

Trajectory Planning for a Six-Legged Walking Robot and its Experimental Validation

*Thesis submitted in partial fulfilment
of the requirements for the degree of*

Master of Mechanical Engineering

BY

BHASKAR GUIN

Examination Roll No: M4MEC19017.
Registration No.: 140874 of 2017-18.

UNDER THE GUIDANCE OF

**Dr. Nipu Modak
&
Prof. Tarun Kanti Naskar**

**DEPARTMENT OF MECHANICAL ENGINEERING
FACULTY OF ENGINEERING & TECHNOLOGY
JADAVPUR UNIVERSITY
KOLKATA – 700032
MAY 2019**

**FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY**

CERTIFICATE OF APPROVAL*

This foregoing thesis is hereby approved as a credible study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

COMMITTEE

ON FINAL EXAMINATION FOR

EVALUATION OF THE THESIS

***Only in case the thesis is approved**

**FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY**

CERTIFICATE OF RECOMMENDATION

*I hereby recommend that the thesis presented under my supervision by **Mr. Bhaskar Guin** entitled “**Trajectory Planning for a Six-Legged Walking Robot and its Experimental Validation**” be accepted in partial fulfillment of the requirements for the degree of **Master of Mechanical Engineering**.*

Countersigned:

Head of the Department
Department of Mechanical Engineering

Dr. Nipu Modak
(Thesis Supervisor)

Dean
Faculty of Engineering and Technology

Prof. Tarun Kanti Naskar
(Thesis Supervisor)

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC ETHICS

I hereby declare that the thesis contains literature survey and original research work by the undersigned candidate, as a part of his *MASTER OF MECHANICAL ENGINEERING* studies. All information in this document have been obtained and presented in accordance with the academic rules and ethical conduct. I also declare that, as required by these rules of conduct, I have fully cited and referenced all the material and results that are not original to this work.

Name: **BHASKAR GUIN.**

Examination Roll Number: **M4MEC19017.**

Class Roll Number: **001711202014.**

University Registration No: **140874 of 2017-18.**

Thesis Title: ***“Trajectory Planning for a Six-Legged Walking Robot and its Experimental Validation”.***

Signature with Date.

ACKNOWLEDGEMENT

I would take this opportunity to convey my sincere gratitude and appreciation for my supervisors **Dr. Nipu Modak** and **Prof. Tarun Kanti Naskar**. I am indebted to them for their methodical guidance, motivation and constant support they have given me throughout the coursework. They have identified my potentials and kept faith in me. It is because of their confidence that I was able to successfully accomplish the targets. I am extremely thankful to them, without them my thesis could not have reached this state.

A word of thanks to the technical staff in the Machine Elements laboratory for their help and technical assistance. A special thanks to all my seniors for their advice and support.

I would also like to convey my thanks to the Head of the Department, Faculty members and all Academic and Technical staffs of Mechanical Engineering Department, Jadavpur University.

Finally, I acknowledge the countless efforts of my parents. They have stood by me through thick and thin, and I am forever indebted to them.

Date: -----

(Bhaskar Guin)

Table of Contents

	Page no.
Certificate of approval	ii
Certificate of Recommendation	iii
Declaration of Originality and Compliance of Academic Ethics	iv
Acknowledgement	v
Table of Contents	vi
List of Nomenclatures	x
List of Figures	xi
List of Tables	xvi
1. Introduction and Literature Review	1-12
1.1. The Family of Walking Robots	2
1.2. Advantage and Disadvantage of Walking Robots	3
1.3. Literature Review	4
1.3.1. Study of Gait Patterns	4
1.3.2. Kinematic and Dynamic Analysis	6
1.3.3. Terrain Adaptation	7
1.3.4. Hybrid Designs	8
1.3.5. Obstacle Avoidance	9
1.3.6. Trajectory Optimization	9

1.4. Gap Finding	11
1.5. Aim and Objective	12
2. The Hexcrawler – a six legged Robot	13-25
2.1. Technical Specification	14
2.2. Upgrading the Control System	15
2.3. Design of a Tracing Attachment	18
2.4. Calibration of the Modified System	19
2.4.1. Determining p_d (pulse/degree)	19
2.4.2. Equalising Foot-Tip Pressure	21
2.5. Conclusion	25
3. Gait Patterns for the Hexcrawler	26-37
3.1. Overview of Periodic Gaits	26
3.2. Tripod Gait	27
3.3. Ripple Gait	31
3.4. Wave Gait	33
3.5. Result and Discussion	36
3.6. Conclusion	37
4. Leg Mechanism – Kinematic Model	38-48
4.1. Description	38
4.2. Analysis using DH parameters	39
4.3. Generating Leg Workspace	45

4.4. Conclusion	48
5. Kinematic Analysis of the Hexcrawler	49-77
5.1. Motions of the Hexcrawler	49
PART – I. Mathematical Relations	
5.2. Frames of Reference	50
5.3. Inter-relationship between $\{B\}$, l_i and f_i	51
5.3.1. Calculating l_i when $\{B\}$ is defined	52
5.3.2. Calculating f_i when l_i is defined	52
5.3.3. Calculating l_i when f_i is defined	54
5.3.4. Calculating $\{B\}$ when l_i is defined	54
5.3.5. Estimating $\{\theta_{hi}, \theta_{vi}\}$ when l_i and f_i is defined	55
PART– II. Forward Kinematics	
5.4. Analysis of Tripod Gait	56
5.5. FK Solver Algorithm	58
5.6. Analysis of Stance Phase	58
PART – III. Inverse Kinematics	
5.7. IK Solver Algorithm	64
5.8. Results and Discussion	66
5.9. Conclusion	77
6. Complex Trajectories – planning and robot guidance	78-98
PART – I. Trajectory Construction.	
6.1. Problem Formulation	79

6.2. Parametric Trajectories	79
6.2.1. Definition of a Bezier Curve	81
6.2.2. Joining Two Bezier Curve Segments	82
6.2.3. Subdivision of Bezier Curves	82
6.2.4. Computation of Arc Length	83
6.2.5. Curve fitting using Bezier Curves	84
PART – II. Robot Guidance.	
6.3. IK Solver for Parametric Trajectories.	87
6.4. Results and Discussion	88
PART - III. Trajectory Optimization.	
6.5. Need for Optimization	90
6.6. Evaluating Time-Dependent Kinematic Parameters	90
6.7. Minimizing Jerk	92
6.8. Results and Discussion	93
6.9. Conclusion	98
7. Conclusion and future scope of work	99-100
7.1. Conclusion	99
7.2. Future scope of work	100
List of references	101-108

Nomenclature

p_d	pulse/degree
θ_i	Rotation angle in degrees
pw	Pulse width
L_i	Link lengths in mm
ξ_i	Fixed angles in a linkage
$\{B\}$	Body Frame
l_i	Leg pivot position
f_i	Foot-tip position
x_i, y_i, z_i	Position coordinates in 3D space
γ	Yaw angle
d_i	Length of the vector joining $\{B\}$ with l_i
λ_i	Orientation of the vector joining $\{B\}$ with l_i measured w.r.t $\{B\}$
g, f, c	Coefficients of generalised circle equation
J	Bernstein basis
B_i, C_i, D_i	Vertices of the control polygon
P	Point on a Bezier curve
L_B, L_P, L_C	Arc length, perimeter of control polygon, chord length
u	Parameter of Bezier curve
t	time
s, v, a, j	Displacement, velocity, acceleration, jerk.

List of Figures

	Page No.
1. Fig 1.1. Classification of Robots, with pictures of a few selected types of robots.	1
2. Fig 1.2. (a) Asymmetrical, (b) Omnidirectional 6 legged robot.	2
3. Fig 1.3. Advantage of walking robots when traversing through an uneven terrain.	3
4. Fig 2.1. The Hexcrawler robot, showing the 3 major segments as viewed from various angles.	13
5. Fig. 2.2. Original Controller Setup with Parallax Boards.	15
6. Fig. 2.3. Components of Controller System with wire connections.	17
7. Fig. 2.4. Mounting Tray for (a). Arduino Microcontroller and (b). Servo Controller – designed in Solidworks 2018, (c) Controller Setup after installation.	18
8. Fig. 2.5. Tracing Attachment – (a) sectional view, (b) rendered image and(c) after installation on the Hexcrawler.	19
9. Fig. 2.6. (a) Setup and Measurement of rotation angles. Recording Pulse at (b) 0°, (c) +30°, (d) -30° rotation of the legs.	20
10. Fig. 2.7. Plots of θ_{hi} Vs (pw) from the Calibration Data obtained in Table 4. (a) for Right Sided Legs and (b) for Left Sided Legs.	22
11. Fig. 2.8. (a) Hexcrawler Robot in Standing Posture (b) Hexcrawler Robot in Squatting Posture.	23
12. Fig. 2.9. Electronic circuit for interfacing Flexiforce pressure sensor.	24
13. Fig. 2.10. (a) Setup for equalising the Foot-Tip pressure, (b) Flexiforce Pressure Sensor placed between the glass surface and Foot-Tip.	25

14.	Fig. 3.1. Stability criteria illustrated for instantaneous position of the Tripod Gait. The Grounded Foot-Tips form the vertices of the support polygon, and the CG is projected on the plane of the support polygon. The stability margin is marked ensuring the Hexcrawler is statically stable for this particular position.	27
15.	Fig. 3.2. Two possible position of Legs, with their corresponding angles.	28
16.	Fig. 3.3. Stepping Sequence of Native-Tripod Gait.	29
17.	Fig. 3.4. Stepping Sequence of Spinning-Tripod Gait	30
18.	Fig. 3.5. Stepping Sequence for Half-Step Tripod Gait.	30
19.	Fig. 3.6. Five possible position of Legs, with their corresponding angles.	31
20.	Fig. 3.7. Steps per cycle of Ripple Gait for Straight or Curved path walking.	33
21.	Fig. 3.8. Six possible position of Legs, with their corresponding angles.	34
22.	Fig. 3.9. Steps per cycle of Ripple Gait for Straight or Curved path walking.	35
23.	Fig. 3.10. Experimentally obtained Trace Curves of Tripod, Ripple and Wave Gaits for straight walking, with programmed actuation as mentioned in Table 3.1.	36
24.	Fig.4.1. (a) Front view of the Leg mechanism showing the position of Servo-1 & 2. (b) Schematic Line Diagram drawn neglecting Servo-1 to obtain a planar mechanism.	39
25.	Fig. 4.2. Decomposing the planar mechanism into 2 loops and 1 serial linkage representing the loops (a) P1-A-B-P2-P1 and (b) P2-C-D-P3-P2 respectively, (c) represents the serial linkage D-E-F.	40
26.	Fig. 4.3. Splitting Loop-1 into two serial linkage segments, P1-A-B-P2 and P1-P2.	41

27.	Fig. 4.4. Splitting Loop-2 into two serial linkage segments, P2-C-D-P3 and P2-P3.	42
28.	Fig. 4.5. Simplified Serial Linkage for the planar Leg Mechanism.	43
29.	Fig. 4.6. 3D position vector joining the leg-pivot and foot-tip, obtained by rotation about the z-axis by an angle θ_h .	45
30.	Fig. 4.7. 3D plot of leg workspace, showing all possible positions of the foot-tip within the actuation limits $\theta_h \in [-80^\circ, +80^\circ]$ and $\theta_v \in [-30^\circ, +30^\circ]$.	46
31.	Fig.4.8. (a) 3D plot of Leg workspace considering all 6 legs, (b) Top-view of leg-workspace showing the non-intersecting nature of the workspaces within the working limits of θ_h and θ_v .	47
32.	Fig. 5.1. Six possible motions of the Hexcrawler.	49
33.	Fig. 5.2. Location of the Body Frame $\{B\}$ with respect to the Earth fixed Frame $\{W\}$.	51
34.	Fig. 5.3. Hierarchy of relationship between $\{B\}, l_i$ and f_i .	52
35.	Fig. 5.4. Locating leg pivots l_i when $\{B\}$ is defined.	52
36.	Fig. 5.5. Sign conventions of θ_{hi} and θ_{vi} for Right and Left Sided Legs. (a). Schematic top-view of Robot body viewed from the rear side, showing the leg at 3 different positions for both right and left sided legs. (b). Front-view of Robot body viewed from the front side showing the crank at 3 different positions for both right and left sided legs.	53
37.	Fig. 5.6. Estimating Yaw angle three known positions of the leg-pivot.	54
38.	Fig.5.7. Torque distribution diagram of legs 1,3,5 having equal actuation $\theta_{h1} = \theta_{h3} = \theta_{h5}$. The initial and final leg positions are shown for the stance phase.	60
39.	Fig. 5.8. Cases arising due to unequal angular displacements, showing 3 distinct segments (i), (ii) and (iii) in each case. $\Delta\theta_{h1} > \Delta\theta_{h5}$, and $\theta_{h1} _{initial} > \theta_{h5} _{initial} > \theta_{h5} _{final} > \theta_{h1} _{final}$ $\Delta\theta_{h5} > \Delta\theta_{h1}$, and $\theta_{h5} _{initial} > \theta_{h1} _{initial} > \theta_{h1} _{final} > \theta_{h5} _{final}$	61

40.	Fig. 5.9. Dragging of Leg-5 caused when Leg 1,3 is actuated, but Leg-5 remains unactuated (acting as a rigid link), as in Segments (i) and (iii) of Case 2.a.	61
41.	Fig. 5.10. Torque distribution diagram of legs 1,3,5 for spinning motion. The initial and final leg positions are shown for the stance phase.	63
42.	Fig. 5.11. Utility of keeping legs at 0° for starting position- allows convergence to either Standard Walking or Spin Mode.	66
43.	Fig. 5.12. (a) Simulating a straight line path using FK Algorithm. (b) Experimental verification of Straight line walking using Tripod Gait.	67
44.	Fig. 5.13. (a) Simulating a Circular path using Spinning FK Algorithm, (b) Experimental verification of Spinning using Tripod Gait.	68
45.	Fig. 5.14. (a) Simulating a circular path using FK Algorithm as per inputs in Case 3.a, (b) Experimentally obtained Trace curve obtained from experiment using the actuation values in Case 3.a.	70
46.	Fig. 5.15. (a) Simulating a circular path using FK Algorithm as per inputs in Case 3.b, (b) Experimentally obtained Trace curve obtained from experiment using the actuation values in Case 3.b.	71
47.	Fig. 5.16. (a) Superimposing simulation curves obtained in Fig. 5.14 (a) and 5.15 (a) (b) Superimposing the trace curves obtained in Fig. 5.14 (b) and 5.15 (b).	72
48.	Fig. 5.17. (a) Proposed straight line trajectory with knots at 30mm intervals, (b) Proposed straight line trajectory with knots at 20mm intervals, (c) Experimentally obtained Trace Curves obtained by programming the results obtained in Table 5.4 and 5.5.	74
49.	Fig. 5.18. (a) Proposed circular trajectory with knots at 5° intervals, (b) Experimentally obtained Trace curve obtained by programming the results obtained in Table 5.6.	76

50. Fig. 6.1. Trade-off between Perfect Fit and Distortion. 86
 (a). Curve fit using 3rd degree showing that it does not pass through all target points, (b) Curve fir using 5th degree showing that it passes through all target points, (c) Increasing degree of fitted curve results in localized distortion.
51. Fig. 6.2. Hexcrawler maneuvering along a complex trajectory. 87
52. Fig. 6.3. Complex Path Planning and Robot Guidance using Parametric Curve, (a) Parametric Curve generation that fits the target points, (b) Proposed sinusoidal trajectory with knots at 0.02 parametric interval. (c) Trace Curve obtained by programming the robot with joint angles estimated by the IK Solver. 89
53. Fig. 6.4. Displacement-Time function after normalization. A constant velocity profile with sinusoidal blends at start and end. 91
54. Fig. 6.5. Plots of Case Study 1, (a) Curve fitting using 6th degree Bezier curve, (b) Displacement Vs Time, (c) Velocity Vs Time, (d) Acceleration Vs Time, (e) Jerk Vs Time, (f) Trace Curve generated from disp-time function. 94
55. Fig. 6.6. Plots of Case Study 2, (a) Curve fitting using 6th and 4th degree Bezier curve segments, (b) Displacement Vs Time, (c) Velocity Vs Time, (d) Acceleration Vs Time, (e) Jerk Vs Time, (f) Trace Curve generated from disp-time function. 96

List of Tables

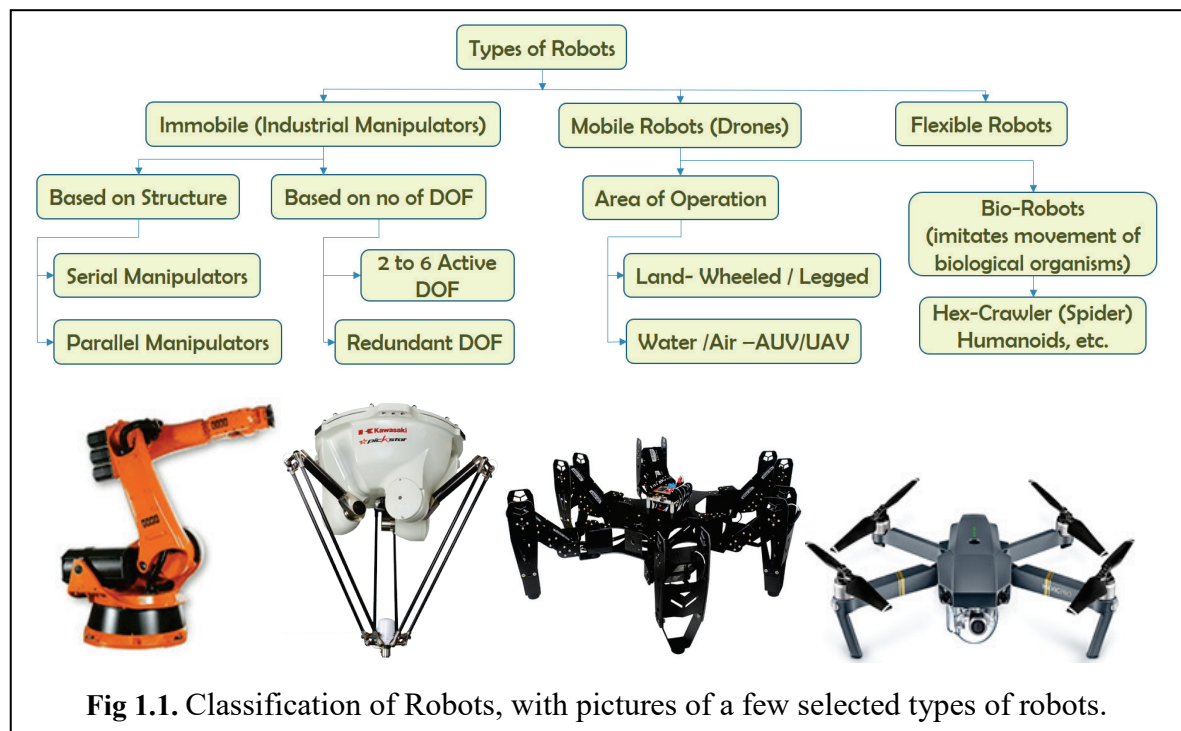
	Page No.
1. Table 1.1. Walking robots based on the number of legs.	2
2. Table 2.1. Technical Specifications of Hexcrawler Robot.	14
3. Table 2.2. Comparison Chart of Parallax vs Arduino.	16
4. Table 2.3. Wiring and Pin-connections.	17
5. Table 2.4. Horizontal Servo Calibration.	21
6. Table 2.5. Vertical Servo Calibration Results.	25
7. Table 3.1. Actuation Details to obtain the Trace curves for 3 fundamental gaits.	36
8. Table 4.1. D-H parameters for serial linkage $P1-A-B-P2$.	41
9. Table 4.2. Joint parameters for serial linkage $P1-P2$.	41
10. Table 4.3. D-H parameters for serial linkage $P1-A-B-P2$.	42
11. Table 4.4. D-H parameters for serial linkage $P1-P2$.	42
12. Table 4.5. D-H parameters for simplified serial linkage.	44
13. Table 5.1. Types of motion of the Hexcrawler.	50
14. Table 5.2. Values of d_i and λ_i (obtained from geometry of the Hexcrawler).	52
15. Table 5.3. Step Sequence per cycle for Tripod Gait.	57
16. Table 5.4. Cases that may arise with the values of θ_{h1} and θ_{h5} .	59
17. Table 5.4. Results of IK Solver for a single walking cycle with knots taken at 30mm intervals.	73
18. Table 5.5. Results of IK Solver for a single walking cycle with knots taken at 20mm intervals.	73
19. Table 5.6. Results of IK Solver for a single walking cycle.	75

20.	Table 6.1. Shape change of Bezier and B-Spline curves due to addition of Control Vertices.	81
21.	Table 6.2. Optimization Results for Case-1.	94
22.	Table 6.3. Optimization Results for Case-2.	95
23.	Table 6.4. Variation of Jerk with Blend Time.	97
24.	Table 6.5. Variation of Jerk with Actuator Limits.	97

Chapter 1.

Introduction and Literature Review.


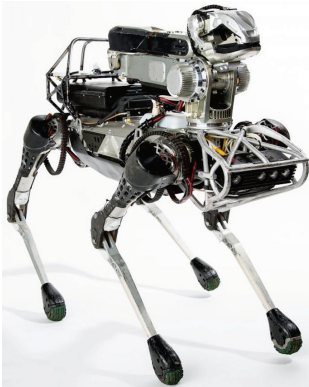

Robots have become an indispensable part of life in the 21st century. Intense research and development are being carried out in the field of robotics. Robotics is a multi-disciplinary subject evolving expertise from the fields of mechanical, electronics and computer science engineering. Hence, projects on robotics can be challenging. Robots were primarily developed as a vital part of industrial automation systems where they are commonly referred to as Industrial manipulators. However, researchers have realised that robots need not be confined to industries only. A new class of robots called Mobile robots have enormous potentials in various fields. A detailed classification of robots have been presented in Fig. 1.1. In the current work, detailed analysis of walking robots will be discussed.



1.1. The Family of Walking Robots

Often working environments are unhealthy for humans to work in, such as nuclear power plants, oil rigs, space explorations, etc. In these cases, robots can be designed to largely replace human beings. Walking robots can be used for these purposes. These are biologically inspired with 2 or more legs. More the number of legs greater is the stability though at the expense of mobility. Table 1.1 shows the different types of walking robots based on the number of legs. Further, six legged robots are of two types: asymmetrical and omnidirectional, shown in Fig. 1.2. In the current thesis work, 6-legged symmetrical robot is considered for analysis.

Table 1.1. Walking robots based on the number of legs:

2 Legged	4 Legged	6 Legged
Humanoid	Quadruped	Hexapod
Honda Asimo	Boston Dynamics – Robot Dog	NI - Hexapod
		

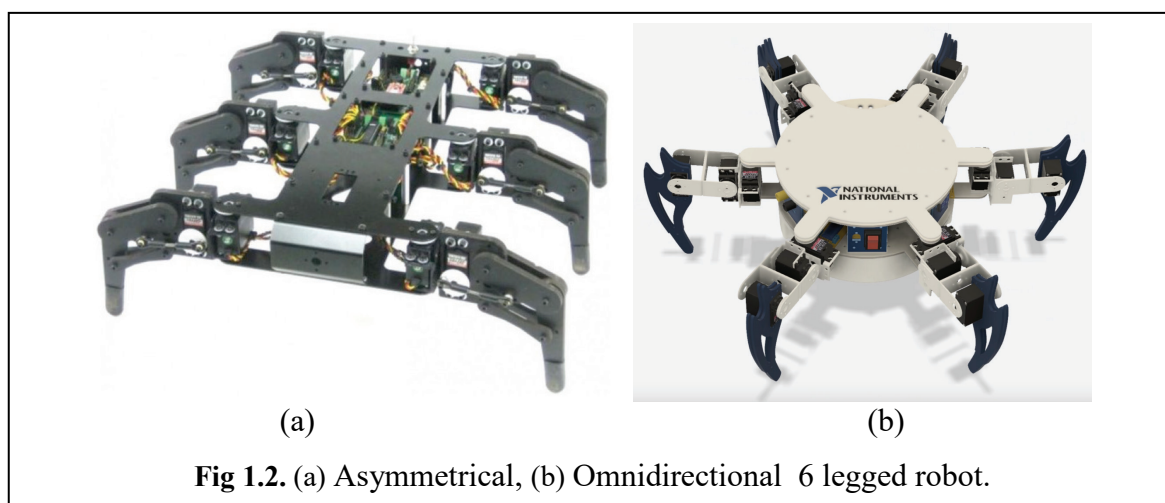


Fig 1.2. (a) Asymmetrical, (b) Omnidirectional 6 legged robot.

1.2. Advantages and Disadvantages of Walking Robots

The main advantage of legged robots is their ability to access places impossible for wheeled robots. Wheeled robots require a smooth terrain, while legged robots can travel on irregular surface using suitable selection of footholds. The legs can be equipped with active suspensions which would damp out any shocks. The robot platform can be held at a constant roll and pitch angle irrespective of the undulations of the terrain, shown in Fig.1.3. This property is beneficial for material handling operations. The legged robots having higher degree of freedom, have better obstacle avoidance capability, while some of them can even jump over obstacles.

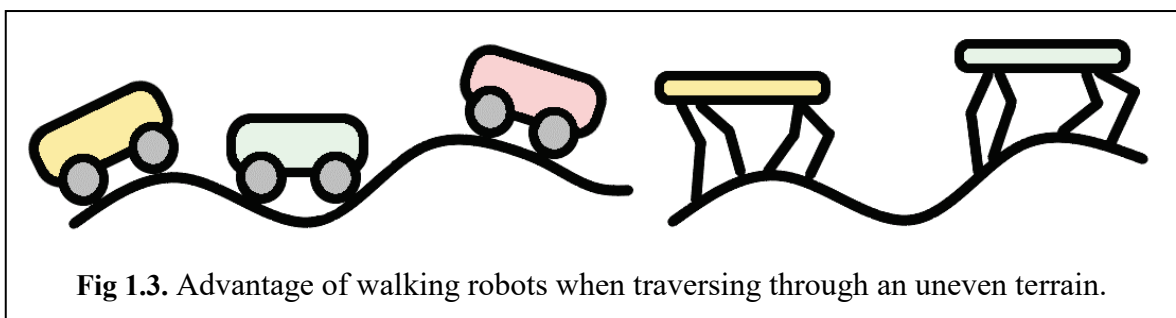


Fig 1.3. Advantage of walking robots when traversing through an uneven terrain.

Many environments are inaccessible or hazardous for humans in which some sort of manipulation needs to be done. For example, handling of remaining debris after earthquake and fire, attempting search and rescue operations, inspection, repair, and maintenance in oil rigs, etc. Generally, these locations suffer from lack of appropriate visual, respiratory, and safety conditions for human workers. Therefore, it is more appropriate to use robots instead of humans in these environments. These robots are also being used for underwater and outer-space exploration. They are also used for specialised tasks such as landmine detection and removal, transportation and handling of radioactive and toxic chemicals, rescue and exploration missions.

However, walking robots also have some limitations. The control of legged robots is extremely complex. The payload capacity is much smaller than that of wheeled robots. Legged robots are also much slower than wheeled robots. However, the advantages overrule the limitations. Researchers try to impose smoother controls and develop new gait patterns so that these robots perfectly mimic the animal or human locomotion, such as walking, running, and jumping.

1.3. Literature Review

At the outset, it is necessary to review literatures on legged robots. Through this review, the earlier works by researchers in this field are studied. This review is beneficial for further improvements in the work and to introduce new developments (if possible) in the related field of study. This review has been subdivided into few broad categories.

1.3.1. Study of Gait Patterns

Hexapods are biologically inspired robots that mimic the locomotion characteristics of 6 legged animals (mainly insects). Each animal has their own distinct walking characteristics. Hence, the gait patterns are analysed and implemented by programming the robot to walk accordingly. McGhee [1] provides fundamental definition of key parameters pertaining to gaits of legged robots. Zhu et al. [2] has focussed on turning of a hexapod robot using tripod gait. They have introduced a gait-coefficient that controls the swing and support phases. By suitable selection of this gait-coefficient it is possible for the hexapod to follow a circular path. Further, the simulation and experimental results have been superimposed to show the maximum deviation. Wang et al. [3] used model based analysis of an omni-directional robot. They analysed the leg trajectories and body dynamics for periodic gaits. They simulated the joint kinematic parameters as a function of stride for different gaits. Burkus and Odry [4] designed a hexapod robot with each legs having 2 d.o.f. They simulated the leg trajectories and used hybrid-neuro fuzzy technique to estimate the best route in an unknown environment and to generate a suitable adaptive gait. Youcef and Pierre [5] used distributed Q-learning to instil the ability of self-learning. The robot maps the terrain and trains itself to obtain the optimal leg movements for precise trajectory tracking and maximum stability. The simulation and experiments have been performed for Tripod gait. Yang and Juang [6] used fuzzy controller to select the foothold position in swing mode. It uses Particle Swarm optimization to obtain a solution that guides the fuzzy controller. The proposed method has been tested on a hexapod robot by making it trace a circular trajectory and a rectangular path having a physical barrier on one side. Campos et al. [7] used central pattern generators to replicate the gait patterns for the Chira hexapod robot and the simulation results has been presented. The central pattern generators are special neural structures. Inagaki et al. [8] also used the CPG model to generate different gaits. The results show that by suitably controlling the energy of the oscillators it is possible to vary the walking speed. Lewis et al. [9] used genetic algorithm to alter the neural parameters instead of using a learning algorithm.

Soyguder and Alli [10] modelled a hexapod robot on the basis of kinematics and dynamics, and implemented it on walking and running gaits. Bounding gaits have been controlled using PD controller and using a spring loaded inverted pendulum leg model. Carbone and Ceccarelli [11] classifies legged robots on the basis of number of legs. It provides information regarding maximum speeds attainable for each class of robots and finally it gives an overview of gait patterns for multi-legged robots. Ding et al. [12] also provides mathematical description of gaits for six legged robots. The analysis of a gait resulting due to the loss of one or two legs has been presented in this paper. Parker and Mills [13] used cyclic genetic algorithms to develop metachronal gaits with the objective of maximizing stability. Deng et al. [14] optimized the leg trajectory for minimizing the power consumption of a hydraulically actuated leg. The proposed idea was that, minimizing the power consumption per leg would effectively reduce the overall power consumption for the robot. Deng et al. [15] formulated a real life problem of material transportation using hexapods. They considered the case when two limbs would be used to hold an object and the robot would move on the remaining 4 legs. A motion planner was developed which would estimate the required joint actuation values based on the trajectory it has to follow. Jin et al. [16] emphasised on minimizing the energy consumption of the robot by minimizing the joint torques. It has been suggested that, suitable selection of gait parameters like walking velocity, duty factor, and stride length could help reduce the energy consumption. Porta and Celaya [17] describes the gradual convergence of free gait to tripod gait. Free gaits are suited for rough terrain only, for, as the roughness increases the number of steps increases making the locomotion slower. However, as the roughness decreases the maneuvering speed increases and finally for smooth terrain the free gait converges to tripod gait. Moore et al. [18] developed an algorithm by which a 6 legged robot RHex was made to climb a flight of stairs. The power consumption in the process was recorded. Saranlı et al. [19] describes the mechanical structure and gait pattern controller for the RHex which is an under actuated six legged robot. Kecskés et al. [20] used particle swarm optimization technique to optimize the kinematic parameters - velocity and acceleration for straight walking. They compared it with the results obtained from genetic algorithm and the results of PSO were found to be superior. Zhang et al. [21] used cubic Bezier curves to construct the trajectory and guided a quadruped robot along the trajectory using free gait. Pal et al. [22] used wave gait to guide a robot on a rough terrain. They used a heuristic approach for foothold planning, and the result was a combination of free and wave gaits. A parallel heuristic method has been formulated with

graph search using A* algorithm, in [23]. Duan et al. [24] developed a new turning gait using tripod sequence, which would provide additional agility to avoid obstacles and walk on rough terrains. The proposed theory is supported by experimental results.

1.3.2. Kinematic and Dynamic Analysis

To manoeuvre a robot along a desired trajectory, its motion characteristics must be determined at first. The kinematics and kinetic (dynamic) analysis provides the guiding equations of motion of a robot. Goodwine and Burdick [25] formulated a basic kinematic model of a hexapod and performed a simulation for walking along straight and curved paths. Olaru and Nițulescu [26] modelled a rectangular hexapod with each leg having 3 d.o.f. A detailed leg workspace analysis has been presented to check for possible collision zones. Further considering the effect of gravity, simulation results have been shown where the robot walks on sloping surface and climbs stairs. Yang and Geng [27] provides an analytical solution to a parallel hexapod manipulator. Although it is not a mobile robot, yet the methodology is valuable from the aspect of the current thesis work. Silva et al. [28] studied the foot-tip interaction with the maneuvering surface. They modelled the surface conditions using spring damper assembly, and simulated the interaction characteristics for different gait patterns. The study of foothold forces is crucial, since it dictates the overall motion of the robot. In [29] they have used Genetic algorithm to obtain the optimum robot parameters when it is desired to maneuverer the robot at a specific speed using Wave Gait. Bo et al. [30] emphasises on the mechanical design and control system. It highlights the fact that for robots having legs of 2 d.o.f, only planar maneuvering is possible, and to travel on uneven terrain, legs with 3 d.o.f is required along with additional sensors. Straight walking using tripod gait has been tested for the designed hexapod. Tedeschi and Carbone [31] gives an overview of Hexapod robots. The paper addresses the key performance measurement parameters and discusses the basic periodic gaits in detail. Zhong et al. [32] considered legs actuated by linear actuators. The dynamics of the system was modelled using decentralised control strategy. The robot leg has been actuated using different control modes - PD, PID and the trajectory has been obtained using a dynamic model. Schneider and Schmucker [33] [34] identified that the foot-tip force is critical parameter in locomotion of legged robots. The designed legs were capable of sensing the foot-tip force and used it as a feedback to the controller. The force distribution patterns on different surface conditions was obtained by testing it on a hexapod robot. Netto et al. [35] formulated the inverse kinematics problem

and developed a solver using neural and fuzzy system that guides the robot in real-time. The reduction in computation time and accuracy in trajectory tracking has been highlighted. Skaburskyte et al. [36] studied the stability for 3 primary gaits- tripod, ripple and wave. Their observation was that instability occurs at the instance when supporting legs are interchanged, and the simulation results show that wave gait provides maximum stability. Lopez et al. [37] used DH-parameters to obtain the state space model of the leg mechanism of a hexapod robot and used it to simulate the leg trajectories. Orin et al. [38] is a novel literature showing the kinematic and kinetic analysis of an open chained mechanism, and has been implemented on a hexapod robot. It uses Newton-Euler approach for establishing the robot dynamics. Gonzalez et al. [39] developed algorithms for foothold selection by minimizing the energy consumption. They developed a model for the energy consumption per half gait cycle, and minimized it. The algorithm was simulated for hexapod robot travelling along a straight line trajectory at a constant speed using tripod gait. Mahapatra et al. [40] modelled a hexapod robot using Catia and used Adams software to simulate the robot using wave gait. With the objective of making the robot to follow a straight line path, the variation of joint torques and the movement of the center of mass were recorded. Zhu et al. [41] developed an inverse kinematic solver by coupling it with an energy consumption index and workspace feasibility. The solution was obtained by optimizing the problem using genetic algorithm. The simulation shows lowered joint torques along with minimum power consumption. Mahfoudi et al. [42] developed an inverse kinematic solver by optimizing the force distribution at the legs considering the effect of friction. Such a solution would provide superior manoeuvring over rough terrains by proper weight distribution and achieving greater stability.

