between $C_5$ and $C_7$ is a single point, $p_4$. The point $p_7$ coincides with point $p_4$ since $C_5$ is a single point and not a curve. There is no auxiliary point or curve. This example demonstrates how the placement of the cardinal points can vary the boundary shape of the workspace.

### 4.2.4 Application

Once the analytical solution to the boundary of the workspace of the mechanism is given, it is possible to find the maximum rectangular shape inside the workspace of the mechanism as shown in Figure 4.15.

From Figure 4.15, the rectangular area is computed by

$$Area = 2(x - p_{0x}) \cdot (y_2 - y_1) \quad (4.17)$$
where $y_2$ locates either on $C_{01}$ or $C_{12}$, and $y_1$ locates on $C_{34}$. The equations for $C_{01}$, $C_{12}$, and $C_{34}$ already exist as a function of $x$ as already described. Assuming that $y_2$ locates on the curve $C_{12}$, Equation 4.17 will turn into the following equation:

$$\text{Area} = 2(x - \frac{d}{2}) (\sqrt{\left(l_{\text{max}} + \frac{b}{2}\right)^2 - x^2} \pm \frac{1}{6} \sqrt{\pm 8 \sqrt{4(x-d)^2 - 3c^2 + 3b^2 (x-d) - 20(x-d)^2 + 12c^2 - 3b^2}})$$

(4.18)

Therefore, the area is only a function of $x$. To find the maximum rectangular area shown in Figure 4.15, Equation 4.18, as the cost function, needs to be maximized with respect to $x$. If the resultant point $B(x, y_2)$ is not on $C_{12}$, which is the case in this example, it means $y_2$ is on $C_{01}$ not $C_{12}$. Therefore, Equation 4.17 needs to be maximized for the case that $y_2$ locates on $C_{01}$. By maximizing Equation 4.17 for this case, the optimized parameters are found as $x = 353\ mm$, $y_1 = 42.8\ mm$, and $y_2 = 159\ mm$. Hence, $l = 106\ mm$ and $w = 116.2\ mm$. Doing the same analysis without having the analytical solution for the boundary of the workspace needs huge numerical effort to find the maximum rectangle within the workspace.

### 4.3 Spatial Workspace

Generally, a closed-form solution for the workspace boundary of a spatial parallel mechanism is difficult due to a complex surface boundary. The author believes that among the existing algorithms for the workspace of parallel mechanisms, the algorithm provided by Gosselin [81] is a good solution. Gosselin’s methodology is general and usable for different configurations of the robot with any sort of structural parameters. However, no equations of boundary are presented as the final solution and one has to go through a numerical algorithm using the provided equations of
workspace spheres. Also, it can only be used for COW, not RW. The presented methodology in this section overcomes these issues for axially symmetric hexapod robots. The methodology, by itself, is able to be employed for both COW and RW of the robot for both lateral and spatial (3D) cases [120,121]. Also, it is able to present analytical solution of the lateral workspace of the robot without need to calculate for the whole 3D workspace. On the other hand, using this methodology, the equations of boundary surfaces, curves, and points are all given in a closed-form solution such that one does not need to follow a numerical algorithm to find them. One can plug the structural parameters into these equations to find the workspace boundary. In other words, the equation of every single point, curve, and surface for the boundary is given. Also, in the case that an optimization of the 3D workspace needs to be done, once the lateral workspace of the robot is optimized, the 3D workspace will be optimized accordingly. For example, as it will be shown, if the volume of the 3D workspace needs to be maximized, it can be achieved by maximizing the lateral workspace of the robot. Therefore, the optimization process becomes concise using the presented methodology. All of these claims are shown in this section by some examples. The presented methodology is general and can be used for any axially symmetric $n$-legged robots, where $n$ is even, with non-symmetrical and non-identical kinematic chains. Without losing the generality, for demonstration purposes, the methodology is used to solve for the COW of axially symmetric hexapod robots since they are widely used in practice due to their simplicity and innate static balance [31,32].

4.3.1 Closed-form Solution for the Workspace Boundary

The following methodology can be adapted to find the spatial workspace of the axially symmetric hexapod robot. The methodology is general enough to be
employed for Stewart Platform (SP)-type robots as well as any \( n \)-legged parallel robot where \( n \) is even. The proposed methodology is as follows:

First, the problem domain is modeled similar to the hexapod robot as a SP. The only difference is the way in which the leg lengths change. With SPs, the lengths of the legs are changed as prismatic joints in order to change the position and orientation of the platform. With HWRs, the revolute joints of the articulated legs are rotated in order to achieve the desired position and orientation of the platform. Since the knee and ankle joints of the hexapod articulated leg form a planar mechanism and the hip joint is assumed to remain radially aligned, each articulated leg can be theoretically replaced with a virtual prismatic joint with an extendable leg length as in SPs. This theoretical replacement is shown in Figure 4.16. Therefore, if each entire articulated leg in a HWR is replaced with a prismatic joint, the HWR will become similar to a normal SP as shown in Figure 4.16. In this case, the HWR needs an extra inverse kinematic analysis whose complete solution was presented in section 3.2 and can also be found in [122].

Second, knowing that each articulated leg can be replaced with a prismatic joint, the HWR can be modeled as combination of three leg pairs, each forming 2-\( RPR \)
mechanisms, since opposite legs are aligned in the home position. This is shown in Figure 4.16. Therefore, each pair of opposite legs creates a 2-\textit{RPR} mechanism. Each mechanism has its own workspace in 2D (from its front view) and thereby in 3D space. Then, the total workspace of the HWR is a result of patching all three workspaces of the leg pairs (2-\textit{RPR} mechanisms). Hence, the workspace of axially symmetric hexapod robots can be calculated by the following procedures:

1. Break the hexapod down into three 2-\textit{RPR} mechanisms as shown in Figure 4.16.

2. Calculate 2D COW of each 2-\textit{RPR} mechanism.

3. Calculate 3D COW of a 2-\textit{RPR} mechanism by rotating the 2D one around the line connecting the two foot contact points.

4. Sum COW of all three 3D 2-\textit{RPR} mechanisms.

5. Find patches (common space) COW of all three 3D 2-\textit{RPR} mechanisms analytically.

The resultant workspace from the last step will be the COW of the initial axially symmetric hexapod robot. The methodology procedure is general and can be used for any other types of workspace such as RW and for any axially symmetric \textit{n}-pod robot where \textit{n} is even. In this work, the mathematical solution of the methodology procedure is presented in a closed-form manner.

\textbf{2D COW of 2-\textit{RPR} Planar Parallel Mechanism}

As shown in Figure 4.17, a general 2-\textit{RPR} planar parallel mechanism is constructed by two prismatic links that are grounded at one end and connected to a moving platform (triangle efb) at the other end. The COW of such a mechanism is
then the whole area which can be reached by a reference point on the platform while $a$, $c$, and $\theta_2$ are free to change and $\theta_3$, as the orientation of the moving platform, is kept constant [98]. From geometrical point of view in 2D space, this area is a surface bounded by four circular curves called boundary curves; two of them are related to the minimum legs extension and two others are related to the maximum legs extension. As shown in Figure 4.17, the solid area is a general shape of COW of the mechanism with an arbitrary given orientation of the platform, $\theta_3$. The boundary curves are as follows: the LLC (Left Lower Curve) and RLC (Right Lower Curve) are the circular curves created by the minimum left and right leg extensions, respectively; and the RUC (Right Upper Curve) and LUC (Left Upper Curve) are created by the maximum left and right leg extensions, respectively. While the methodology and the solution do not lose generality, due to using the same minimum and maximum extension on the link lengths $a$ and $c$, the 2D workspace will be symmetric about y axis as shown in Figure 4.18 and Figure 4.19. This reduces the analysis to half of the workspace. The solution can then be mirrored about the symmetry axis,
Figure 4.18: The COW is constrained by multiple circles.

...to obtain the full 2D workspace. Assuming $\theta_1 = 0$ and no offset for the center of the platform ($e = b/2$ and $\beta = 0$) as shown in Figure 4.19, the centers of LLC and RUC will be located at $\langle k, -l \rangle$ or $\langle D/2, 0 \rangle$ and the centers of RLC and LUC are located at $\langle -k, l \rangle$ or $\langle -D/2, 0 \rangle$ in $X-Y$ or $x-y$ coordinate systems, respectively where $k = \frac{1}{2}(d - b \cos(\theta_3))$ and $l = \frac{b}{2} \sin(\theta_3)$. If there is any amount of offset, it can be always added to the final solution. The distance between two circle centers is $D = 2\sqrt{k^2 + l^2}$. Before presenting the analytical solution, the following conditions should be always satisfied for having a mechanism with continuous workspace:

- $D < 2\sqrt{l_{max}^2 - \left(\frac{Dk}{2k}\right)^2}$: because otherwise the workspace will be null.
- $D > l_{max} - l_{min}$: because otherwise RLC and LLC will be contained within RUC and LUC, respectively, so there would not be any points like $p_r$ and $p_l$.
- $D < l_{max} + l_{min}$: because otherwise RUC and LUC will be separate from RLC and LLC, respectively.
Figure 4.19: The 2-RPR mechanism with symmetric COW.

Assuming that these three conditions are satisfied, the equations for all boundary curves can be given as follows in $X-Y$ coordinate system:

\[ RLC : (X - k)^2 + (Y + l)^2 = l_{min}^2 \]  \hspace{1cm} (4.19a)

\[ RUC : (X + k)^2 + (Y - l)^2 = l_{max}^2 \]  \hspace{1cm} (4.19b)

\[ LLC : (X + k)^2 + (Y - l)^2 = l_{min}^2 \]  \hspace{1cm} (4.19c)

\[ LUC : (X - k)^2 + (Y + l)^2 = l_{max}^2 \]  \hspace{1cm} (4.19d)

Then, the problem of finding 2D workspace of the mechanism can be divided into two sub-problems as follows:

1. \[ D \leq 2\sqrt{l_{min}^2 - \left(\frac{Dl}{2k}\right)^2} \]

2. \[ D > 2\sqrt{l_{min}^2 - \left(\frac{Dl}{2k}\right)^2} \]
I. $D \leq 2\sqrt{l_{min}^2 - \left(\frac{D}{2k}\right)^2}$:

This condition means RLC and LLC has an intersection: In this case, calculating $p_o$ in $x - y$ coordinate system (Solving RLC=LLC) will be as follows:

\[(X - k)^2 + (Y + l)^2 - l_{min}^2 = (X + k)^2 + (Y - l)^2 - l_{min}^2 \rightarrow X = \frac{l}{k} Y \quad (4.20)\]

Substitute $X = \frac{l}{k} Y$ into RLC for solving $Y = p_{oY}$ gives

\[p_{oY} = \pm \frac{2k\sqrt{l_{min}^2 - D^2/4}}{D} \quad (4.21)\]

However, $Y$ cannot be negative because of the physics of the robot. Therefore, $p_{oY} = \frac{2k\sqrt{l_{min}^2 - D^2/4}}{D}$. From Equation 4.20, $p_{ox}$ will be $p_{ox} = \frac{2l\sqrt{l_{min}^2 - D^2/4}}{D}$. Therefore, point $p_o$ will be as follows in the $X - Y$ coordinate system:

\[p_o = \left\langle \frac{2l\sqrt{l_{min}^2 - D^2/4}}{D}, \frac{2k\sqrt{l_{min}^2 - D^2/4}}{D} \right\rangle \quad (4.22)\]

The same equations can be done for point $p_t$ by intersecting RUC and LUC. With the same strategy, calculating $p_r$ and $p_l$ can be done in the $X - Y$ coordinate system by solving RLC=RUC and LLC=LUC, respectively. The following equations are the solution for all of the points.

\[p_o = \left\langle \frac{2l\sqrt{l_{min}^2 - D^2/4}}{D}, \frac{2k\sqrt{l_{min}^2 - D^2/4}}{D} \right\rangle \quad (4.23a)\]

\[p_t = \left\langle \frac{2l\sqrt{l_{max}^2 - D^2/4}}{D}, \frac{2k\sqrt{l_{max}^2 - D^2/4}}{D} \right\rangle \quad (4.23b)\]
\[ p_r = \frac{k(l_{max}^2 - l_{min}^2)}{D^2} + \frac{2l}{D} \sqrt{\frac{l_{min}^2}{2} - \left( \frac{l_{max}^2 - l_{min}^2 - D^2}{2D} \right)^2}, \quad (4.23c) \]

\[ p_l = \frac{k(l_{min}^2 - l_{max}^2)}{D^2} + \frac{2l}{D} \sqrt{\frac{l_{max}^2}{2} - \left( \frac{l_{max}^2 - l_{min}^2 - D^2}{2D} \right)^2}, \quad (4.23d) \]

II. \( D > 2 \sqrt{l_{min}^2 - \left( \frac{Dl}{2k} \right)^2} \):

In this case, all the points will have the same equations, but \( p_o \) will disappear because RLC and LLC will not have any intersection. Therefore, the bottom of the workspace will be created by a line parallel to \( X \) axis with equation of \( Y = \frac{b}{2} \sin(\theta_3) \) bounded by a left and a right point. This case can be analyzed in two different conditions as follows.

A. \( l_{max} \geq \sqrt{(2k - l_{min})^2 + b^2 \sin^2(\theta_3)} \) and \( l_{min} \leq \sqrt{(2k - l_{max})^2 + b^2 \sin^2(\theta_3)} \):

This condition means that the left and right leg lengths do not limit the motion of the mechanism to the right and left side of the bottom of the workspace, respectively. In this case, if \( \theta_3 \geq 0 \), the points \( \langle l_{min} - k, \frac{b}{2} \sin(\theta_3) \rangle \) and \( \langle l_{max} - k, \frac{b}{2} \sin(\theta_3) \rangle \) will be the left and the right points of the bottom of the workspace, respectively. If \( \theta_3 < 0 \), the bounding points will be mirrored about \( Y \) axis such that the points \( \langle k - l_{max}, \frac{b}{2} \sin(\theta_3) \rangle \) and \( \langle k - l_{min}, \frac{b}{2} \sin(\theta_3) \rangle \) will create the left and right points of the bottom of the workspace, respectively.
B. \( l_{\text{max}} < \sqrt{(2k - l_{\text{min}})^2 + b^2 \sin^2(\theta_3)} \) and \( l_{\text{min}} > \sqrt{(2k - l_{\text{max}})^2 + b^2 \sin^2(\theta_3)} \):

This condition means that the left and right leg lengths limit the motion of the mechanism to the right and left side of the bottom of the workspace, respectively. In this case, if \( \theta_3 \geq 0 \), then the left point will be \( \langle k - \sqrt{l_{\text{max}}^2 - b^2 \sin^2(\theta_3)}, \frac{b}{2} \sin(\theta_3) \rangle \) and the right point will be \( \langle l_{\text{max}} - k, \frac{b}{2} \sin(\theta_3) \rangle \). However, if \( \theta_3 < 0 \), then the left point will be \( \langle k - l_{\text{max}}, \frac{b}{2} \sin(\theta_3) \rangle \) and the right point will be \( \langle -k + \sqrt{l_{\text{max}}^2 - b^2 \sin^2(\theta_3)}, \frac{b}{2} \sin(\theta_3) \rangle \).

3D COW of 2-RPR Parallel Mechanism

A 3D 2-RPR mechanism is the 2D one shown in Figure 4.19 with the ability of rotating around the line connecting two ground contact points as the rotation axis (X axis). For 3D analysis, it is assumed that the \( X - Y \) plane is attached to the ground and Z axis is then perpendicular to it. Therefore, the amount of rotation of 2D workspace about X, to get 3D one, is between 0 and \( \pi \). Consider a 2D workspace of the 2-RPR mechanism as shown in Figure 4.20-a. Then, Figure 4.20-b and Figure 4.20-c illustrate the 3D workspace of the 2-RPR mechanism in top and isometric view, respectively. Suppose the closed-form solution of the 2D one is given according to the solution presented in the previous section. Each curve can be written in the form \( Y^2 = g(X) \). Since the 2D workspace is rotating about X axis, each point of the 2D workspace will then follow a circle according to the following equation:

\[
Y^2 + Z^2 = g(X), Z \geq 0
\]

(4.24)

Equation 4.24 represents the general closed-form solution for the 3D workspace of the 2-RPR mechanism. Since \( Y^2 = g(X) \) is a circular curve, Equation 4.24 represents a sphere.
Consider the right half of the boundary of the 2D workspace of 2-RPR planar parallel mechanism which is constructed by RLC and RUC connected to each other by point \( p_r \). As a result, the boundary of 3D one will be created by two surfaces: \( RLS_1 \) (Right Lower Surface) and \( RUS_1 \) (Right Upper Surface), which are connected by a curve \( C_{p_r} \). Curve \( C_{p_r} \) is created as a result of rotating point \( p_r \) and is part of a circle. To illustrate this more clearly, consider Figure 4.21 which shows a general axially symmetric hexapod robot from its top view. As shown, each two opposite legs, which create a 2-RPR mechanism, are connected using the rotation axis. Points \( c_i \) and \( c'_i \) represents the centers of spheres LUS/RLS and RUS/LLS, respectively. Centers of these spheres can be calculated as follows.

\[
\begin{align*}
    c_i &= (k_i \cos(\varphi_i), k_i \sin(\varphi_i), -l_i) \\
    c'_i &= (k'_i \cos(\varphi_i), k'_i \sin(\varphi_i), l_i)
\end{align*}
\]

(4.25)

where \( l_i = \frac{b}{2} \sin(\theta_{3i}) \) and \( k_i = m_i - \frac{b}{2} \cos(\theta_{3i}) \) and \( k'_i = -m'_i + \frac{b}{2} \cos(\theta_{3i}) \), \( \varphi_1 = 0 \), and \( \theta_{3i} \) is \( \theta_3 \) regarding to \( i \)th 2-RPR mechanism. Therefore, the 3D workspace of each
2-RPR mechanism will be surrounded by the following four surfaces:

\[
RLS_i : (X - c_{ix})^2 + (Y - c_{iy})^2 + (Z - c_{iz})^2 = l_{min}^2
\]  \hspace{1cm} (4.26a)

\[
LUS_i : (X - c_{ix})^2 + (Y - c_{iy})^2 + (Z - c'_{iz})^2 = l_{max}^2
\]  \hspace{1cm} (4.26b)

\[
LLS_i : (X - c'_{ix})^2 + (Y - c'_{iy})^2 + (Z - c_{iz})^2 = l_{min}^2
\]  \hspace{1cm} (4.26c)

\[
RUS_i : (X - c'_{ix})^2 + (Y - c'_{iy})^2 + (Z - c'_{iz})^2 = l_{max}^2
\]  \hspace{1cm} (4.26d)

where \( i = 1, 2, 3 \).

Considering the same minimum and maximum leg extensions for all six legs of the hexapod will cause the same \( k_i = k'_i \) for all three mechanisms. This is a common case for hexapod robots. Only for illustration purposes and without losing the generality
of the solution, assumes $m_i$ is the same for all three mechanisms and is the same as $m'_i$ as well ($m_i = m'_i = \frac{d}{2}$). Also assume that $\theta_{3i} = 0$. Therefore, the 2D workspace of all three mechanisms will be the same as shown in Figure 4.22. However, the solution presented in this section is still general and does not consider the assumptions which are considered for illustration purposes. All three 3D workspaces, workspaces number 1, 2, and 3, and their own rotation axis ($X_i$) are shown in Figure 4.22. Let’s call them as originator workspaces since the final workspace is originated by rotation of them about their rotation axis and finally patches of all three 3D workspaces of 2-$RPR$ mechanisms will give the final 3D workspace of the hexapod robot. The $Y_i$ axes are not shown for ambiguity avoidance.
After each originator workspace shown in Figure 4.22 rotates about its own rotation axis ($X_i$), all three created 3D surfaces, which one of them is shown in Figure 4.20, will intersect. The summation of the workspace of all three 3D 2-RPR mechanisms is shown in Figure 4.23-a. However, this is not the final workspace because in reality each workspace is constrained by two other mechanisms. Thus, the patches of all three 3D 2-RPR mechanisms will be the final result of the workspace of the robot which is shown in Figure 4.23-b. In other words, the final 3D workspace of the hexapod robot is the space that all three 3D workspaces encompass simultaneously i.e. belongs to all three workspaces at the same time. Figures 4.23-c and 4.23-d illustrate Figure 4.23-a (before patch) and Figure 4.23-b (after patch) at the same time from the top and front view, respectively, in the same scale. Let’s divide the final workspace shown in Figure 4.23-b into Upper and Lower sub-workspaces. Then, from the top view (Figure 4.23-c and Figure 4.24-b) they are exactly at top of each other and each is created by six identical surfaces ($S_i$) constrained between six identical curves ($C_i$) and bounded by six other identical curves ($B_i$) separated by six points ($p_i$). In other words, from top view, the workspace boundary of an axially symmetric HWR is created by six identical areas for upper and six for lower sub-workspace. Therefore for surfaces $S_i$ and curves $C_i$, two solutions exist; one for upper sub-workspace and one for lower sub-workspace which are exactly at top of each other in top view. Thus, define $FUS_i$ and $FUC_i$ as the Final Upper Surfaces and Curves as well as $FLS_i$ and $FLC_i$ representing the Final Lower Surfaces and Curves, respectively. Then, the final 3D workspace of the hexapod robot can be represented as $\bigcup_{i=1}^{6} [W_i(FUS_i, FLS_i, FUC_i, FLC_i, B_i, p_i)]$. Once the solution for $i = 1$ is found, the whole workspace can be solved by multiplying the equations in a rotation matrix to get for $i = 2, \ldots, 6$. Therefore, by finding $FUC_1, FLC_1, FUS_1, FLS_1, B_1$,
Figure 4.23: The summation of the 3D 2-RPR mechanisms before and after Boolean operation.

Figure 4.24: The symmetric workspace of hexapod robot.

and \( p_1 \), all other sections of the final workspace can be calculated. It should be emphasized that the presented solution is still general to solve for non-symmetric workspace. In the case of having a non-symmetric workspace, each \( W_i \) has to be calculated separately while in the case of symmetric workspace, one is calculated and is rotated to get the other ones.

**Calculating \( C_i \) (\( FUC_i \) and \( FLC_i \)):**

The planar curves \( FUC_1 \) and \( FLC_1 \) are the same as curve \( C_1 \) from top view shown in Figure 4.24-b. \( FUC_1 \) is created as a result of intersecting \( LUS_2 \) and \( RUS_3 \) which are created as a result of rotation of \( LUC_2 \) and \( RUC_3 \) about \( X_2 \) and \( X_3 \), respectively, as shown in Figure 4.22. The intersection of \( LUS_2 \) and \( RUS_3 \) can be found by solving Equations 4.26b and 4.26d simultaneously. For \( FLC_1 \), the
intersecting of $LLS_2$ and $RLS_3$ should be solved. The similar methodology works for any $C_i$.

**Calculating $B_i$:**

The envelope boundary of the workspace from top view in Figure 4.24-b is shown by solid lines creating a hexagon. The sides of this hexagon, $B_i$, are parts of the dashed lines shown in Figure 4.24-a which are the boundaries of the 3D 2-$RPR$ workspaces from the top view. $B_1$ and $B_4$ are created as a result of intersection of $RUS_3$-$RLS_3$ and $LUS_3$-$LLS_3$, respectively. With the same strategy, $B_2$ and $B_5$ are created as a result of intersection of $RUS_1$-$RLS_1$ and $LUS_1$-$LLS_1$, respectively. Also, $B_3$ and $B_6$ are created as a result of intersection of $RUS_2$-$RLS_2$ and $LUS_3$-$LLS_3$, respectively.

**Calculating $S_i$ ($FUS_i$ and $FLS_i$):**

The hatched surface $S_1$ shown in Figure 4.24-a is representing both $RUS_3$ and $RLS_3$. Therefore, $FUS_1$ is part of $RUS_3$ and $FLS_1$ is part of $RLS_3$.

**Calculating $p_i$ ($i = 0, \ldots, 6$) and $p_t$:**

For $i = 1, \ldots, 6$, once $B_i$ is calculated, $p_i$ will be as a result of intersection of $B_i$ and $B_{i-1}$. When $i = 1$, $B_{i-1} = B_6$. Therefore, the equation $B_i = B_{i-1}$ will give $p_i(i = 1, \ldots, 6)$. For $p_0$ and $p_t$, Equation 4.23 can be used.

**Summary of the Solution**

The following tables show the exact closed-form solution for the boundary workspace of an axially symmetric hexapod robot in the general case. For $i = 0$ and $i = t$, $p_i$ is defined the same as Equation 4.23 but with $\theta_3 = \max\{\theta_{3i}, i = 1, 2, 3\}$. Next section will show how these equations can be derived for an axially symmetric hexapod robot.
Table 4.5: Equations of the final workspace solution for final upper surfaces.

<table>
<thead>
<tr>
<th>(i)</th>
<th>(FUS_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(RUS_3 : (X - c'<em>{3X})^2 + (Y - c'</em>{3Y})^2 + (Z - c'<em>{3Z})^2 = l</em>{max}^2)</td>
</tr>
<tr>
<td>2</td>
<td>(RUS_1 : (X - c'<em>{1X})^2 + (Y - c'</em>{1Y})^2 + (Z - c'<em>{1Z})^2 = l</em>{max}^2)</td>
</tr>
<tr>
<td>3</td>
<td>(RUS_2 : (X - c'<em>{2X})^2 + (Y - c'</em>{2Y})^2 + (Z - c'<em>{2Z})^2 = l</em>{max}^2)</td>
</tr>
<tr>
<td>4</td>
<td>(LUS_3 : (X - c_{3X})^2 + (Y - c_{3Y})^2 + (Z - c_{3Z})^2 = l_{min}^2)</td>
</tr>
<tr>
<td>5</td>
<td>(LUS_1 : (X - c_{1X})^2 + (Y - c_{1Y})^2 + (Z - c_{1Z})^2 = l_{min}^2)</td>
</tr>
<tr>
<td>6</td>
<td>(LUS_2 : (X - c_{2X})^2 + (Y - c_{2Y})^2 + (Z - c_{2Z})^2 = l_{min}^2)</td>
</tr>
</tbody>
</table>

Table 4.6: Equations of the final workspace solution for final lower surfaces.

<table>
<thead>
<tr>
<th>(i)</th>
<th>(FLS_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(RLS_3 : (X - c_{3X})^2 + (Y - c_{3Y})^2 + (Z - c_{3Z})^2 = l_{min}^2)</td>
</tr>
<tr>
<td>2</td>
<td>(RLS_1 : (X - c_{1X})^2 + (Y - c_{1Y})^2 + (Z - c_{1Z})^2 = l_{min}^2)</td>
</tr>
<tr>
<td>3</td>
<td>(RLS_2 : (X - c_{2X})^2 + (Y - c_{2Y})^2 + (Z - c_{2Z})^2 = l_{min}^2)</td>
</tr>
<tr>
<td>4</td>
<td>(LLS_3 : (X - c'<em>{3X})^2 + (Y - c'</em>{3Y})^2 + (Z - c'<em>{3Z})^2 = l</em>{min}^2)</td>
</tr>
<tr>
<td>5</td>
<td>(LLS_1 : (X - c'<em>{1X})^2 + (Y - c'</em>{1Y})^2 + (Z - c'<em>{1Z})^2 = l</em>{min}^2)</td>
</tr>
<tr>
<td>6</td>
<td>(LLS_2 : (X - c'<em>{2X})^2 + (Y - c'</em>{2Y})^2 + (Z - c'<em>{2Z})^2 = l</em>{min}^2)</td>
</tr>
</tbody>
</table>

Table 4.7: Equations of the final workspace solution for final upper curves.

<table>
<thead>
<tr>
<th>(i)</th>
<th>(FUC_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((X - c'<em>{3X})^2 + (Y - c'</em>{3Y})^2 + (Z - c'<em>{3Z})^2 - (X - c</em>{2X})^2 - (Y - c_{2Y})^2 - (Z - c_{2Z})^2 = 0)</td>
</tr>
<tr>
<td>2</td>
<td>((X - c'<em>{3X})^2 + (Y - c'</em>{3Y})^2 + (Z - c'<em>{3Z})^2 - (X - c'</em>{1X})^2 - (Y - c'<em>{1Y})^2 - (Z - c'</em>{1Z})^2 = 0)</td>
</tr>
<tr>
<td>3</td>
<td>((X - c'<em>{2X})^2 + (Y - c'</em>{2Y})^2 + (Z - c'<em>{2Z})^2 - (X - c'</em>{1X})^2 - (Y - c'<em>{1Y})^2 - (Z - c'</em>{1Z})^2 = 0)</td>
</tr>
<tr>
<td>4</td>
<td>((X - c'<em>{2X})^2 + (Y - c'</em>{2Y})^2 + (Z - c'<em>{2Z})^2 - (X - c</em>{3X})^2 - (Y - c_{3Y})^2 - (Z - c_{3Z})^2 = 0)</td>
</tr>
<tr>
<td>5</td>
<td>((X - c_{3X})^2 + (Y - c_{3Y})^2 + (Z - c_{3Z})^2 - (X - c_{1X})^2 - (Y - c_{1Y})^2 - (Z - c_{1Z})^2 = 0)</td>
</tr>
<tr>
<td>6</td>
<td>((X - c_{1X})^2 + (Y - c_{1Y})^2 + (Z - c_{1Z})^2 - (X - c_{2X})^2 - (Y - c_{2Y})^2 - (Z - c_{2Z})^2 = 0)</td>
</tr>
</tbody>
</table>
Table 4.8: Equations of the final workspace solution for final lower curves.