1.3.3. Terrain Adaptation

The advantages of legged robots over conventional wheeled robots are that it can walk over rough terrains. This capability is not inherent, but depends upon the control algorithm. Isvara et al. [43] developed an effective algorithm which would guide a hexapod along a predefined trajectory on a rough terrain. The terrain adaptation is achieved using tactile sensor at the foot-tip and using it as a feedback to the controller. Song and Choi [44] developed a gait suitable for traversing an uneven terrain. They assumed that the legs will be placed on the footprints made by the legs ahead of them, in this way the complexities of foothold selection will be minimised. This type of gait was named as follow-the-leader gait. The stability has been verified using simulations. McGhee and Iswandhi [45] also concentrates on locomotion

over rough terrain in which they have considered dead-zones where the foot-tip cannot be placed. An adaptive algorithm decides where to place the foot-tip, and is based on free gait. Belter [46] developed an adaptive method to estimate the foothold for a hexapod robot when travelling on a rough surface. It uses terrain maps to generate a surface and searches for points on that surface which would provide maximum stability to the robot, and yet maintain its target path. The self-learning capability had been tested on a real robot. Hauser et al. [47] developed a foothold planner based on graph search technique and used probabilistic planning to ensure a continuous stepping sequence. The suggested planner would greatly improve locomotion on a rough terrain. Kumar and Waldron [48] concentrated on a real-time adaptive gait generator based on wave gait, which would accommodate the changes in foothold after each cycle. The duty factor was varied to obtain the optimal leg cycling frequency, and hence making the regular wave gait into a free gait capable of guiding the robot along terrain irregularities. Kalakrishnan et al [49] gives an extensive detail of the required control architecture for locomotion on rough terrain. They used terrain maps for foothold selection, and calculated the stability at the assumed position, if the position is favourable, then an inverse kinematics solver is used to obtain the desired actuation values. A similar work has also been presented by Medrano et al. [50], where they considered the presence of obstacles and a terrain with forbidden zones [51]. Deng et al. [52] developed a hexapod controller for rescue operations. The paper emphasises on a zero-moment point (ZMP) which would offer perfect static stability to the robot. The proposed algorithm helps to estimate the ZMP and modify any primary gait to obtain the required free gait.

1.3.4. Hybrid Designs

Hoover et.al. [53] designed an under actuated robot with legs made of smart composite microstructures. These legs are actuated using wires made of shape memory alloys which allow the legs to change stiffness and shape. Using this feature different gaits have been experimented and results of improved stability and manoeuvrability have been presented. Plamondon and Nahon [54] replaced the legs with paddles to develop an underwater exploration robot. The paddles oscillates providing the required thrust to travel underwater. Using a model based non-linear controller, sinusoidal trajectories have been traced and validated. Aggelopoulou et al. [55] formulated an algorithm to obtain the leg sequence to walk on sloping surfaces for a hexapod robot maneuvering underwater. The model is based on the vehicle dynamics since it is subjected to external forces. Grand et al. [56] used a

hybrid wheel-leg system. A decoupled posture and trajectory control algorithm was used to manoeuvre the robot on a rough terrain. Simulation and experimental results were provided to prove the performance of the algorithm. Jun et al. [57] [58] developed a hexapod robot capable of walking on a seabed. Undoubtedly the robot is subjected to a harsh environment. The design and test results have been elaborately shown. Autumn et al. [59] developed a 6 legged robot capable of climbing walls by adhering to the surface. The design and construction of this bio-inspired robot have been presented in this paper. Zhu et al. [60] used semi-round foot-tips to reduce the impacts on touch down and smoothen the stance phases. A leg trajectory correction method was suggested that significantly improves the path deviation and overall power consumption.

1.3.5. Obstacle Avoidance

Obstacle avoidance is a vital research segment in mobile robotics. The hexapod while in a mission must be able to avoid collisions with an obstacle. Seljanko [61] developed a locomotion strategy for an omnidirectional hexapod in order to avoid obstacles in its path. The model was simulated using tripod gait with different trajectories and positions of obstacles. Han et al. [62] developed a method of avoiding moving obstacles in real-time, by measuring the minimum distance of the obstacle from the robot using Gilbert-Johnson-Keerthi algorithm. The developed algorithm can guide the robot and prevent head on collisions. Skaff et al. [63] used sensor guided closed loop controller to guide a hexapod by visual sensing of a line defining its trajectory. The paper gives valuable information regarding visual sensing, to be used extensively in obstacle detection. Zhao et al. [64] developed a novel path planar which can account for large obstacles in between the start and end locations. The robot will be guided along the path maintaining the stability and kinematic feasibility. Yang [65] derived the modified crab and turning gaits (spinning and circular) for an omni-directional hexapod with one leg having locked joints. The gait was simulated for conditions of obstacle avoidance presented in the form of case studies.

1.3.6. Trajectory Optimization

Irrespective of the terrain condition and presence of obstacles, the motion of robot is guided by a trajectory. The motion along the trajectory must be optimized to derive maximum benefits. For example lowering the energy consumption would require a smaller battery to power the robot, minimizing the jerk would provide a smooth motion, while maximizing

payload would provide greater load carrying capacity. The literatures on trajectory optimization can be classified on the basis of:

- i. Minimization of total execution time,
- ii. Minimization of total energy consumption
- iii. Minimization of jerk,
- iv. Maximizing the payload.

Some researchers have used combined optimization of the above mentioned parameters. In context to the present work, a few of the literatures have been reviewed. Piazzzi and Visioli [66] used cubic splines to define the trajectories and optimized it for minimum jerk. They used an algorithm based on interval analysis, which assured the convergence to a global minima. The algorithm was simulated for an m - joint robotic manipulator, and the variation of joint kinematic parameters were shown. Saramago and Steffen Jr. [67] formulated an optimization problem by combining the travel time and energy consumption by using weighing factors. It was simulated for a 3 and 6 d.o.f manipulator considering the dynamics and physical constraints. Gasparetto and Zanotto [68] used a combined optimization of the total execution time and the square of jerk. A minimum jerk trajectory would ensure a smooth motion, while the execution time need not be defined beforehand. Korayem et al. [69] formulated an optimization problem to maximize the payload of a mobile manipulator, considering the dynamics. Simon and Isik [70] relied on trigonometric splines to define trajectories and highlights its advantages over algebraic splines. They stated that by minimizing the jerk it is possible to obtain a closed form solution, and presented a trajectory interpolation algorithm. Cao et al. [71] used cubic splines to interpolate the trajectory and used combined optimization of travel time and jerk using weight factors. Using simulation on a SCARA manipulator, the deviations obtained by varying the weight factors have been presented in the paper. Lin et al. [72] used piecewise cubic polynomials to define the trajectory and proposed an algorithm to obtain the optimal travel time from one knot to the other. The kinematic limits were considered and the simulation results for optimum joint trajectories was presented. Gasparetto and Zanotto [73] used 5th order B-Spline to define the trajectory and performed an optimization of the integral of the squared jerk combined with total execution time. Boryga and Grabos [74] presented a comparison of the joint trajectories by using polynomial splines of order 5, 7 and 9. Lin [75] used particle swarm optimization to generate a minimum jerk trajectory. The simulation results along with the effectiveness of the algorithm are highlighted in this paper. Huang et al. [76] used 5th order B-Spline to

define the trajectory and performed a combined minimization of jerk and execution time. They used NSGA-II based solver to obtain the optimization results. Further, a selection parameter to choose the best solution from the Pareto optimal front was presented. Alshahrani et al. [77] used 3-4-5 polynomial splines to define the trajectory and compared it with the cubic splines with parabolic blend trajectories. Yang and Choi [78] used Bezier curve to construct the trajectory and performed a path planning for a two-wheeled robot. Hwang et al. [79] used cubic Bezier curves to interpolate a user drawn trajectory on a 2D touch space. The paper discusses the required space planning to guide a mobile robot along the designed path at any programmed velocity.

1.4. Gap Finding

An extensive survey of literatures based on hexapods and trajectory optimization have been presented in section 1.3 and a few research gaps have been found. The thesis work is based on a hexapod commercially available by the name of Hexcrawler, manufactured by CrustCrawler USA. The Hexcrawler is a six-legged robot, with each leg having 2 d.o.f. As stated in the literatures, a robot with each leg having 2 d.o.f can be manoeuvred along planar surface only. It has limited mobility to walk on rough terrains. Moreover, there is no provision of attaching tactile sensors at the foot-tip. So the robot cannot sense the foothold forces. Considering these limitations, there will be an attempt to fill the gaps in available research works, in the light of the available resources and scope. The possible areas of research are:

- i. The forward kinematics has not been sufficiently dealt with. Though forward kinematics has no direct application, yet it is important from the point of view of studying the motion characteristics of a particular robot.
- ii. Inverse Kinematics solvers suggested in the literatures are very complex and computationally expensive. There should be an attempt to develop a far more simplified solver, capable of providing solutions for any trajectory having a mathematical definition.
- iii. Researchers have focused upon navigation over rough terrain. However, navigation along complex paths or zig-zag irregular trajectories is yet to be explored. Hence there should be an attempt to navigate the Hexcrawler along complex trajectories.
- iv. Literatures on trajectory optimization focus on minimizing jerk, total time, energy etc. Some works have also combined two or more parameters as the cost function.

However, if it is considered that the total time of travel is known along with the velocity and acceleration limits, then the trajectory can be optimized for minimum jerk. The result of the optimization should be the optimal blend times. For, blending time is a critical control parameter for the robot controller.

- v. In most literatures the trajectory has been designed using polynomials of lower order which reduces the accuracy of the trajectory. Higher order parametric curves could be used to define the trajectory with greater accuracy.

1.5. Aim and Objective

Considering the above gaps in the reported works and the available resources this work is planned as follows.

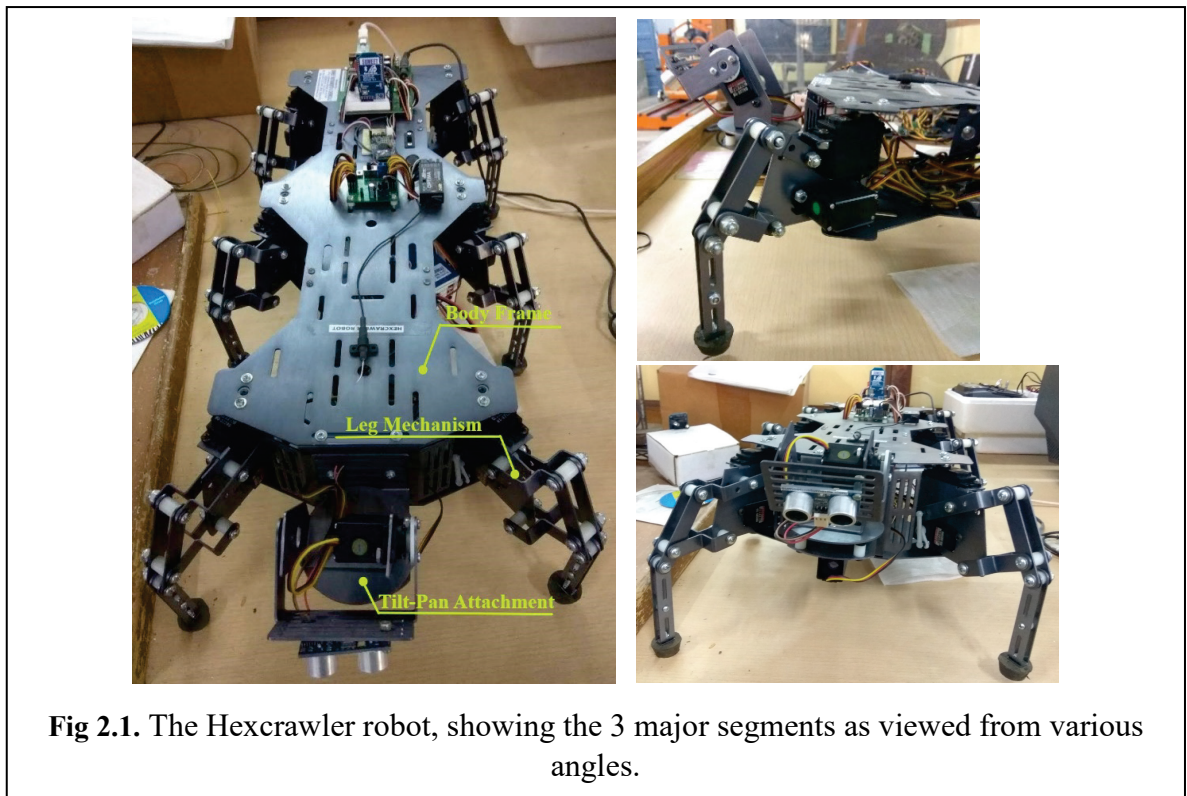
- i. To, prepare the Hexcrawler for complex manoeuvring tasks, and to design a tracing attachment, which can be used to draw trace curves as the robot travels along a programmed path.
- ii. To study the fundamental periodic gaits and to program them on the Hexcrawler.
- iii. To make an analysis of the leg mechanism and to generate the workspace of the foot-tip. Also to check for joint actuation limits considering a non-overlapping workspace between two adjacent legs.
- iv. To formulate a forward kinematics solver that can predict the trajectory of the robot for a given set of actuation angles and number of walking cycles.
- v. To, formulate an Inverse kinematics solver that can estimate the required joint actuation angles for the robot to travel from one predefined location to another. The solver must be independent of the nature of the curve.
- vi. To, mathematically define complex trajectories and to apply the inverse kinematics solver to obtain a solution. In all cases, the simulation results must be validated through experimentation.
- vii. To, define trajectories using n^{th} order parametric curve and to optimize it by minimizing the jerk, and obtaining the optimal blend time.

The above work plan has been executed and is presented with the results and validation of the proposed mathematical models in the subsequent chapters.

Chapter 2.

The Hexcrawler – a six legged robot.

The Hexcrawler robot shown in Fig. 2.1 is available in the Machine Elements Laboratory, Mechanical Engineering Department of Jadavpur University. This robot is used for present analysis and experiment. It belongs the family of walking robots which has 6 legs. The robot is driven by 12 servo motors. It is manufactured by CrustCrawler Robotics, USA and has been locally purchased from Sumeet Instruments and Chemicals, India. The robot has 3 segments – robot frame, 6 set of legs and tilt-pan attachment to mount additional sensors. It is controlled by the Basic Stamp microcontroller. The figure shows the Hexcrawler robot, as imported from the manufacturer.



2.1. Technical Specification

The technical specification as obtained from the user guide provided by the manufacturer has been enlisted in Table 2.1.

Table 2.1. Technical Specifications of Hexcrawler Robot:

Sl. No.	Item	Description
1.	Material	5052-Aluminium with type-II anodizing. Plate thickness of 0.063 gauge (1.29 mm)
2.	Dimensions	Overall dimensions – 19.56" x 15.75" (49.68cm X 40.00cm) Measured from leg to leg.
3.	Height	Standing: 6" (15.24cm) Squatting/Sitting: 4.86" (12.34cm)
4.	Ground Clearance	3.5" (8.89cm)
5.	Leg Movement	2 degrees of freedom/leg (vertical and horizontal)
6.	Payload Capacity	7.5 lbs. (3.4 kg)
7.	Servo Motors	Vertical Servos: Make: HiTech Delux HS-322HD Type: Cored Metal Brush Motor Stall Torque: 3.0 kg.cm at 4.8V 3.7 kg.cm at 6.0V Horizontal Servos: Make: HiTech Delux HS-485HD Type: Cored Metal Brush Motor Stall Torque: 4.8 kg.cm at 4.8V 6.0 kg.cm at 6.0V
8.	Control System	Parallax Board of Education, with Basic Stamp II-Microcontroller

2.2. Upgrading the Control System

The Hexcrawler robot is first upgraded for making it suitable for the present work for the reasons described below. The robot is programmable using the Parallax Board of Education and Basic Stamp II-Microcontroller, as shown in Fig. 2.2. The accessories include - servo controller, ultrasonic sensor, batteries and radio control system. The control electronics of the Hexcrawler Robot was re-modelled to overcome the following limitations:

- a. The previous controller was required to be replaced, since it was damaged and was malfunctioning. However, it was not available in the local market, so there was a need to shift to another controller.
- b. The memory capacity of Basic Stamp was too small (32bytes of RAM), making the controller unsuitable for executing programs using large number of variables and data, such as path generation using inverse kinematics.
- c. The Basic Stamp Editor software is a variant of the BASIC language, making programming more difficult and incapable of handling complex programs.

For this, the Arduino based micro-controller has been selected, having greater memory and processing power. The comparison of the two microcontrollers is enlisted in Table 2.2 and it clearly shows that the Arduino based controller is more suited for accomplishing the objectives of the thesis.

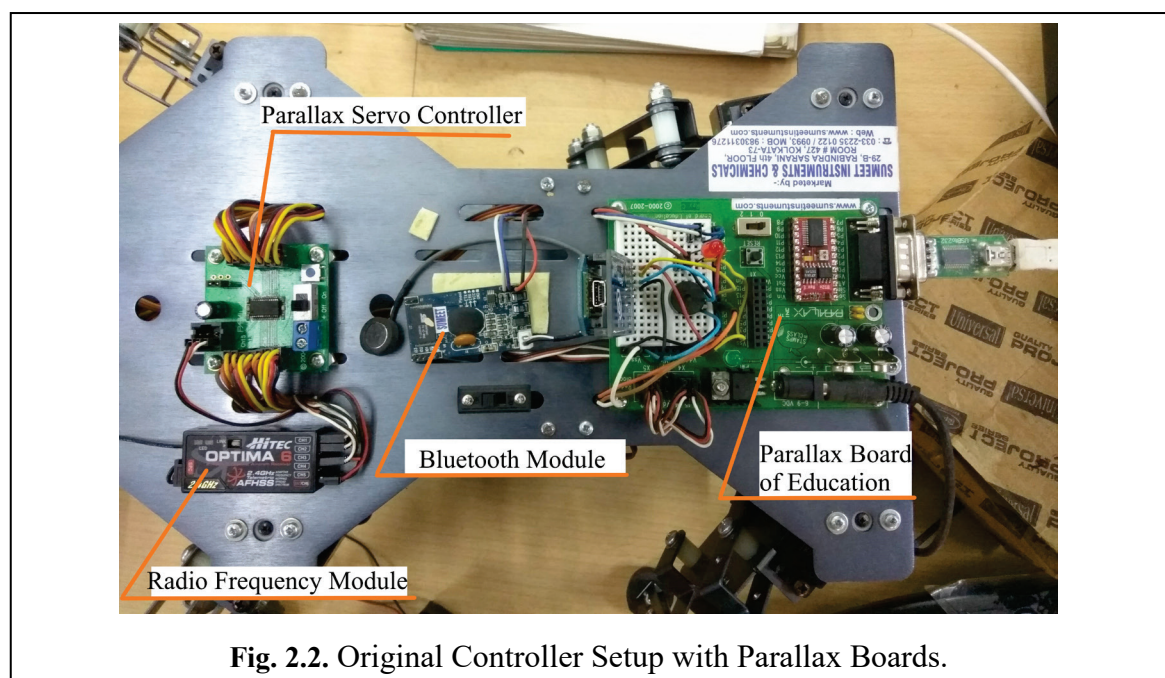


Fig. 2.2. Original Controller Setup with Parallax Boards.

Table 2.2. Comparison Chart of Parallax vs Arduino:

Manufacturer	Parallax	Arduino
Processor	Basic Stamp II	ATmega2560
Clock Speed	20MHz	16MHz
Flash Memory	32 Bytes	256 KB
EEPROM	2 KB	4 KB
I/O Pins	16 + 2 dedicated serial	54 (with 15 PWM, and 16 analogue)
Software	Basic Stamp Editor	Arduino IDE, can be interfaced with MatLab 2017 or higher versions.
Sensors	Limited	Wide range of sensors are available in the form of Shields.

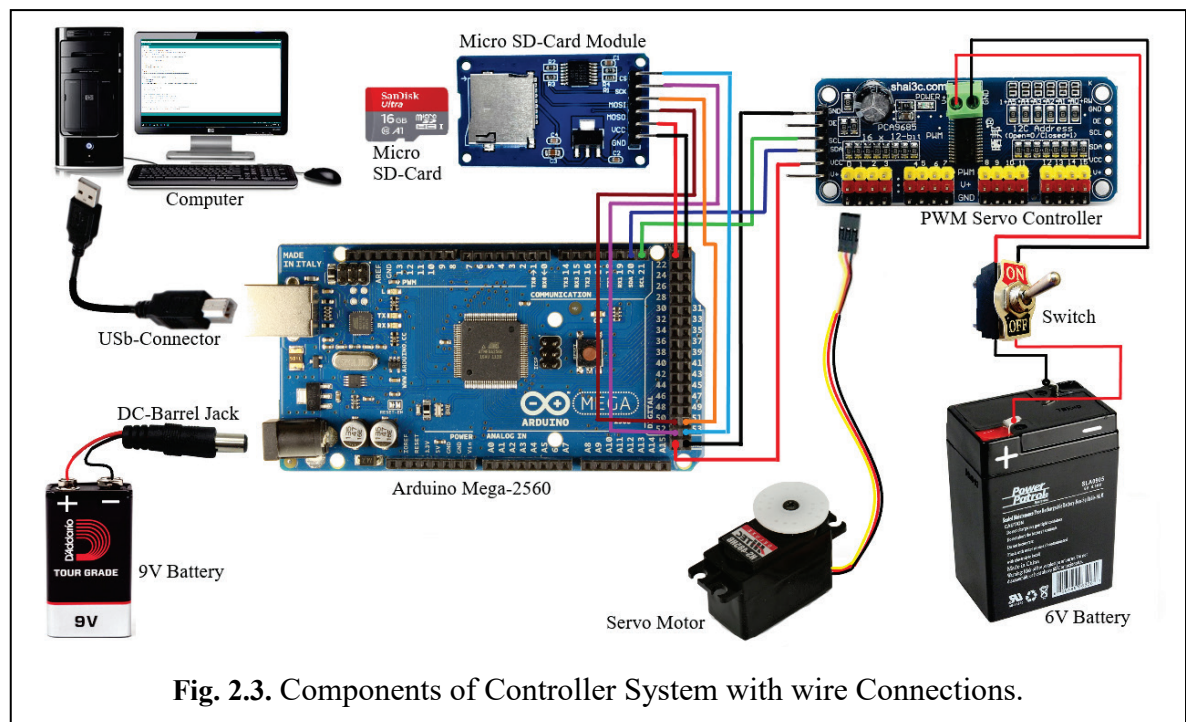
The components of the modified setup include:

- Arduino Mega 2560, Rev3.
- PCA-9685, Adafruit PWM servo controller (12 bit, 16 channel)
- Micro SD card reader module, for data logging
- 6V, 4.5AH Lead Acid Battery, to power the servos
- 9V battery, to power the Arduino mega
- Switch and jumper wires.

In this context, it is worth highlighting that the Radio frequency controller and Bluetooth modules have been removed from the system, since they are irrelevant considering the objective of the present work. The pin connections of different modules with the microcontroller have been listed in Table 2.3. Further, Fig. 2.3 shows the schematic layout of the components and its wiring.

Table 2.3. Wiring and Pin-connections:

Module Name	Pins on Module	Pin on Arduino Mega
Adafruit PWM servo controller	GND	GND
	SCL	SCL
	SDA	SDA
	VCC	+5V
Micro-SD Card Module	CS	53
	SCK	52
	MOSI	51
	MISO	50
	VCC	+5V
	GND	GND

**Fig. 2.3.** Components of Controller System with wire Connections.

The Hexcrawler frame has mounting provisions for the Basic Stamp controller, which do not match with the Arduino controller. Hence, mounting trays have been designed and fabricated using acrylic sheet, as shown in Fig. 2.4. This allows the new setup to be installed on the existing frame.

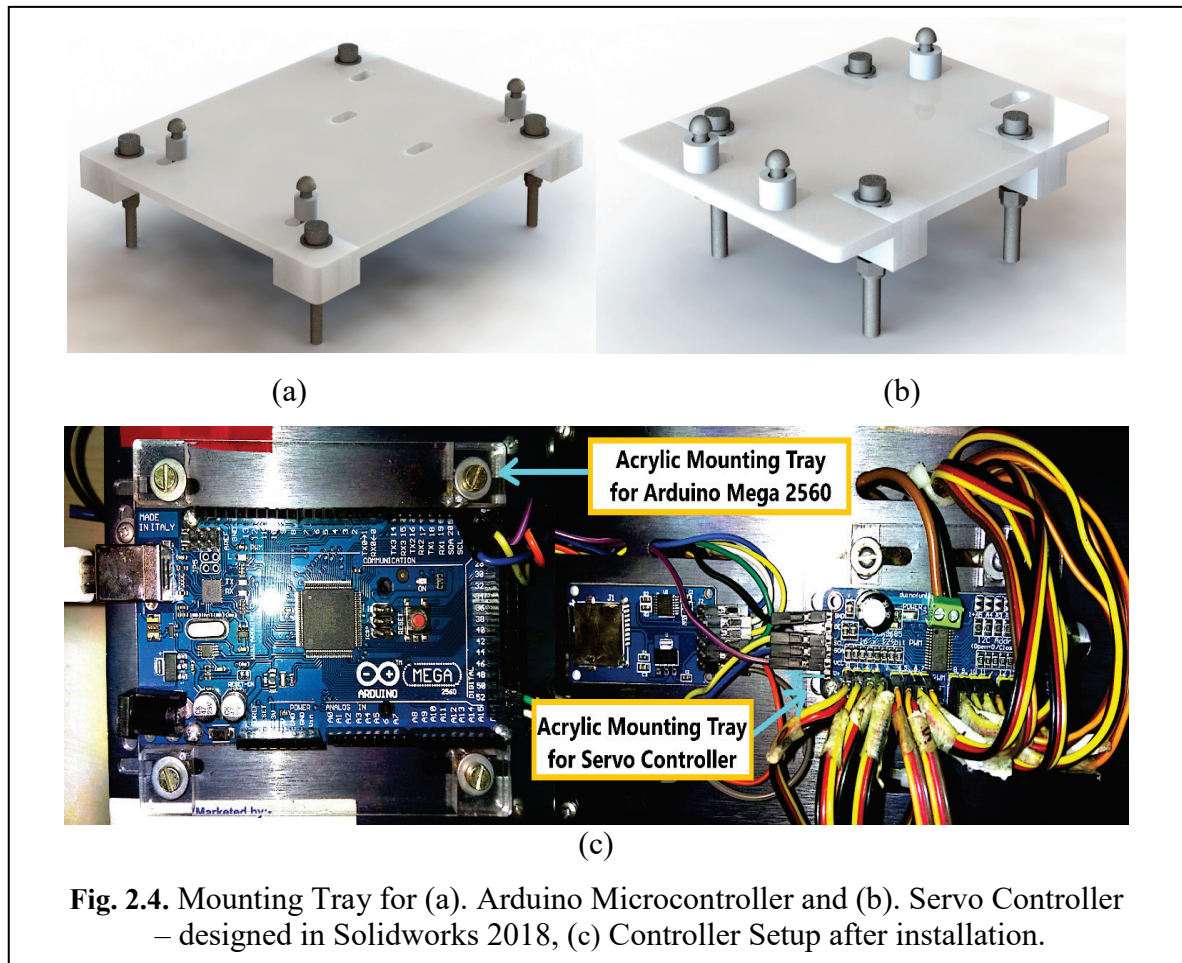


Fig. 2.4. Mounting Tray for (a). Arduino Microcontroller and (b). Servo Controller – designed in Solidworks 2018, (c) Controller Setup after installation.

2.3. Design of a Tracing Attachment

The Hexcrawler robot walks along a flat and levelled surface. The trajectory of the robot needs to be recorded for analysis. A tracing attachment is designed and shown in Fig. 2.5. It is fabricated using aluminium since it is light in weight and adds minimum load to the robot.

The attachment comprises of the following parts:

- a. Primary Barrel: houses the spring and pencil.
- b. Counter barrel: prevents bending of the pencil and avoids the chances of breaking.
- c. Spring: allows the pencil to press upon the tracing surface with a uniform pressure, and maintains contact of the pencil tip with the surface when the robot height fluctuates during a programmed motion, refer Fig. 2.8.
- d. Pencil: used to make tracing marks as the robot moves.
- e. Fasteners (nut-bolts): hold and fasten the two halves of the barrel with the robot frame.

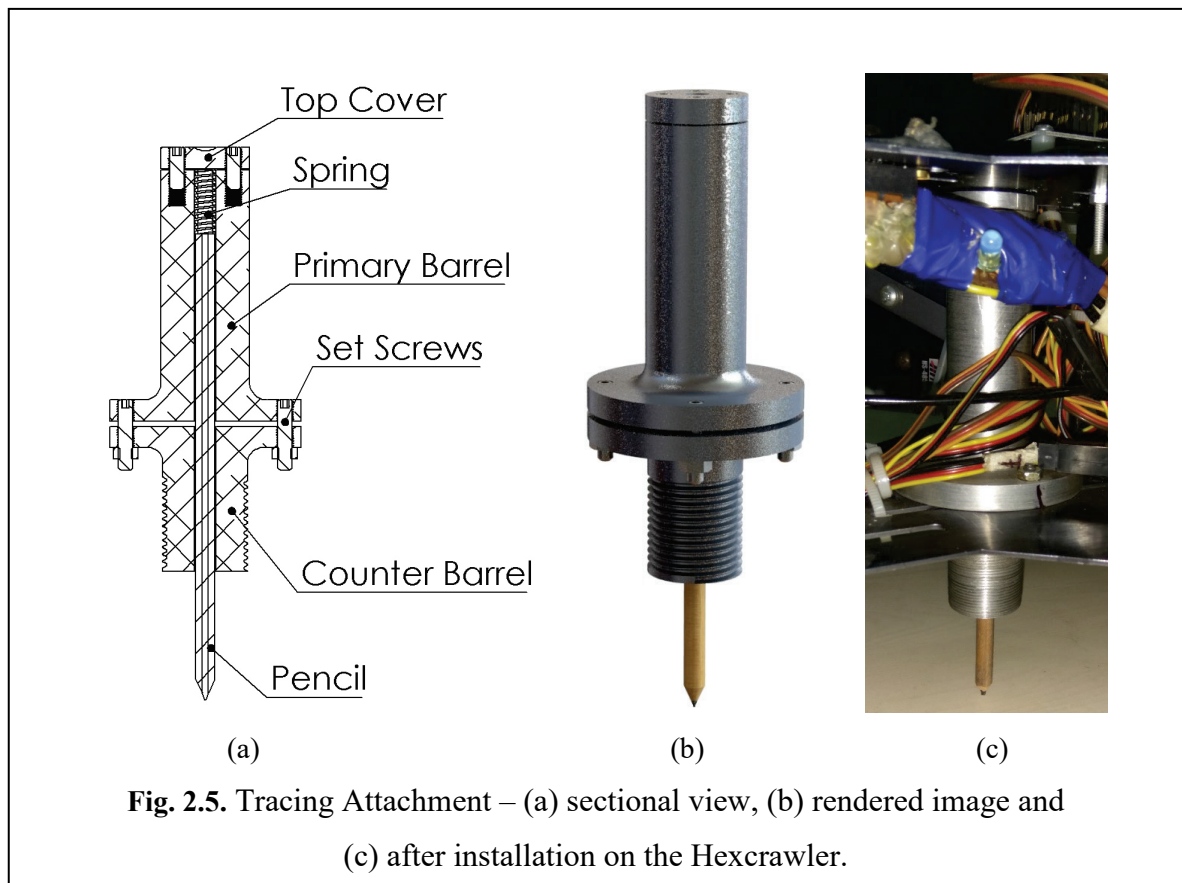


Fig. 2.5. Tracing Attachment – (a) sectional view, (b) rendered image and (c) after installation on the Hexcrawler.

2.4. Calibration of the Modified System

The Hexcrawler robot has a new controller system and it needs to be calibrated before it can be programmed. The servo motors work on PWM control. The calibration process can be divided into two parts – determining the pulse required for rotation of the servo shaft by 1° , and equalising the foot-tip pressures.

2.4.1. Determining p_d (pulse/degree):

After installing the new controller setup, the system is required to be calibrated for precise motion generation. The servo motors are actuated using pulse-width-modulation (PWM) control. The rotation angle of a servo motor is limited to $\pm 90^\circ$ about the mean (0°) position. The operating voltage is applied in pulse to signal the motor to turn in the clockwise or anticlockwise about the mean position. The pulse width (in milliseconds) defines by how much angle the motor should rotate. So a relation between the pulse-width (pw) and rotation angle must be established. Let p_d be the pulse required to rotate the leg by 1 degree.

An arrangement to measure the rotation angle has been shown in Fig. 2.6. Considering a particular leg, a protractor is fixed to the robot frame at the leg hinge, and using the microcontroller, the pulse-width is varied. The leg under consideration is lifted, while other legs are grounded to ensure that the robot is stationary. The supplied pulse-width at which the legs achieve the 0° , $+30^\circ$, -30° positions are recorded. The process is repeated for each leg and the results are recorded in Table 2.4. Here, $\theta_{hi}|_m$, $\theta_{hi}|_f$ and $\theta_{hi}|_b$ represent the angles corresponding to the mean (0°), forward ($+30^\circ$) and backward (-30°) positions of the legs. Although the calibration has been done for the horizontal servos, the results are applicable for vertical servos also.