<table>
<thead>
<tr>
<th>( i )</th>
<th>( FLC_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((X - c_{3X})^2 + (Y - c_{3Y})^2 + (Z - c_{3Z})^2 - (X - c'<em>{2X})^2 - (Y - c'</em>{2Y})^2 - (Z - c'_{2Z})^2 = 0)</td>
</tr>
<tr>
<td>2</td>
<td>((X - c_{3X})^2 + (Y - c_{3Y})^2 + (Z - c_{3Z})^2 - (X - c_{1X})^2 - (Y - c_{1Y})^2 - (Z - c_{1Z})^2 = 0)</td>
</tr>
<tr>
<td>3</td>
<td>((X - c_{2X})^2 + (Y - c_{2Y})^2 + (Z - c_{2Z})^2 - (X - c_{1X})^2 - (Y - c_{1Y})^2 - (Z - c_{1Z})^2 = 0)</td>
</tr>
<tr>
<td>4</td>
<td>((X - c_{2X})^2 + (Y - c_{2Y})^2 + (Z - c_{2Z})^2 - (X - c'<em>{1X})^2 - (Y - c'</em>{1Y})^2 - (Z - c'_{1Z})^2 = 0)</td>
</tr>
<tr>
<td>5</td>
<td>((X - c'<em>{3X})^2 + (Y - c'</em>{3Y})^2 + (Z - c'<em>{3Z})^2 - (X - c'</em>{1X})^2 - (Y - c'<em>{1Y})^2 - (Z - c'</em>{1Z})^2 = 0)</td>
</tr>
<tr>
<td>6</td>
<td>((X - c'<em>{1X})^2 + (Y - c'</em>{1Y})^2 + (Z - c'<em>{1Z})^2 - (X - c'</em>{2X})^2 - (Y - c'<em>{2Y})^2 - (Z - c'</em>{2Z})^2 = 0)</td>
</tr>
</tbody>
</table>

Table 4.9: Equations of the final workspace solution for boundary side curves.

<table>
<thead>
<tr>
<th>( i )</th>
<th>( B_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((X - c'<em>{3X})^2 + (Y - c'</em>{3Y})^2 + (Z - c'<em>{3Z})^2 - (X - c</em>{3X})^2 - (Y - c_{3Y})^2 - (Z - c_{3Z})^2 + l_{min}^2 - l_{max}^2 = 0)</td>
</tr>
<tr>
<td>2</td>
<td>((X - c'<em>{1X})^2 + (Y - c'</em>{1Y})^2 + (Z - c'<em>{1Z})^2 - (X - c</em>{1X})^2 - (Y - c_{1Y})^2 - (Z - c_{1Z})^2 + l_{min}^2 - l_{max}^2 = 0)</td>
</tr>
<tr>
<td>3</td>
<td>((X - c'<em>{2X})^2 + (Y - c'</em>{2Y})^2 + (Z - c'<em>{2Z})^2 - (X - c</em>{2X})^2 - (Y - c_{2Y})^2 - (Z - c_{2Z})^2 + l_{min}^2 - l_{max}^2 = 0)</td>
</tr>
<tr>
<td>4</td>
<td>((X - c'<em>{3X})^2 + (Y - c'</em>{3Y})^2 + (Z - c_{3Z})^2 - (X - c'<em>{3X})^2 - (Y - c'</em>{3Y})^2 - (Z - c'<em>{3Z})^2 + l</em>{min}^2 - l_{max}^2 = 0)</td>
</tr>
<tr>
<td>5</td>
<td>((X - c'<em>{1X})^2 + (Y - c</em>{1Y})^2 + (Z - c_{1Z})^2 - (X - c'<em>{1X})^2 - (Y - c</em>{1Y})^2 - (Z - c_{1Z})^2 + l_{min}^2 - l_{max}^2 = 0)</td>
</tr>
<tr>
<td>6</td>
<td>((X - c'<em>{2X})^2 + (Y - c</em>{2Y})^2 + (Z - c_{2Z})^2 - (X - c'<em>{2X})^2 - (Y - c</em>{2Y})^2 - (Z - c_{2Z})^2 + l_{min}^2 - l_{max}^2 = 0)</td>
</tr>
</tbody>
</table>

Table 4.10: Equations of the final workspace solution for final boundary points.

<table>
<thead>
<tr>
<th>( i )</th>
<th>( p_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( B_1 - B_6 = 0 )</td>
</tr>
<tr>
<td>2</td>
<td>( B_2 - B_1 = 0 )</td>
</tr>
<tr>
<td>3</td>
<td>( B_3 - B_2 = 0 )</td>
</tr>
<tr>
<td>4</td>
<td>( B_4 - B_3 = 0 )</td>
</tr>
<tr>
<td>5</td>
<td>( B_5 - B_4 = 0 )</td>
</tr>
<tr>
<td>6</td>
<td>( B_6 - B_5 = 0 )</td>
</tr>
</tbody>
</table>
Effect of Symmetry

The equations provided in Table 4.5 through Table 4.10 are related to the general case of a hexapod robot, where the leg contacts with the ground are free to be an arbitrary distance from CG of the robot, which causes a non-symmetrical shape of foot contact distribution and the workspace. A common and usual axially symmetric hexapod robot has a symmetric configuration where $\varphi_1 = 0$, $\varphi_2 = \varphi_3 = \varphi$, and $m_i = m'_i = d/2$ for any number of $i$ (Figure 4.21). To use the equations presented in Table 4.5 through Table 4.10 for having a symmetric 3D workspace about $Z$ axis, a horizontal platform is chosen in this section where $\theta_{3i} = 0$ giving $k_i = -k'_i = k = \frac{d-b}{2}$, and $l_i = 0$. However, the generality of the solution will not be lost. The effect of any arbitrary orientation for the platform can be applied by calculating $\theta_{3i}$ based on the normal vector of the platform and thereby calculating $k_i$ and $l_i$ and substituting into the equations presented in Table 4.5 through Table 4.10. By choosing a horizontal platform and using Table 4.5 through Table 4.10, the workspace of the axially symmetric hexapod robot can be determined using the equations provided in Table 4.11 through Table 4.17. These equations are very useful for deriving both the workspace of the robot and optimization process. Next sections will show how one can use these equations to calculate the workspace of the robot and optimize the workspace and/or the robot parameters based on the desired workspace.

4.3.2 Optimization and Design

As an example to solve for the workspace of an axially symmetric hexapod robot, let $d = 230 \text{ mm}$, the diameter of the platform be $50 \text{ mm}$, and the replacement prismatic leg has $l_{\text{max}} = 173 \text{ mm}$ and $l_{\text{min}} = 90 \text{ mm}$. Therefore, $k = 90 \text{ mm}$ and $\varphi_1 = 0$
Table 4.11: Equations of the final workspace solution for final upper surfaces.

<table>
<thead>
<tr>
<th>( i )</th>
<th>( FUS_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((X + k \cos(\varphi))^2 + (Y + k \sin(\varphi))^2 + Z^2 = l_{\text{max}}^2)</td>
</tr>
<tr>
<td>2</td>
<td>((X + k)^2 + Y^2 + Z^2 = l_{\text{max}}^2)</td>
</tr>
<tr>
<td>3</td>
<td>((X + k \cos(\varphi))^2 + (Y + k \sin(\varphi))^2 + Z^2 = l_{\text{max}}^2)</td>
</tr>
<tr>
<td>4</td>
<td>((X - k \cos(\varphi))^2 + (Y - k \sin(\varphi))^2 + Z^2 = l_{\text{max}}^2)</td>
</tr>
<tr>
<td>5</td>
<td>((X - k)^2 + Y^2 + Z^2 = l_{\text{max}}^2)</td>
</tr>
<tr>
<td>6</td>
<td>((X - k \cos(\varphi))^2 + (Y + k \sin(\varphi))^2 + Z^2 = l_{\text{max}}^2)</td>
</tr>
</tbody>
</table>

Table 4.12: Equations of the final workspace solution for final lower surfaces.

<table>
<thead>
<tr>
<th>( i )</th>
<th>( FLS_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((X - k \cos(\varphi))^2 + (Y - k \sin(\varphi))^2 + Z^2 = l_{\text{min}}^2)</td>
</tr>
<tr>
<td>2</td>
<td>((X - k)^2 + Y^2 + Z^2 = l_{\text{min}}^2)</td>
</tr>
<tr>
<td>3</td>
<td>((X - k \cos(\varphi))^2 + (Y + k \sin(\varphi))^2 + Z^2 = l_{\text{min}}^2)</td>
</tr>
<tr>
<td>4</td>
<td>((X + k \cos(\varphi))^2 + (Y + k \sin(\varphi))^2 + Z^2 = l_{\text{min}}^2)</td>
</tr>
<tr>
<td>5</td>
<td>((X + k)^2 + Y^2 + Z^2 = l_{\text{min}}^2)</td>
</tr>
<tr>
<td>6</td>
<td>((X + k \cos(\varphi))^2 + (Y - k \sin(\varphi))^2 + Z^2 = l_{\text{min}}^2)</td>
</tr>
</tbody>
</table>

Table 4.13: Equations of the final workspace solution for final upper curves.

<table>
<thead>
<tr>
<th>( i )</th>
<th>( FUC_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((Y + k \sin(\varphi))^2 + Z^2 = l_{\text{max}}^2 - (k \cos(\varphi))^2, \ X = 0)</td>
</tr>
<tr>
<td>2</td>
<td>((X + k)^2 + Y^2 + Z^2 = l_{\text{max}}^2, \ Y = \frac{1 - \cos(\varphi)}{\sin(\varphi)} X)</td>
</tr>
<tr>
<td>3</td>
<td>((X + k)^2 + Y^2 + Z^2 = l_{\text{max}}^2, \ Y = \frac{\cos(\varphi) - 1}{\sin(\varphi)} X)</td>
</tr>
<tr>
<td>4</td>
<td>((Y - k \sin(\varphi))^2 + Z^2 = l_{\text{max}}^2 - (k \cos(\varphi))^2, \ X = 0)</td>
</tr>
<tr>
<td>5</td>
<td>((X - k)^2 + Y^2 + Z^2 = l_{\text{max}}^2, \ Y = \frac{1 - \cos(\varphi)}{\sin(\varphi)} X)</td>
</tr>
<tr>
<td>6</td>
<td>((X - k)^2 + Y^2 + Z^2 = l_{\text{max}}^2, \ Y = \frac{1 - \cos(\varphi)}{\sin(\varphi)} X)</td>
</tr>
</tbody>
</table>

Table 4.14: Equations of the final workspace solution for final lower curves.

<table>
<thead>
<tr>
<th>( i )</th>
<th>( FL_{\text{C}}_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((Y - k \sin(\varphi))^2 + Z^2 = l_{\text{min}}^2 - (k \cos(\varphi))^2, \ X = 0)</td>
</tr>
<tr>
<td>2</td>
<td>((X - k)^2 + Y^2 + Z^2 = l_{\text{min}}^2, \ Y = \frac{1 - \cos(\varphi)}{\sin(\varphi)} X)</td>
</tr>
<tr>
<td>3</td>
<td>((X - k)^2 + Y^2 + Z^2 = l_{\text{min}}^2, \ Y = \frac{\cos(\varphi) - 1}{\sin(\varphi)} X)</td>
</tr>
<tr>
<td>4</td>
<td>((Y + k \sin(\varphi))^2 + Z^2 = l_{\text{min}}^2 - (k \cos(\varphi))^2, \ X = 0)</td>
</tr>
<tr>
<td>5</td>
<td>((X + k)^2 + Y^2 + Z^2 = l_{\text{min}}^2, \ Y = \frac{1 - \cos(\varphi)}{\sin(\varphi)} X)</td>
</tr>
<tr>
<td>6</td>
<td>((X + k)^2 + Y^2 + Z^2 = l_{\text{min}}^2, \ Y = \frac{1 - \cos(\varphi)}{\sin(\varphi)} X)</td>
</tr>
</tbody>
</table>
Table 4.15: Equations of the final workspace solution for boundary side curves.

<table>
<thead>
<tr>
<th>(i)</th>
<th>(B_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( (X + k\cos(\varphi))^2 + (Y + k\sin(\varphi))^2 + Z^2 = l_{\text{max}}^2, Y = -X \cot(\varphi) + \frac{l_{\text{max}}^2 - l_{\text{min}}^2}{4k\sin(\varphi)} )</td>
</tr>
<tr>
<td>2</td>
<td>( Y^2 + Z^2 = l_{\text{max}}^2 - \left( \frac{l_{\text{max}}^2 - l_{\text{min}}^2}{4k} + k \right)^2, X = \frac{l_{\text{max}}^2 - l_{\text{min}}^2}{4k} )</td>
</tr>
<tr>
<td>3</td>
<td>( (X + k\cos(\varphi))^2 + (Y - k\sin(\varphi))^2 + Z^2 = l_{\text{max}}^2, Y = X \cot(\varphi) - \frac{l_{\text{max}}^2 - l_{\text{min}}^2}{4k\sin(\varphi)} )</td>
</tr>
<tr>
<td>4</td>
<td>( (X - k\cos(\varphi))^2 + (Y - k\sin(\varphi))^2 + Z^2 = l_{\text{max}}^2, Y = -X \cot(\varphi) - \frac{l_{\text{max}}^2 - l_{\text{min}}^2}{4k\sin(\varphi)} )</td>
</tr>
<tr>
<td>5</td>
<td>( Y^2 + Z^2 = l_{\text{max}}^2 - \left( \frac{l_{\text{max}}^2 - l_{\text{min}}^2}{4k} - k \right)^2, X = \frac{l_{\text{min}}^2 - l_{\text{max}}^2}{4k} )</td>
</tr>
<tr>
<td>6</td>
<td>( (X - k\cos(\varphi))^2 + (Y + k\sin(\varphi))^2 + Z^2 = l_{\text{max}}^2, Y = X \cot(\varphi) + \frac{l_{\text{max}}^2 - l_{\text{min}}^2}{4k\sin(\varphi)} )</td>
</tr>
</tbody>
</table>

Table 4.16: Equations of the final workspace solution for final boundary points in \(x\) and \(y\) axes.