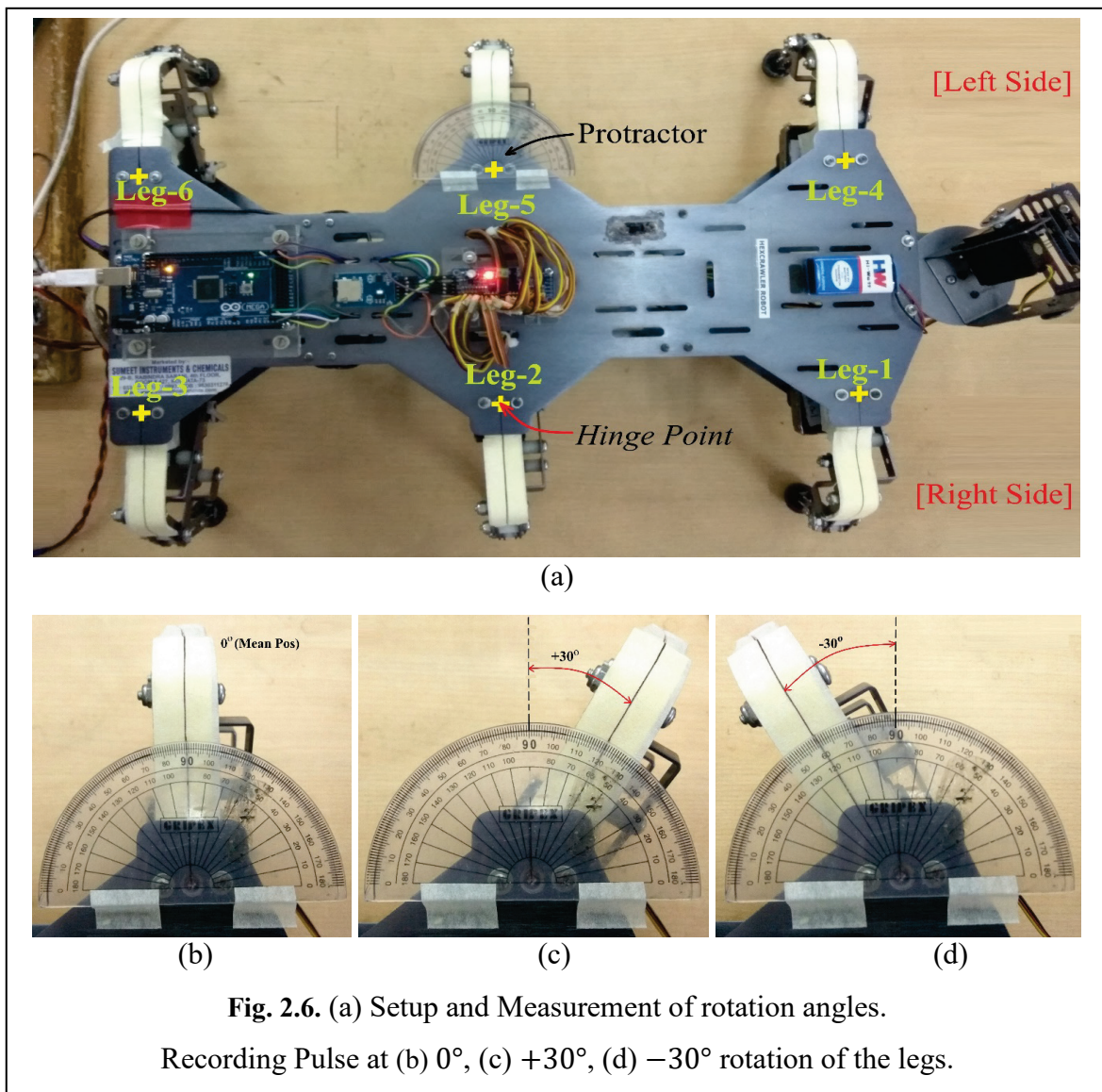


Table 2.4. Horizontal Servo Calibration:

Side	Leg No. (i)	Pulse-width (pw) for $\theta_{hi} _m = 0^\circ$	Pulse-width (pw) for $\theta_{hi} _f = +30^\circ$	Pulse-width (pw) for $\theta_{hi} _b = -30^\circ$	Pulse per Degree (p_d)
Right	1	345	412	278	2.2333
	2	370	437	303	2.2333
	3	340	407	273	2.2333
Left	4	330	263	397	-2.2333
	5	345	278	412	-2.2333
	6	365	298	432	-2.2333

The pulse-width vs rotation angle θ_{hi} can be plotted graphically for each leg as shown in Fig. 2.7. It can be observed that the curve is a straight line and the slope of the curve is p_d . For right and left sided legs, p_d has opposite signs. Using the equation of a straight line, the slope p_d is given by: $p_d = \frac{pw|_f - pw|_b}{\theta_{hi}|_f - \theta_{hi}|_b}$ (2.1)

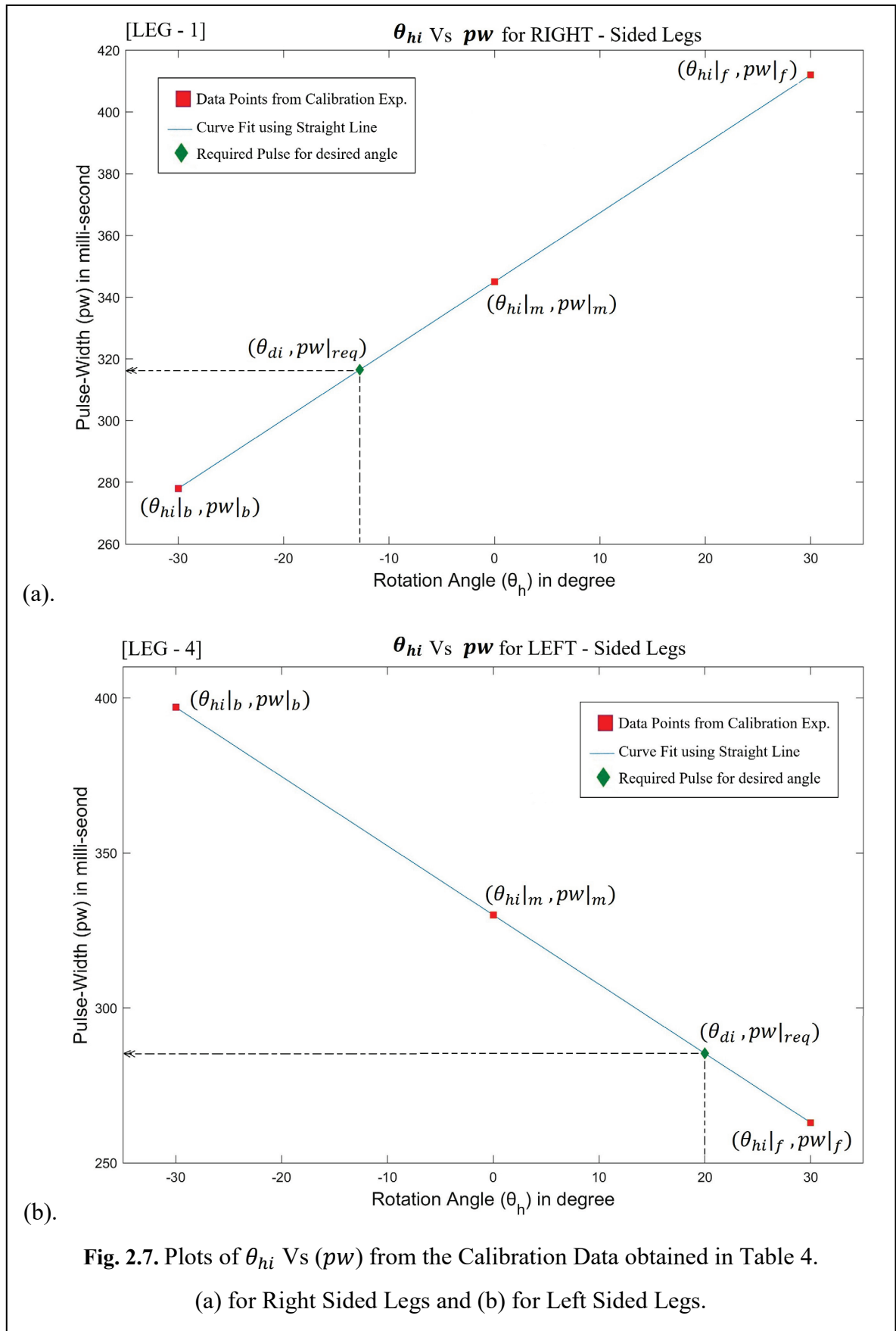
The required pulse $pw|_{req}$ for the desired angle θ_{di} can be calculated from the relation:

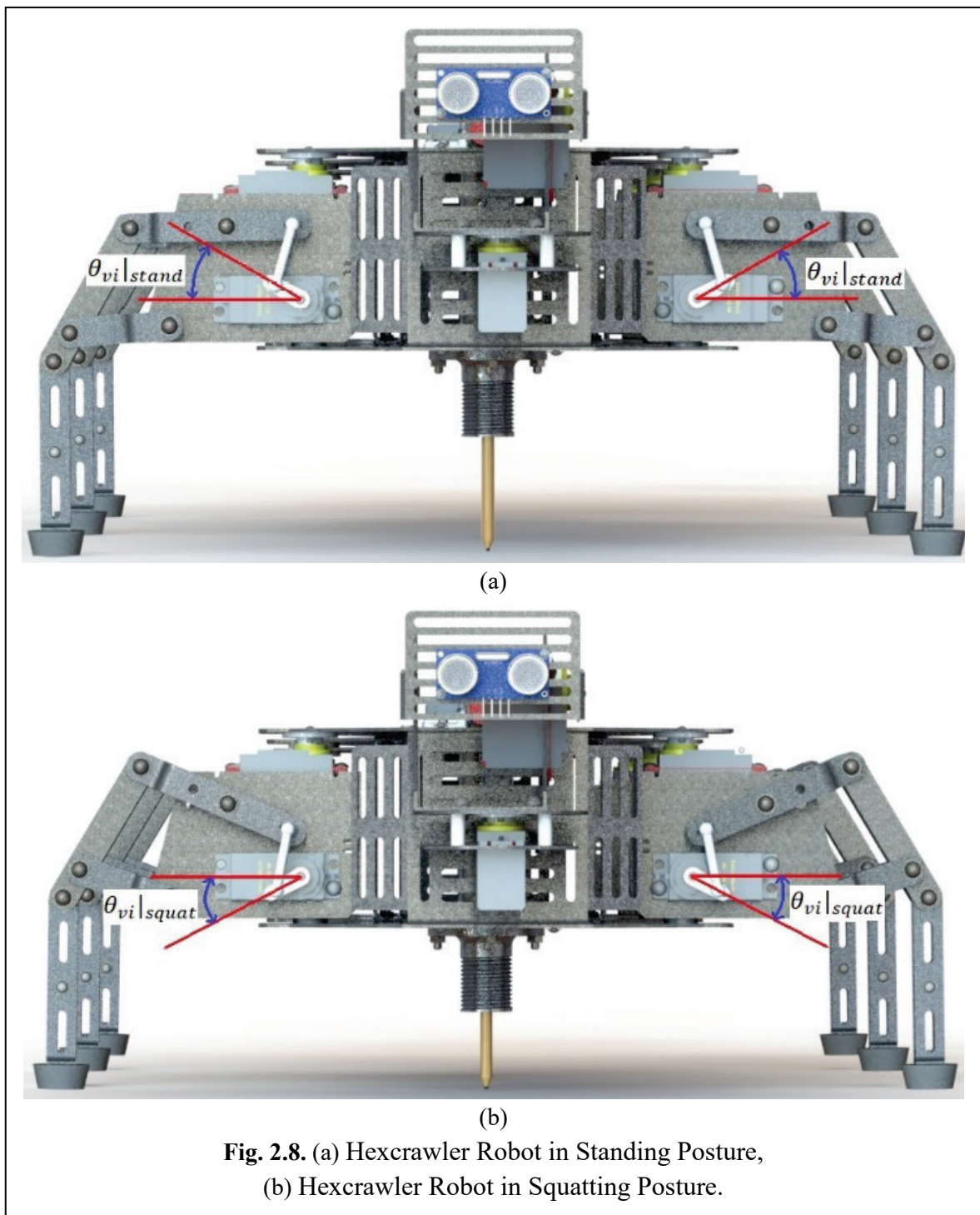
$$(pw|_{req} - pw|_f) = p_d \cdot (\theta_{di} - \theta_{hi}|_f) \quad \text{..... (2.2)}$$

2.4.2. Equalising Foot-Tip Pressure:

In the previous section, the relation between pulse-width and rotation angle was determined primarily for the horizontal servos. The pulse/degree is a constant for both the horizontal and vertical servos. However to interpolate the pulse for a desired angle, the pulse at any two positions must be defined. Hence, the pulse required to position the vertical servos at $\pm\theta_{vi}$ is determined experimentally. The problem is more complex in this case. The two positions result in two distinct posture of the robot – Standing and Squatting as shown in Fig. 2.8. At these postures, the following conditions must be satisfied for planar maneuvering:

- i. The plane of the robot body must be parallel to the ground surface to ensure zero roll and pitch,
- ii. The foot-tips for each legs must exert equal amount of force on the ground surface, ensuring that the body weight is evenly distributed on all six legs.





To perform the experiment, the following components are required:

- a. Flexiforce pressure sensor,
- b. Spirit level indicator,
- c. Glass surface which is smooth and levelled.

Calibration Method:

1. Setup the electronic circuit as shown in Fig. 2.9.
2. Place the spirit level indicator on the top surface of the robot body, and place the Flexiforce sensor between the glass surface and the foot-tip of a particular leg.
3. Set the robot in standing posture using set of pulse-width values chosen arbitrarily,
4. Observe the spirit level indicator and the reading of the Flexiforce sensor from the Serial-monitor of the Arduino software.
5. Place the Flexiforce sensor on the other legs and observe the pressure readings,
6. Adjust the pulse-width for each leg so that the Foot-tip pressures are equal and the spirit level indicating that the robot body is parallel to the glass surface.

The experimental setup has been shown in Fig. 2.10. The calibration process is tedious and iterative. The experiment is conducted for Standing and Squatting posture. Finally, the magnitude of $\theta_{vi}|_{stand}$ and $\theta_{vi}|_{squat}$ and their corresponding pulse-width is obtained. The results of the calibration has been enlisted in Table 2.5.

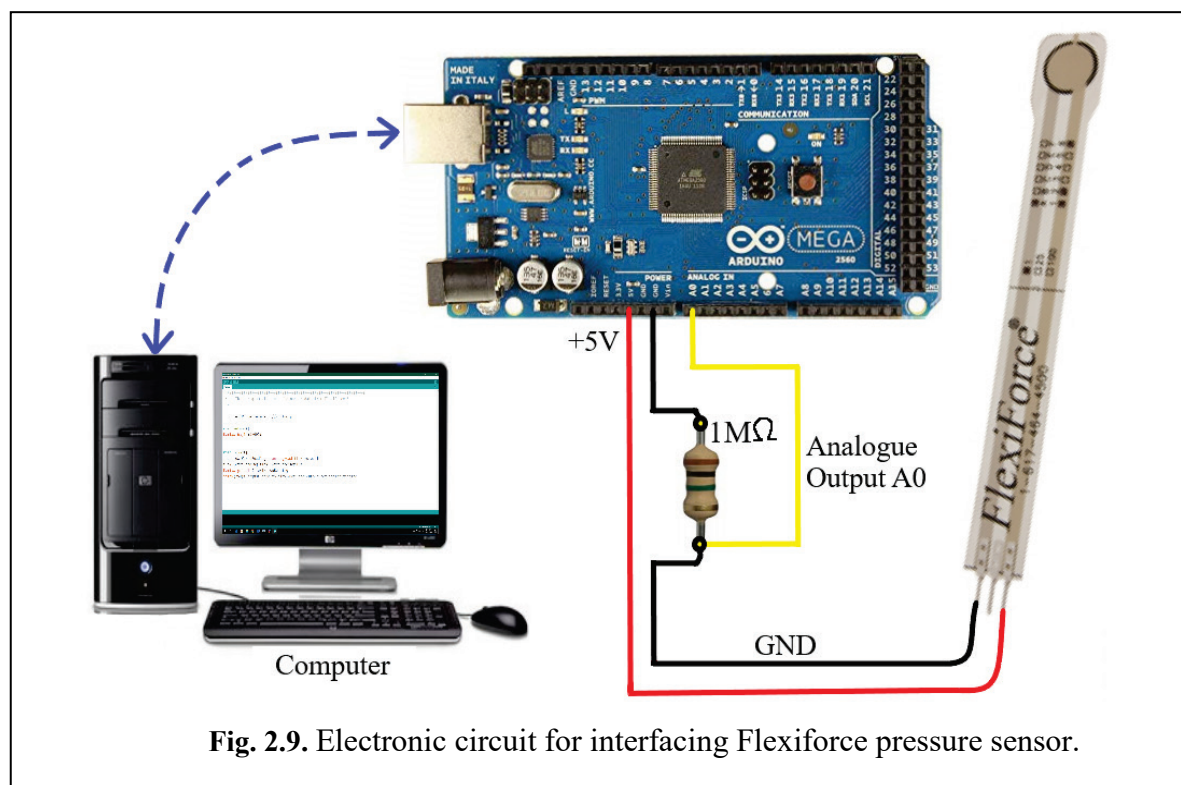
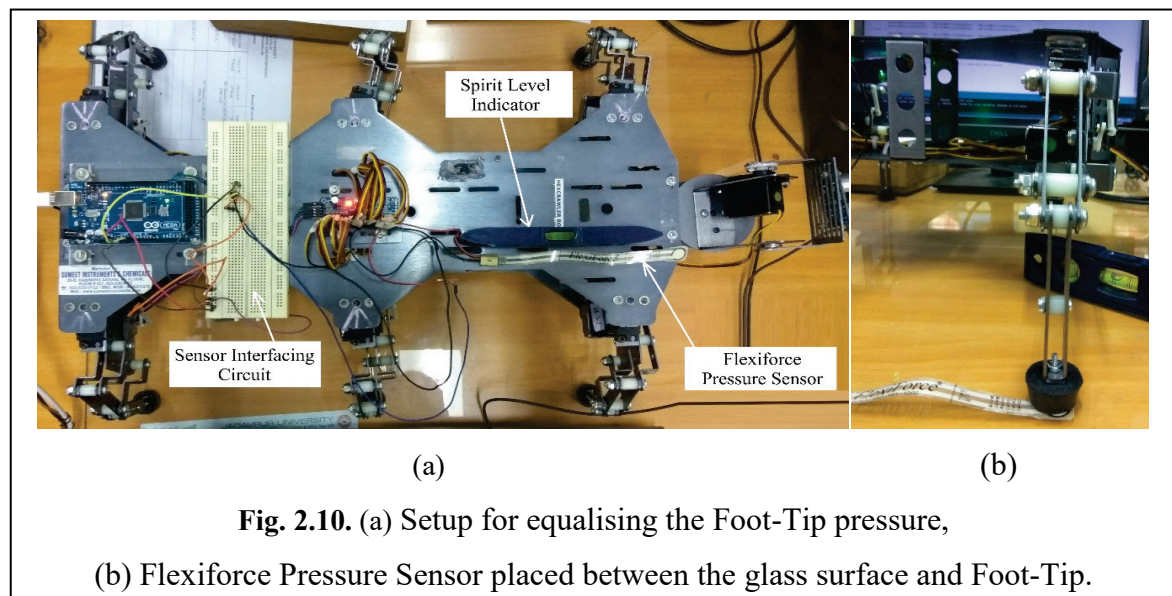


Table 2.5. Vertical Servo Calibration Results:

Side	Leg No. (<i>i</i>)	Pulse-width (<i>pw</i>) for $\theta_{vi} _{stand}$	Pulse-width (<i>pw</i>) for $\theta_{vi} _{squat}$
Right	1	235	475
	2	265	460
	3	275	510
Left	4	360	180
	5	315	170
	6	400	155



2.5. Conclusion

By upgrading the Hexcrawler it is now possible to program complex trajectories, add different sensors and even interface it with MatLab software. The mounting trays allow easy installation of the new electronic components on the existing frame. The Tracing attachment helps in drawing Trace curves which will be used in the later sections of the thesis. Finally, calibrating the new control system allows accurate movement of the robot for programmed motion. The robot is now ready for analysis and experimentation.

Chapter 3.

Gait Patterns for the Hexcrawler.

Gait is defined as the manner of walking or the stepping sequence of the legs. Gaits can be periodic or random (free gait). In periodic gait a particular stepping sequence is repeated, while in Free Gait the stepping sequence is random and is designed to adapt to terrain undulations and ruggedness. In the present work, it is desired to manoeuvre the robot on a smooth and levelled surface (planar manipulation), hence the periodic gaits are of interest.

3.1. Overview of Periodic Gaits

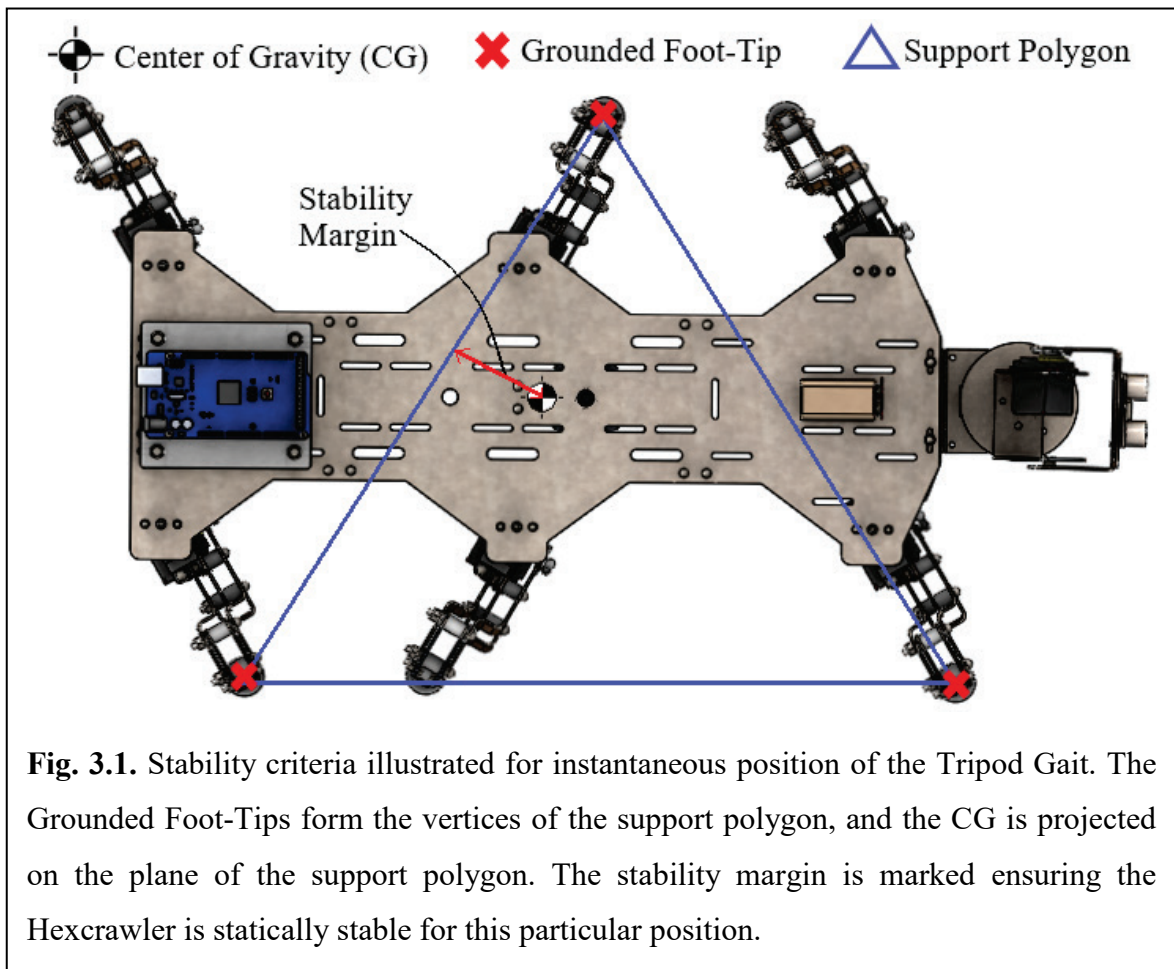
Periodic gaits have a unique stepping sequence constituting a cycle. The gait cycle when iteratively programmed helps the robot to walk continuously. The gaits patterns are dependent on the number of legs of the walking robot. Considering the Hexcrawler which is a six-legged robot, 3 fundamental periodic gaits are identified. There are:

- i. Tripod Gait,
- ii. Ripple Gait,
- iii. Wave Gait.

Each of these gaits have their own merits and demerits. The nature of these gaits have been discussed in detail in the subsequent section. The three gaits are differentiable from each other on the basis of:

- a. **Duty factor:** which is defined as the ratio of time taken for stance phase to the total cycle time for any given leg. The duty factor gives an indication of how fast or slow the body travels at each cycle. For example, tripod gait having duty factor 0.5 is much faster compared to wave gait having duty factor of 0.66.

- b. **Stability:** the static stability of a gait is ensured if the CG of robot lies within the support polygon at any instant. As the CG approaches the edges of the polygon, the stability decreases, as shown in Fig. 3.1. Stability is quantified by defining a stability margin, which is the shortest distance from the projected CG to any of the support polygon edges. The legs of the Hexcrawler are directed away from the body and hence, it ensures higher stability margin for all three gaits.



3.2. Tripod Gait

In the Tripod Gait the Hexcrawler has 3 legs on the ground and remaining 3 legs are in air. So at any instant the weight of the robot is supported by three legs forming a triangular support polygon. Considering the stability, legs 1, 3, 5 and legs 2, 4, 6 form stable tripods ensuring static stability. Fig 3.2 shows the leg at 2 different positions marked P_1 and P_2 at

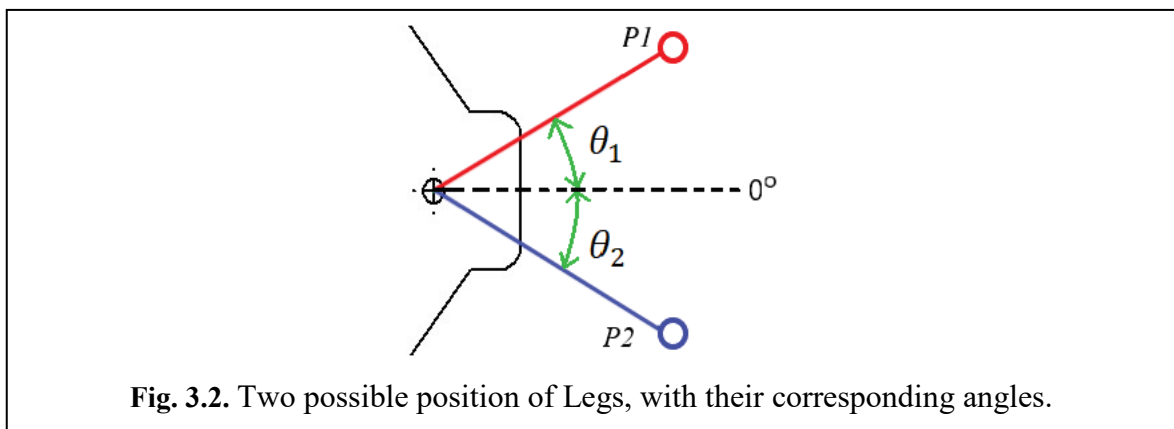
$\theta_h = \{\theta_1, \theta_2\}$ respectively. The Tripod gait is periodic in nature and the stepping sequence starts with the initial condition followed by steps (i) to (vi).

Starting Condition:

- a. Legs 1,3,5 - Grounded and at $\theta_h|_{forward}$
- b. Legs 2,4,6 - In-air and at $\theta_h|_{backward}$

Step sequence per cycle:

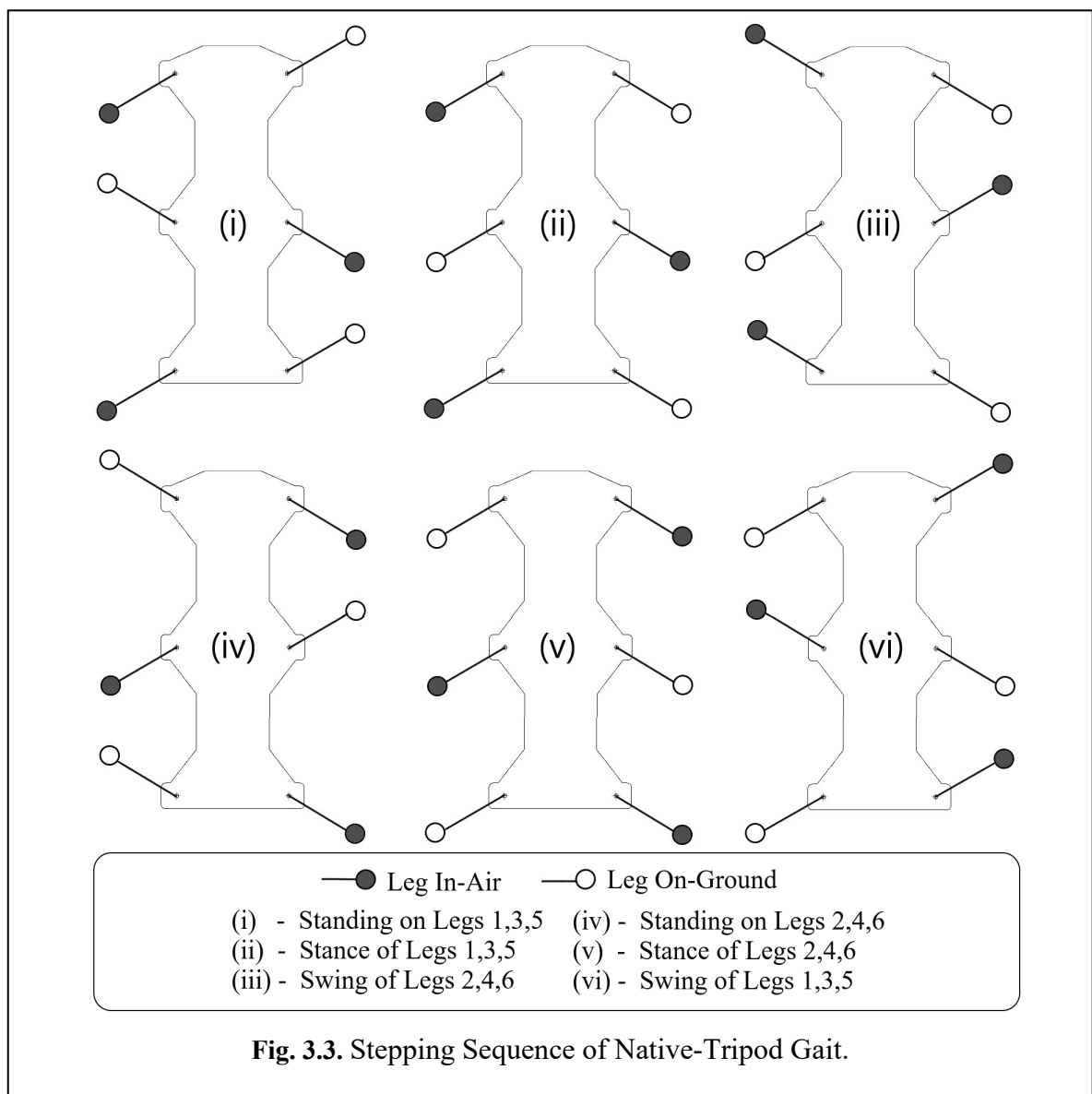
- i. Stance of legs 1,3,5
- ii. Swing of legs 2,4,6
- iii. Change support from legs 1,3,5 to legs 2,4,6
- iv. Stance of legs 2,4,6
- v. Swing of legs 1,3,5
- vi. Change support from legs 2,4,6 to legs 1,3,5

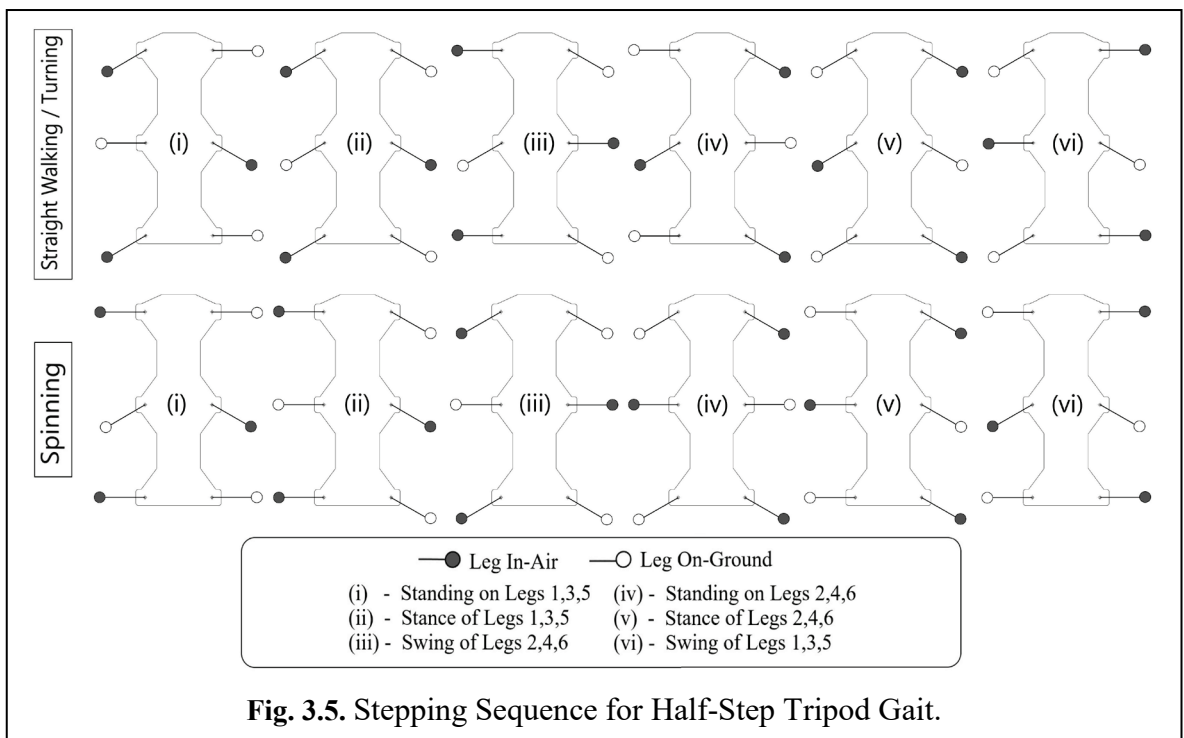
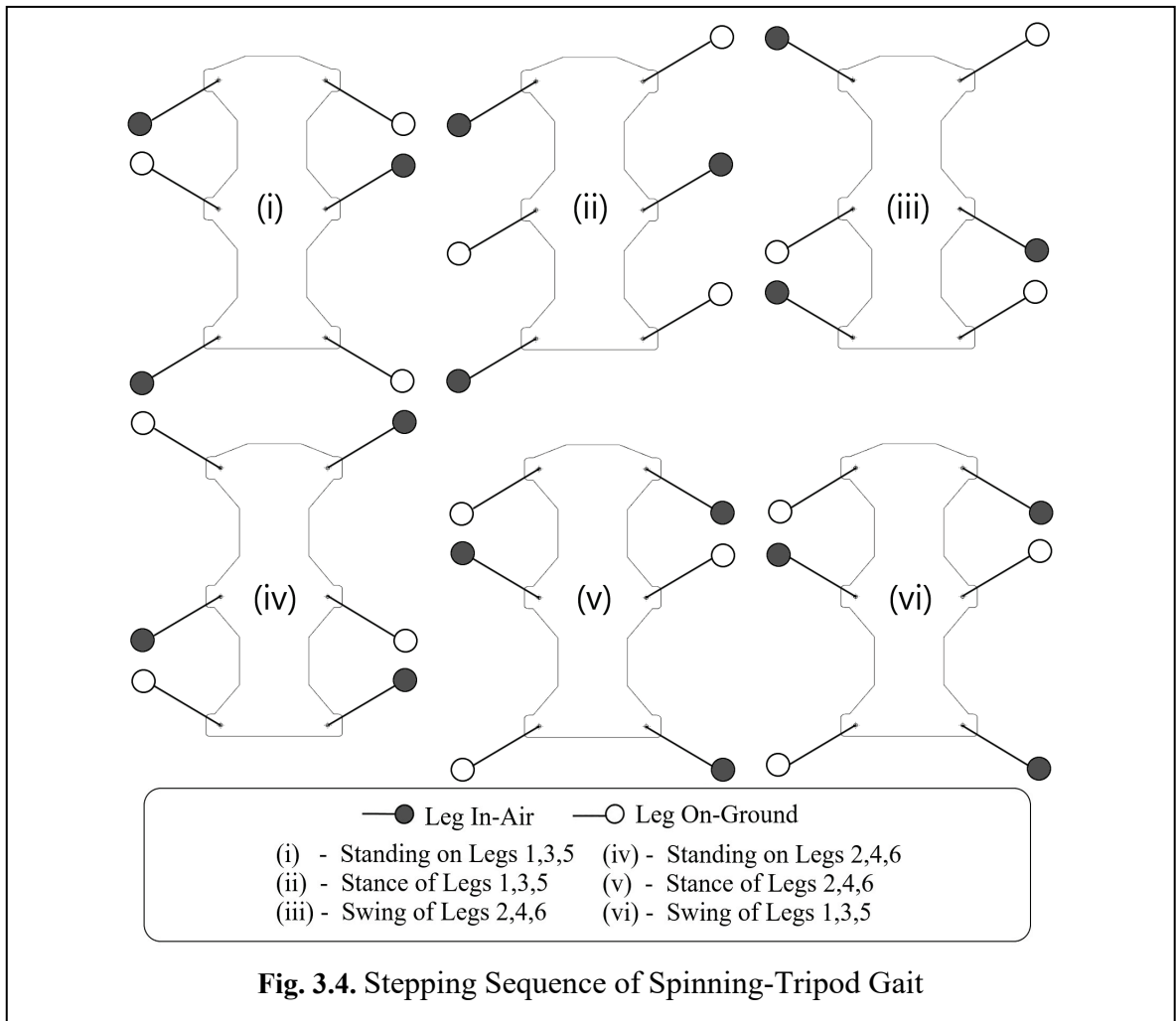


Steps (i) to (vi) constitutes one cycle. For each cycle the robot advances by two steps (during the stance phase). The minimum duty factor is 0.5, which can be achieved by overlapping the stance and swing phases, making the tripod gait the fastest gait. However, in the current thesis work, emphasis is laid on the kinematic analysis, so the stance and swing phases are considered independently and is non-overlapping.

There are three variants of the Tripod Gait:

- **Native Tripod:** applicable for straight walking and turning along a large radius of curvature, refer Fig. 3.3.
- **Spinning Tripod:** applicable for sharp turns with negligible radius of curvature, refer Fig. 3.4.
- **Half-Step Tripod:** adaptable gait which can readily switch between Native and Spinning Tripod gaits. Ideal for traversing irregular trajectories, and foot-step planning using inverse kinematics, refer Fig. 3.5.





3.3. Ripple Gait

In Ripple Gait, the Hexcrawler has 4 legs on the ground and remaining 2 legs in air. So at any instant the weight of the robot is supported by 4 legs forming a quadrilateral support polygon. The Ripple gait ensures greater stability as compared to the Tripod gait enabling the robot to carry higher payloads. However, it is slower, as more number of steps are required to travel the same distance, as compared to the Tripod gait. Fig. 3.6 shows the leg at 5 different positions marked P_1, P_2, \dots, P_5 at $\theta_h = \{\theta_1, \theta_2, \dots, \theta_5\}$ respectively. If θ_1 and θ_5 be the extreme positions then, the remaining angles are computed as:

$$\begin{cases} \theta_2 = 0.75\theta_1 + 0.25\theta_5 \\ \theta_3 = 0.50\theta_1 + 0.50\theta_5 \\ \theta_4 = 0.25\theta_1 + 0.75\theta_5 \end{cases} \dots\dots\dots (3.1)$$

For each stance phase, the corresponding legs move by $\Delta\theta_h$, i.e. a leg initially at θ_2 moves to θ_3 . For each swing phase, the corresponding legs move by $2\Delta\theta_h$, i.e. a leg initially at θ_5 moves to θ_3 . The stance and swing phases are considered in overlapping condition to reduce number of steps per cycle. The ripple gait can be further subdivided into 3 variants- native ripple, spinning ripple, and half-step ripple. The variants are exactly similar to those of the tripod variants and hence not shown separately. Fig 3.7 shows the steps per cycle for straight or curved path walking. Ripple gait is periodic in nature and the stepping sequence starts with the initial condition followed by steps (i) to (xii).

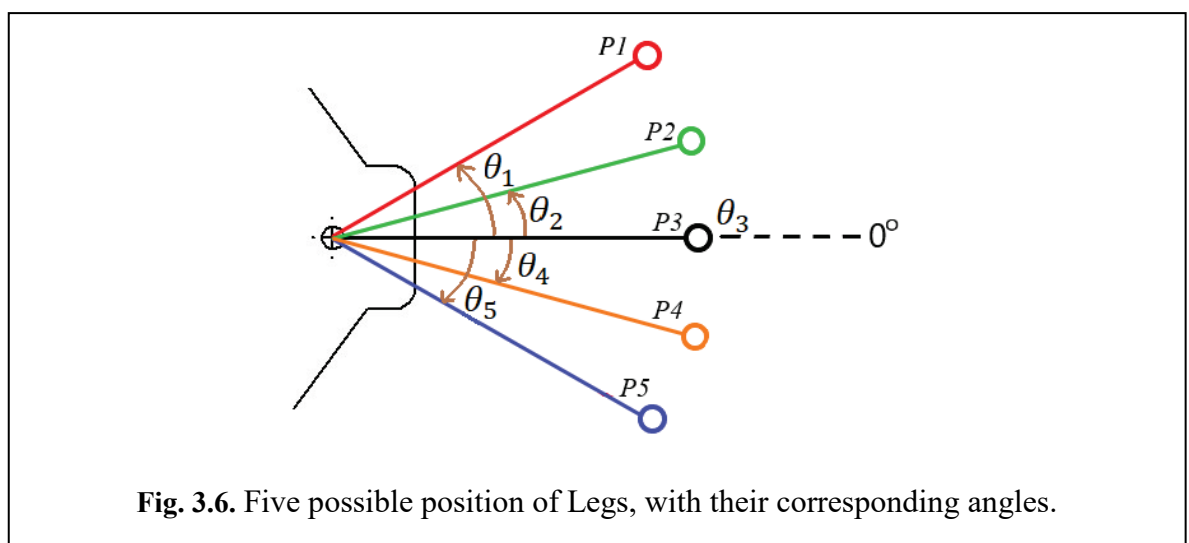


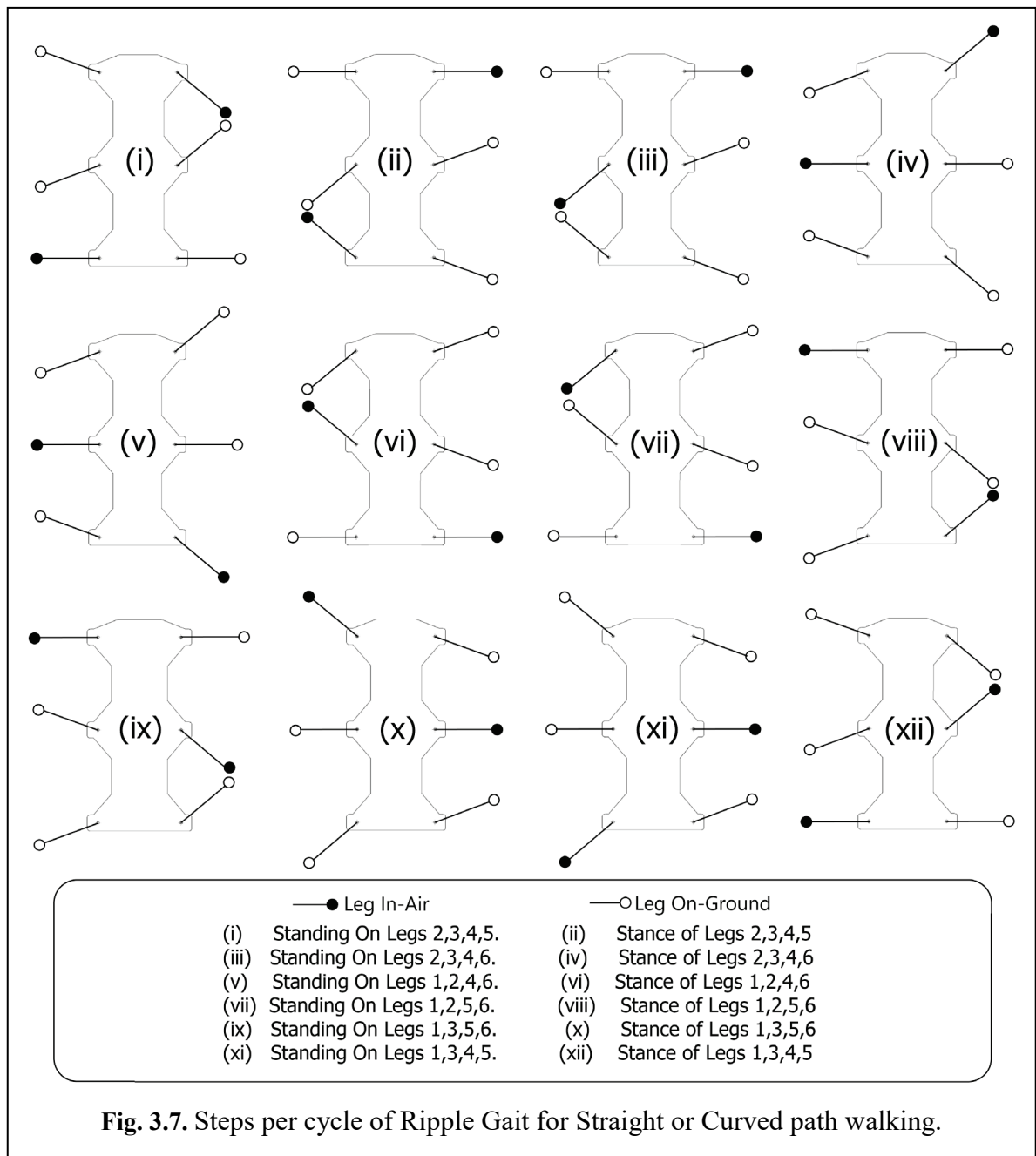
Fig. 3.6. Five possible position of Legs, with their corresponding angles.

Starting Condition:

- a. Legs 2,3,4,5 - Grounded with corresponding θ_{hi} at $\{\theta_1, \theta_3, \theta_2, \theta_4\}$ respectively,
- b. Legs 1,6 - In-air with corresponding θ_{hi} at $\{\theta_5, \theta_3\}$ respectively.

Steps Sequence per cycle:

- i. Stance of legs 2,3,4,5 with Swing of legs 1,6.
- ii. Change support from legs 2,3,4,5 to legs 2,3,4,6.
- iii. Stance of legs 2,3,4,6 with Swing of legs 1,5.
- iv. Change support from legs 2,3,4,6 to legs 1,2,4,6.
- v. Stance of legs 1,2,4,6 with Swing of legs 3,5.
- vi. Change support from legs 1,2,4,6 to legs 1,2,5,6.
- vii. Stance of legs 1,2,5,6 with Swing of legs 3,4.
- viii. Change support from legs 1,2,5,6 to legs 1,3,5,6.
- ix. Stance of legs 1,3,5,6 with Swing of legs 2,4.
- x. Change support from legs 1,3,5,6 to legs 1,3,4,5.
- xi. Stance of legs 1,3,4,5 with Swing of legs 2,6.
- xii. Change support from legs 1,3,4,5 to legs 2,3,4,5.



3.4. Wave Gait

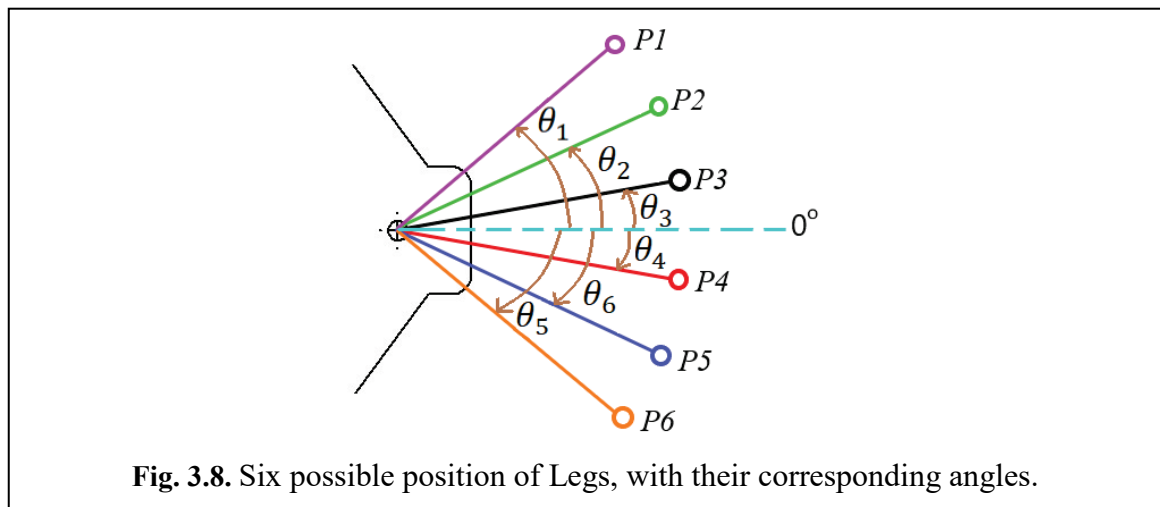
In wave gait, the Hexcrawler has 5 legs on the ground and remaining 1 leg are in air. So at any instant the weight of the robot is supported by 5 legs forming a pentagonal support polygon. Wave gait ensures maximum stability as compared to any other gait enabling the robot to carry highest payload. However, it is also the slowest, as it requires more number of steps to travel the same distance, as compared to the other gaits. Fig 3.8 shows the leg at

5 different positions marked P_1, P_2, \dots, P_6 at $\theta_h = \{\theta_1, \theta_2, \dots, \theta_6\}$ respectively. If θ_1 and θ_6 be the extreme positions then, the remaining angles are computed as:

$$\begin{cases} \theta_2 = \theta_1 - \Delta\theta \\ \theta_3 = \theta_1 - 2\Delta\theta \\ \theta_4 = \theta_1 - 3\Delta\theta \\ \theta_5 = \theta_1 - 4\Delta\theta \end{cases} \dots\dots\dots (3.2)$$

Where, $\Delta\theta = \frac{(\theta_1 - \theta_6)}{5}$

For each stance phase, the corresponding legs move by $\Delta\theta$, i.e. a leg initially at θ_2 moves to θ_3 . For each swing phase, the corresponding legs move by $5\Delta\theta$, i.e. a leg initially at θ_6 moves to θ_1 . The stance and swing phases are considered in overlapping condition to reduce number of steps per cycle. The wave gait can be further subdivided into 3 variants- native wave, spinning wave, and half-step wave. The variants are exactly similar to those of the tripod and ripple variants and hence not shown separately. Fig 3.9 shows the steps per cycle for straight or curved path walking. Wave gait is periodic in nature and the stepping sequence starts with the initial condition followed by steps (i) to (xii).



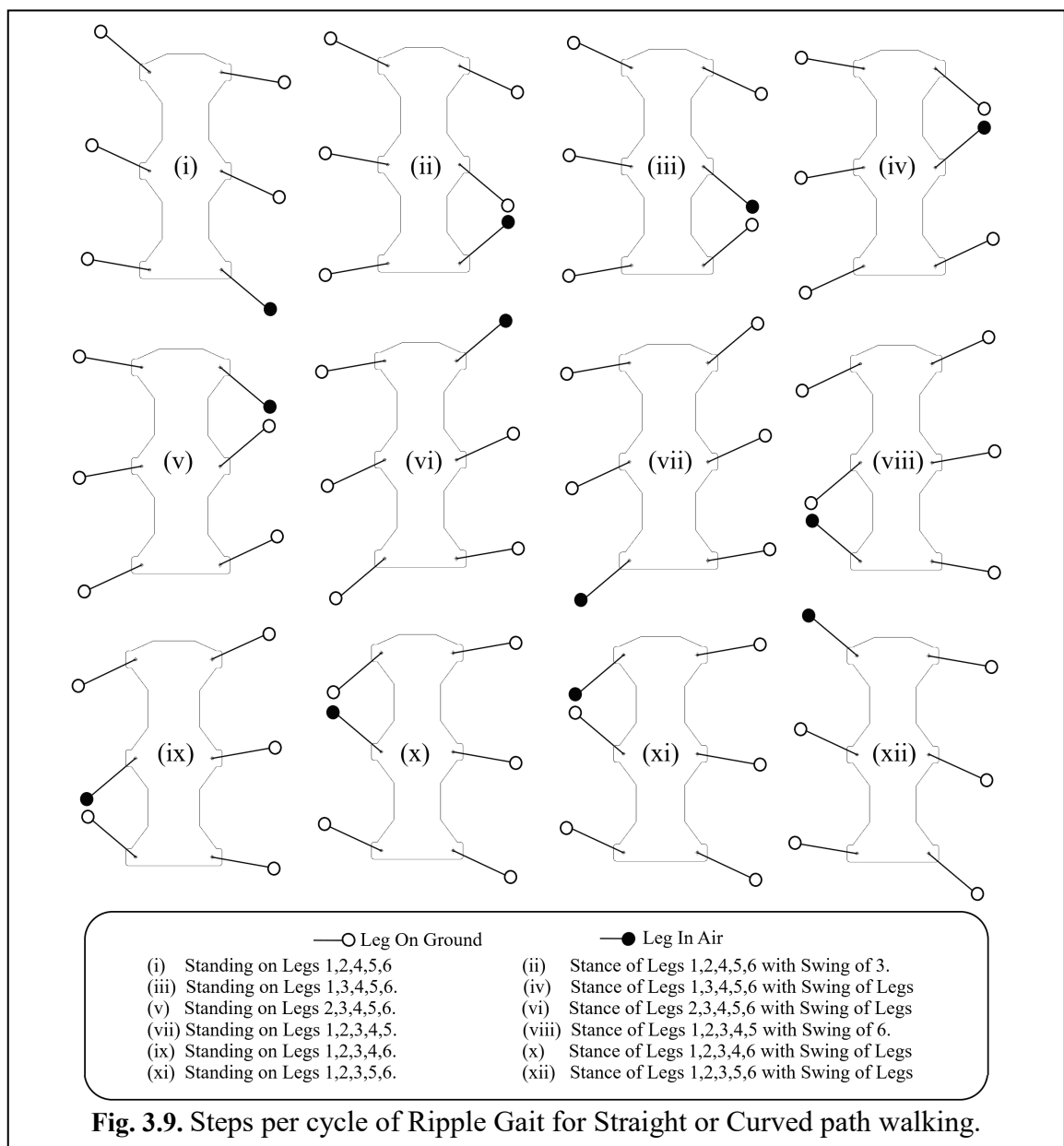
Starting Condition:

- a. Legs 1,2,4,5,6 - Grounded with corresponding θ_{hi} at $\{\theta_4, \theta_5, \theta_1, \theta_2, \theta_3\}$ respectively,
- b. Leg 3 - In-air with corresponding θ_{hi} at $\{\theta_6\}$ respectively.

Steps Sequence per cycle:

- i. Stance of legs 1,2,4,5,6 with Swing of leg 3.
- ii. Change support from legs 1,2,4,5,6 to legs 1,3,4,5,6.

- iii. Stance of legs 1,3,4,5,6 with Swing of leg 2.
- iv. Change support from legs 1,3,4,5,6 to legs 2,3,4,5,6.
- v. Stance of legs 2,3,4,5,6 with Swing of leg 1.
- vi. Change support from legs 2,3,4,5,6 to legs 1,2,3,4,5.
- vii. Stance of legs 1,2,3,4,5 with Swing of leg 6.
- viii. Change support from legs 1,2,3,4,5 to legs 1,2,3,4,6.
- ix. Stance of legs 1,2,3,4,6 with Swing of leg 5.
- x. Change support from legs 1,2,3,4,6 to legs 1,2,3,5,6.
- xi. Stance of legs 1,2,3,5,6 with Swing of leg 4.
- xii. Change support from legs 1,2,3,5,6 to legs 1,2,4,5,6.



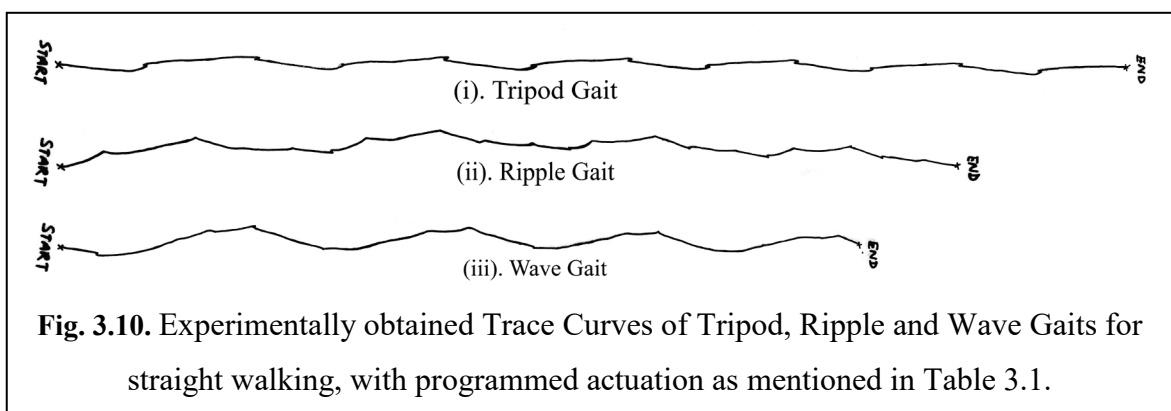
3.5. Results and Discussion

The 3 primary periodic gaits- tripod, ripple and wave gaits has been tested on the Hexcrawler robot. Using the tracing attachment, it is possible to obtain the trace curves as the robot moves along the programmed path. The stepping sequence per cycle has been programmed in Arduino IDE and loaded in the microcontroller for execution.

For comparison, the Hexcrawler robot has been programmed to walk along a straight line (equal actuation for left and right sided legs). The actuation details has been presented in Table 3.1. The trace curves have been shown in Fig. 3.10. It is observed that for tripod gait, the stance phases produce aggressive surges even though the actuation angle is halved as compared to the rest. However the sway is minimum. The magnitude of surge and sway is moderate in ripple gait. Wave gait has small surge and larger sway. Aggressive surge value ensures faster walking speeds, but the system becomes highly unstable when the actuation angles are large. Smaller surge value makes the walking slower but ensured more stability, allowing the robot to carry higher payloads. The sway in each case is inevitable, and occurs due to the unequal torque distribution at the legs and the stepping sequence. However, by joining the start and end points by a straight line, it can be observed that the overall motion is linear in nature.

Table 3.1. Actuation Details to obtain the Trace curves for 3 fundamental gaits:

Gait Pattern	Servo Actuation	No of Cycles
Native Tripod	$\theta_1 = +15^\circ, \theta_2 = -15^\circ$	6
Ripple	$\theta_1 = +30^\circ, \theta_5 = -30^\circ$	4
Wave	$\theta_1 = +30^\circ, \theta_6 = -30^\circ$	4



3.6. Conclusion

In this chapter, the stability of the Hexcrawler robot has been analysed. Different walking techniques along with their stepping sequence per cycle have been illustrated. The proposed gaits have been tested on the Hexcrawler by programming it to walk along a straight line. The trace curves obtained has been analysed and it can be concluded that:

- i. Tripod gait is suitable where faster walking speeds is desired at the expense of lower payload capacity.
- ii. Wave gait is suitable where higher payload is desired at the expense of slower walking speeds.
- iii. Ripple gait is suited for moderate payloads at moderate speed.

So, each of these gaits have their own advantages and disadvantages. The choice of gait is dependent on the field of application. A detailed analysis of the tripod gait has been done in the subsequent chapters. This includes the kinematic motion analysis of the robot with numerical simulation using predefined trajectories. Further, experiments have been conducted which validate the analysis and simulation results. The same can be applied for ripple and wave gaits also and can be considered for future scope of work.

Chapter 4.

Leg Mechanism – kinematic model.

Before developing a kinematic model of the robot, it is essential to understand the working of each leg. The legs of the Hexcrawler robot are a closed chain manipulator having two degrees of freedom (d.o.f). In this chapter the aim is to develop a kinematic model of the leg, and generate the workspace using numerical simulation.

4.1. Description

The Hexcrawler Robot has six legs attached to the body frame, each of which is powered by 2 servo motors, refer Fig 4.1(a). Servo-1 is responsible for swinging the leg back and forth, while servo-2 is responsible for lifting and lowering the leg, making each leg a 2 d.o.f system. Each of the legs can be considered as a closed kinematic chain, and the 6 legs together make the robot a parallel manipulator having 12 degrees of freedom. Before developing the forward and inverse kinematic model of the robot it is essential to develop a mathematical relation comprising of servo actuator angles and foot-tip / leg pivot positions.

A simplified line-diagram of the leg mechanism has been shown in Fig 4.1(b). Let θ_h and θ_v be the actuation angles for servo-1 and servo-2 respectively. The leg pivot is considered as a point of intersection of the rotational axes of servo- 1 and servo-2. The axes of servo-1 and servo-2 are assumed to be intersecting, although the physical construction suggests an offset of 4mm (approx.), which can be neglected for simplification of the problem. The foot-tip is a point which makes contact with the ground surface. These two points – leg pivot and foot-tip are of key interest as position of one with respect to the other can be used to obtain the position and orientation of the Hexcrawler robot.

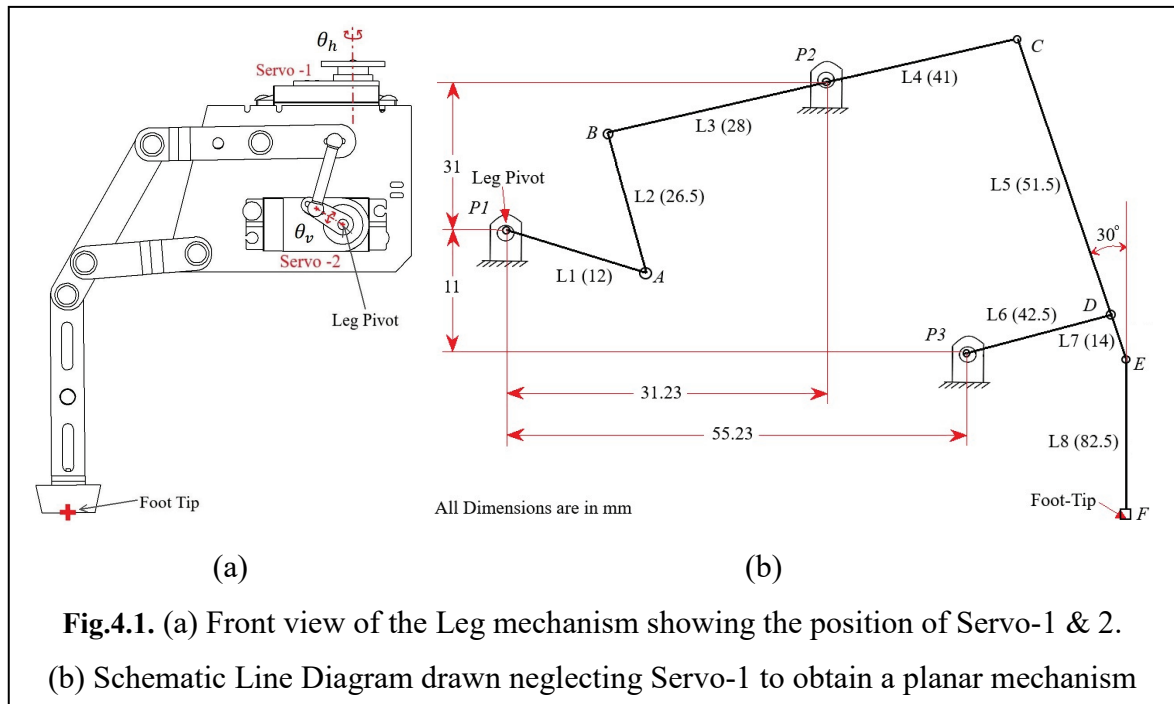
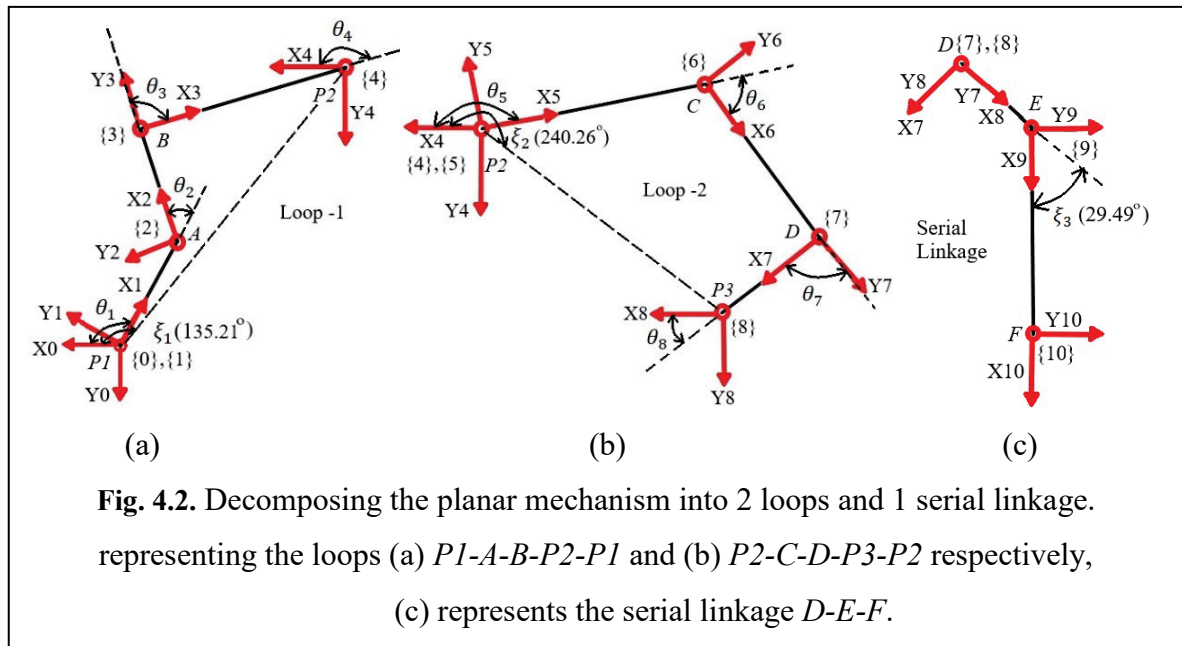


Fig.4.1. (a) Front view of the Leg mechanism showing the position of Servo-1 & 2.
 (b) Schematic Line Diagram drawn neglecting Servo-1 to obtain a planar mechanism

4. 2. Analysis using DH parameters

The relation between the leg pivot l_i and the foot-tip f_i is a function of the servo actuation angle θ_h , where i indicates the leg number, $i \in \{1,2 \dots 6\}$. This relation can be obtained using Denavit-Hartenberg (DH) algorithm for serial manipulators. However the problem is more complex in this case, as it is a closed chain mechanism. Hence the problem is broken down into 2 loops $\{P1-A-B-P2-P1, P2-C-D-P3-P2\}$ and 1 serial linkage $\{D-E-F\}$. Fig. 4.2 shows the frames and Euler angles for each segment. Frames are numbered within $\{ \}$, Joint angles are represented by θ_i , linkage constant angles are represented by ξ_i and link lengths by L_i .

If the reference frame coincides with the leg-pivot then the coordinates of the foot-tip can be obtained by a Homogeneous Transformation matrix, when the actuation angle θ_v is known. Initially servo-1 is neglected and this reduces the problem to a planar mechanism. The planar linkage is now solvable using DH-parameters. The effect of actuation of servo-1 by angle θ_v will be considered later on, and the initially obtained solution will be modified accordingly.



The composite transformation matrix is given by:

$${}^{i-1}T_i = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots (4.1)$$

Where,

- a_i is the link length, measured as the shortest distance between the consecutive joint axes Z_{i-1} and Z_i ,
- α_i is the angular twist, measured as the angle between Z_{i-1} and Z_i measured about X_i in the right hand sense,
- θ_i is the link angle, measured as the angle between X_{i-1} and X_i measured about Z_i in the right hand sense,
- d_i is the link offset, measured as the distance along Z_{i-1} to the point of common normal to Z_i .

Analysis of Loop-1:

For closed loop linkages, D-H method can be applied by splitting the closed chain into 2 open chains and re-combining them using ‘loop-closure’ technique, refer Ghosal [80]. Hence for Loop-1, $P1-A-B-P2$ is the 1st serial linkage and $P1-P2$ is the 2nd serial linkage, as shown in Fig 4.3. The D-H parameters are enlisted in Table 4.1 and Table 4.2.