<table>
<thead>
<tr>
<th>(i)</th>
<th>(p_{ix})</th>
<th>(p_{iy})</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>( \frac{l_{\text{max}}^2 - l_{\text{min}}^2}{4k\sin(\varphi)} )</td>
</tr>
<tr>
<td>2</td>
<td>( l_{\text{max}}^2 - l_{\text{min}}^2 )</td>
<td>( \frac{l_{\text{max}}^2 - l_{\text{min}}^2}{4k} ) (1 - (\cos(\varphi)))</td>
</tr>
<tr>
<td>3</td>
<td>( l_{\text{max}}^2 - l_{\text{min}}^2 )</td>
<td>( \frac{l_{\text{max}}^2 - l_{\text{min}}^2}{4k} ) (1 - (\cos(\varphi)))</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>( \frac{l_{\text{max}}^2 - l_{\text{min}}^2}{4k\sin(\varphi)} )</td>
</tr>
<tr>
<td>5</td>
<td>( l_{\text{min}}^2 - l_{\text{max}}^2 )</td>
<td>( \frac{l_{\text{min}}^2 - l_{\text{max}}^2}{4k\sin(\varphi)} ) (1 - (\cos(\varphi)))</td>
</tr>
<tr>
<td>6</td>
<td>( l_{\text{min}}^2 - l_{\text{max}}^2 )</td>
<td>( \frac{l_{\text{min}}^2 - l_{\text{max}}^2}{4k\sin(\varphi)} ) (1 - (\cos(\varphi)))</td>
</tr>
<tr>
<td>(t)</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 4.17: Equations of the final workspace solution for final boundary points in $z$ axis.

<table>
<thead>
<tr>
<th>$i$</th>
<th>$P_{iz}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$\sqrt{l_{\text{min}}^2 - k^2}$</td>
</tr>
<tr>
<td>1</td>
<td>$\sqrt{l_{\text{max}}^2 - k^2 - \frac{l_{\text{max}}^2 - l_{\text{min}}^2}{2} - \left(\frac{l_{\text{max}}^2 - l_{\text{min}}^2}{4ksin(\phi)}\right)^2}$</td>
</tr>
<tr>
<td>2</td>
<td>$\sqrt{l_{\text{max}}^2 - k^2 - \frac{l_{\text{max}}^2 - l_{\text{min}}^2}{2} - \frac{2(1 - cos(\phi))(l_{\text{max}}^2 - l_{\text{min}}^2)}{4ksin(\phi)}^2}$</td>
</tr>
<tr>
<td>3</td>
<td>$\sqrt{l_{\text{max}}^2 - k^2 - \frac{l_{\text{max}}^2 - l_{\text{min}}^2}{2} - \frac{2(1 - cos(\phi))(l_{\text{max}}^2 - l_{\text{min}}^2)}{4ksin(\phi)}^2}$</td>
</tr>
<tr>
<td>4</td>
<td>$\sqrt{l_{\text{max}}^2 - k^2 - \frac{l_{\text{max}}^2 - l_{\text{min}}^2}{2} - \left(\frac{l_{\text{max}}^2 - l_{\text{min}}^2}{4ksin(\phi)}\right)^2}$</td>
</tr>
<tr>
<td>5</td>
<td>$\sqrt{l_{\text{max}}^2 - k^2 - \frac{l_{\text{max}}^2 - l_{\text{min}}^2}{2} - \frac{2(1 - cos(\phi))(l_{\text{max}}^2 - l_{\text{min}}^2)}{4ksin(\phi)}^2}$</td>
</tr>
<tr>
<td>6</td>
<td>$\sqrt{l_{\text{max}}^2 - k^2}$</td>
</tr>
</tbody>
</table>

Table 4.18: Equations of the final workspace solution for final upper surfaces.

<table>
<thead>
<tr>
<th>$i$</th>
<th>$FUS_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$(X + 45)^2 + (Y + 78)^2 + Z^2 = 29929$</td>
</tr>
<tr>
<td>2</td>
<td>$(X + 90)^2 + Y^2 + Z^2 = 29929$</td>
</tr>
<tr>
<td>3</td>
<td>$(X + 45)^2 + (Y - 78)^2 + Z^2 = 29929$</td>
</tr>
<tr>
<td>4</td>
<td>$(X - 45)^2 + (Y - 78)^2 + Z^2 = 29929$</td>
</tr>
<tr>
<td>5</td>
<td>$(X - 90)^2 + Y^2 + Z^2 = 29929$</td>
</tr>
<tr>
<td>6</td>
<td>$(X - 45)^2 + (Y + 78)^2 + Z^2 = 29929$</td>
</tr>
</tbody>
</table>

and $\phi_2 = \phi_3 = 60^\circ$. The workspace boundary can be calculated by substituting values of $l_{\text{max}}, l_{\text{min}}, k$, and $\phi$ into Table 4.11 through Table 4.17. The results are shown in Table Table 4.18 through Table 4.23. These equations show the closed-form solution of the COW of the axially symmetric HWR with a horizontal platform.

**COW-based Design of HWR**

One advantage of having a closed-form solution for the workspace of the hexapod is to design a robot based on a given desired workspace in an optimum manner. With
Table 4.19: Equations of the final workspace solution for final lower surfaces.

<table>
<thead>
<tr>
<th>(i)</th>
<th>(F_{LS_i})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((X - 45)^2 + (Y - 78)^2 + Z^2 = 8100)</td>
</tr>
<tr>
<td>2</td>
<td>((X - 90)^2 + Y^2 + Z^2 = 8100)</td>
</tr>
<tr>
<td>3</td>
<td>((X - 45)^2 + (Y + 78)^2 + Z^2 = 8100)</td>
</tr>
<tr>
<td>4</td>
<td>((X + 45)^2 + (Y + 78)^2 + Z^2 = 8100)</td>
</tr>
<tr>
<td>5</td>
<td>((X + 90)^2 + Y^2 + Z^2 = 8100)</td>
</tr>
<tr>
<td>6</td>
<td>((X + 45)^2 + (Y - 78)^2 + Z^2 = 8100)</td>
</tr>
</tbody>
</table>

Table 4.20: Equations of the final workspace solution for final upper curves.

<table>
<thead>
<tr>
<th>(i)</th>
<th>(F_{UC_i})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((Y + 78)^2 + Z^2 = 27904, X = 0)</td>
</tr>
<tr>
<td>2</td>
<td>((X + 90)^2 + Y^2 + Z^2 = 29929, Y = 0.577X)</td>
</tr>
<tr>
<td>3</td>
<td>((X + 90)^2 + Y^2 + Z^2 = 29929, Y = -0.577X)</td>
</tr>
<tr>
<td>4</td>
<td>((Y - 78)^2 + Z^2 = 27904, X = 0)</td>
</tr>
<tr>
<td>5</td>
<td>((X - 90)^2 + Y^2 + Z^2 = 29929, Y = -0.577X)</td>
</tr>
<tr>
<td>6</td>
<td>((X - 90)^2 + Y^2 + Z^2 = 29929, Y = 0.577X)</td>
</tr>
</tbody>
</table>

Table 4.21: Equations of the final workspace solution for final lower curves.

<table>
<thead>
<tr>
<th>(i)</th>
<th>(F_{LC_i})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((Y - 78)^2 + Z^2 = 6075, X = 0)</td>
</tr>
<tr>
<td>2</td>
<td>((X - 90)^2 + Y^2 + Z^2 = 8100, Y = 0.577X)</td>
</tr>
<tr>
<td>3</td>
<td>((X - 90)^2 + Y^2 + Z^2 = 8100, Y = -0.577X)</td>
</tr>
<tr>
<td>4</td>
<td>((Y + 78)^2 + Z^2 = 6075, X = 0)</td>
</tr>
<tr>
<td>5</td>
<td>((X + 90)^2 + Y^2 + Z^2 = 8100, Y = -0.577X)</td>
</tr>
<tr>
<td>6</td>
<td>((X + 90)^2 + Y^2 + Z^2 = 8100, Y = 0.577X)</td>
</tr>
</tbody>
</table>

78
Table 4.22: Equations of the final workspace solution for boundary side curves.

<table>
<thead>
<tr>
<th>i</th>
<th>( B_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((X + 45)^2 + (Y + 78)^2 + Z^2 = 29929, Y = -0.577X + 70)</td>
</tr>
<tr>
<td>2</td>
<td>(Y^2 + Z^2 = 7429, X = 61)</td>
</tr>
<tr>
<td>3</td>
<td>((X + 45)^2 + (Y - 78)^2 + Z^2 = 29929, Y = 0.577X - 70)</td>
</tr>
<tr>
<td>4</td>
<td>((X - 45)^2 + (Y - 78)^2 + Z^2 = 29929, Y = -0.577X + 70)</td>
</tr>
<tr>
<td>5</td>
<td>(Y^2 + Z^2 = 7429, X = -61)</td>
</tr>
<tr>
<td>6</td>
<td>((X - 45)^2 + (Y + 78)^2 + Z^2 = 29929, Y = 0.577X + 70)</td>
</tr>
</tbody>
</table>

Table 4.23: Equations of the final workspace solution for final boundary points.

<table>
<thead>
<tr>
<th>i</th>
<th>( p_i )</th>
<th>i</th>
<th>( p_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0, 0, 0)</td>
<td>4</td>
<td>(0, -70, 78)</td>
</tr>
<tr>
<td>1</td>
<td>(0, 70, 78)</td>
<td>5</td>
<td>(-61, -35, 78)</td>
</tr>
<tr>
<td>2</td>
<td>(61, 35, 78)</td>
<td>6</td>
<td>(-61, 35, 78)</td>
</tr>
<tr>
<td>3</td>
<td>(61, -35, 78)</td>
<td>t</td>
<td>(0, 0, 148)</td>
</tr>
</tbody>
</table>

a given desired workspace, solving for the optimum structural parameters, which includes foot placements, of a HWR to satisfy the given workspace is difficult using numerical methods. In this section, a specific optimized design problem is presented using the presented closed-form solution to show how the solution is useful to get the optimum design of the robot based on the design objectives. The cost function can vary based on the design objectives.

Usually it is difficult for the designer of the robot to initially give the whole boundary of the desired workspace distinctly because it is created by infinite points. That is why a designer usually presents basic requirements in a concise way. The presented solution methodology can help designer to do so. It can be said that dealing with 2D workspace of the 2-\(RPR\) mechanism presented in section 4.3.1 as the lateral COW of the hexapod robot will directly affect the 3D final workspace as
Figure 4.25: The lateral COW of the hexapod robot covering the desired workspace.

The same thing is true for the presented lateral RW of the robot as well. In other words, optimizing the lateral workspace will optimize the 3D workspace and vice versa.

Assume a designer needs a robot as small as possible but covering a specific desired working area. Looking at the lateral COW of the hexapod robot, as shown in Figure 4.25, the designer will need a desired workspace. The desired workspace can be represented as a 2D area bounded within a rectangle. Let’s call it workspace rectangle. Then the problem to be solved becomes designing the smallest robot whose COW, for a given orientation, covers the desired workspace rectangle which is one of the problems that designers of parallel robots usually encounter. There are four unknown structural parameters to be derived as the design goal. The points \((x_1, y_1)\) and \((x_2, y_2)\) should be located on the RLC and RUC, respectively as shown in Figure 4.25. The parameters \(l, w, \) and \(h_{min}\) are given initially based on the design goals. This gives two equations as follows based on Equation 4.19 \(x_1 = x_2 = l/2,\)
\( y_1 = h_{\text{min}}, \ y_2 = h_{\text{min}} + w \):

\[
\text{RLC : } (x_1 - k)^2 + (y_1)^2 - l_{\text{min}}^2 = 0 \quad (4.27a)
\]

\[
\text{RUC : } (x_2 + k)^2 + (y_2)^2 - l_{\text{max}}^2 = 0 \quad (4.27b)
\]

For the smallest sized robot, the least square criterion can be used to give the following equations:

\[
\frac{\partial D}{\partial l_{\text{min}}} = 0 \quad (4.28a)
\]

\[
\frac{\partial D}{\partial l_{\text{max}}} = 0 \quad (4.28b)
\]

\[
\frac{\partial D}{\partial b} = 0 \quad (4.28c)
\]

\[
\frac{\partial D}{\partial d} = 0 \quad (4.28d)
\]

where the cost function, \( D \), is given as

\[
D = d_0^2 + d_r^2 + d_l^2 \quad (4.29)
\]

**Numerical Example:** Assume the design of a robot is desired to cover a designated rectangular workspace with \( w = 40\text{mm}, \ l = 80\text{mm}, \) and \( h_{\text{min}} = 40\text{mm} \). This gives \((x_1, y_1)\) and \((x_2, y_2)\) equal to \((40, 40)\) and \((40, 80)\), respectively. Then,

\[
\text{RLC : } (40 - k)^2 + (40)^2 - l_{\text{min}}^2 = 0 \quad (4.30a)
\]

\[
\text{RUC : } (40 + k)^2 + (80)^2 - l_{\text{max}}^2 = 0 \quad (4.30b)
\]

81
and

\[ d_o = 40 - \sqrt{l_{min}^2 - k^2} \]  
(4.31a)

\[ d_r = \frac{l_{max}^2 - l_{min}^2}{4k} - 40 \]  
(4.31b)

\[ d_t = \sqrt{l_{max}^2 - k^2} - 80 \]  
(4.31c)

The final answer derived from MATLAB is \( l_{min} = 40.5 \, mm \), \( l_{max} = 109.93 \, mm \), \( b = 22.92 \, mm \), and \( d = 93.83 \, mm \). Given the numerically optimized design, the exact workspace boundary of the robot can be derived using the closed-form equations of the workspace presented in Table 4.11 through Table 4.17.