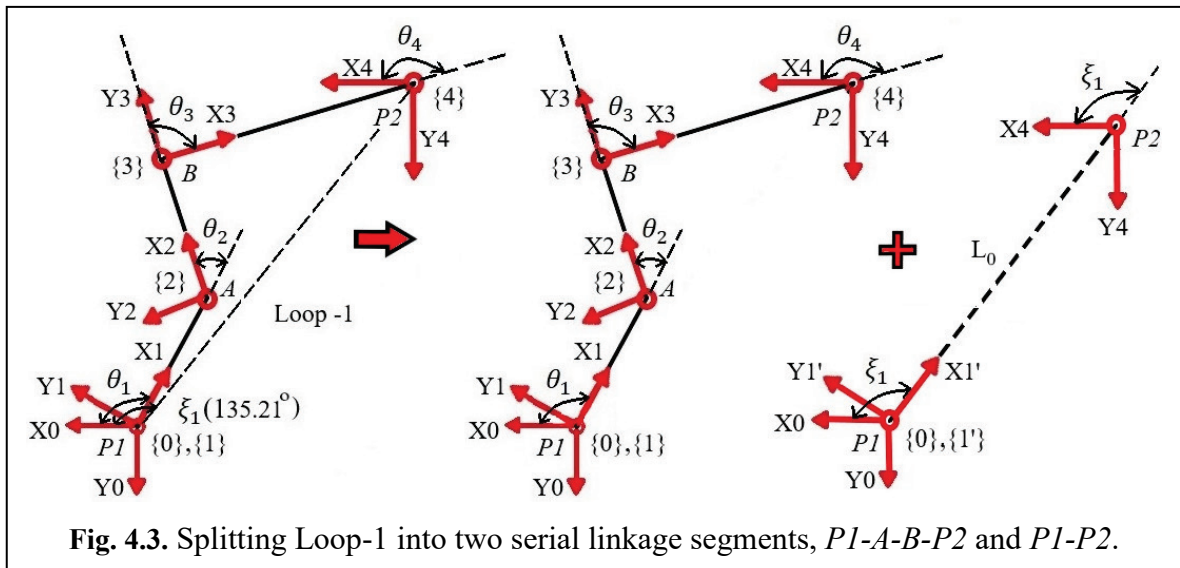


Table 4.1. D-H parameters for serial linkage $P1-A-B-P2$:

Axes	a_i	α_i	d_i	θ_i
1	$L_1 = 12$	0	0	$+\theta_1$
2	$L_2 = 26.5$	0	0	$-\theta_2$
3	$L_3 = 28$	0	0	$+\theta_3$
4	0	0	0	$-\theta_4$

Table 4.2. Joint parameters for serial linkage $P1-P2$:

Axes	a_i	α_i	d_i	θ_i
1'	$L_0 = 44$	0	0	$+\xi_1$
2	0	0	0	$-\xi_1$

For Serial Linkage $P1-A-B-P2$: ${}^0_4T_{s1} = {}^0_1T \cdot {}^1_2T \cdot {}^2_3T \cdot {}^3_4T$ (4.2)

For Serial Linkage $P1-P2$: ${}^0_4T_{s2} = {}^0_1'T \cdot {}^1'_4T$ (4.3)

By Loop-closure: ${}^0_4T_{s1} = {}^0_4T_{s2}$ (4.4)

Comparing each element of the matrix ${}^0_4T_{s1}$ with those of ${}^0_4T_{s2}$ yields the following results:

$$\theta_3 = \cos^{-1} \left(\frac{L_0^2 + L_1^2 - L_2^2 - L_3^2 - 2L_0L_1c(\xi_1 - \theta_1)}{2L_2L_3} \right) \text{ (4.5)}$$

$$\theta_2 = \cos^{-1} \left(\frac{L_0L_2c(\xi_1 - \theta_1) + L_0L_3c(\theta_1 + \theta_3 - \xi_1) - L_1L_2 - L_1L_3c\theta_3}{L_2^2 + L_3^2 + 2L_2L_3c\theta_3} \right) \text{ (4.6)}$$

$$\theta_4 = \theta_1 - \theta_2 + \theta_3 \quad \dots\dots\dots (4.7)$$

Where, $\theta_1 = \theta_v =$ actuation angle of servo-2, and $|\xi_1| = 135.21^\circ$.

Analysis of Loop-2:

A similar analysis is applied for Loop-2, where it is split into two parts, $P2-C-D-P3$ being the 1st serial linkage and $P2-P3$ being the 2nd serial linkage, as shown in Fig 4.4. The D-H parameters are enlisted in Table 4.3 and Table 4.4.

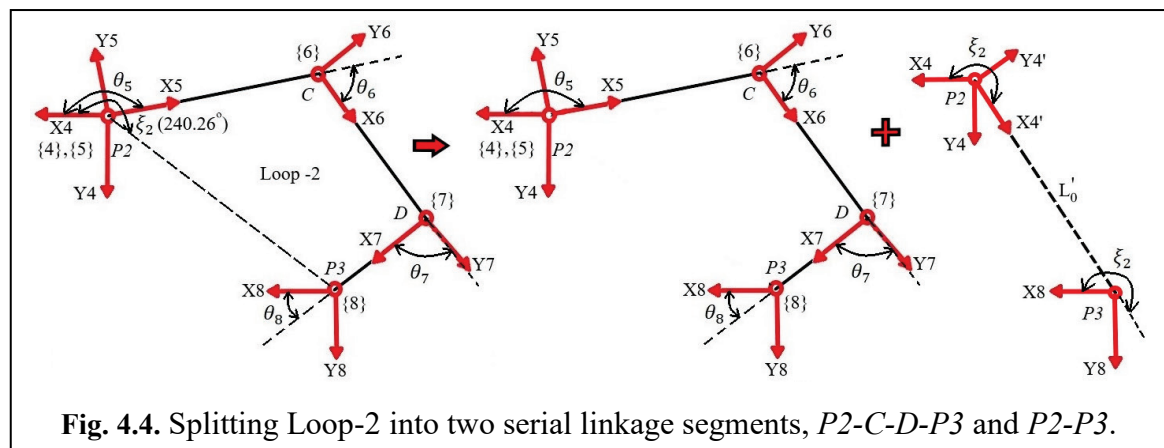


Fig. 4.4. Splitting Loop-2 into two serial linkage segments, $P2-C-D-P3$ and $P2-P3$.

Table 4.3. D-H parameters for serial linkage $P1-A-B-P2$:

Axes	a_i	α_i	d_i	θ_i
5	$L_4 = 12$	0	0	$+\theta_5$
6	$L_5 = 26.5$	0	0	$+\theta_6$
7	$L_6 = 28$	0	0	$+\theta_7$
8	0	0	0	$+\theta_8$

Table 4.4. D-H parameters for serial linkage $P1-P2$:

Axes	a_i	α_i	d_i	θ_i
1'	$L_0' = 48.37$	0	0	$+\xi_2$
2	0	0	0	$-\xi_2$

For Serial Linkage $P2-C-D-P3$: ${}^4T_{s3} = {}^4T_5 \cdot {}^5T_6 \cdot {}^6T_7 \cdot {}^7T_8$ (4.8)

For Serial Linkage $P2-P3$: ${}^4T_{s4} = {}^8T_4 \cdot {}^4T_8$ (4.9)

By Loop-closure: ${}^4_8T_{S3} = {}^4_8T_{S4}$ (4.10)

Comparing each element of the matrix ${}^0_4T_{S3}$ with those of ${}^0_4T_{S4}$ yields the following results:

$\theta_5 = -\theta_4$ (4.11)

$\theta_7 = \cos^{-1} \left(\frac{L_0'^2 + L_4^2 - L_5^2 - L_6^2 - 2L_0'L_4c(\xi_2 - \theta_5)}{2L_5L_6} \right)$ (4.12)

$\theta_6 = \cos^{-1} \left(\frac{L_0'L_5c(\xi_2 - \theta_5) + L_0'L_6c(\theta_5 + \theta_7 - \xi_2) - L_4L_5 - L_4L_6c\theta_7}{L_5^2 + L_6^2 + 2L_5L_6c\theta_7} \right)$ (4.13)

$\theta_8 = 2\pi - (\theta_5 + \theta_6 + \theta_7)$ (4.14)

Where, $|\xi_2| = 240.26^\circ$.

Now that the mathematical expressions of joint angles θ_2 to θ_8 has been obtained for given value of θ_1 , the mechanism can be further simplified by converting into a serial linkage *PI-A-B-C-E-F* as shown in Fig 4.5. The D-H parameters are enlisted in Table 4.5.

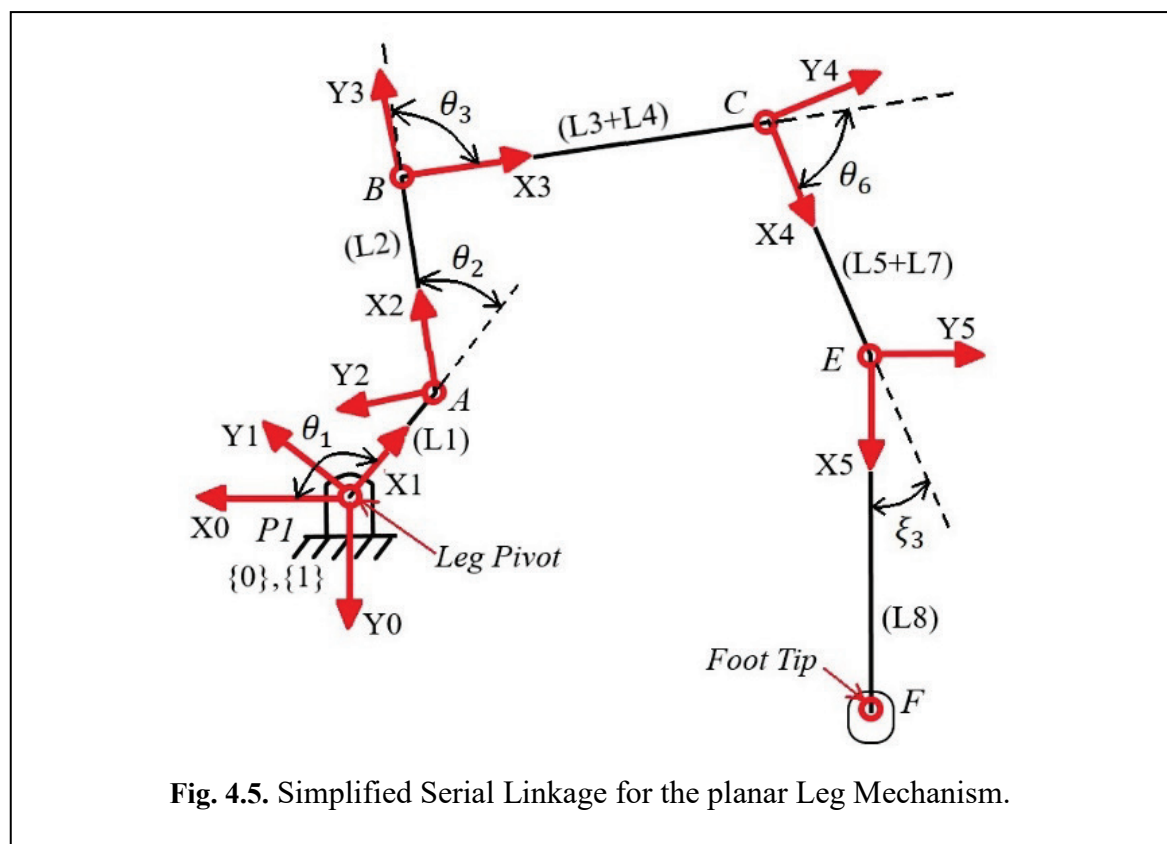


Table 4.5. D-H parameters for simplified serial linkage:

Axes	a_i	α_i	d_i	θ_i
1	$L1 = 12$	0	0	$\theta_1 = \theta_v$
2	$L2 = 26.5$	0	0	θ_2
3	$L3+L4 = 69$	0	0	θ_3
4	$L5+L6 = 94$	0	0	θ_6
5	$L8 = 82.45$	0	0	$\xi_3 = 29.49^\circ$

The coordinates of the foot-tip with respect to the leg pivot can be calculated by substituting the values from Table 4.5 in the relation:

$${}_{leg-pivot}^{foot-tip}T = {}_5^0T = {}_1^0T \cdot {}_2^1T \cdot {}_3^2T \cdot {}_4^3T \cdot {}_5^4T \quad \dots\dots\dots (4.15)$$

Where, ${}_5^0T = \begin{bmatrix} [Rot] & [p] \\ 0 & 1 \end{bmatrix}$ and $[p] = [p1 \ p2 \ 0]^T$

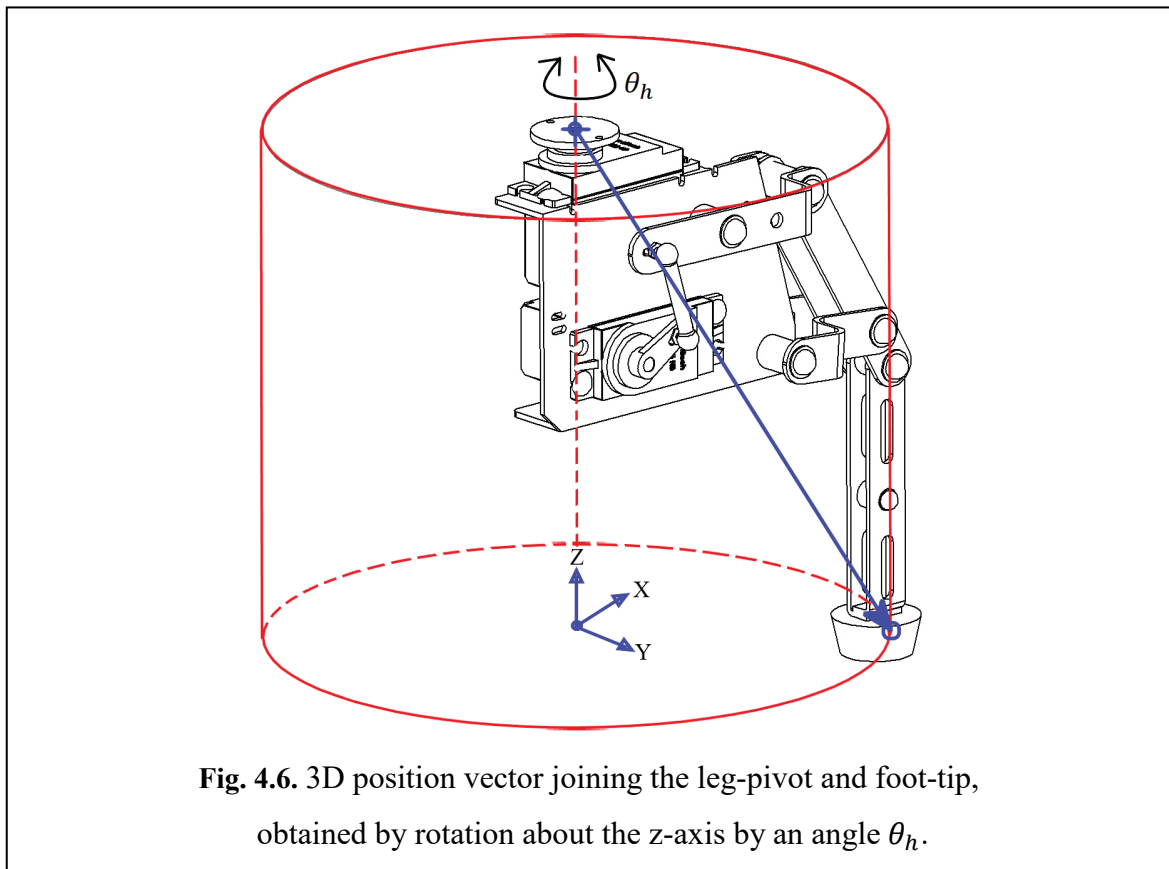
$[Rot]$ and $[p]$ being the rotation matrix and position vector respectively.

So far, leg mechanism had been considered as a planar mechanism neglecting the effect of servo-1. Now, the actuation of servo-1 by angle θ_h needs to be considered. If it is assumed that the planar mechanism lies on the y-z plane, and a vector is drawn that joins l_i and f_i . Futher, it can be considered that the vector is rotated by an angle θ_h about the z-axis, as shown Fig. 4.6. Hence, the position coordinates of foot-tip can be calculated by resolving into components.

The coordinates of the foot-tip f_i is calculated as:

$$f_i = [p1\sin\theta_h \ p1\cos\theta_h \ p2]^T \quad \dots\dots\dots (4.16)$$

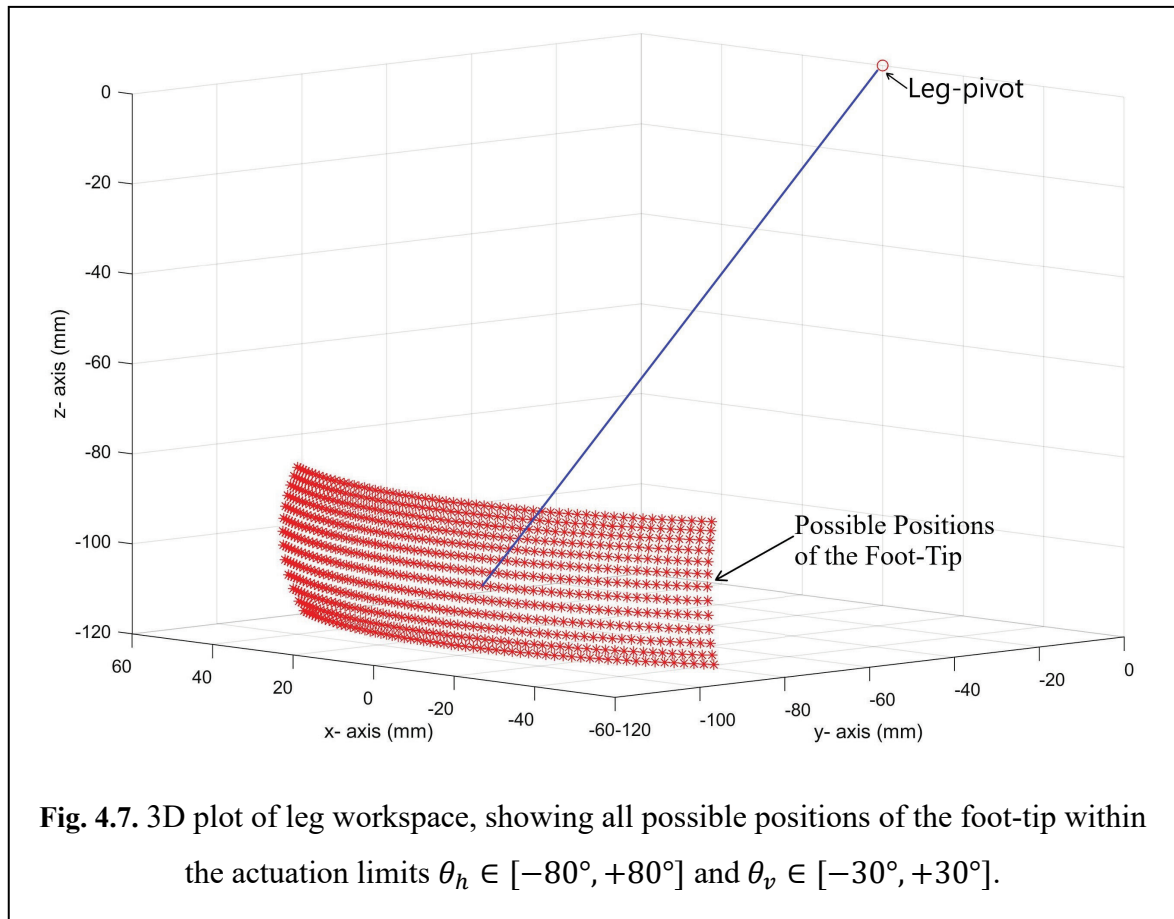
Where, $l_i = [0 \ 0 \ 0]^T$ ∴ Leg pivot is considered as origin.



4. 3. Generating Leg-Workspace

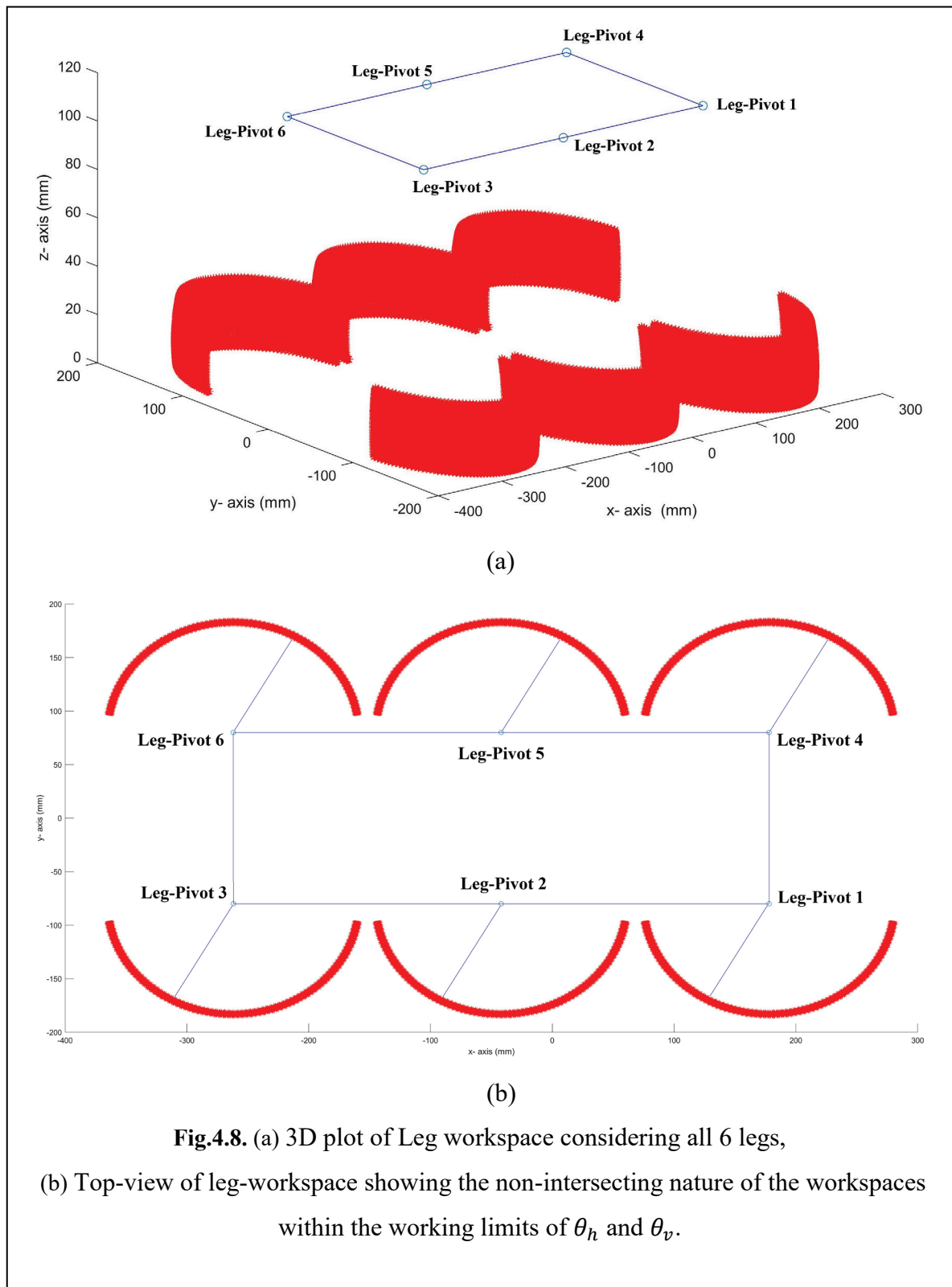
Case – I. Workspace for an individual leg:

The leg-pivot is considered as the origin and coincides with the global frame of reference. Actuation angles θ_h and θ_v is varied within the angular limits -80° to $+80^\circ$ and -30° to $+30^\circ$ respectively. Every combination of θ_h , θ_v values yields a set of data points for the foot-tip. Considering the leg-pivot as the origin, all possible foot-tip positions are recorded and these data points plotted, as shown in Fig 4.7. The leg-mechanism being a 2 degree of freedom system, the workspace is a toroidal surface.



Case – II. Workspace considering all 6 legs:

The Hexcrawler robot has six legs working in parallel. The actuation angles θ_h and θ_v must be such that any two adjacent legs must have non-intersecting workspaces to avoid collision. Hence, the workspace of each of the 6 legs are plotted in a single graph as shown in Fig. 4.8(a) and Fig. 4.8(b). To generate the graph, the leg pivots must coincide with the true geometric positions with respect to the global frame of reference. So, initially, the position of the foot-tip is calculated assuming the leg-pivot as origin and then using ‘shifting of origin’ the true positions are obtained. The simulation result shows that the servo-1 can be safely operated within the angular limits -80° to $+80^\circ$ without having any risk of collisions.



4.4. Conclusion

From the kinematic analysis of the leg mechanism it is possible to obtain the positions of foot-tip f_i when leg-pivot l_i and actuation angles θ_h and θ_v are defined. This relation will be vital for the establishing the kinematics of the Hexcrawler. Further, it has been identified that actuation of servo-1 within the limits -80° to $+80^\circ$ yields a non-intersecting workspace and collision can be avoided. The working limits for servo-2 will be taken as -30° to $+30^\circ$ by default unless mentioned otherwise.

Chapter 5.

Kinematic Analysis of the Hexcrawler.

In this chapter, the aim is to develop a kinematic model capable of describing the motion of the robot for different gait patterns. The Hexcrawler robot has six legs each of which is a serial manipulator. The six legs together is responsible for the motion of the robot body, making the system a parallel manipulator. So, 12 independent actuators result into a unique motion of the robot. For simplicity, the chapter has been divided into 3 parts:

- I. Mathematical Relations
- II. Forward Kinematic Model
- III. Inverse Kinematic Model

5.1. Motions of the Hexcrawler

The Hexcrawler robot possess 6 degrees of freedom- surge, sway, heave, roll, pitch and yaw, as shown in Fig. 5.1. These motions are achievable by providing suitable inputs to the 12 servo actuators. Table. 1 describes the motions in detail.

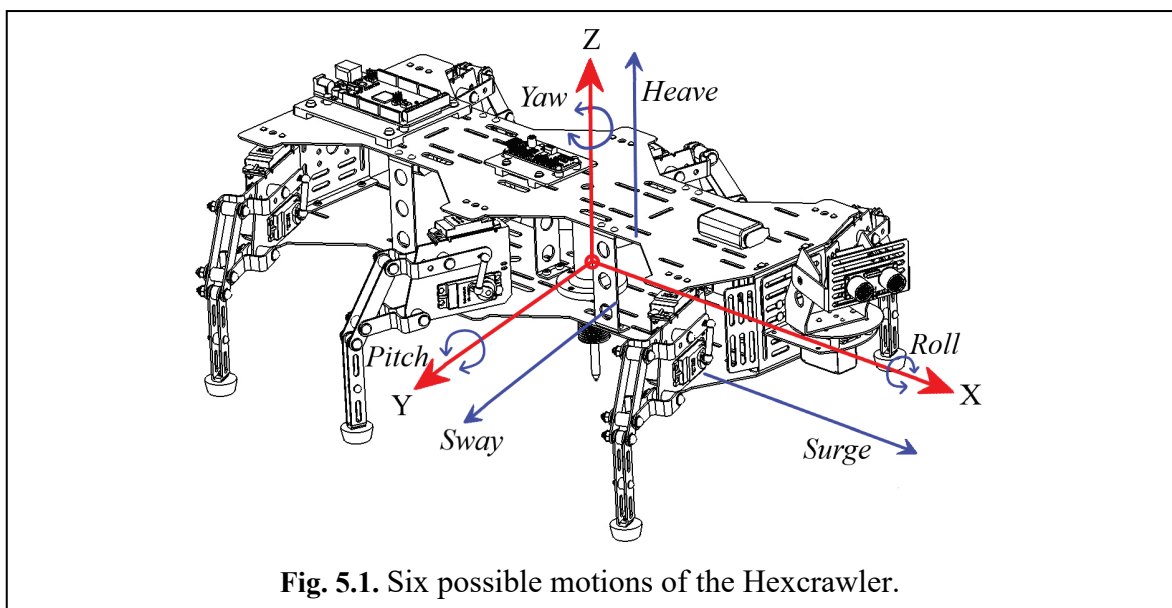


Fig. 5.1. Six possible motions of the Hexcrawler.

Table 5.1. Types of motion of the Hexcrawler:

D.O.F	Motion Type	Description	Position/Euler Angle
1	Surge	Translation along x-axis	X
2	Sway	Translation along y-axis	Y
3	Heave	Translation along z-axis	Z
4	Roll	Rotation about x-axis	α
5	Pitch	Rotation about y-axis	β
6	Yaw	Rotation about z-axis	γ

PART –I. Mathematical Relations.

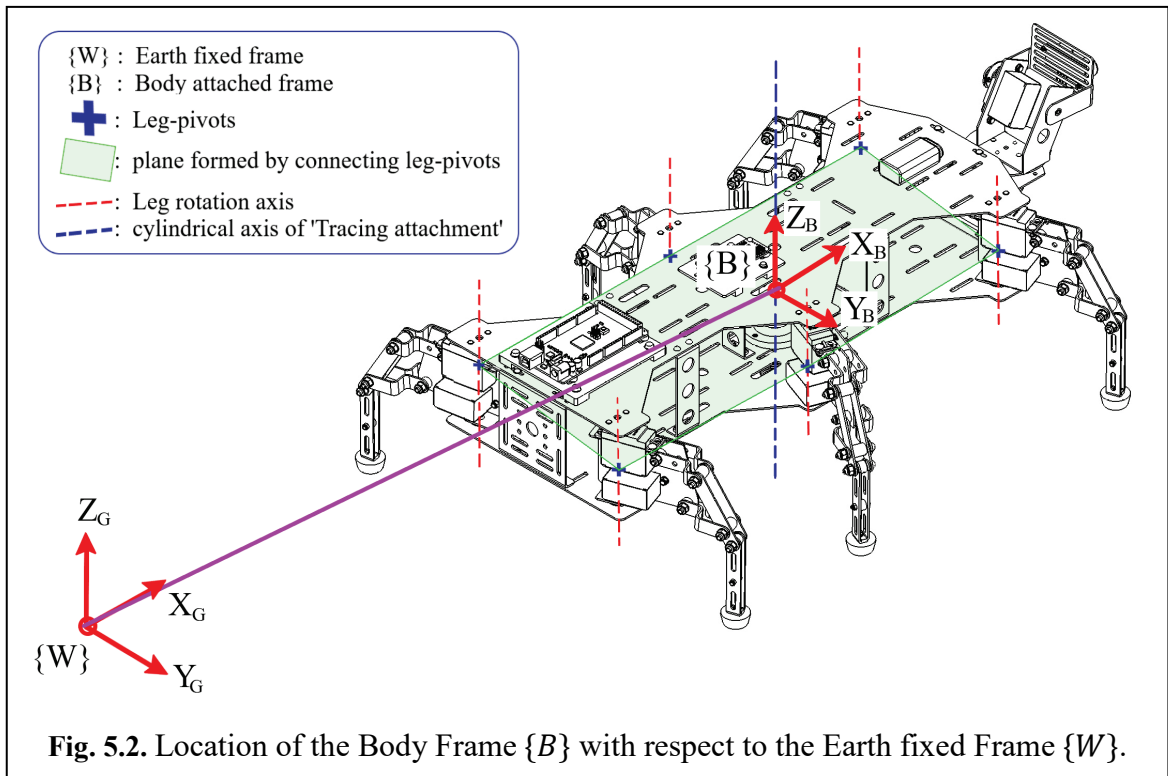
To establish the forward and inverse kinematic model, a few mathematical relations must be derived. These relations would help relate the motion of an individual leg to the motion of the Hexcrawler at large.

5.2. Frames of Reference

The Hexcrawler robot being a mobile system, requires two frames of reference to describe the motion of the system.

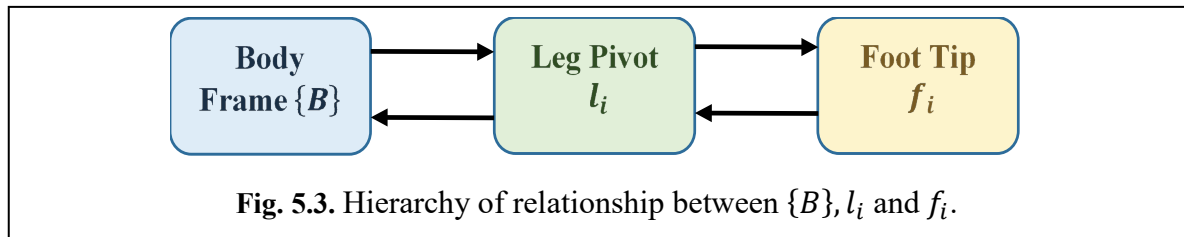
1. Body frame of reference $\{B\}$: a non-inertial, moving frame attached to the robot. It coincides with the geometric center of the body. Position and orientation described with respect to this frame is designated by $[X_B \ Y_B \ Z_B \ \gamma_B]$.
2. Earth fixed frame of reference $\{W\}$: an inertial frame of reference with respect to which the Body frame $\{B\}$ is defined. Position and orientation described with respect to this frame is designated by $[X_G \ Y_G \ Z_G \ \gamma_G]$.

Both $\{W\}$ and $\{B\}$ frames are in right-handed Cartesian coordinate system as shown in Fig. 5.2. The body frame of reference $\{B\}$ lies on the plane formed by joining the leg-pivots, and coincides with the cylindrical axis of ‘Tracing attachment’. Although $\{B\}$ does not coincide with the geometric center of the system, but it simplifies calculations for forward and inverse kinematics of the robot.



5.3. Inter-relationship between $\{B\}$, l_i and f_i

To control the motion of the robot and enable it to walk, forward and inverse kinematic solutions of the robot is required to be calculated. Forward kinematics involves finding the location of the robot platform when the time history of joint angles are known. While Inverse kinematics involves computing the joint angles when the initial and final positions are defined. To establish the forward and inverse kinematics of the robot, it is required to establish a few important relations. To specify the position and posture of the robot three parameters are required – the position and orientation of the body frame $\{B\}$ w.r.t. $\{W\}$, the position of leg pivots l_i and the position of foot-tips f_i for all six legs. Let θ_{hi} and θ_{vi} be the actuation angle of Servo-1 and Servo-2 of the i^{th} leg defined w.r.t. $\{B\}$. Leg pivot is defined as $l_i = [x_{li} \ y_{li} \ z_{li}]^T$, foot tip is defined as $f_i = [x_{fi} \ y_{fi} \ z_{fi}]^T$, and the body frame is defined as $\{B\} = [x_B \ y_B \ z_B \ \gamma]^T$. Mathematical relations must exist to evaluate $\{B\}$, l_i or f_i when any one is defined and the actuation angles θ_{hi} and θ_{vi} are known. Fig. 5.3 shows the hierarchy of relation between $\{B\}$, l_i and f_i . The subsequent section establishes these relationship mathematically.