**Maximizing the Workspace Volume**

The design and optimization process is not always unique since the design objectives may differ. For example, the problem solved in the previous section is different from the problem of maximizing the workspace volume. For maximizing the volume of the hexapod workspace, inscribed inside a rectangular shape, the surface area of the lateral workspace (2-RPR mechanisms workspace) needs to be maximized according to the following procedure:

1. Locate points \( p_0 \), \( p_r \), and \( p_t \) on the boundary of the desired workspace as shown in Figure 4.26.

2. Connect points \( p_0 \), \( p_r \) and \( p_r \), \( p_t \) using two lines as shown by dashed lines in Figure 4.26.

3. From the centers of the circular curves RUC and RLC (\( c_l \) and \( c_r \), respectively), consider the radius perpendicular to the lines as shown by centerlines in Figure 4.26.
4. Derive the point as a result of intersecting RLC and the line parallel to the radius passing the point \((x_1, y_1)\). Do the same thing for RUC and point \((x_2, y_2)\).

5. Derive the distance between the points on RLC and the point \((x_1, y_1)\) as well as the distance between RUC and the point \((x_2, y_2)\).

6. Minimize those distances using the Least Square method. When minimizing, consider equations derived from step 1 at the same time.

Once the surface area of the lateral workspace is maximized, it is guaranteed that the volume of the corresponding hexapod workspace is maximized since the hexapod workspace, as mentioned before, is created as a result of the intersection of the three rotated lateral workspaces.

Many different optimization processes can be applied to the presented solution in this work. Two examples were given just to show how to use the presented solution.
However, different optimization objectives require different procedures.

**RW-based Design of HWR and Optimization**

The presented methodology and closed-form solution lend itself to RW-based design and optimization problems as well. In this section, an optimized design of the 2-RPR planar parallel mechanism is presented to widen the workspace of the mechanism as much as possible (in $x$ direction) but restrict it to be within a desired rectangle as shown in Figure 4.27. By doing this, it is guaranteed that the workspace of the corresponding hexapod robot is widen as much as possible. To do so, the points $p_0$ and $p_4$ are constrained to the top and bottom edges of the rectangle, respectively, as shown in Figure 4.27. A single dimension optimization is required to minimize the distances $d_i$ while keeping points $p_0$ and $p_4$ on the rectangle. The rectangle parameters, $l$ and $w$, can be initially defined based on the design objectives such as the desired minimum and maximum height expected from the mechanism.
To optimize the workspace, the mechanism needs to be optimized. Optimization of the mechanism focuses on determining the optimized structural parameters, \( l_{\text{min}} \), \( b \), \( l_{\text{max}} \), and \( d \), to produce the optimized workspace. From the rectangle parameters, the desired locations of points \( p_0 \), \( p_4 \), and \( p_7 \) are known and provide Equations 4.32, 4.33, and 4.34, respectively.

\[
\begin{align*}
  l_{\text{max}} \sin \left( \cos^{-1} \left( \frac{d-b}{2l_{\text{max}}} \right) \right) - w &= 0 \quad (4.32) \\
  b^2 + (d - l_{\text{min}})^2 - l_{\text{min}}^2 - 2b(d - l_{\text{min}}) &= 0 \quad (4.33) \\
  d - b - 2l_{\text{min}} &= 0 \quad (4.34)
\end{align*}
\]

Assume that the desired rectangle has the following dimensions: \( w = 60 \ mm \) and \( l = 80 \ mm \), and the robot structural constraints are: \( 20.0 \ mm < l_{\text{min}} < 25.0 \ mm \), \( 60.0 \ mm < l_{\text{max}} < 70.0 \ mm \), \( 55.0 \ mm < b < 65.0 \ mm \), and \( 100.0 \ mm < d < 110.0 \ mm \). The closed-form equations of the workspace boundary are already given. Equation 4.28 is used again where the cost function is \( D = d_1^2 + d_2^2 + d_3^2 \). From Equations 4.32, 4.33, 4.34, and the analytical solution for points \( p_1 \), \( p_2 \), and \( p_3 \), from Table 4.3, the following optimization conditions can be derived.
Figure 4.28: The final optimized workspace with all cardinal points.

\[
l_{\text{max}} \sin \theta_2^0 - 60 = 0 \quad (4.35)
\]

\[
b^2 + (d - l_{\text{min}})^2 - l_{\text{min}}^2 - 2b(d - l_{\text{min}}) = 0 \quad (4.36)
\]

\[
d - b - 2l_{\text{min}} = 0 \quad (4.37)
\]

\[
d_1 = \left(\frac{d + 80}{2}\right) - (l_{\text{max}} + \frac{b}{2}) \frac{(l_{\text{max}} + b)^2 + d^2 - l_{\text{max}}^2}{2(l_{\text{max}} + b)d} \quad (4.38)
\]

\[
d_2 = \left(\frac{d + 80}{2}\right) - (l_{\text{max}} + \frac{b}{2}) \frac{(l_{\text{max}} + b)^2 + d^2 - l_{\text{min}}^2}{2(l_{\text{max}} + b)d} \quad (4.39)
\]

\[
d_3 = \left(\frac{d + 80}{2}\right) - l_{\text{max}} - \frac{b^2 + (d - l_{\text{max}})^2 - l_{\text{min}}^2}{4(d - l_{\text{max}})} \quad (4.40)
\]

The tolerance for points \(p_0\) and \(p_4\) is considered to be \(\pm 0.5\, mm\) although any tolerance may be selected based on design needs. The final optimum answer derived using the Least Squares method is \(l_{\text{min}} = 23.2\, mm\), \(l_{\text{max}} = 64.0\, mm\), \(b = 61.0\, mm\), and \(d = 107.4\, mm\). Using the optimized structural parameters, the exact RW, shown in Figure 4.28, can be calculated using the closed-form equations. In Figure 4.28, the two of cardinal points, \(p_5\) and \(p_6\), are located within the workspace and not on
In a hexapod robot and based on the McGee theorem \cite{71}, the static stability boundary is the support polygon (hexagon) from the top view. Therefore, the static stable workspace is that part of the workspace which is inside of the stability hexagon.
Figure 4.29 shows three different possible static stable workspace from the top view. The hatched region represents the static stable workspace of the robot. When a HWR is being designed, the concept of static stable workspace should be considered such that the boundary of the workspace from the top view should be located inside of the support polygon.
Chapter 5

Stability and Control

Stability analysis of multi-legged robots is necessary for control especially under dynamic situations over irregular terrain. This chapter analyzes the stability of multi-legged/wheeled robots and presents a novel stability margin based on the normal foot forces of the robot used in a bio-inspired reactive controller.

5.1 Background

Stability of multi-legged/wheeled robots has been investigated for more than four decades. The main concept of the stability of this kind of vehicles is that the CG of the robot has to be kept inside of a stable region to prevent the robot from tipping over. Hence, under both static and dynamic situations when walking, moving or manipulating, it is essential to monitor robot stability at each instant through the use of a stability criterion in terms of control especially while traversing over irregular terrain.

There are several widely used stability criteria in the field of robotics and wheeled systems which can be generally divided into static and dynamics-based criteria. However, they can be further classified into five categories based on their stability
Distance-based criteria \([4, 63, 71, 123]\) focus on either the distance between the support polygon and projection of the CG \([4, 63, 71]\) or the distance between the support polygon and the net force vector acting at the CG \([123]\) as the metric for stability. The Stability Margin (SM) by McGhee \([4, 71]\), shown in Figure 5.1, is the most notable distance-based stability criterion since it is the first presented stability margin. The SM is defined to be the minimum distance between the projection of the CG and the support polygon boundary.

Angle-based criteria \([5, 124–126]\) use the angle between the support polygon and the net force acting at the CG to represent the stability of the system. Relying on the number of citations to their work, the most notable criterion in this category is the Force Angle Stability Margin (FASM) \([5]\) which is shown in Figure 5.2. The FASM is defined to be the minimum angle between the net force and the line connecting the center of mass to the rotation point (for 2D case) or rotation axis (for spatial case).

Energy-based criteria \([127–131]\) look at the difference between the energy of the robot in two different situations: the current configuration and when tipping over. The first statics energy-based stability margin (ESM) was presented by Messuri and Klein \([127]\) and the first dynamic one (DESM) was presented by Ghasempoor and

Figure 5.1: An illustration of SM presented by McGhee \([4]\).
Sepehri [128] were normalized by S. Hirose et al. (NESM) [129] and Garcia et al. (NDESM) [130], respectively.

Most of the stability criteria fall under moment-based criteria [132–140] since tipping over occurs when moments are exceeded about one of the tipping over axes. However, moment-based criteria can be rather difficult to implement especially on irregular terrain since they require knowledge of the moments about each tip over axis and the foot positions of each leg. The most notable moment-based criteria are the Dynamics Stability Margin (DSM) [132], Zero Moment Point (ZMP) [133–137], and Tumble Stability Margin (TSM) [138]. A less commonly known moment-based criterion is the Moment-Height Stability (MHS) [139,140] which, regardless of rather high calculation costs and low sensitivity, it is more useful for mobile wheeled robots on planar terrain and becomes very difficult to implement with a walking system over irregular terrain.

Force-based criteria [123,141–144] focus on the body contact forces of the robot. With legged robots, contact generally occurs at the feet. When tipping over, the foot forces, except those creating the tipping over axis (or point in planar robots [123]), become zero. Although attempts were made to use this concept for robot control [141–143], there is no distinctly defined stability margin or extent in this
category.

Garcia et al. [145] proposed a classification of several stability criteria and finally showed that, there is no optimal criterion to be applied for real applications in terms of taking into consideration the terrain unevenness, static, dynamic, and inertial effects, and being cost effective. Also they did a comparative study in terms of computational complexity of the criteria and showed that FASM (TOSM) is the most complex of the compared margins. In another comparative study, Roan et al. [146] compared ZMP, MHS, and FASM using some experimental tests and showed that FASM is the best and ZMP is the worst in terms of tipping over prediction. However there cannot be seen big differences between FASM and MHS since the tests are not comprehensive.

As a summary, no criterion presents a quantitative stability extent that measures how close or far the robot is to the unstable or the maximum possible stable state. Also, since the stability is constantly monitored in a selected time period, dynamics margins usually tend to require a considerable amount of input sensor data and calculations.

For the control purposes, in general, there are a series of different control methods such as position and attitude control, impedance control, force control, neural networks, stiffness control, damping control, fuzzy control, posture control, locomotion control, and gait control [147–163]. These controllers are usually embedded within the high-level controllers of the walking robots. However, these concepts are not the focus of this research. In general it can be said that every multi-legged walking robot is controlled in three levels; trunk level, leg level, and joint level as shown in Figure 5.3. This is critical for any walking robot. Then, each level may employ some different types of controllers mentioned above. Different types of sensors may be needed for different controllers accordingly. For more information, readers are
Figure 5.3: The general control architecture of a multi-legged robot [6].

referred to [164]. Later in this chapter, a control architecture is presented taking into consideration the workspace and stability used for reactive control.

5.2 Foot Force Based Stability

In this section, a new stability criterion called the Foot Force Stability Margin (FFSM) is presented. The FFSM is a force-based criterion that represents system stability in an extent bounded by the unstable state and the maximum stable state and is applicable to dynamical scenarios [165, 166]. The FFSM has been modified (MFFSM) for top-heaviness and geometrical sensitivity. The accuracy, conciseness, low calculation cost, and sensitivity of the criterion make it efficient for use in an on-line and real-time controller. One of the main merits of FFSM is that it needs less sensor information compared to existing margins since only the measured foot forces are necessary to provide a measure of system stability.

5.2.1 Foot Force Stability Margin

Under dynamic situations, it is essential to monitor robot stability at each instant through the use of a stability criterion especially while traversing over irregular
terrain as shown in Figure 5.4. According to [130], a walking machine is dynamically stable if the moment about \( j \)th edge of the support polygon due to the robot/ground forces and moments is positive (in the clockwise direction). Note that [130] uses \( F \times R \), not \( R \times F \). Using \( R \times F \), the definition for dynamic stability can be rewritten as follows: a walking machine is dynamically stable if the moment about the \( j \)th edge of the support polygon due to the robot/ground forces and moments is negative (in the counterclockwise direction). From Newton’s law, the following must be satisfied about every edge of the support polygon:

\[
M_{in,j} = M_{gr,j} + M_{man,j} + M_{sup,j}
\]  

(5.1)
where $M_{\text{in},j}$ is the moment due to the inertial force and moment, $M_{\text{gr},j}$ is the moment due to the gravitational force and moment, $M_{\text{man},j}$ is moment due to the manipulation (external) forces and moments, and $M_{\text{sup},j}$ is the moment due to the foot contacts forces and moments. All of the moments are calculated about the $j$th edge of the support polygon. From Equation 5.1, the following can be written:

$$M_{\text{sup},j} = -(M_{\text{gr},j} + M_{\text{man},j} - M_{\text{in},j}) \quad (5.2)$$

The term in the parenthesis in the right hand side of Equation 5.2 is the net moment acting about the $j$th edge of the support polygon due to all of the gravitational, manipulation, and inertial forces and moments. Hence, it can be replaced with $M_{\text{Net},j}$, which must be compensated by the moment due to robot/ground forces and moments ($M_{\text{sup},j}$). Therefore, to have a dynamically stable robot, the following must be satisfied:

$$M_{\text{sup},j} = -M_{\text{Net},j} \quad (5.3)$$

which indicates that in order for a robot to be dynamically stable, the net moment about the $j$th edge of the support polygon must be positive (in the clockwise direction), but with the same magnitude as the moment due to the support forces and moments. Otherwise, the robot will tumble. Assuming point contacts for the supporting legs, the following equation can be written to represent the $M_{\text{sup},j}$:

$$M_{\text{sup},j} = \sum_{i=1}^{n} R_i \times F_i = \sum_{i=1}^{n} f_i \cdot R_i \times (\sqrt{1 + \mu_i^2} \cdot e_{F,i}) \quad (5.4)$$

where $n$ is the number of supporting legs, $R_i$ is the position vector of $i$th foot contact perpendicular and with respect to the $j$th support edge, $F_i$ is the $i$th contact foot force vector, $f_i = \|f_i\|$ is the $i$th normal foot force magnitude where $f_i$ is the normal
component of $F_i$ as shown in Figure 5.4, $\mu_i$ is the $i$th foot/ground friction coefficient, and $e_{F_i}$ is the $i$th unit vector of the contact foot force $F_i$.

Therefore, the dynamic stability of the robot directly correlates with the normal foot forces ($f_i$) as well as the friction coefficient ($\mu_i$) and foot positions ($R_i$) according to [130], Equation 5.3, and Equation 5.4. However, to analyze the stability of an ideal multi-legged robot, the friction coefficient is typically considered to be large enough to prevent the robot from slipping. Therefore, instability is considered as tipping over (tumbling) not slipping. It is assumed that the foot distribution is non-collinear, $\sum_{i=1}^{n} ||R_i|| \neq 0$. Also, all contact between the legs and the contact surfaces are assumed to be point contacts.

Given a multi-legged ($n \geq 3$) robot with only two strictly positive forces, indicating that only two legs are in contact with the ground, creating the $j$th support edge, states that $M_{sup,j}$ is zero which requires the $M_{Net,j}$ to be zero as well. Otherwise ($M_{Net,j} \neq 0$), Equation 5.3 will not be satisfied and the robot will tumble. Therefore, to be considered dynamically stable, the robot must have, at least, one more leg on the ground with a strictly positive normal foot force to cause a negative moment about the $j$th support edge and compensate for the positive $M_{Net,j}$. If $M_{Net,j}$ is negative, the robot is unstable. With the above discussion and assumptions, the following definition is proposed:

**Definition 1**: An ideal spatial multi-legged walking robot with $n$ supporting legs ($n \geq 3$) is dynamically stable at time $t$ if and only if there are at least three non-collinear legs with strictly positive normal foot forces ($f_i > 0$) at time $t$.

Definition 1 provides a quick measurable method for determining the stability of the system. However it does not consider $R_i$ which greatly influences the $M_{sup,j}$. To consider the foot distribution, and other geometry and heaviness factors, a modified Foot Force Stability Margin (MFFSM) will be presented. As indicated by Definition
1, the current relation between stability and foot forces requires strictly positive normal foot forces \((f_i > 0)\). However, the relation can be modified to account for walking on walls, ceilings, and highly inclined surfaces considering friction as well which are not within the scope of this work.

**The Metric of FFSM**

As declared in Definition 1, stability occurs when there are at least three legs with strictly positive foot forces. Intuitively, maximum stability occurs when the magnitude of the foot forces are all the same i.e. the forces are equally distributed across all of the feet. It is desired to have a stability measure that provides a normalized understanding of the current stability of the system based on the foot force magnitudes. The FFSM uses all supporting foot forces to describe the stability status of the system. Let \(f_1, f_2, \ldots, f_n\) be the normal foot force magnitudes of the supporting legs where \(n\) denotes the number of supporting legs. The product of all foot forces, \(\prod_{i=1}^{n} f_i\), is used as a base for defining the FFSM since it satisfies Definition 1 for instability. For the FFSM to satisfy the maximum stability state of the robot, the product is normalized between 0 and 1. For this purpose, the individual foot force ratio to the total measured force, \(\frac{f_i}{f_{tot}}\), is used where \(f_{tot} = \sum_{i=1}^{n} f_i\). Note that \(\sum_{i=1}^{n} \frac{f_i}{f_{tot}} = 1\). The maximum magnitude of \(\prod_{i=1}^{n} \frac{f_i}{f_{tot}}\) is \(\frac{1}{n^n}\) which correlates with the maximum stability state of the robot. In order for the FFSM to result in a number between 0, for the unstable state, and 1, for the maximum stable state, the term \(n^n\) is multiplied by the product. The FFSM at time \(t\) for a multi-legged robot with \(n\) supporting legs is then defined as

\[
FFSM = S = \prod_{i=1}^{n} \frac{f_i}{f}, 0 \leq S \leq 1
\]  

(5.5)
where \( n \) is the number of supporting legs with strictly positive foot forces, and 
\( \bar{f} = \frac{f_{\text{tot}}}{n} \) is the average of all normal foot force magnitudes. Hence, the FFSM is defined as the product of the fractions of foot forces to the average of all foot forces.

Equation 5.5 provides a stability margin magnitude between zero and one, \( 0 \leq S \leq 1 \), indicating how close the system is to the unstable or maximum stability state. As expected, Equation 5.5 indicates that a more even distribution of foot forces enhances the stability of the whole system. Hence, the maximum stability, \( FFSM = 1 \), only occurs when the foot forces are evenly distributed i.e. the standard deviation of foot force magnitudes is zero.

Given a system with \( n \geq 4 \) and \( m \) legs, \( m \leq n - 3 \), lose contact with the ground, which generally occurs on irregular terrain, Equation 5.5 would indicate a zero stability margin while the system may still be stable with \( n - m \) supporting legs. For example, when a walking robot changes from a quadruped, \( n = 4 \), to a tripod, \( n = 3 \), configuration, one leg loses contact with the ground while the tripod supporting configuration maintains stability. To account for purposeful loss of ground contact, \( n \), in the calculation of the FFSM, should be updated accordingly, \( n \leftarrow n - m \), at each iteration within the controller. To guarantee that the robot will be stable after switching from \( n \) to \( n - m \) legs, the FFSM of both states should be calculated simultaneously as the robot switches. In this way, if the \( n - m \) configuration is not stable, the robot will know instead of falling.

5.2.2 Modified Foot Force Stability Margin

Since the FFSM only focuses on the magnitude of the normal component of the foot forces, there are multiple robot configurations that would produce the same stability margin but, intuitively, should be different. Consider the cross section of a legged robot as shown in Figure 5.5. Assuming that \( \alpha > 1 \) is a constant, the
Figure 5.5: Multiple planar robot configurations with the same Foot Force Stability Margin.

FFSM is the same for all four cases. However, this is not conceptually true from the point of view of system energy since some have a higher tip over potential in the presence of lateral noise. For a specific instance, suppose a lateral force is acting upon the system. Thus, conceptually, Figure 5.5-a and Figure 5.5-d have the lowest and highest geometrical tip over potential, respectively. On the other hand, in Figure 5.5-(a-d), the tip over potential is reduced if the vertical force $F$ is replaced with $\alpha F$. This represents sensitivity to top-heaviness. However, the FFSM neglects the geometry and heaviness of the robot. Also, assuming a tripod robot with evenly distributed foot forces while all the foot contacts are collinear, FFSM gives $S = 1$ while the robot is in the threshold of tumbling. Since these parameters directly affect the tip over potential of the system, a general Modified FFSM (MFFSM) is developed to enhance the sensitivity of the stability margin to these parameters and an example is given in the next section. A general MFFSM is given as

\begin{equation}
MFFSM = m(t)S
\end{equation}
where $S$ is the FFSM and $m(t)$ is an application specific scaling factor which varies over the terrain and could be defined to consider any desired parameters reflecting the stability of the robot.

Conceptually, to enhance the stability from a physical perspective, the CG height of the robot should be reduced as much as possible while the distance between the feet contact points and the CG should be maximized or spread out. However, the stability sensitivity of a system is not constant under changes to system parameters and across different loading conditions. For example, if a system is used for an application where the robot CG is only under lateral loading parallel to the ground, the stability of the system will be infinite if the height of the robot becomes zero. However, this is not true in the presence of a moment, under CG offset loading. Therefore, the stability of the system will be more sensitive to the height of the robot compared to the loading force or leg placement. Accordingly, if a vertical force, perpendicular to the ground, is applied to the robot, the stability will be more sensitive to top-heaviness as compared to other parameters. This implies that when modifying the FFSM, the scaling factor should be taken into consideration along with the geometrical configuration and mass of the robot. The following section provides an example of MFFSM developed for spatial and planar legged robots and considers the application of MFFSM to wheeled systems.

**Spatial Robot**

Consider a general spatial $n$-legged robot traversing an irregular terrain as shown in Figure 5.6, where the support polygon is not restricted to be planar. In Figure 5.6, $\mathbf{M}_t$ is the net moment vector acting at the CG and $\mathbf{F}_t$ is the net force vector acting at the CG where $\mathbf{F}_t = - \sum_{i=1}^{n} \mathbf{F}_i$ and $\mathbf{F}_i$ is $i$th foot force vector whose normal component is $\mathbf{f}_i$ with a magnitude of $f_i = \| \mathbf{f}_i \|$. Hence, $\bar{f} = \frac{1}{n} \sum_{i=1}^{n} f_i$ directly correlates to the
net force acting on the system and can be used for top-heaviness sensitivity with the FFSM [125]. On the other hand, the stability enhancement due to height reduction and foot placement spread could be directly taken into consideration by scaling the FFSM by $\frac{P_i}{h_i}$, where $P_i = \|P_i\|$, $P_i$ is the tip over axis normal vector created by perpendicularly connecting the CG to the tip over axis, and $h_i$ is the height of the CG with respect to the tip over axis aligned with the gravity vector which is the vertical portion of the tip over axis normal vector, $P_i$. The use of $\frac{P_i}{h_i}$ is only directly applicable as long as the CG lies within the support polygon. Otherwise, $\frac{P_i}{h_i}$ should be replaced with $\frac{1}{P_i h_i}$. Hence, the MFFSM considering top-heaviness and geometrical sensitivities can be described using

$$MFFSM = a f^{\alpha} b P_i^{\beta} \frac{1}{c h_i^{\gamma}} S \quad (5.7)$$
where \(a, b, c, \alpha, \beta, \) and \(\gamma\) are the application specific constants based on robot loading and environmental conditions, and \(i\) is iterated for all foot contacts, \(i = 1, \ldots, n\). Generally, the MFFSM need only be calculated for the two feet with the largest measured foot forces normal to the contact surface. If the loading condition and environment of the robot are unknown, unity constants may be used. Note that the exponent \(\beta\) must be positive if the projection of the CG is inside or on the support polygon and must be negative if it is outside of the support polygon.

The height, \(h_i\), is given as

\[
h_i = P_{iz} - P_i \cdot \left(-\hat{k}\right) \tag{5.8}
\]

where \(\hat{k}\) is the unit vector of the \(z\)-axis of the global coordinate system. The vector \(P_i\) is the portion of the vector \((C_{i+1} - C_g)\) which is perpendicular to the tip over axis where \(C_i\) is the position vector of \(i\)th ground contact point and \(C_g\) is the position vector of CG. This indicates that \(P_i\) can be obtained by subtracting that portion of the vector \((C_{i+1} - C_g)\) which is along the tip over axis as follows

\[
P_i = (C_{i+1} - C_g) - [(C_{i+1} - C_g) \cdot \hat{t}_i] \hat{t}_i \tag{5.9}
\]

where \(\hat{t}_i = \hat{t}_i / \|\hat{t}_i\|\) is the unit vector of \(t_i\), the \(i\)th tip over axis vector, given by

\[
t_i = C_{i+1} - C_i, i = 1, \ldots, n - 1 \tag{5.10}
\]

In Equation 5.9, when \(i = n\), \(C_{i+1}\) will become \(C_1\).

In a spatial robot, tipping over occurs about an axis called the tip over axis. The tip over axis is chosen as the axis created by connecting the two surface contacts which have the largest foot force magnitudes amongst all foot forces. The subscript \(i\) in Equations 5.7-5.10 relates to that tip over axis. This reduces the calculations to
only one axis. However, if there are two such axes with the same set of foot forces at the same time, the measure of MFFSM could be different for them since \( \frac{bP_i\beta}{ch_i\gamma} \) varies. In this case, a lower MFFSM indicates a higher tip over potential about the tip over axis and vice versa. In a very specific case, when the MFFSM is the same for two axes, tip over will occur about the single foot contact point with maximum foot force instead of an axis. One example is a tripod robot with a lateral force applied along the bisector of the supporting isosceles triangle.

**Planar Robot**

Consider a planar robot on irregular terrain as shown in Figure 5.7. Equation 5.7 is applicable with minor changes to the parameter definitions. In the planar case, \( P_i = C_i - C_g \) is the ith tip over vector with magnitude \( P_i = \|P_i\| \), and \( h_i \) is the height of the CG with respect to ith ground contact point in the gravity direction. In Equation 5.7, the choice of subscript \( i, i \in 1, \ldots, n \), is based on the foot with the largest measured foot force. For example, if \( f_1 > f_2 \), then \( i = 1 \), if \( f_1 < f_2 \), then \( i = 2 \), and if \( f_1 = f_2 \), then \( i \) could be either 1 or 2. In the latter case, FFSM is 1 but MFFSM for \( i = 1 \) and \( i = 2 \) could be different since \( \frac{bP_i\beta}{ch_i\gamma} \) could be different. In this case, a lower MFFSM indicates a higher tip over potential about that foot.

**Mobile Wheeled Robots**

In mobile wheeled robots, the locations of the surface contacts do not change with respect to the CG. Therefore, \( P_i \) is fixed and does not change. Hence, the term \( \frac{bP_i\beta}{ch_i\gamma} \) in Equation 5.7 becomes \( \frac{1}{ch_i\gamma} \) when dealing with mobile wheeled robots. For mobile wheeled robots, the MFFSM is given as

\[
MFFSM = \frac{af^\alpha}{ch_i\gamma}S
\]  
(5.11)
Assume that there is a planar robot over an uneven terrain as shown in Figure 2. From Figure 2, $\mathbf{p}$ is the number of supporting legs or DOS, $\mathbf{f}$ is the net force vector acting at the center of gravity (COG), $\mathbf{d}$ is the $d$th foot force vector and $\mathbf{m}$ is the net moment vector acting at the center of gravity. $\mathbf{n}$ and $\mathbf{v}$ are normal foot forces. $\mathbf{p}_d$ is the $d$th foot placement, $\mathbf{h}_d$ is height of COG with respect to $\mathbf{p}_d$ placement. Therefore, the following can be derived:

\[ \mathbf{f} = \mathbf{p} \]

The Normalized FFSM (NFFSM) is given as follows:

\[ n_{FFSM} = \frac{\mathbf{a} \cdot \mathbf{f}}{\mathbf{a} \cdot \mathbf{S}} = 1 \]

In the figure above, three different situations can exist:

1. $n_{FFSM} \neq 1$: in this status, the tipping over point is $\mathbf{n}$ and so $\mathbf{p}_d = \mathbf{n}$ in the Equation (2).

2. $n_{FFSM} = 1$: in this status, the tipping over point could be either $\mathbf{n}$ or $\mathbf{v}$ because $\mathbf{f} = \mathbf{p}_d \mathbf{p}_v$.

3. $n_{FFSM} < 1$: this status is opposite of 1.