5.3.1. Calculating l_i when $\{B\}$ is defined:

The leg pivots can be located locally i.e. w.r.t $\{B\}$ using a combination of d_i and λ_i as shown in Fig. 5.4. From geometry, the values of d_i and λ_i are enlisted in Table 5.2.

If, $\{B\} = [x_B \ y_B \ z_B \ \gamma]^T$

Then, $l_i = [(x_B + d_i \cos(\lambda_i + \gamma)) \ (y_B + d_i \sin(\lambda_i + \gamma)) \ z_B]^T \dots\dots (5.1)$

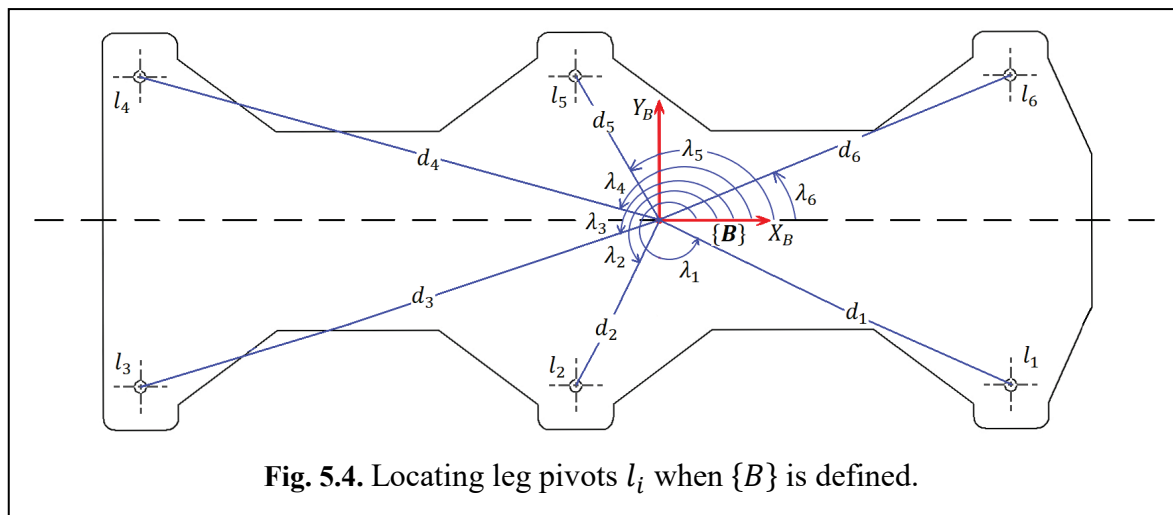


Table 5.2. Values of d_i and λ_i (obtained from geometry of the Hexcrawler):

d_i in mm			λ_i in degrees		
$d_1 = 195.15$	$d_2 = 90.35$	$d_3 = 273.94$	$\lambda_1 = 335.8^\circ$	$\lambda_2 = 242.3^\circ$	$\lambda_3 = 197.0^\circ$
$d_4 = 195.15$	$d_5 = 90.35$	$d_6 = 273.94$	$\lambda_4 = 24.2^\circ$	$\lambda_5 = 117.7^\circ$	$\lambda_6 = 163.0^\circ$

5.3.2. Calculating f_i when l_i is defined:

The method of calculating the position of foot-tip f_i when the leg pivot l_i and actuator angles θ_{hi} and θ_{vi} is known, has been elaborately described in the chapter 4. However a few modifications are necessary. The Hexcrawler robot when viewed from the rear end: the legs

1,2,3 are the right sided legs and legs 4,5,6 are left sided legs. To reduce complexity, the angles θ_{hi} and θ_{vi} are defined in the same way for both right and left sided legs, refer to Fig. 5.5. This requires a slight modification to the formulas derived previously.

Let the leg pivot be defined as: $l_i = [x_{li} \ y_{li} \ z_{li}]^T$

The foot-tips can be evaluated using the relation:

$$f_i|_{\text{right-sided legs}} = [(x_{li} - p1\sin\theta_h) \ (y_{li} + p1\cos\theta_h) \ (z_{li} + p2)]^T \quad \dots\dots (5.2a)$$

$$f_i|_{\text{left-sided legs}} = [(x_{li} - p1\sin\theta_h) \ (y_{li} - p1\cos\theta_h) \ (z_{li} + p2)]^T \quad \dots\dots (5.2b)$$

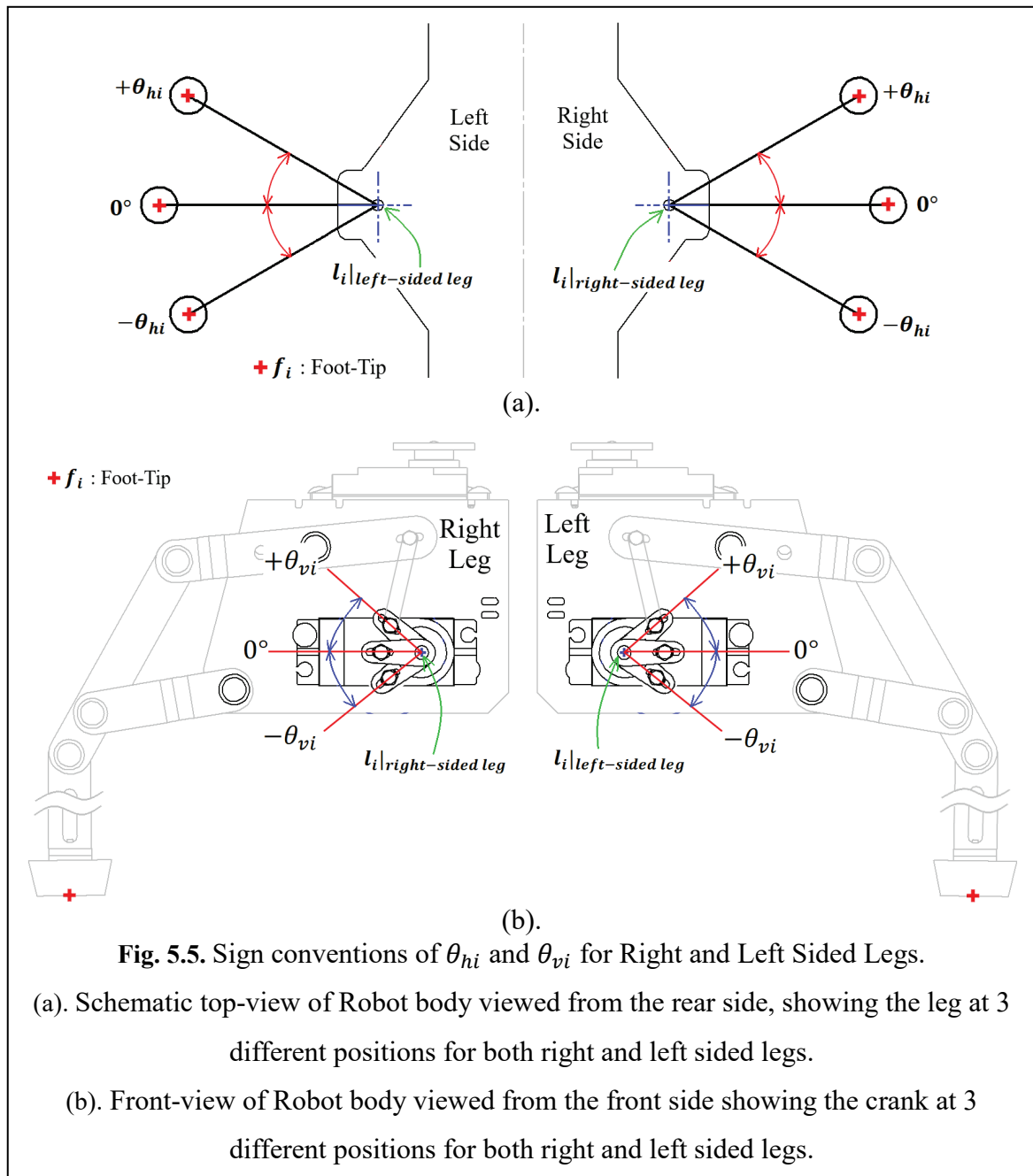


Fig. 5.5. Sign conventions of θ_{hi} and θ_{vi} for Right and Left Sided Legs.

(a). Schematic top-view of Robot body viewed from the rear side, showing the leg at 3 different positions for both right and left sided legs.

(b). Front-view of Robot body viewed from the front side showing the crank at 3 different positions for both right and left sided legs.

5.3.3. Calculating l_i when f_i is defined:

The method of evaluating l_i when f_i , θ_{hi} and θ_{vi} is known is simple. Let \vec{p} be a vector joining points l_i to f_i . Then the vector from l_i to f_i will be $-\vec{p}$. The position of l_i is calculated using this analogy with the previously derived relations.

Let the foot-tip be defined as: $f_i = [x_{fi} \ y_{fi} \ z_{fi}]^T$

The leg-pivot can be evaluated using the relation:

$$l_i|_{right} = [(x_{li} + p1\sin\theta_h) \ (y_{li} - p1\cos\theta_h) \ (z_{li} - p2)]^T \quad \dots\dots (5.3a)$$

$$l_i|_{left} = [(x_{li} + p1\sin\theta_h) \ (y_{li} + p1\cos\theta_h) \ (z_{li} - p2)]^T \quad \dots\dots (5.3b)$$

5.3.4. Calculating $\{B\}$ when l_i is defined:

Evaluating $\{B\}$ for a known l_i can be tricky and is required only in the Stance phase. $\{B\}$ is a combination of position coordinates $[x_B \ y_B \ z_B]$ and yaw angle γ . The first challenge is to calculate γ . For this, three positions of l_i is must be known. On actuating a grounded leg by θ_{hi} , let the initial leg pivot position be given by $l_i|_{initial}$. After actuation the new position is $l_i|_{final}$. A third point is required, which can be obtained by evaluating the leg-pivot at $\theta_{hi}/2$. Let this position be given by $l_i|_{mid}$. Now using the points $l_i|_{initial}$, $l_i|_{mid}$ and $l_i|_{final}$ a circle is constructed that passes through all three points. The tangent drawn to the circle at $l_i|_{final}$ gives the direction or the yaw angle γ of the robot. Fig 5.6 illustrates the idea of yaw angle calculation. Finally, the position coordinates $[x_B \ y_B \ z_B]$ can be calculated using an inverse relation.

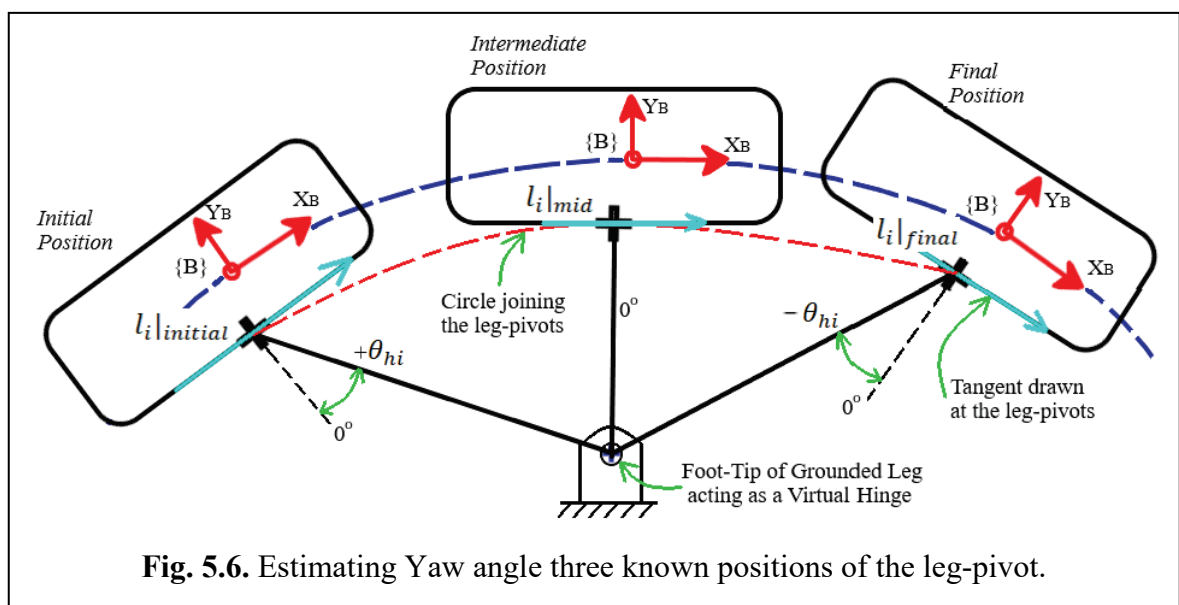


Fig. 5.6. Estimating Yaw angle three known positions of the leg-pivot.

Let,

$$l_i|_{initial} = [x_1 \ y_1 \ z_B]^T, \quad l_i|_{mid} = [x_2 \ y_2 \ z_B]^T, \quad l_i|_{final} = [x_3 \ y_3 \ z_B]^T$$

∴ $z_{li} = z_B$, for planar maneuvering.

In the x-y plane, the equation of circle is given by:

$$x^2 + y^2 + 2gx + 2fy + c = 0 \quad \dots\dots (5.4)$$

Passing through points: (x_1, y_1) , (x_2, y_2) , (x_3, y_3)

Substituting the points in the circle eq. 5.4, the coefficients g , f and c is given by:

$$\begin{bmatrix} g \\ f \\ c \end{bmatrix} = - \begin{bmatrix} 2x_1 & 2y_1 & 1 \\ 2x_2 & 2y_2 & 1 \\ 2x_3 & 2y_3 & 1 \end{bmatrix}^{-1} \begin{bmatrix} (x_1^2 + y_1^2) \\ (x_2^2 + y_2^2) \\ (x_3^2 + y_3^2) \end{bmatrix} \quad \dots\dots (5.5)$$

The tangent equation is given by:

$$xx_T + yy_T + g(x + x_T) + f(y + y_T) + c = 0 \quad \dots\dots (5.6)$$

Where, (x_T, y_T) is the point on the circle at which the tangent is to be constructed.

Setting, $(x_T, y_T) = (x_3, y_3)$

The slope of the tangent, $m = -\left(\frac{x_T+g}{y_T+f}\right) \quad \dots\dots (5.7)$

And, Yaw angle, $\gamma = \tan^{-1}(m) \quad \dots\dots (5.8)$

Finally,

$$\{B\} = [x_B \ y_B \ z_B \ \gamma]^T = [(x_3 - d_i \cos(\lambda_i + \gamma)) \ (y_3 - d_i \sin(\lambda_i + \gamma)) \ z_B \ \gamma]^T \quad \dots\dots (5.9)$$

5.3.5. Estimating Actuation Angles $\{\theta_{hi}, \theta_{vi}\}$ when l_i and f_i is defined:

Inverse kinematics is requires to computing the servo actuation values, when the position of the foot-tip and leg-pivot is defined. For the i^{th} leg, θ_{hi} is obtained from the projection of the vector joining l_i and f_i on the x-y plane.

Let, $\{B\} = [x_B \ y_B \ z_B \ \gamma]^T$, $l_i = [x_{li} \ y_{li} \ z_{li}]^T$ and $f_i = [x_{fi} \ y_{fi} \ z_{fi}]^T$

Considering the projection on the x-y plane,

$$\theta_{hi} = -\tan^{-1}\left(\frac{x_{fi} - x_{li}}{y_{fi} - y_{li}}\right) - \gamma, \quad \text{for Right sided legs} \quad \dots\dots (5.10a)$$

$$= \tan^{-1}\left(\frac{x_{fi} - x_{li}}{y_{fi} - y_{li}}\right) + \gamma, \quad \text{for Left sided legs} \quad \dots\dots (5.10a)$$

Estimating θ_{vi} is more complex since, the relation ${}_{foot-tip}^{leg-pivot}T$ cannot be used directly. An alternate approach is the use of root-search using optimization techniques. The search

domain is $\theta_{vi} \in \{-80^\circ, 80^\circ\}$. The foot-tip position f_{i-calc} is calculated w.r.t leg-pivot l_i for an arbitrary value of θ_{vi} . The distance between f_{i-calc} and f_i is the error ϵ .

$$\text{Error, } \epsilon = |f_i - f_{i-calc}| \quad \dots\dots (5.11)$$

Minimizing the square of error would give a possible solution for θ_{vi} . Golden section method is used as the optimization technique, since it converges rapidly and does not require evaluating derivatives. The solver has been programmed using MatLab R2017b and produces fairly acceptable results.

Mathematically, the optimization problem is stated as:

$$\text{Minimize } \epsilon^2$$

$$\text{With } \theta_{vi} \text{ as the optimization variable and } -80^\circ \leq \theta_{vi} \leq 80^\circ$$

PART – II. Forward Kinematics

To control the motion of the robot and to enable it to walk, forward and inverse kinematic solutions of the robot is required to be calculated. Forward kinematics (FK) involves finding the location of the robot platform when the time history of joint angles are known. While formulating the forward kinematic model it is assumed that the robot undergoes planar maneuvering. So roll and pitch can be neglected ($\alpha = \beta = 0$) and z_B is constant, where z_B represents the height of the robot body, measured as the distance of the virtual plane on which the leg pivots lie, from the earth fixed reference frame along Z_G .

5.4. Analysis of Tripod Gait

For any gait pattern there are two basic phases – Stance and Swing. A programmed sequence of alternate Stance and Swing phases results in the walking motion of the body. The position/orientation of body frame $\{B\}$, leg pivot l_i or foot-tip f_i is updated at each phase. Where i indicates leg number $i \in \{1,2 \dots 6\}$. The two phases can be described as:

1. **Stance phase:** When an input is provided to the actuators of a grounded leg, the foot-tip acts as hinge, about which the leg pivot is displaced. The leg-pivot being rigidly connected with the robot frame, results in a motion of the robot platform. It is in this phase that the robot moves from one position to the other. So mathematically the

foot-tips f_j remaining fixed but body frame $\{B\}$ and leg pivot l_j changes in this motion phase, j indicating the legs under Stance phase.

2. **Swing phase:** When an input is provided to the actuators of a leg in-air (ungrounded), the leg pivot acts as a hinge about which the foot-tip is displaced. Hence the body frame also remains fixed. This phase can be mathematically realised as a motion that results in the body frame $\{B\}$ and leg pivot l_k remaining fixed but foot-tips f_k changes, k indicating the legs under Swing phase.

Tripod Gait is used as the primary walking technique. In this gait, the weight of the robot shifts alternately from one tripod to another. Legs 1,3,5 and 2,4,6 form two stable tripod configurations. Stability is assured as the center of gravity remains well inside the support triangle formed by the tripods 1,3,5 and 2,4,6. So for both stance and swing phases, the three legs of the tripod are simultaneously actuated. However the actuation may be equal or unequal in magnitude and direction. This leads to straight line and curved path motion respectively. The step sequence for tripod Gait has been enlisted in Table 5.3. This sequence is especially important to establish the proposed algorithm. The motion analysis in these two phases for the tripod gait is discussed in the subsequent section.

Table 5.3. Step Sequence per cycle for Tripod Gait:

Step	Sequence	Description
1	Stance of Legs 1,3,5	Grounded legs 1,3,5 are actuated resulting in the robot body being displaced by a proportionate amount.
2	Swing of Legs 2,4,6	In-Air legs 2,4,6 are actuated, preparing them for the next Stance phase.
3	Change Support	Support tripod changes from legs 1,3,5 to legs 2,4,6.
4	Stance of Legs 2,4,6	Grounded legs 2,4,6 are actuated resulting in the robot body being displaced by a proportionate amount.
5	Swing of Legs 1,3,5	In-Air legs 1,3,5 are actuated, preparing them for the next Stance phase.
6	Change Support	Support tripod changes from legs 2,4,6 to legs 1,3,5.

5.5. FK Solver Algorithm

A simplified approach towards evaluating the forward kinematics (FK) of the robot has been presented using an algorithm. Given an initial state of the robot, the FK Solver will estimate the final state of the robot for a given set of joint actuation. The algorithm utilizes relations between $\{B\}$, l_i and f_i which has been derived previously.

Proposed Algorithm:

Step 1. Define a 'Starting state' for the robot by setting values of θ_{hi} and θ_{vi} for $i \in \{1, \dots, 6\}$.

Set a feasible value of z_B .

Obtain values of l_i and f_i .

Coincide $\{B\}$ with $\{W\}$. Hence $\{B\} = [0 \ 0 \ z_B \ \gamma]^T$,

The yaw angle γ can be set to any desired value.

Step 2. For Stance 1-3-5, keeping f_1, f_3, f_5 constant, l_i is evaluated for a given θ_{hi} .

The body frame $\{B\}$ is located for the new positions of l_i .

Step 3. In Swing of 2-4-6, $\{B\}$ and l_i remain constant.

f_2, f_4, f_6 is evaluated for a given θ_{hi} .

Step 4. Legs 1,3,5 is raised in-air and legs 2,4,6 are grounded by changing values of θ_{vi} .

$\{B\}$ and l_i remain constant.

f_i is updated.

Step 4. For Stance 2-4-6, keeping f_2, f_4, f_6 constant, l_i is evaluated for a given θ_{hi} .

The body frame $\{B\}$ is located for the new positions of l_i .

Step 5. In Swing of 1-3-5, $\{B\}$ and l_i remain constant.

f_1, f_3, f_5 is evaluated for a given θ_{hi} .

Step 6. Legs 2-4-6 are raised in-air and legs 1-3-5 are grounded by changing values of θ_{vi} .

$\{B\}$ and l_i remain constant.

f_i is updated.

5.6. Analysis of Stance Phase

The stance phase is responsible for motion of the Hexcrawler robot. For tripod gait, 3legs are simultaneously actuated. The actuation may be equal or unequal in both magnitude and direction. By direction, clockwise (CW) or counter-clockwise (CCW) rotation is suggested. Legs 1,3,5 and 2,4,6 form the two stable tripods. Here it is assumed that legs 1,3 and legs

4,6 have equal actuation values i.e. $\theta_{h1} = \theta_{h3}$ and $\theta_{h4} = \theta_{h6}$. So, while working with Stance of legs 1,3,5: leg-1 and leg-5 will be considered, and while working with Stance of legs 2,4,6: leg-2 and leg-4 will be considered for analysis. Table 5.4 enlists the possible cases that may arise with the values of θ_{h1} and θ_{h5} for Stance of 1,3,5. Stance 2,4,6 can be analysed in an exactly similar way, hence details is omitted.

Table 5.4. Cases that may arise with the values of θ_{h1} and θ_{h5} :

Sl. No.	Magnitude	Direction
Case - 1	$\theta_{h1} = \theta_{h5}$	θ_{h1}, θ_{h5} have CW rotation,
Case - 2.	a. $\theta_{h1} > \theta_{h5}$	OR, θ_{h1}, θ_{h5} have CCW rotation.
	b. $\theta_{h1} < \theta_{h5}$	
Case - 3	$\theta_{h1} \neq \theta_{h5}$	If, θ_{h1} is CW, then θ_{h5} is CCW. If, θ_{h1} is CCW, then θ_{h5} is CW.

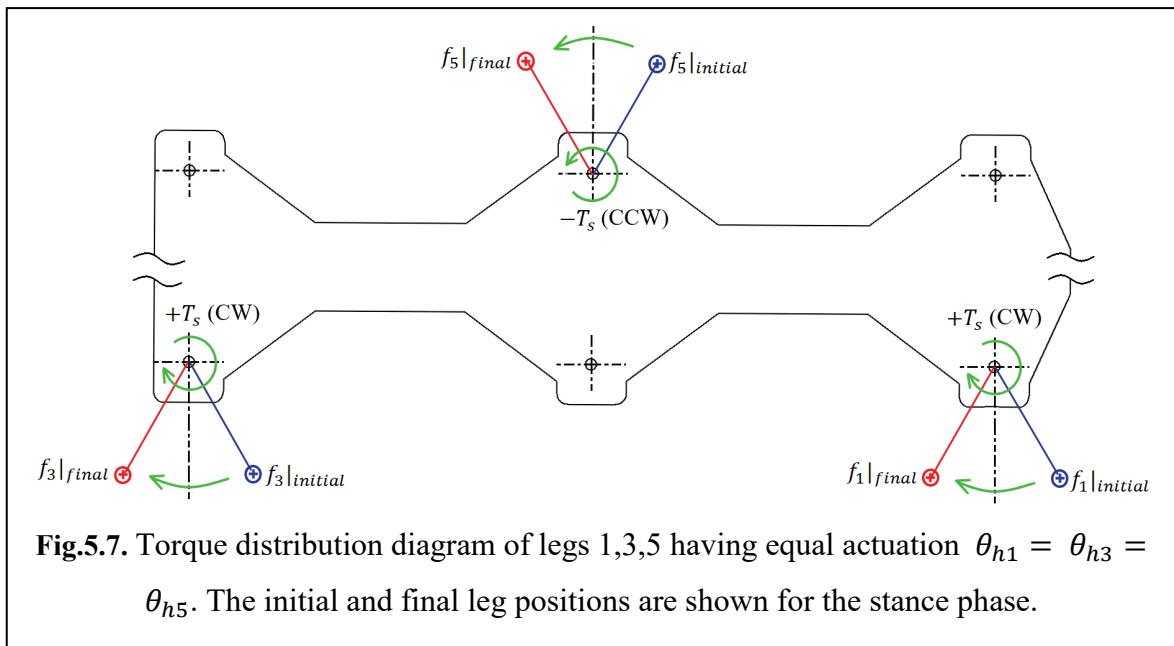
Case – 1: Actuation of equal magnitude and same direction.

In this case $\theta_{h1} = \theta_{h3} = \theta_{h5}$. The servos when actuated generate a torque proportional to the rotation angle. So, in this case equal amounts of torque is generated by legs 1,3,5. If T_s be the torque generated by each servo, then $2T_s$ is the total torque generated in the right side, and $-T_s$ torque is generated in the left side of the robot body, refer Fig 5.7. Hence the net torque will be T_s acting at the right side. So, the motion analysis of either leg-1 or leg-3 is sufficient to describe the motion of the robot body.

Considering leg-1 for stance 1,3,5. The steps of analysis are:

- Keeping f_1 fixed,
Calculate $l_1|_{initial}$, $l_1|_{mid}$ and $l_1|_{final}$ for $\theta_{h1}|_{initial}$, $\theta_{h1}/2$ and θ_{h1} .
- Calculate $\{B\}|_{new}$ from $l_1|_{initial}$, $l_1|_{mid}$ and $l_1|_{final}$.
- Calculate the updated positions of the leg pivots l_i , $i \in \{1,2, \dots, 6\}$ w.r.t $\{B\}|_{new}$.
- Locate the new foot-tips $\{f_2, f_4, f_6\}$ for the legs in-air.

The foot-tips $\{f_1, f_3, f_5\}$ remain unchanged since the legs are grounded (legs 2-4-6 for Stance 2-4-6). The analysis for Stance of Legs 2,4,6 can be done by replacing Leg-1 by Leg-4, Leg-3 by Leg-6 and Leg-5 by Leg-2.



Case – 2: Actuation of unequal magnitude but same direction.

The legs 1 & 5 will be considered for analysis as in case-1. Let $\theta_{h1}|_{initial}, \theta_{h5}|_{initial}$ be the angles of the legs before the Stance 1,3,5 phase, and $\theta_{h1}|_{final}, \theta_{h5}|_{final}$ be the angle of the legs after the Stance 1,3,5 phase. Since the net angular displacement of legs 1 & 5 are different in magnitude but have the same direction, two conditions arise:

- a. $\Delta\theta_{h1} > \Delta\theta_{h5}$, and $\theta_{h1}|_{initial} > \theta_{h5}|_{initial} > \theta_{h5}|_{final} > \theta_{h1}|_{final}$
- b. $\Delta\theta_{h5} > \Delta\theta_{h1}$, and $\theta_{h5}|_{initial} > \theta_{h1}|_{initial} > \theta_{h1}|_{final} > \theta_{h5}|_{final}$

The two cases are analysed separately as Case 2.a. and Case 2.b and shown in Fig 5.8.

Case 2.a. The motion can be divided into 3 distinct segments:

- i. $\theta_{h1}|_{initial}$ to $\theta_{h5}|_{initial}$: actuation for legs 1,3 only.
- ii. $\theta_{h5}|_{initial}$ to $\theta_{h5}|_{final}$: simultaneous actuation for legs 1,3,5.
- iii. $\theta_{h5}|_{final}$ to $\theta_{h1}|_{final}$: actuation for legs 1,3 only.

For segment i. & iii. Leg-5 is unactuated but it is grounded, so it can be thought of as a rigid link. The robot moves due to actuation at legs 1,3. Leg-5 being a rigid link attached to the robot frame, also moves with the frame. The motion of leg-5 in this case can be called “Dragging” as shown in Fig. 5.9. The situation is exactly similar to human walking, where a person does not lift his feet up from the ground when walking, or ‘he drags his feet while

walking'. Dragging is a dynamic mechanism where friction plays a vital role. The effect of friction is neglected since, the dynamics is not being considered. In dragging, the foot-tip is updated w.r.t. the leg pivot for the leg being dragged.

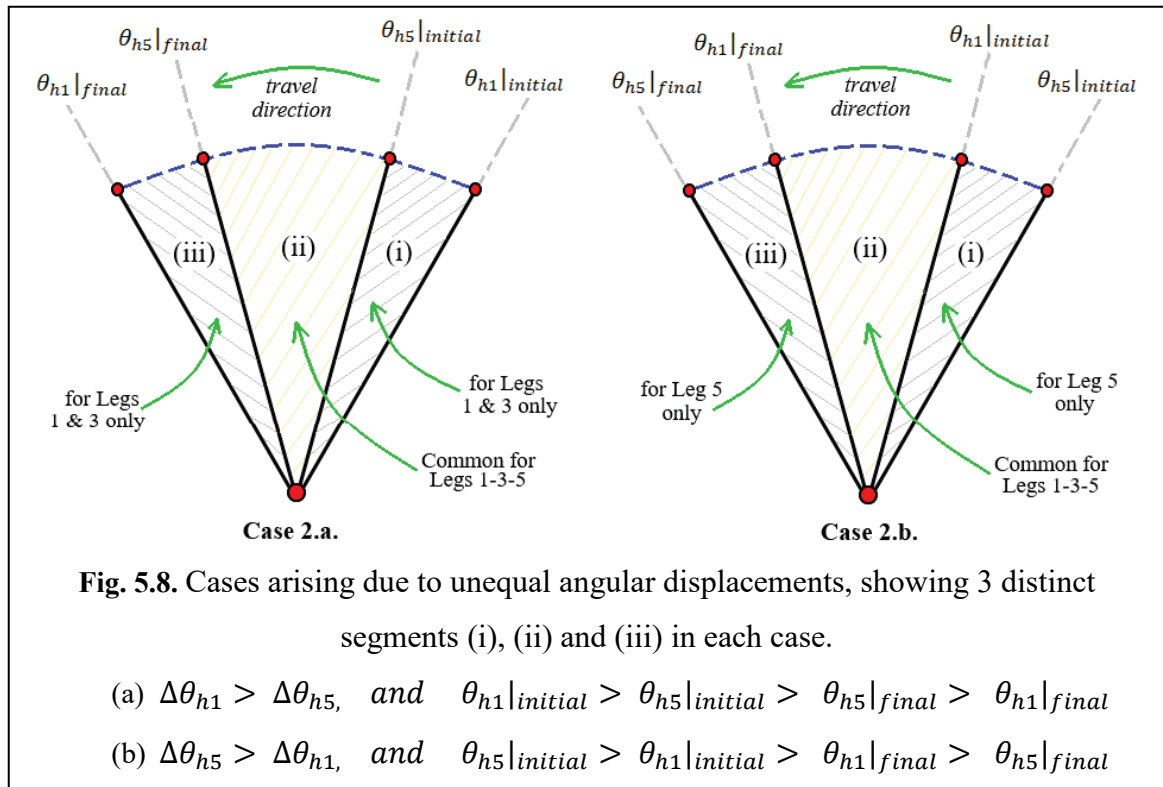
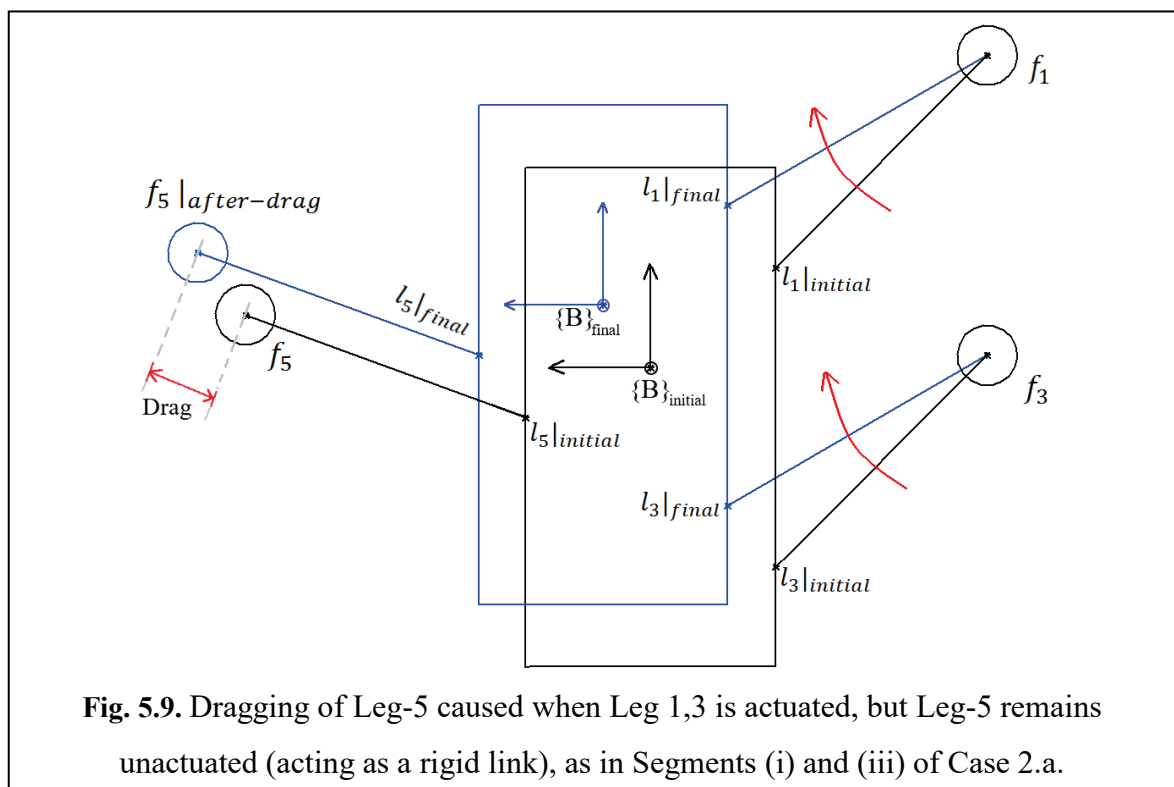


Fig. 5.8. Cases arising due to unequal angular displacements, showing 3 distinct segments (i), (ii) and (iii) in each case.

- (a) $\Delta\theta_{h1} > \Delta\theta_{h5}$, and $\theta_{h1}|_{initial} > \theta_{h5}|_{initial} > \theta_{h5}|_{final} > \theta_{h1}|_{final}$
- (b) $\Delta\theta_{h5} > \Delta\theta_{h1}$, and $\theta_{h5}|_{initial} > \theta_{h1}|_{initial} > \theta_{h1}|_{final} > \theta_{h5}|_{final}$



The steps of analysis are:

- For Case 2.a.i. the foot-tips f_1 and f_3 is kept fixed.
 1. Considering leg-1, calculate $l_1|_{initial}$, $l_1|_{mid}$ and $l_1|_{final}$ for $\theta_{h1}|_{initial}$, $(\theta_{h1}|_{initial} + \theta_{h5}|_{initial})/2$ and $\theta_{h5}|_{initial}$
 2. Calculate $\{B\}|_{new-i}$ from $l_1|_{initial}$, $l_1|_{mid}$ and $l_1|_{final}$
 3. Calculate the updated positions of the leg pivots l_i , $i \in \{1,2, \dots, 6\}$ w.r.t $\{B\}|_{new-i}$
 4. Locate the new foot-tips $\{f_2, f_4, f_6\}$ for the legs in-air, and for grounded leg-5 $\{f_5\}$ considering the drag.
- For Case 2.a.ii. perform analysis as stated in Case -1, with the limits $\theta_{h5}|_{initial}$ to $\theta_{h5}|_{final}$, since the legs 1-3-5 have equal actuations.
 1. Obtain the new positions of $\{B\}|_{new-ii}$ and l_i , $i \in \{1,2, \dots, 6\}$ w.r.t $\{B\}|_{new-ii}$
 2. The foot-tips $\{f_1, f_3, f_5\}$ remains unchanged
- For Case 2.a.iii. perform an exactly similar analysis as in Case 2.a.i. with the limits $\theta_{h5}|_{final}$ to $\theta_{h1}|_{final}$.

Case 2.b. The motion can be divided into 3 distinct segments:

- i. $\theta_{h5}|_{initial}$ to $\theta_{h1}|_{initial}$: actuation for leg 5 only.
- ii. $\theta_{h1}|_{initial}$ to $\theta_{h1}|_{final}$: simultaneous actuation for legs 1,3,5.
- iii. $\theta_{h1}|_{final}$ to $\theta_{h5}|_{final}$: actuation for leg 5 only.

For segment i. & iii. Leg-1,3 is unactuated but it is grounded, so it can be thought of as a rigid link. The robot moves due to actuation of leg 5 only. Legs 1,3 undergoes dragging similar to that described in Case 2.a.

The steps of analysis are:

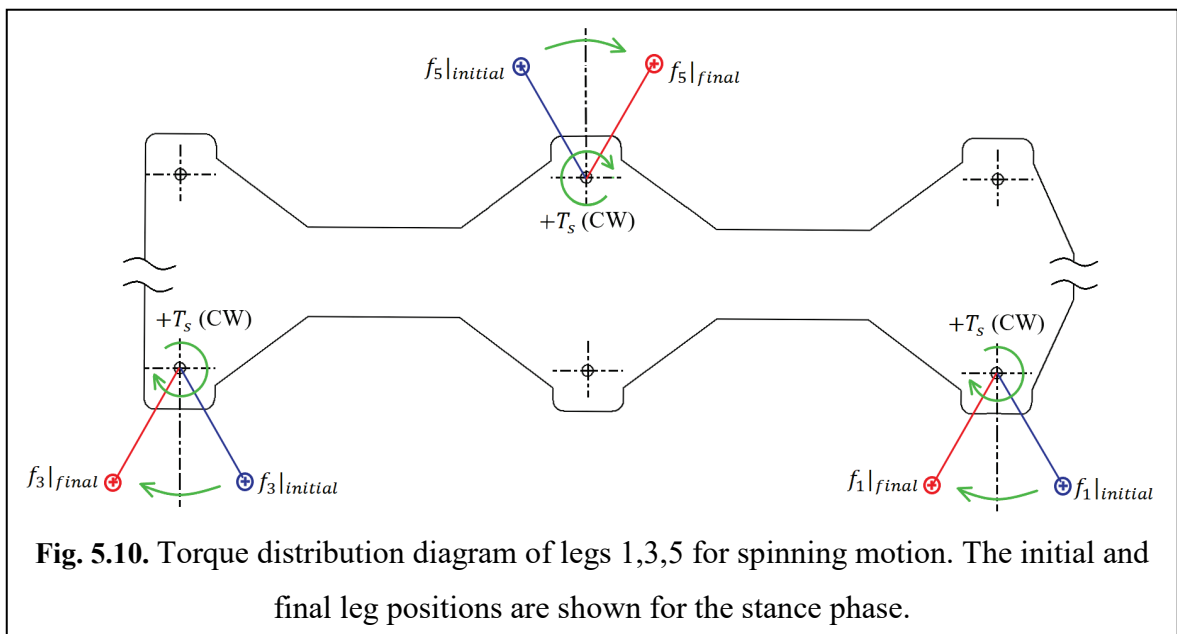
- For Case 2.b.i. only foot-tip f_5 is kept fixed.
 1. Considering leg-5, calculate $l_5|_{initial}$, $l_5|_{mid}$ and $l_5|_{final}$ for $\theta_{h5}|_{initial}$, $(\theta_{h5}|_{initial} + \theta_{h1}|_{initial})/2$ and $\theta_{h1}|_{initial}$
 2. Calculate $\{B\}|_{new-i}$ from $l_5|_{initial}$, $l_5|_{mid}$ and $l_5|_{final}$
 3. Calculate the updated positions of the leg pivots l_i , $i \in \{1,2, \dots, 6\}$ w.r.t $\{B\}|_{new-i}$
 4. Locate the new foot-tips $\{f_2, f_4, f_6\}$ for the legs in-air, and for grounded legs-1 & 3 $\{f_1, f_3\}$ considering the drag.

- For Case 2.b.ii. perform analysis as stated in Case -1 with the limits $\theta_{h1}|_{initial}$ to $\theta_{h1}|_{final}$, since the legs 1-3-5 have equal actuations.
 1. Obtain the new positions of $\{B\}|_{new-ii}$ and l_i , $i \in \{1,2, \dots, 6\}$ w.r.t $\{B\}|_{new-ii}$
 2. The foot-tips $\{f_1, f_3, f_5\}$ remains unchanged
- For Case 2.b.iii. perform an exactly similar analysis as in Case 2.b.i. with the limits $\theta_{h5}|_{final}$ to $\theta_{h1}|_{final}$

Case – 3: Actuation of unequal magnitude and opposite direction.

This case is applicable for spinning gaits only. In spinning the robot is able to make sharp turns. The torque direction is same for right and left sided legs. As a result the net torque is $3T_s$ as shown in Fig. 5.10. The analysis technique for such an actuation value is a two-step process.

- i. Keeping the left sided leg-5 at $\theta_{h5}|_{initial}$, the right sided Leg 1,3 is actuated from $\theta_{h1}|_{initial}$ to $\theta_{h1}|_{final}$,
- ii. Then keeping the right sided Legs 1,3 at $\theta_{h1}|_{final}$, the left sided leg-5 is actuated from $\theta_{h5}|_{initial}$ to $\theta_{h5}|_{final}$.



The analysis for Cases 1, 2 & 3 has been shown for Stance 1,3,5. However Stance 2,4,6 can be analysed in exactly the same way by substituting the data of leg-4 in leg-1, leg-6 in leg-3 and leg-2 in leg-5. These motions have been programmed using MatLab R2017b using a set of actuation values. The trajectories of the robot obtained using forward kinematics has been discussed in detail in the later section.

PART – III. Inverse Kinematics

A more practical approach to maneuvering a mobile robot such as the Hexcrawler is by making it follow a predefined path. The problem in this case is to find the time history of joint angles, which when fed to the controller, would drive the robot along the desired path with minimum deviation. Inverse Kinematic model would provide the required joint actuation values. Assuming that the trajectory can be defined by a geometric curve, and it must have finite derivatives at each point along the curve. The existence of derivatives indicate that a tangent can be drawn at any point on the curve, which in turn gives the yaw angle of the robot. A generalised algorithm is presented which employs inverse kinematics using the inter-relations between $\{B\}$, l_i and f_i . The assumption of planar maneuvering is valid for this model also with the trajectory being defined in the x-y plane.

5.7. IK Solver Algorithm

A simplified approach towards evaluating the inverse kinematics (IK) of the robot has been presented using an algorithm. Given an initial and final state of the robot, the IK Solver will estimate required joint actuation angles.

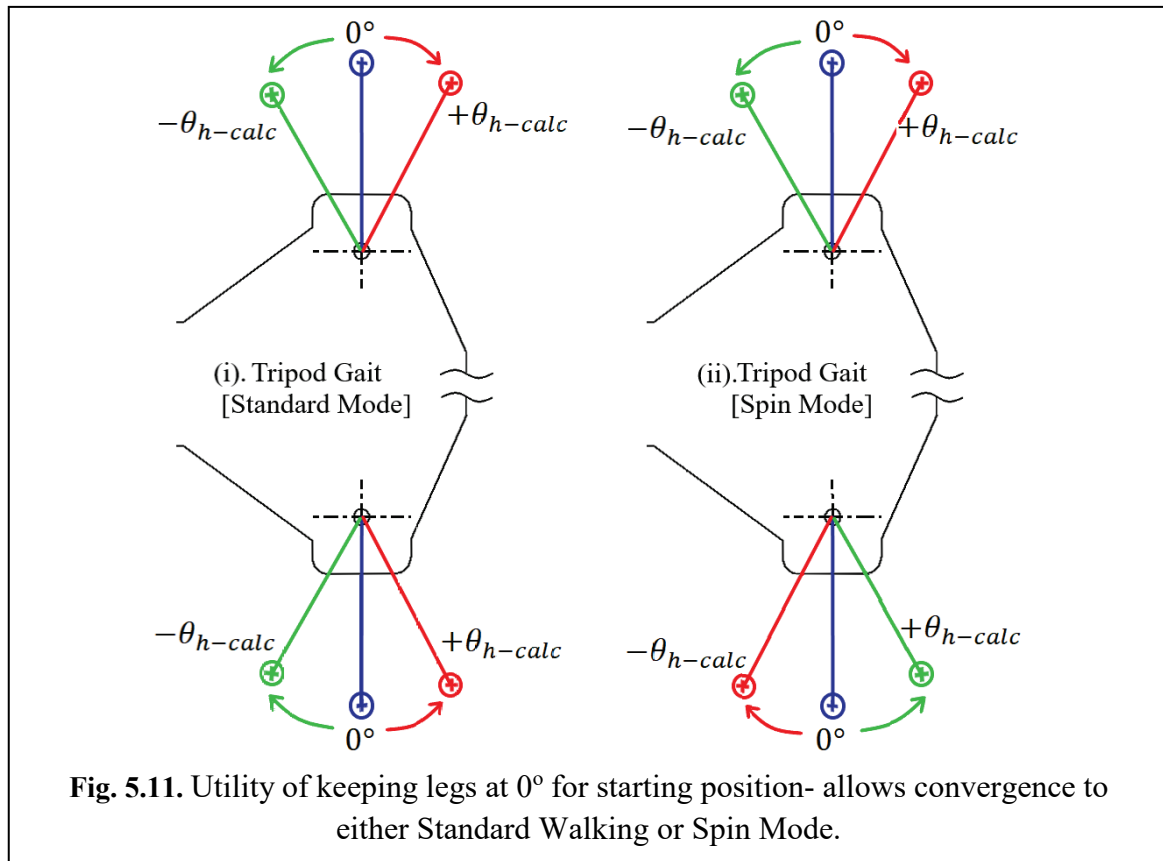
Proposed Algorithm:

Step 1. The trajectory is defined by a curve C , and is divided into n segments. So, there must exist n numbers of $\{B\}_k$ values, $k \in \{1,2, \dots \dots n\}$. The choice of n depends upon the span of the curve. Taking higher values of n ensures feasible solution at the expense of number of steps required to travel along the trajectory. Using smaller values of n could result in an infeasible solution.

Step 2. Set the ‘Starting state’ of the robot by setting $\theta_{hi} = 0^\circ$, for $i \in \{1,2, \dots \dots 6\}$, with legs 1,3,5 grounded and legs 2,4,6 in-air. The legs are set at 0° or the neutral position, this gives the IK solver more chances of converging to a feasible solution. The solution can result in both a standard Tripod gait or a Spin based Tripod gait, as shown in Fig. 5.11.

Step 3. For $k = 1$ to n

- i. For Stance of 1,3,5:
 - Obtain value of $\{B\}_k = [x_B \ y_B \ z_B \ \gamma]^T$
 - Obtain the positions of leg-pivots $l_{i,k}$ w.r.t. $\{B\}_k$
 - Use IK Solver to obtain values of $\theta_{h1}, \theta_{h3}, \theta_{h5}$.
- ii. For Swing 2,4,6: obtain the new foot-tip positions for Legs 2,4,6, $\{f_2, f_4, f_6\}$ w.r.t $l_{i,k}$. $\theta_{h2} = \theta_{h4} = \theta_{h6} = 0^\circ$, which is the default position, preparing the legs for next Stance phase.
- iii. Interchange Support: Legs 1,3,5 is lifted in-air and Legs 2,4,6 is lowered to form the support tripod.
- iv. For Stance 2,4,6:
 - Update $k = k + 1$
 - Obtain value of $\{B\}_{k+1} = [x_B \ y_B \ z_B \ \gamma]^T$
 - Obtain the positions of leg-pivots $l_{i,k+1}$ w.r.t. $\{B\}_{k+1}$
 - Use IK Solver to obtain values of $\theta_{h2}, \theta_{h4}, \theta_{h6}$.
- v. For Swing 1,3,5: obtain the new foot-tip positions for Legs 1,3,5 $\{f_1, f_3, f_5\}$ w.r.t $l_{i,k+1}$. $\theta_{h1} = \theta_{h3} = \theta_{h5} = 0^\circ$, which is the default position, preparing the legs for next Stance phase.
- vi. Interchange Support: Legs 2,4,6 is lifted in-air and legs 1,3,5 is lowered to form the support tripod.



5.8. Results and Discussion

The algorithms for Forward and Inverse Kinematics have been used to generate trajectories using simulation in MatLab 2017b. Experiments have been performed that test the success of the algorithms developed. The simulation and experiments have been presented side-by-side for comparison. However, no measurements of the trace curve have been done since the FK Algorithm does not account for the dynamics and friction. Only the nature of the curve is compared for the simulated and experimental curves. Different trajectories have been presented in the form of case studies.

Case 1. Straight Line Walking using Forward Kinematics:

It is desired to manoeuvre the Hexcrawler robot along a straight line trajectory. As stated previously, equiangular actuation for all 6 legs would allow the robot to move in a straight line path. The amount of surge and sway per stance phase is proportional to the magnitude of actuation.

The Forward Kinematic algorithm has been applied for the following input parameters:

- Actuation angle: $\theta_{hi} = \pm 15^\circ$, $i = \{1, 2, \dots, 6\}$.
- No of Cycles: 6.

After simulating for the above inputs, the body frame $\{B\}$ is recorded after each stance phase. The values of x_B and y_B have been plotted and shown in Fig. 5.12(a). The inputs have been used to program the Hexcrawler and executing it produces a trace curve shown in Fig. 5.12(b). The simulation and experimental trace curve are a close match which justifies the success of the FK Algorithm.

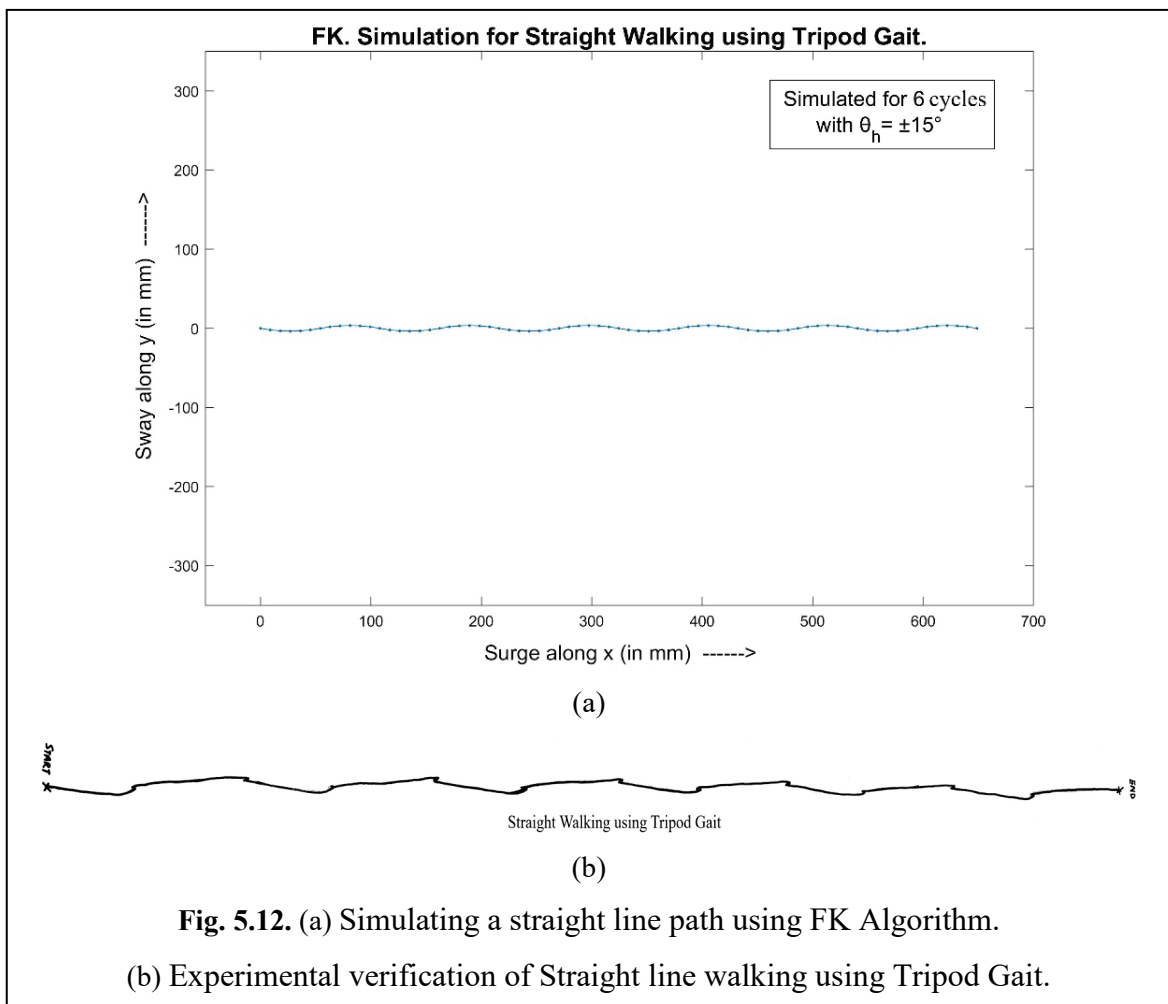


Fig. 5.12. (a) Simulating a straight line path using FK Algorithm.
(b) Experimental verification of Straight line walking using Tripod Gait.

Case 2. Spinning using Forward Kinematics:

In spinning mode the robot turns with a very small turning radius, it can be used when it is required to make sharp turns. This happens when the actuation is equal in magnitude but opposite in direction for right and left sided legs. Numerical simulation and experimental validation has been performed for the following inputs:

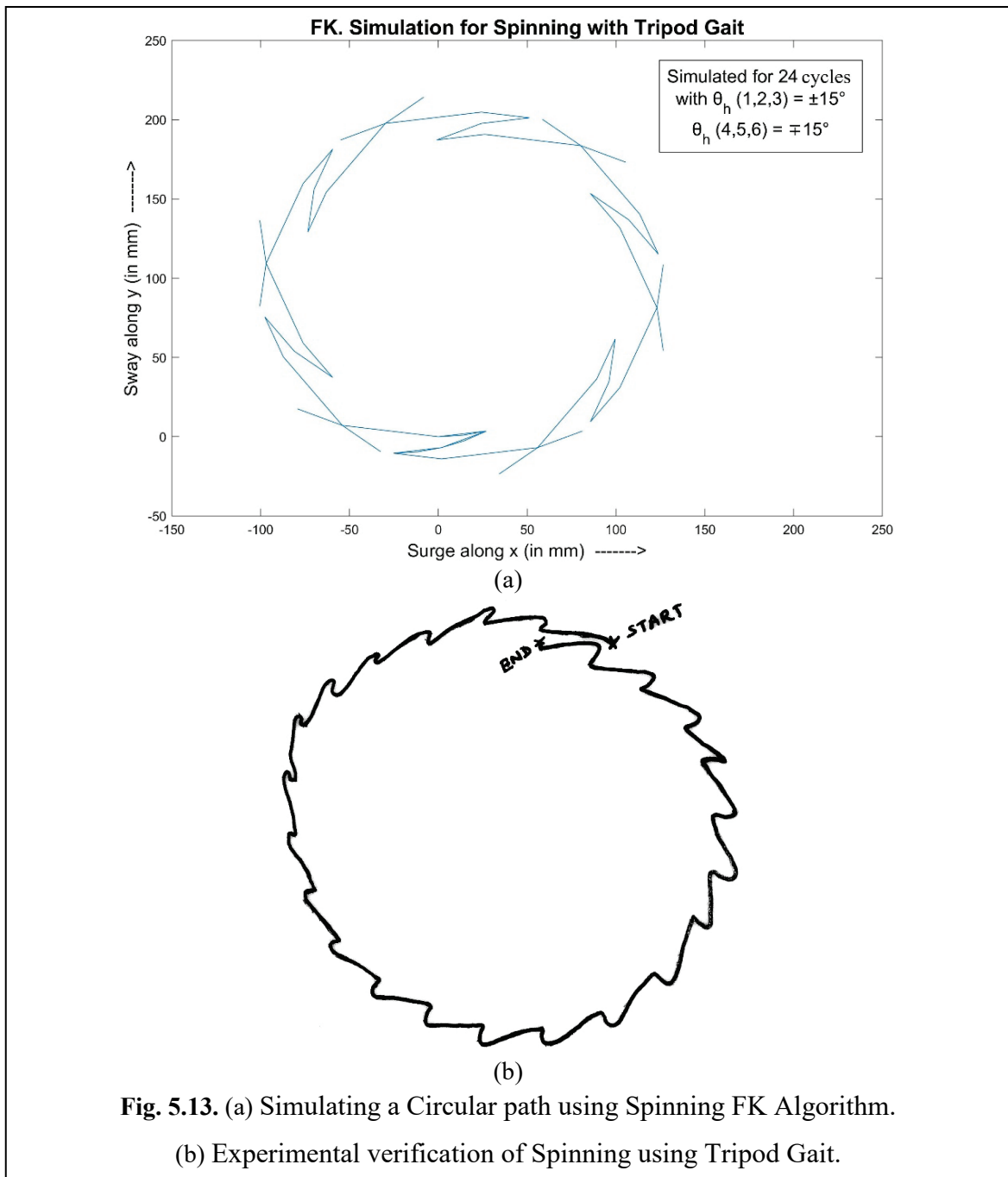
- Actuation angles:

$$\theta_{hi} = \pm 15^\circ, i = \{1,2,3\} \text{ for Right sided legs}$$

$$\theta_{hi} = \mp 15^\circ, i = \{4,5,6\} \text{ for Left sided legs}$$

- No of cycles: 24.

After simulating for the above inputs, the body frame $\{B\}$ is recorded after each stance phase. The values of x_B and y_B have been plotted and shown in Fig. 5.13 (a). The inputs have been used to program the Hexcrawler and executing it produces a trace curve shown in Fig. 5.13 (b). The simulation and experimental trace curve are a close match.



Case 3. Turning using Forward Kinematics:

Curved paths can be realized with tripod gaits using turning. In this case, the actuation magnitudes are different for left and right sided legs, the direction may or may not be the same. Both these cases have been addressed and simulated using the FK Solver. The Hexcrawler robot has been programmed using the corresponding actuation values and the trace curves are obtained which are a close match to those of the simulation plots. Numerical simulation and experimental validation has been performed for the following set of inputs:

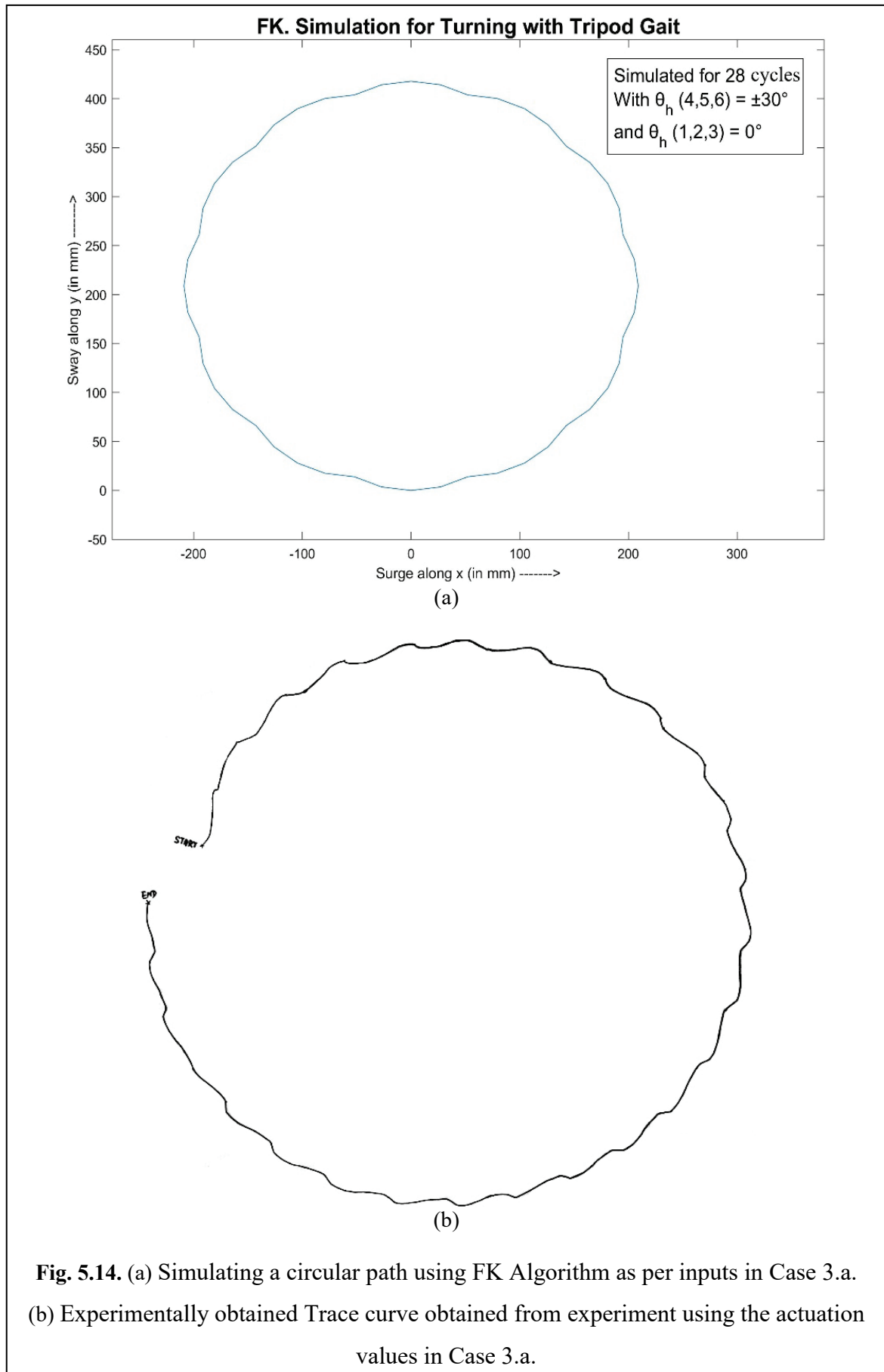
Case 3.a.

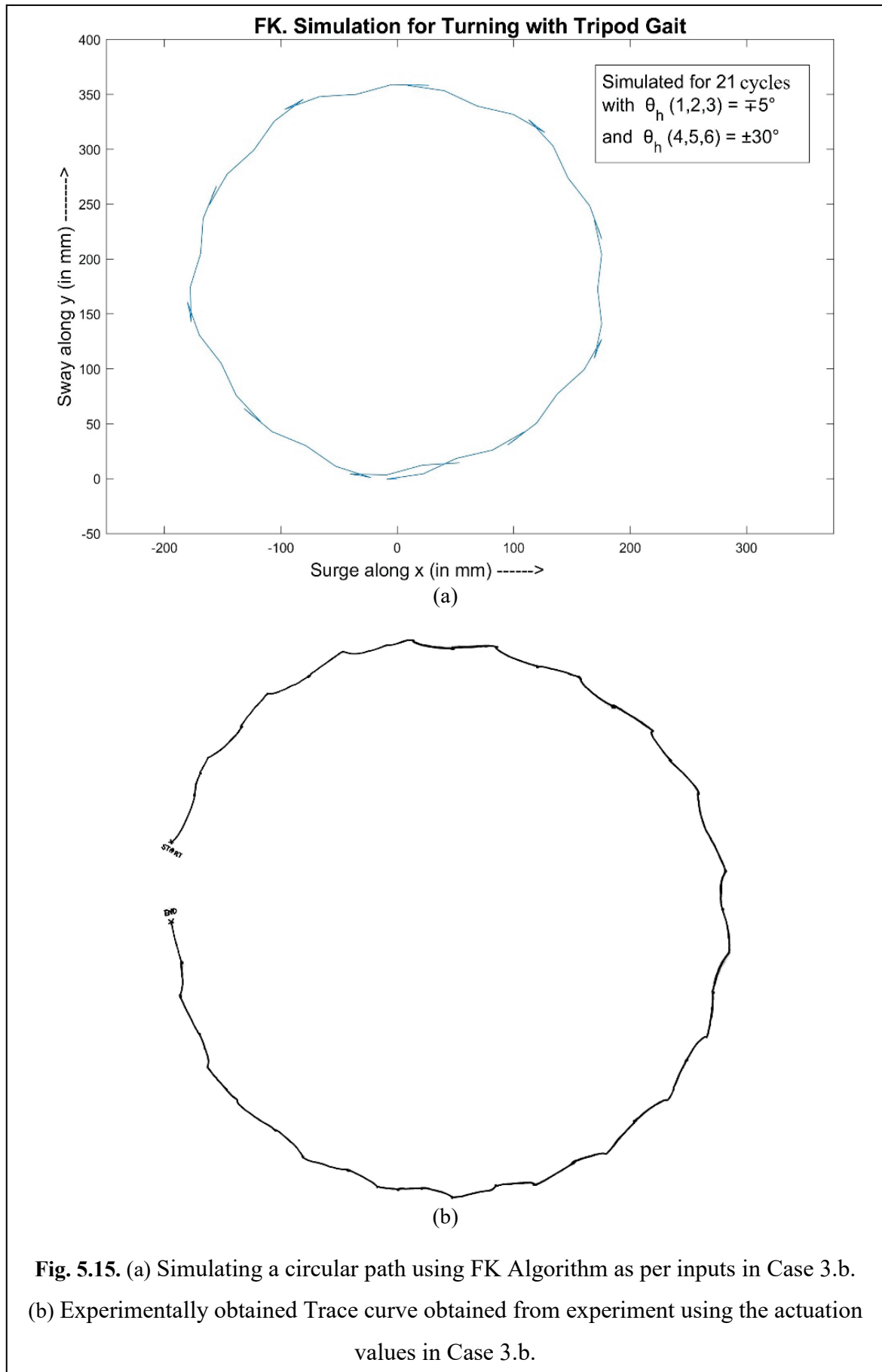
- Actuation angles:
 $\theta_{hi} = \pm 30^\circ$, $i = \{1,2,3\}$ for Right sided legs
 $\theta_{hi} = 0^\circ$, $i = \{4,5,6\}$ for Left sided legs
- No of cycles: 28.