3.2 Spatial Case

If the ground is level, the height $h_i$ will also be fixed and the ratio $\frac{h \beta}{ch}$ becomes $1$. Equation 5.11 can then be simplified to

\[ MFFSM = a \mathbf{f} \mathbf{S}. \]

Discussion on the FFSM and MFFSM

Simplicity and expressiveness are attractive characteristics of FFSM. The FFSM provides a magnitude between zero and one which indicates how far the system is from instability or maximum stability state. Since all of the effects of gravity, external forces, inertial forces and disturbances can be observed in foot forces, the dimensionless FFSM represents a dynamics stability margin in this perspective. In a practical dynamics controller of a robot, once the foot forces are known, the FFSM can be calculated using Equation 5.5. The tip over axis is found by determining the surface contact positions of the two feet with the highest magnitude of foot force. Hence, there is no need to calculate all foot positions and heights when calculating MFFSM. For the developed example of MFFSM, Equation 5.9 and Equation 5.10 are the only geometrical calculations which are needed to calculate the stability of
the system at any instant. On the other hand, the foot force magnitudes are provided at any instant by force sensors and substituted in Equation 5.5. Eventually, substituting Equations 5.5, 5.9, and 5.10 into Equation 5.7 will provide the MFFSM which can be used in a controller in an on-line and real-time manner. This conciseness of calculation is very helpful and cost effective when dealing with quadruped or hexapod robots in irregular terrain since with real-time control, all of these calculations must be deterministic and occur at high frequency. Also, with the FFSM and the MFFSM, no further intermediate calculations nor information like moments about all the edges of the supporting polygon are needed.

The MFFSM enhances the sensitivity of the system to any desired system characteristic. The term \( a \bar{f} \alpha bP_i \beta \) was multiplied to Equation 5.5 to include sensitivity to top-heaviness, foot placement with respect to CG, and height. When using MFFSM, one should simultaneously consider FFSM to take advantage of having a measure between 0 and 1 which provides knowledge of how far or close the robot is to the most stable and unstable states. A lower MFFSM represents a higher tip over potential and vice versa.

The FFSM does not take into consideration the possibility of recovering from instability nor does it handle dynamic balance of a system with at most two legs in contact with the ground. FFSM is also not applicable to running robots since there are periods of time in which no leg is in contact with the ground.

### 5.3 Validation

Multiple studies comparing stability criteria have been completed [139, 140, 145, 146]. However, no optimal criterion exists for complex situations that include irregular terrain, inertial forces, manipulation forces, and cost efficiency [145].
From [145, 167], each criterion has its own characteristics with both advantages and disadvantages. The following sections discuss the simulations and experiments conducted to validate the FFSM and MFFSM using a hexapod robot. During both the simulation and experiment, the MFFSM was calculated using

$$MFFSM = \int \frac{P_i}{h_i} S$$  \hspace{1cm} (5.13)

which uses Equation 5.7 with $a = b = c = 1$, and $\alpha = \beta = \gamma = 0$. The selected coefficients are applicable to any legged/wheeled robot under any loading situation.

5.3.1 Numerical Validation of FFSM and MFFSM

while in the experiment, the foot forces are measured, to determine the simulated forces acting on the feet of the robot for simulation purposes, a Foot Force Distribution (FFD) method needs to be utilized.

Foot Force Distribution Calculations

To simulate the forces acting on the robot due to contact with the ground, a FFD method is required. As with the kinematics, dynamics of parallel legged robots can be divided into forward and inverse dynamics with the same concept. Calculation of FFD and joint torque distribution of legged robots are important for control purpose. In general, there are two main theoretical methods for calculating the joint torque and FFD of a multi-legged robot: the Newton-Euler method [168, 169] and the Lagrange method [170]. Although both methods can be used for any multi-legged robot, they are generally not used in practical applications due to high computation costs. In practice, the FFSM and MFFSM directly use the measured foot forces. For simulation purposes in this section, the Newton-Euler method is used.
The solution is regarding $i$th leg and $j$th link or segment. Note that $i=1,2,3,4,5,$ and $6$ and $j=1,2,$ and $3.$ However, link "0" (link "$j-1$" for $j=1$) means the main body of the robot and link 4 (link "$j+1$" for $j=3$) means the ground. Also note that:

\[

ci_j(j-1) = -c_i_j(j-1) - c_i_j(j+1) \times \vec{F}_{i(j+1)}
\]

Newton Law:

\[
\sum \vec{F} = m\vec{a} \quad (5.14)
\]
\[
\sum \vec{M} = I\vec{\alpha} \quad (5.15)
\]

and considering Figure 5.8, Newton’s laws can be written for $i$th leg and $j$th link as follows:

\[
m_{ij}\vec{a}_{ij} = m_{ij}\vec{g} - \vec{F}_{ij(j-1)} + \vec{F}_{i(j+1)j} \quad (5.16)
\]
\[
J_{ij}\vec{\alpha}_{ij} + \vec{\omega}_{ij} \times (J_{ij}\vec{\omega}_{ij}) = -\vec{M}_{ij(j-1)} - \vec{M}_{i(j+1)j}
\]
\[
-\vec{c}_{ij(j-1)} \times \vec{F}_{ij(j-1)} - \vec{c}_{ij(j+1)} \times \vec{F}_{i(j+1)j} \quad (5.17)
\]
and can be written for the main body (subscript $b$) as follows:

$$m_b \ddot{a}_b = m_b \ddot{g} + \sum_{i=1}^{6} \vec{F}_{ib}$$  \hspace{1cm} (5.18)

$$J_b \ddot{\alpha}_b + \vec{\omega}_b \times (J_b \vec{\omega}_b) = \sum_{i=1}^{6} \vec{M}_{ib} + \sum_{i=1}^{6} \vec{r}_{ib} \times \vec{F}_{ib}$$  \hspace{1cm} (5.19)

where the subscripts $i$ and $j$ refers to $i$th leg and $j$th link, respectively, $m$ is the mass, $J$ is the moment of inertia, $\vec{\omega}$ is the rotational velocity vector, $\vec{\alpha}$ is the rotational acceleration vector, $\vec{r}$ is the foot contact position vector, $\vec{g}$ is the gravitational acceleration vector, $\vec{c}$ is the joint vectors calculated from the CG of the link, and $\vec{F}$ and $\vec{M}$ are the force and moment vectors, respectively, shown in Figure 5.8.

The inclusion of the leg dynamics complicates the computation requiring an indirect method for finding the FFD [169]. For simplicity, the legs are generally considered to be massless [168]. The simplified Newton-Euler equations for finding the FFD in an inertial coordinate frame, G, considering Figure 5.9, are given by

$$\mathbf{F}_g = m \dot{\mathbf{v}}_g - \mathbf{w}_g - \mathbf{f}_{ext,g}$$  \hspace{1cm} (5.20)

$$\mathbf{M}_g = \dot{\mathbf{H}}_g - \mathbf{m}_{ext,g}$$  \hspace{1cm} (5.21)

where $\mathbf{H}_g$ is the angular momentum, $\mathbf{w}_g$ is the weight vector, $m$ is the total mass of the robot located at the CG of the robot which is the origin of the body frame, $\mathbf{f}_{ext,g}$ and $\mathbf{m}_{ext,g}$ are the total external force and torque acting on the CG, and $\mathbf{F}_g$ and $\mathbf{M}_g$ are the total force and moment which must be applied to the CG by the feet contact forces to satisfy the desired motion. All quantities in the inertial frame are represented by the subscript $g$. It is easier to solve Equation 5.20 in the body frame. All quantities in the body frame, $B$, are represented by the subscript $b$. Rewriting
Figure 5.9: A schematic of a HWR on an uneven terrain with external force and moment applying on the body.

Equation 5.20 into the body frame, B, gives

\[
\begin{align*}
\mathbf{F}_b &= m \dot{\mathbf{v}}_b + \mathbf{\omega}_b \times m \mathbf{v}_b - \mathbf{w}_b - \mathbf{f}_{ext,b} \\
\mathbf{M}_b &= \dot{\mathbf{H}}_b + \mathbf{\omega}_b \times \mathbf{H}_b - \mathbf{m}_{ext,b}
\end{align*}
\]  

Equation 5.22 is obtained by applying a rotation matrix to Equation 5.20. The rotation matrix, \( \mathbf{R}_{G}^B \), relates the inertial frame \( G \) to the body frame \( B \). For simplifying the problem, it is assumed that the legs are massless with no feet contact moments. The current linear and angular velocities (\( \mathbf{v}, \mathbf{\omega} \)) and the accelerations (\( \mathbf{a}, \mathbf{\alpha} \)) are given (or desired). Let \( \mathbf{F}_i \) be the contact forces at the feet contact points with the ground represented in the inertial frame \( G \), whose centers are at \( \mathbf{r}_i(x_i, y_i, z_i) \). Therefore, \( \mathbf{r}_i \) is a vector whose origin is at the CG whose end is at the contact points and is represented in the inertial frame \( G \). Then the equations of equilibrium for
the vehicle body can be written as follows

\[
\sum_{i=1}^{n} F_i = R_B^G F_b
\]  
(5.24)

\[
\sum_{i=1}^{n} (r_i \times F_i) = R_B^G M_b
\]  
(5.25)

where \( n \) is the number of feet on the ground and \( R_B^G \) is the rotation matrix from frame \( B \) to the inertial frame \( G \). Writing it in matrix form gives

\[
Gq = w
\]  
(5.26)

where

\[
G = \begin{bmatrix}
I_3 & \cdots & I_3 \\
R_1 & \cdots & R_n
\end{bmatrix},
\]

\[
R_i = \begin{bmatrix}
0 & -z_i & y_i \\
z_i & 0 & -x_i \\
-y_i & x_i & 0
\end{bmatrix}
\]

and

\[
w = R_B^G \begin{bmatrix}
F_b^T & M_b^T
\end{bmatrix}^T.
\]

The FFD aims to solve Equation 5.26 for \( q \). For this purpose, multiple techniques may be applied [168, 169]. The pseudo inverse method is employed in this work as the optimal technique. Therefore, the FFD can be obtained as follows:

\[
q = G^+ w
\]  
(5.27)

where \( G^+ \) is the pseudo inverse of matrix \( G \).
The above scenario is called “Loaded” scenario since the robot is gripping the rod during steps 2. There is also a “Non-Loaded” scenario as second scenario in which the robot has the same motion as the first one but without gripping the rod. In the loaded scenario, the external force acting at the gripper, because of the rod weight, is assumed to be 0.3m. In the loaded scenario, the external force acting at the gripper, because of the rod weight, is assumed to be 0.3m.

Simulation Setup

For the simulation, a virtual hexapod with a gripper tool, modeled after the Lynxmotion hexapod depicted in Figure 5.10, was simulated under MATLAB going through four motion phases: a negative translation in the z-axis, grabbing a cylindrical rod, a positive translation in the z-axis, and a positive translation in the x-axis. The motion phases were executed with three legs on the ground following a tripod gait. An xyz ground coordinate system is considered. The gripper is attached to the front of the robot along the x direction and is 250 mm off from CG. During the motion, the CG is assumed to be at the center of the platform.

Simulation Results: Comparison with FASM

The first simulation was executed to compare the FFSM and MFFSM against the Force Angle Stability Margin (FASM) [5]. As shown in Figure 5.11, the simulation scenario follows the four motion phases described in the simulation setup. In the first motion phase, 0 ≤ t < 10 sec, the robot will reduce its height from 300 mm to 200 mm. In the second motion phase, 10 ≤ t < 11 sec, the robot will pick up a
25 \text{N} metallic cylindrical rod. In the third motion phase, $11 \leq t < 26 \text{ sec}$, the robot will go to increase its height to 350 \text{mm}. During the final motion phase, $t \geq 26 \text{ sec}$, the robot will move laterally along the $x$-axis with the cylindrical rod. Figure 5.12 shows the results comparing the FFSM, MFFSM and FASM.

From Figure 5.12, all margins indicate the same instability point which validates the FFSM and MFFSM compared to FASM. During the first phase of motion, $0 \leq t < 10 \text{ sec}$, the sensitivity of the FASM to the height of the robot is higher than FFSM and MFFSM. During the second phase of motion, when the robot picks up the cylinder at $t = 10 \text{ sec}$, the sharp increase in the MFFSM indicates it is more sensitive to top-heaviness compared the FASM. The FFSM drops due to irregular FFD from the moment about the CG caused by the weight of the metallic cylindrical rod. During the third phase of motion, the robot returns to the initial height and pass it which reduces overall stability as seen in all three stability margins. The sensitivity of the MFFSM to top-heaviness is indicated by the increase in stability from the starting position even though the robot is at the same height at both times. During the fourth motion phase, as the robot moves laterally, the MFFSM indicates a higher sensitivity to lateral movements of the robot which is a common reason for the tipping over of multi-legged robots. Figure 5.12 shows that under specific situations, the MFFSM is more conservative than the FASM and vice versa.

**Simulation Results: MFFSM Sensitivity**

The second simulation was executed to demonstrate the sensitivity of the MFFSM compared to FFSM. Following the same motion phases indicated in the simulation setup, considering dynamic motion of the system as well, the motion scenario for this simulation is characterized by the motion profile graph in Figure 5.13.

Figure 5.14 shows the stability margin of the system during the motion calcu-
Figure 5.11: Four motion phases described in the simulation setup.

Figure 5.12: The results of stability simulation comparing the FFSM, MFFSM and FASM.
Figure 5.13: The motion scenario for dynamic stability simulation.

Figure 5.14: The results of the numerical dynamic stability simulation for both the loaded and non-loaded scenarios.
lated using FFSM and MFFSM. Two scenarios were carried out to demonstrate the sensitivity of the MFFSM. The initial scenario executed the motion of the robot without having the robot pick up the cylindrical rod. The second scenario had the robot pick up the cylindrical rod. In the loaded scenario, the external force acting at the gripper due to the rod weight was assumed to be $F_z = -1 \text{ N}$ which causes the same force acting at CG plus an external moment of $M_y = 0.5 \text{ N.m}$ acting about CG. Figure 5.14 shows the results of the numerical simulation for both the loaded and non-loaded scenarios.

During the first motion phase, $0 \leq t \leq 18 \text{ sec}$, the robot does not have any lateral motion in the $x$ or $y$ directions. Although the foot force magnitudes change due to the dynamics of motion, the foot forces are evenly distributed at each instant. With an even distribution of foot forces, the FFSM remains static during the first motion phase. The smooth curved changes in the MFFSM are due to changes in foot placement relative to the CG, height, and top-heaviness of the robot. As the robot descends, the net force from the foot forces decreases due to partial free fall, the foot placement vectors are reduced due to closer proximity of the feet to the CG, and the overall height decreases. A negative change in height causes an increase in the MFFSM, a negative change in the net foot force decreases the MFFSM and a negative change in the distance between CG and the foot placements decreases MFFSM. However, as shown in Figure 5.14, the MFFSM increases which shows that the effect of the height dominates that of foot placement and top-heaviness in this situation. As the robot starts to decelerate its descent, there is a positive acceleration that adds a positive effect of top-heaviness into the MFFSM. As the robot nears the final height, the deceleration decreases. Hence, the parabolic effect in the MFFSM when $9 \leq t \leq 18 \text{ sec}$. The change in the MFFSM validates the desired sensitivity to the height, foot placement spread (distance to CG), and top-heaviness.
During the second motion phase, $18 \leq t \leq 19 \text{ sec}$, the cylindrical metallic rod is lifted in the loaded scenario and ignored in the non-loaded scenario. In the loaded case, the weight and moment about the CG increases which causes an uneven FFD. With an uneven foot distribution, the FFSM decreases sharply while with the MFFSM, the effect of increasing the weight (top-heaviness) dominates over the increasing moment. Hence the sharp increase in the MFFSM. In the non-loaded case, there is no change in weight and the MFFSM does not increase which again validates the sensitivity of MFFSM to top-heaviness.

During the third motion phase, $t > 19 \text{ sec}$, the robot moves in the x direction. In the loaded case, the gradual effect of increasing moment and foot placement dominates that of top-heaviness and cause the MFFSM to decrease with a higher rate than the FFSM. The same event happens in the non-loaded case. However, without the added weight, the MFFSM and FFSM appropriately indicate a higher stability at the time where the loaded case reached instability. In both scenarios, the FFD become increasingly irregular as a result of the lateral motion in the x direction. The non-loaded case hits instability at 33 seconds while the loaded case hits instability at 28 seconds. As expected, both the FFSM and MFFSM reach instability at the same time in both scenarios.

In reality and by definition, the FFSM and MFFSM cannot take on negative values. However, since the FFD algorithm can provide negative foot force values, Figure 5.14 shows negative values for the FFSM and MFFSM.

The simulation comparing the loaded and non-loaded scenarios validates the presented criterion. As expected, the MFFSM is sensitive to height, foot placement (distance to CG), and top-heaviness and both the FFSM and MFFSM indicate instability at the same time. As previously mentioned, for a comprehensive result and control of the stability of the robot, both the FFSM and MFFSM should be
interpreted simultaneously. Having both FFSM and MFFSM data can mitigate control decision which may include changing foot placements, height, and/or net force to increase the stability and decrease the tip over potential of the system.

5.3.2 Experimental Validation of FFSM and MFFSM

The experimental validation of the robot is done in both flat terrain and irregular terrain.

Flat Terrain

The same motion phases as were carried out during the simulation were applied to the physical robot. During the motion, the sensor data was measured and the FFSM and MFFSM were calculated. The gripper is located 460 mm in the x-axis and the robot has an initial height of 100 mm. During the first motion phase, the robot drops to 73 mm. During the third motion phase, the robot rises to 115 mm. Figure 5.15 shows the actual robot during different phases of the experiment. The mass of the robot is 45 N and the mass of the cylindrical rod is 9.5 N. Figure 5.16 shows the results of the experiment overlaid with the new simulation results.

Based on the definition, the extent of the FFSM is between zero and one, but the MFFSM can be any positive number based on the geometrical and top-heaviness characteristics of the system. During the experiment, the first configuration of the robot, the home position, was used as the basis for geometrical and top-heaviness sensitivity. Hence, the MFFSM and FASM for \( t = 0 \) sec are given as 1. During the first phase of the motion, \( 0 \leq t \leq 9 \) sec, the MFFSM and FASM are increasing due to the decreasing height of the robot. This is reverse in the third phase for \( 10 \leq t \leq 24 \) sec. However, in both phases, the FFSM is constant since no moments are applied to the robot. The experimental results follow the simulation results closely.
Figure 5.15: The time snapshots at the experimental scenario.
Figure 5.16: The experimental validation of FFSM and MFFSM compared to FASM.

with a few fluctuations due to sensor drift. Both the simulation and experimental results predict the instability occurrence accurately. The rapid decrease in stability at the 10th second mark is due to added moment being applied to the CG from picking up the metallic cylindrical rod. From Figure 5.16, the robot stability is more sensitive to the lateral movement during the final motion phase, $24 \leq t \leq 34$ sec, compared to the vertical motion phase, $10 \leq t \leq 24$ sec, which is true as a result of increasing the moment about the tip over axis due to the lateral motion. That is the reason for a static FFSM during the third motion phase, $10 \leq t \leq 24$ sec.

In general, from Figure 5.16, the FASM is behaving more conservative than FFSM and MFFSM. However, they all follow similar behaviors when encountering different situations and they all go to instability at the same time. This can be considered as a validation for FFSM and MFFSM.
Irregular Terrain

A terrain which is non-flat and/or non-horizontal is considered to be irregular. The terrain shown in Figure 5.17 is an example of irregular terrain. In this experiment, a 3-legged robot is placed on the irregular terrain with two legs on a flat/horizontal surface and one leg on the adjustable inclined plane. The maximum stability occurs when the foot forces are all the same, which occurs only, under no external force and moment, if the plane is adjusted to be flat and horizontal. When the plane is inclined, the CG of the robot shifts and more force will be applied by the feet onto the flat surface. At the same time, the stability of the system decreases until the robot tips over, which occurs when the foot force of the leg on the inclined plane becomes zero and all the weight of the robot is sustained by only those legs on the level ground. The same scenario is true if two legs are on the inclined plane and one is on the flat plane. However, it should be noted that it is possible to make all three foot forces equal with the help of an external force and moment and achieve maximum stability even with an inclined plane. Figure 5.17 depicts the simulation and experimental robot and Figure 5.18 shows the result and comparison of the stability calculation for FFSM and FASM, both theoretically and experimentally. Figure 5.18 shows that FASM is more conservative than FFSM, but they react similarly as the slope of the plane increases and they go to instability at the same time.
even with an inclined plane. Fig. 14 depicts the simulation and external force and moment and achieve maximum stability possible to make all three foot forces equal with the help of an one is on the flat plane. However, it should be noted that it is similarly as the slope of the plane increases and they go to FASM is more conservative than FFSM, but they react both theoretically and experimentally. Fig. 15 shows that comparison of the stability calculation for FFSM and FASM, robot is sustained by only those legs on the level ground. The on the inclined plane becomes zero and all the weight of the over uneven terrain.

Figure 5.17: The stability scenario of a robot over irregular terrain for both simulation and experiment.

Figure 5.18: The simulated and experimental stability results for FFSM and FASM over uneven terrain.
5.4 Bio-Inspired Reactive Stability Control using FFSM

It is necessary for a legged walking/manipulating robot to be stable during its walking or manipulation process. The possibility of losing the stability will increase when using legged robots in unknown environments with uneven terrain or under unknown external stimuli. In such cases, the robot should be able to compensate for the loss of stability.

For this purpose, a bio-inspired reactive control strategy is developed that mimics the way animals or humans prevent from falling. If an animal is pushed or pulled, it will resist against the external force at the beginning by increasing the torques and forces in its legs’ joints and/or changing the position of its body. However, when it cannot resist anymore, it will then move its leg backward or forward to a new location. The direction of the leg motion depends on the direction of pulling or pushing. Hence, if the robot cannot resist against the external force, then the only way of compensating for the loss of stability is to move the legs accordingly. The reactive control strategy can be divided into the following five steps:

1. *Stability Measurement and Prediction*: A stability criterion needs to be defined for the controller in order for the robot to have a sense of stability to be able to measure the stability of the robot and predict when it is going to lose its stability. For this step, the problem is to choose a proper stability margin. In fact, theoretically, one can take advantage of any of the existing stability margins for this purpose, but in practice, the need for calculating the stability of the robot at any instant causes high frequency calculations within the controller which requires concise stability margin with low calculation cost and low sensor input information. Also, the stability margin should be able
to present a quantitative stability extent which measures how close or far the robot is to the unstable or the maximum stable state. For these reasons, in this work, FFSM is chosen for the first step since it satisfies these needs.

2. *Force Direction Detection*: After the robot has an understanding of stability, in case of losing stability, the robot needs to understand which way the force is applying to decide accordingly. This is essential for making the best decision and recovering to the maximum possible stability level. For this step, the direction of the force(s) are determined based on the distribution of the robot foot forces. In a multi-legged robot, conceptually, the direction of an external force is always towards between those two legs with the maximum amount of foot force. Therefore, for finding the direction of the external force, the maximum foot forces should be realized. This can be done by measuring the foot forces using force sensors embedded in the foot.

3. *Deciding which Legs to Move*: The leg(s) with the maximum foot force(s) are supposed to move. For this purpose, a threshold can be defined based on the magnitude of all the foot forces. If any of the foot forces falls below the threshold, that leg must be displaced.

4. *Calculating the Legs Motion Direction and Amount of Displacement*: After the footholds which need to be displaced are chosen, the direction of the leg motion is determined based on the direction of the external force, and the amount of displacement is based on the magnitude of the measured foot force.

5. *Reaction*: Each leg follows a planned trajectory to reach the new foothold.

Sometimes the robot is limited to put its feet within a certain area because of space limitation and is unable to spread its feet out as much as desired. In this case,
if the robot is under external stimuli and the only way for stability recovery is to change the position of CG. However, the direction and amount of the CG motion depends on how the foot forces are distributed. The distribution of foot forces can be used to present a general strategy for body CG motion to react against external stimuli. The concept is the same as described for leg motion. An experiment to validate the bio-inspired reactive stability strategy is discussed in the next section.

5.4.1 Control Strategy

The reactive control strategy considered for walking and/or manipulation of the robot is according to Figure 5.19. As shown in Figure 5.19, the first step is to do path planning to figure out every single discretized path of the robot for walking and/or manipulation. Therefore, to control the robot, one should be able to control and keep the robot in the desired position of CG and orientation of the platform in each discretized path. If the robot is manipulating and not walking, this position and attitude can be controlled using the inverse kinematic algorithm presented in section 3.2. If it walks as well, legs should be divided to supporting and transferring legs which switch their roles. Then, at each interval, CG position and attitude can be done using the same inverse kinematic algorithm for supporting legs. On the other hand, transferring legs should follow a planned path after selecting their next foothold. this needs a gait and foot trajectory planning at the same time. However, to choose the best foothold, two criteria should be satisfied, workspace and stability. The robot should make sure that the robot will have proper stability and will not tip over while making sure that the desired workspace for manipulation purposes is satisfied since the workspace of the HWR strongly depends of the geometry of footholds. The stability controller will use foot forces of supporting legs and calculate the FFSM to make sure the robot is stable. This is done using measured foot forces.
Then, using D-H parameters the foot trajectory is done and robot can move in that specific interval while maintaining proper workspace and stability.

5.4.2 Implementation

For the implementation of the reactive control, it is assumed that the workspace is not a concern when reacting. Therefore, the controller only considers the stability when reacting. Based on the presented control strategy, a threshold of reactive stability was defined based on FFSM and a fuzzy logic controller was written within C++ and sent to the robot. Whenever the stability of the robot (FFSM) falls under the threshold \( S = 0.5 \) in this example), the robot will react to compensate for the lost stability and to recover itself to a more stable situation. The robot was pulled and pushed randomly in different directions with arbitrary forces and the robot reacted well for all of the situations. Figure 5.20 shows how the robot responded to different loading situations.
Figure 5.19: The reactive control schematic for hexapod walking and manipulation.
Figure 5.20: The experimental implementation of the reactive stability.
Chapter 6

Conclusion and Future Work

6.1 Conclusion

This research is motivated by the need for mobile machining systems to remove humans from hazardous and inaccessible environments. The research analyzed the kinematics, workspace, and stability requirements for mobile machining systems based on hexapod walking robots. The major contributions of this dissertation are as follows.

- Developed an accurate and concise analytical inverse kinematic solution for legged robots. It was shown that the solution is applicable to any arbitrary position and orientation of the platform for both walking and manipulation. The conciseness, accuracy, and low calculation cost of the inverse kinematic solution make it applicable and more suitable for real-time controllers. Both simulation and experimental work validated the solution.

- Derived the analytical solutions to the lateral and spatial reachable workspace and constant orientation workspace of axially symmetric hexapod walking robots. For this purpose, a decomposition methodology was developed. The
workspace-based design was presented to show how the solution can be used to design and optimize the robot and its workspace. The solution is very useful for optimization of the design parameters to a prescribed workspace or space. The solution removes the numerical calculation costs for evaluating the workspace of the robot at each instant. Therefore, the workspace of the robot can be monitored in a real-time manner with a very low calculation cost. The concept of stable workspace was introduced to make sure that the robot remains stable while maintaining its required workspace.

- Developed the Foot Force Stability Margin and Modified Foot Force Stability Margin for determining the stability of the system based on the sensed normal foot forces. The FFSM was limited between zero and one for instability and maximum stability of the robot, respectively. The modified version of the margin, MFFSM, was developed and validated to take into consideration the effect of geometry and top-heaviness. By interpreting both FFSM and MFFSM simultaneously within the controller, a complete information on the stability of the robot can be obtained. Several simulations and experiments were done and the results were compared to FASM criterion to validate the accuracy and efficiency of FFSM and MFFSM. The results showed that FFSM, MFFSM, and FASM react similarly and go to instability at the same time. However, under some conditions, one may be more conservative than the other. FFSM needs lower input information comparing to FASM and thereby, it has lower calculation cost which makes it more suitable for real-time reactive stability controller.

- Developed a bio-inspired reactive stability control strategy that utilized the foot force stability margin. To do so, a bio-inspired reactive stability method
was developed to help legged robots to recover from loss of stability and remain stable against unknown external stimuli instead of tipping over. An experiment was conducted to validate the presented bio-inspired reactive control strategy.

6.2 Future Work

It was shown that once the structural parameters of a HWR are selected, the workspace and physical size of the robot will be determined and fixed and the robot cannot satisfy further workspace or physical size requirements. Therefore, to overcome the shortage of inflexible size and workspace of the robot, a scalable hexapod walking robot will be developed for future work as shown in Figure 6.1. The scalable hexapod walking robot includes articulated-extendable legs capable of changing its size and workspace according to the need of the environment. Therefore, it will have a wider workspace for manipulation purpose and will be able to walk with different steps and thereby different speeds. It can do an optimized adjustment for walking and manipulating and the best adjustment for minimum energy usage. The robot can also be used for different unpredicted tasks and environments. The robot can be used to further validate the developed methodology in this thesis with regards to workspace, stability, and integration of them for real-time control.
Figure 6.1: The designed scalable hexapod walking robot for prototype.
Bibliography