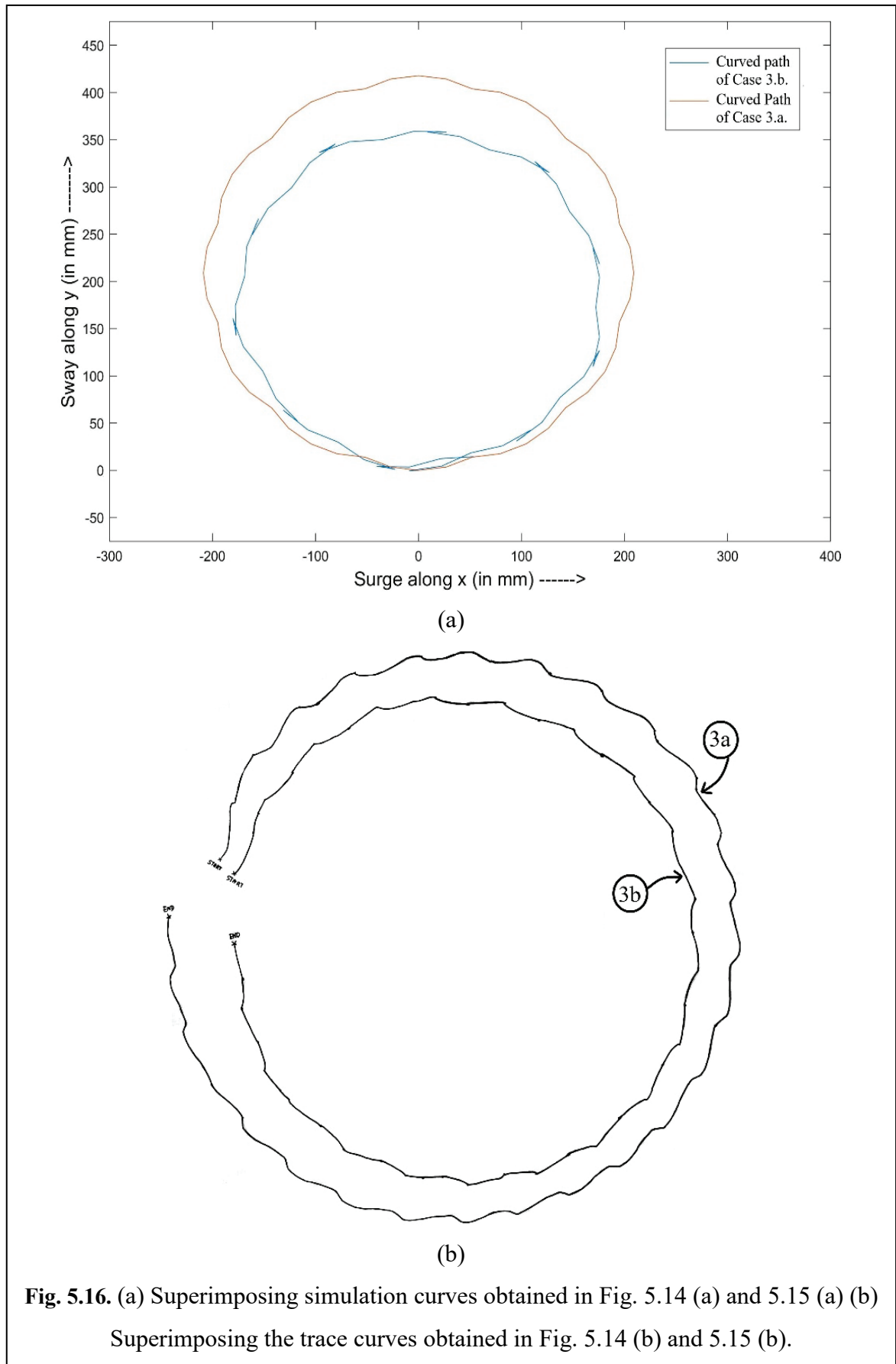
Case 3.b.

- Actuation angles:
 $\theta_{hi} = \pm 30^\circ$, $i = \{1,2,3\}$ for Right sided legs
 $\theta_{hi} = \mp 5^\circ$, $i = \{4,5,6\}$ for Left sided legs
- No of cycles: 21.

The actuations in case 3.a. has been simulated and the trace curve is obtained after experimentation as shown in Fig. 5.14 (a), Fig. 5.14 (b) respectively. While the actuations in case 3.b. has been simulated and the trace curve is obtained after experimentation as shown in Fig. 5.15 (a), 5.15 (b) respectively. The curves have been superimposed and shown in Fig. 5.16(a) and 5.16 (b).







Case 4. Walking along Straight line using Inverse Kinematics:

For problems of Inverse Kinematics the trajectory is pre-defined. It is required to compute the joint actuation angles which would allow the Hexcrawler to trace the desired trajectory.

Two sample tracks have been considered:

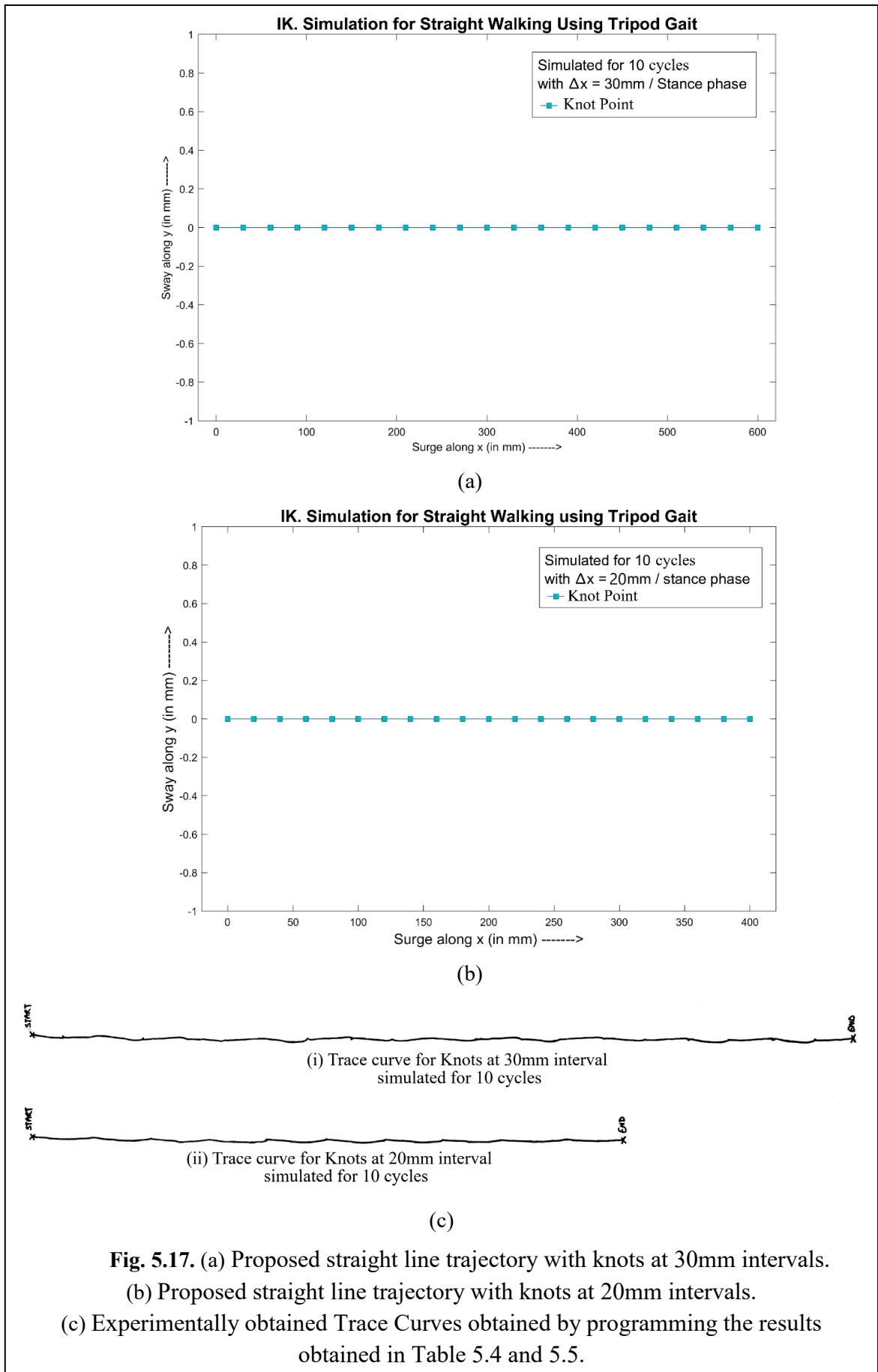
- i. Straight line of span 600mm, with knots taken at 30mm intervals, as shown in Fig. 5.17 (a). At the knots the inverse is calculated using the IK Solver stated previously. The results of IK Solver has been enlisted in Table 5.4. The results are programmed in the Hexcrawler and executed, which generates the trace curve shown in Fig. 5.17 (c).
- ii. Straight line of span 400mm, with knots taken at 20mm intervals, as shown in Fig. 5.17 (b). The results of IK Solver has been enlisted in Table 5.5. The results are programmed in the Hexcrawler and executed, which generates the trace curve shown in Fig. 5.17 (c).

Table 5.4. Results of IK Solver for a single walking cycle with knots taken at 30mm intervals:

Motion	Actuation Angles obtained from IK Solver.											
	Leg - 1		Leg - 2		Leg - 3		Leg - 4		Leg - 5		Leg - 6	
	θ_{h1}	θ_{v1}	θ_{h2}	θ_{v2}	θ_{h3}	θ_{v3}	θ_{h4}	θ_{v4}	θ_{h5}	θ_{v5}	θ_{h6}	θ_{v6}
Starting Position	0	30	0	-30	0	30	0	-30	0	30	0	-30
Stance 1,3,5	-16.02	27.65	0	-30	-16.02	27.65	0	-30	-16.02	27.65	0	-30
Swing 2,4,6	-16.02	30	0	-30	-16.02	30	0	-30	-16.02	30	0	-30
Switch Support	-16.02	-30	0	30	-16.02	-30	0	30	-16.02	-30	0	30
Stance 2,4,6	-16.02	-30	-16.02	27.65	-16.02	-30	-16.02	27.65	-16.02	-30	-16.02	27.65
Swing 1,3,5	0	-30	-16.02	30	0	-30	-16.02	30	0	-30	-16.02	30
Switch Support	0	30	-16.02	-30	0	30	-16.02	-30	0	30	-16.02	-30

Table 5.5. Results of IK Solver for a single walking cycle with knots taken at 20mm intervals:

Motion	Actuation Angles obtained from IK Solver.											
	Leg - 1		Leg - 2		Leg - 3		Leg - 4		Leg - 5		Leg - 6	
	θ_{h1}	θ_{v1}	θ_{h2}	θ_{v2}	θ_{h3}	θ_{v3}	θ_{h4}	θ_{v4}	θ_{h5}	θ_{v5}	θ_{h6}	θ_{v6}
Starting Position	0	30	0	-30	0	30	0	-30	0	30	0	-30
Stance 1,3,5	-10.83	28.85	0	-30	-10.83	28.85	0	-30	-10.83	28.85	0	-30
Swing 2,4,6	-10.83	30	0	-30	-10.83	30	0	-30	-10.83	30	0	-30
Switch Support	-10.83	-30	0	30	-10.83	-30	0	30	-10.83	-30	0	30
Stance 2,4,6	-10.83	-30	-10.83	28.85	-10.83	-30	-10.83	28.85	-10.83	-30	-10.83	28.85
Swing 1,3,5	0	-30	-10.83	30	0	-30	-10.83	30	0	-30	-10.83	30
Switch Support	0	30	-10.83	-30	0	30	-10.83	-30	0	30	-10.83	-30

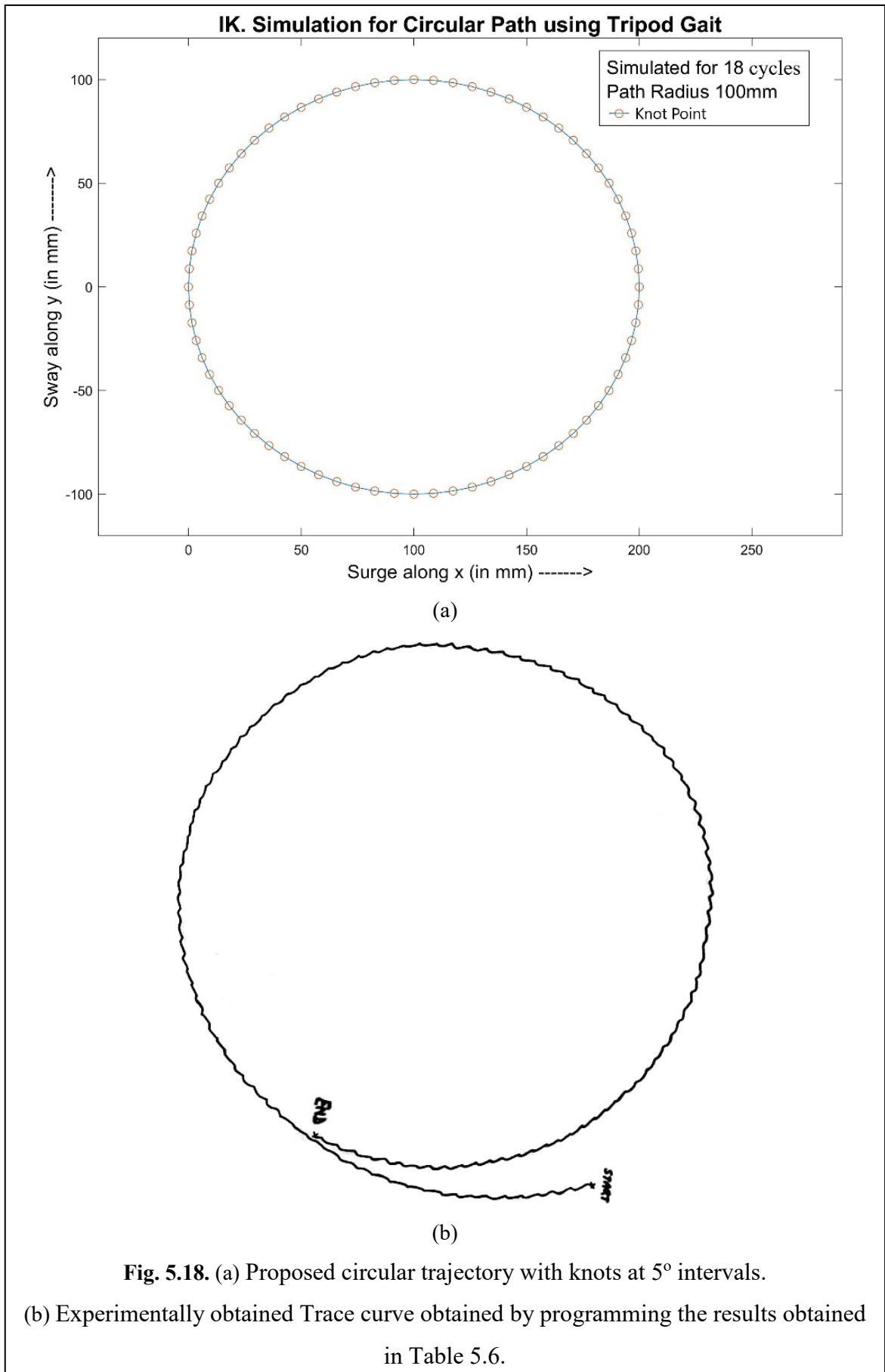


Case 5. Walking along circular path using Inverse Kinematics:

The designed trajectory is circular, and knots are considered at equal spacing at intervals of 5°, shown in Fig. 5.18 (a). Using the IK Solver, the solution has been enlisted in Table 5.6. The results are programmed in the Hexcrawler and executed, which generates the trace curve shown in Fig. 5.18 (b).

Table 5.6. Results of IK Solver for a single walking cycle:

Motion	Actuation Angles obtained from IK Solver.											
	Leg - 1		Leg - 2		Leg - 3		Leg - 4		Leg - 5		Leg - 6	
	θ_{h1}	θ_{v1}	θ_{h2}	θ_{v2}	θ_{h3}	θ_{v3}	θ_{h4}	θ_{v4}	θ_{h5}	θ_{v5}	θ_{h6}	θ_{v6}
Starting Position	0	30	0	-30	0	30	0	-30	0	30	0	-30
Stance 1,3,5	4.31	31.40	0	-30	4.31	31.40	0	-30	-13.87	30.03	0	-30
Swing 2,4,6	4.31	30	0	-30	4.31	30	0	-30	-13.87	30	0	-30
Switch Support	4.31	-30	0	30	4.31	-30	0	30	-13.87	-30	0	30
Stance 2,4,6	4.31	-30	3.99	29.54	4.31	-30	-12.09	28.94	-13.87	-30	-12.09	29.54
Swing 1,3,5	0	-30	3.99	30	0	-30	-12.09	30	0	-30	-12.09	30
Switch Support	0	30	3.99	-30	0	30	-12.09	-30	0	30	-12.09	-30



5.9. Conclusion

The curves obtained in Case 1 to 5 show that the Forward and Inverse Kinematics solver works well. From cases 1, 2 and 3 it can be concluded that:

- i. When $\Delta\theta_{hi}|_{right-sided\ legs} = \Delta\theta_{hi}|_{left-sided\ legs}$, the path is straight or the curvature is infinity.
- ii. When $\Delta\theta_{hi}|_{right-sided\ legs} \neq \Delta\theta_{hi}|_{left-sided\ legs}$, the path is curved and the curvature is finite. The radius of curvature reduces as difference as $|\Delta\theta_{hi}|_{right-sided\ legs} - \Delta\theta_{hi}|_{left-sided\ legs}|$ increases, finally turning radius becomes minimum for spinning when $\Delta\theta_{hi}|_{right-sided\ legs}$ and $\Delta\theta_{hi}|_{left-sided\ legs}$ are equal in magnitude but opposite in sign. This is evident from Fig. 5.16.

The results of simulation are a close match to those experimentally obtained. As stated previously, the dynamics and friction dependencies have not been accounted for in the mathematical model. The displacement of the robot is also dependent on the surface condition or friction coefficient. So the trace curves have not been dimensionally verified with those of the simulation plots. However, the overall nature of the curve of the curve remains invariant and is sufficient to prove the success of the algorithms.

Chapter 6.

Complex Trajectories – planning and optimization.

Legged robots are being used extensively in rescue operations, unmanned exploration, handling of hazardous materials, etc. So, they must be capable of travelling along complex paths and terrains. In context to the present work, the Hexcrawler is required to manoeuvre along a planar levelled surface, due to the technical limitations of the robot. However it is possible for the robot to manoeuvre along complex planar paths. Such a path must be expressible by a geometric curve and quantified a mathematical equation. Parametric curves such as Bezier, B-Spline, and NURBS are very effective in quantifying such paths. These curves are used extensively in computer graphics.

This chapter into 3 major segments:

- I. Trajectory construction,
- II. Robot guidance along the trajectory,
- III. Optimization of trajectory.

The first section deals with constructing a 2D path using Bezier curves followed by developing an Inverse Kinematic solver that estimates the history of joint angles required to manoeuvre along the designed path. These two sections deal with space-planning. The 3rd section deals with time based planning. If time is taken into consideration then the kinematic parameters like velocity, acceleration, jerk, and ping can be calculated. The trajectory can then be optimized with the objective of minimizing one of more factors like travel time, jerk, power consumption, etc. Although the Inverse Kinematic solver is model specific but the methodology can be applied for any mobile robot.

PART – I. Trajectory Construction.

6.1. Problem Formulation

It is desired to manoeuvre the Hexcrawler robot along a complex path. This path should be planar as per the previous assumptions. A problem is formulated considering a real-life scenario of a mission assigned to a mobile robot. A mobile robot is supposed to travel from its initial position to its target location along a pre-defined path. To construct a path, a set of reference points must be defined and the designed path must pass through these points. These points will be referred to as ‘target points’. A mobile robot maneuvering along this path will obviously pass through these target points with minimum or zero deviation.

Let d_i be a set of target points and n_d be the number of target points. Then a path can be constructed by curve-fitting using any spline function. The designed path is then subdivided into a finite number of segments. Inverse Kinematics solver is applied at each segment. So the required actuation to travel from start point to the end point can be determined for each segment. These segments when added up, a sequence of joint actuation angles is obtained which would allow the robot to trace the desired path. The following section presents some mathematical tools which would aid in solving the problem formulated.

6.2. Parametric Trajectories

As stated previously, the trajectory must be expressible using a geometric curve and must have a mathematical definition. Parametric curves are chosen because of the following properties:

- a. axis independent
- b. can be used to represent multiple valued functions,
- c. scalable and extremely flexible,
- d. derivable at all points on the curve.

Parametric Curves are represented in the form,

$$x = f(t), \quad y = f(t), \quad z = f(t) \quad \dots\dots (6.1)$$

Where, t is a parameter and $c_1 \leq t \leq c_2$

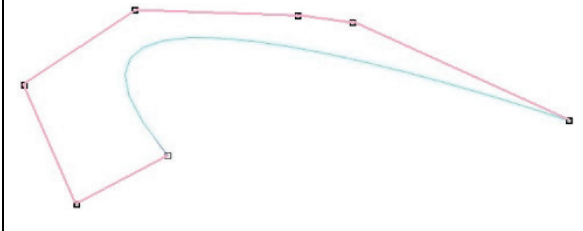
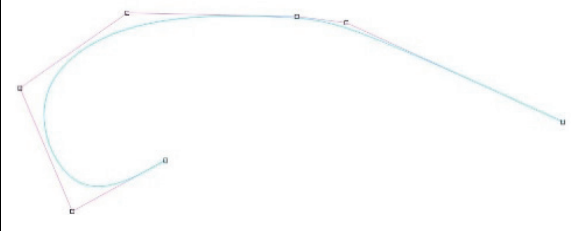
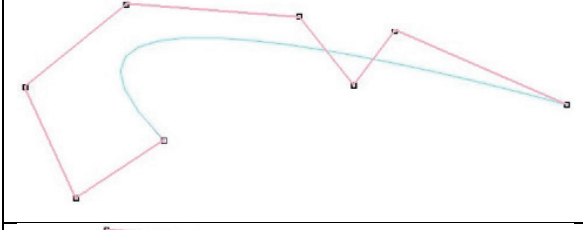
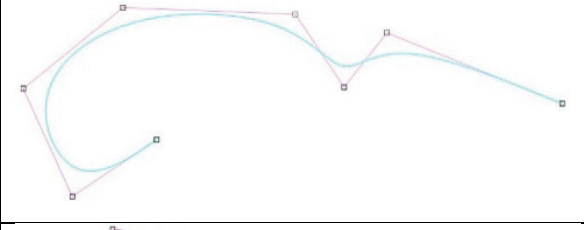
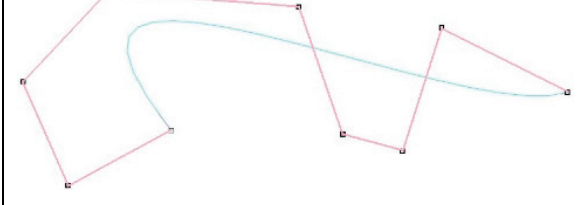
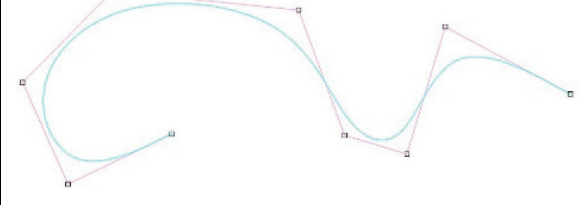
The derivatives of the curve can be calculated as: $\frac{dy}{dx} = \frac{dy/dt}{dx/dt} \quad \dots\dots (6.2)$

Often, the normalized form is more suitable, where the lower and upper bounds c_1, c_2 is converted to 0 and 1 respectively. Two parametric curves can be joined at the point. This point is called a ‘knot’ point. The smoothness of the curves at the knot is ensured by the following continuity relations:

- i. Geometric Continuity: the end point of the 1st segment coincides with the starting point of the 2nd segment, then the two curve segments is said to maintain G^0 continuity. Similarly, G^n continuity exists when the vector obtained by n^{th} derivative of the two segments at the knot has unequal magnitude but same direction.
- ii. Parametric Continuity: two curve segments joined at a knot is said to be C^0 continuous, which is same as G^0 continuity. For higher order continuity C^n the vector obtained by n^{th} derivative of the two segments at the knot has equal magnitude and same direction. Use of higher order continuity ensures smoother blends at the knot.

The available options for constructing trajectory using parametric curves are – Bezier, B-Splines and NURBS. The NURBS being the most generalized and offer maximum control and flexibility, as described in Rogers [81]. On imposing certain conditions, NURBS curve converges to B-Spline. Similarly, imposing constraints on the B-Spline curve it converges to Bezier. The Bezier curve has the simplest mathematical definition but with minimum control and least flexibility, refer Table 6.1. For the current thesis work, it is desired to manoeuvre the Hexcrawler robot along a complex trajectory. The Hexcrawler robot has dimensions 49.68cm x 40.00cm while the dimensions of the table on which the experiment is to be performed is approximately 214cm x 125cm. Hence, the trajectories constructed must be feasible for the robot to follow. Too large a trajectory or a path having sharp bends, irregularities would result in dead-zones. The Inverse Kinematic solver could converge to a solution which the Hexcrawler will not be able to execute. So, trajectories designed with Bezier curves would prove to be the ideal choice, since they are smooth and the show minimum distortion in shape on adding new control vertices.

Table 6.1. Shape change of Bezier and B-Spline curves due to addition of Control Vertices:

Bezier Curve	B-Spline Curve
	
	
	

6.2.1. Definition of a Bezier Curve

Developed in 1960’s by Pierre Bezier for designing surface of cars at Renault Automobiles.

Bezier Curves are parametric in nature and is mathematically defined as:

$$P(u) = \sum_{i=0}^n B_i J_{n,i}(u), \quad 0 \leq u \leq 1 \quad \dots\dots (6.3)$$

Where, $J_{n,i}(u)$ is the Bernstein basis or blending function, defined as:

$$J_{n,i}(u) = \binom{n}{i} u^i (1-u)^{n-i}, \quad 0^0 \equiv 1 \quad \dots\dots (6.4)$$

And
$$\binom{n}{i} = \frac{n!}{i!(n-i)!}, \quad 0! \equiv 1$$

The degree of the polynomial defining the curve segment is $(n - 1)$

Where, n is the number of control polygon vertices.

The 1st derivative of the curve can be written as:

$$P'(u) = \sum_{i=0}^n B_i J'_{n,i}(u), \quad \dots\dots (6.5a)$$

The 2nd derivative of the curve can be written as:

$$P''(u) = \sum_{i=0}^n B_i J''_{n,i}(u), \quad \dots\dots (6.5b)$$

The derivative of the basis function are obtained by differentiating J with respect to u :

$$J'_{n,i} = \frac{i-nu}{u(1-u)} J_{n,i}(u), \quad \dots\dots (6.6a)$$

$$J''_{n,i}(u) = \left(\frac{(i-nu)^2 - nu^2 - i(1-2u)}{u^3(1-u)^2} \right) J_{n,i}(u), \quad \dots\dots (6.6b)$$

In a similar manner the higher order derivatives can also be computed.

The 1st and 2nd order derivatives at the end points can be written as:

$$\begin{aligned} P'(0) &= n(B_1 - B_0), \\ P'(1) &= n(B_n - B_{n-1}), \\ P''(0) &= n(n-1)(B_2 - 2B_1 + B_0), \\ P''(1) &= n(n-1)(B_n - 2B_{n-1} + B_{n-2}). \end{aligned} \quad \dots\dots (6.7)$$

The above relations will be used to establish continuity relations.

6.2.2. Joining Two Bezier Curve Segments

Two Bezier curve segments can be joined at a knot point satisfying geometric or parametric continuity relation of any order. Let, the 1st Bezier curve $P(u_1)$ of degree n , and 2nd Bezier curve $Q(u_2)$ of degree m . Let, B_i and C_i be vertices of the control polygon for the 1st and 2nd curve segment respectively.

$$\begin{aligned} \text{For } G^0 \text{ or } C^0 \text{ continuity,} \quad & P(1) = Q(0), \\ \text{For } G^1 \text{ continuity,} \quad & P'(1) = gQ'(0), \quad \text{where, } g \text{ is a scalar.} \\ \text{Using Equation () we get,} \quad & C_1 - C_0 = 1/g \binom{n}{m} (B_n - B_{n-1}) \end{aligned} \quad \dots\dots (6.8)$$

For C^1 continuity, we set $g = 1$,
i.e. derivatives are equal in magnitude and direction.

$$\text{For } C^2 \text{ continuity,} \quad m(m-1)(C_2 - 2C_1 + C_0) = n(n-1)(B_n - 2B_{n-1} + B_{n-2}) \quad \dots\dots (6.9)$$

6.2.3. Subdivision of Bezier Curves

Just as two curve segments can be joined, it is also possible to split an existing Bezier curve into two segments at any point along the curve. The task is to determine the control polygons for the two new curves which when combined will be identical to the original curve. Let the original curve is given by $P(u_0)$, the new curves after split are $Q(u_1)$ and $R(u_2)$. Let B_i, C_i , and D_i be the vertices of the control polygon for curves P, Q and R respectively. For simplifying calculations, the point of split is chosen at $u_0 = 0.5$. C_i and D_i can then be expressed as:

$$C_i = \sum_{j=0}^i \binom{i}{j} \frac{B_j}{2^i}, \quad i = 0, 1, \dots\dots n \quad \dots\dots (6.10)$$

$$D_i = \sum_{j=i}^n \binom{n-i}{n-j} \frac{B_j}{2^{n-i}}, \quad i = 0, 1, \dots, n \quad \dots (6.11)$$

Using De Casteljau's algorithm, it is possible to split the curve at any value of u in the range $0 \leq u \leq 1$.

De Casteljau's equation is given by: $B_i^k(t_0) = (1 - u_0)B_{i-1}^{k-1} + u_0B_i^{k-1} \quad \dots (6.12)$

Equation. () is used recursively to obtain C_i and D_i respectively.

Where, $k = 1, 2, \dots, n$; $i = k, k + 1, \dots, n$; and u_0 is the parametric split point.

The new control vertices are given by:

$$C_i = \{B_0^0, B_1^1, \dots, B_n^n\} \quad \text{and} \quad D_i = \{B_n^n, B_{n-1}^{n-1}, \dots, B_0^0\}.$$

6.2.4. Computation of Arc Length

Arc Length is the distance between two points on the curve measured along the curve. The computation of Arc length is especially important from the point of view that it can be used to calculate time derivative parameters like velocity and acceleration. The arc-length from point $P(u_1)$ to $P(u_2)$ can be obtained by calculating the line-integral as:

$$s = \int_{u_1}^{u_2} \left| \frac{dP}{du} \right| \cdot du \quad \dots (6.13)$$

Where, $\left| \frac{dP}{du} \right| = \sqrt{\left(\frac{dx(u)}{du}\right)^2 + \left(\frac{dy(u)}{du}\right)^2 + \left(\frac{dz(u)}{du}\right)^2}$ and s is the arc-length.

Such an integral can be evaluated using Numerical Integration techniques such as the Gaussian Quadrature method or Adaptive Gaussian Quadrature method. However, these methods can become quite difficult to implement and become computationally expensive in cases where higher accuracy is desired. An alternative method is the adaptive subdivision method. If a Bezier curve is subdivided into finite number of segment, then the arc length of each segment will be greater than the chord length L_c , but less than the sum of lengths of sides of the control polygon L_p , as suggested in Gravesen [82] and Guenter [83].

The 'true-arc-length' L_B can be computed using a convex combination of L_c and L_p given by: $L_B = \frac{2L_c + (n-1)L_p}{(n+1)} \quad \dots (6.14)$

Let P be a Bezier curve with control points $B_i, i = 0, 1, \dots, n$.

The Arc-Length of the curve is given by: $L_B = \int_0^1 |P'(u)| du \cong \frac{2L_c + (n-1)L_p}{(n+1)} \quad \dots (6.15)$

Sum of lengths of sides of the control polygon: $L_p = \sum_{i=0}^{n-1} |P_{i+1} - P_i| \quad \dots (6.16)$

Chord length between the end-points: $L_c = |P_n - P_0| \quad \dots (6.17)$

Using an Adaptive approach reduces the computation time significantly. The number of subdivisions are not only dependent on the error tolerance ϵ but also on the nature of the curve. The choice of ϵ denotes how fast or slow the algorithm will converge to a solution. Taking a very small value of ϵ ensures higher accuracy at the expense of computation time.

Algorithm to compute Arc Length:

Step-1. Calculate L_P , and L_C .

Step-2. Calculate the error bound, $err = L_P - L_C$.

Step-3. If($err < \epsilon$)

Then, calculate L_B , and STOP recursion.

Else, subdivide P into two equal halves and evaluate Step 1 to 3 for each segment (start recursion).

Return $\sum L_{B-segment}$

Step-4. End.

Let $ARCLLEN(u_1, u_2)$ be a function that computes the arc length along the space curve from points $P(u_1)$ to $P(u_2)$ according to the algorithm stated above then it is possible to solve two types of problem:

- i. Given the parametric values of u_1 and u_2 , the arc-length L_B can be found,
- ii. Given the arc-length L_B^{req} and value of u_1 , the value of u_2 can be found by solving the equation:

$L_B^{req} - ARCLLEN(u_1, u_2) = 0$, using any root search algorithm such as Bisection method (gradient based) or Golden section method (based on region elimination), etc. The latter algorithm is suitable where calculating derivatives can become cumbersome.

6.2.5. Curve fitting using Bezier Curves

A set of target points d_i where $i \in \{1,2,.. n_d\}$ is defined. It is required to fit a Bezier curve that fits the points d_i in the total least square sense. The fitted curve is of degree $(n - 1)$, where n is the number of control polygon vertices. If a target point lies on the Bezier curve then it must satisfy the relation:

$$d_i = B_0 J_{0,n}(t_i) + B_1 J_{1,n}(t_i) + \dots + B_n J_{n,n}(t_i), \quad i \in \{1,2,.. n_d\} \quad \dots (6.18)$$

In the matrix form, $[D] = [J][B]$ (6.19)

Where,

$$[D] = [d_1 \ d_2 \ \dots \ d_{n_d}]^T$$

$$[B] = [B_0 \ B_1 \ \dots \ B_n]^T$$

$$[J] = \begin{bmatrix} J_{1,n}(t_1) & \dots & J_{1,n}(t_1) \\ \vdots & \ddots & \vdots \\ J_{1,n}(t_{n_d}) & \dots & J_{1,n}(t_{n_d}) \end{bmatrix}$$

Solve for $[B]$.

The matrix $[J]$ is a rectangular matrix and hence non-invertible. However, the pseudo-inverse can be used to solve such a system of equations. The Moore-Penrose pseudo-inverse of a matrix A is defined as:

$$\text{pinv}(A) = (A^T A)^{-1} A^T \quad \dots (6.20)$$

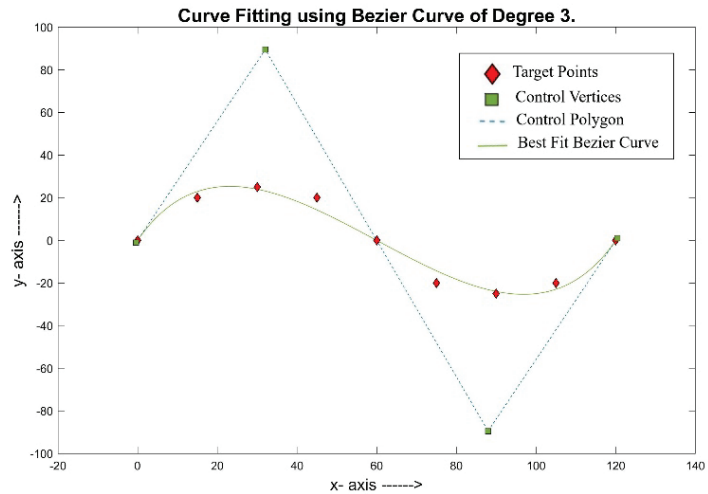
Then the solution for $[B]$ becomes, $[B] = \text{pinv}(J) \cdot [D]$ (6.21)

The above method would yield an exact solution, the solution is not always guaranteed. An approximate method is minimizing the residual $R(t)$, given as: $R = [J][B] - [D]$... (6.22)

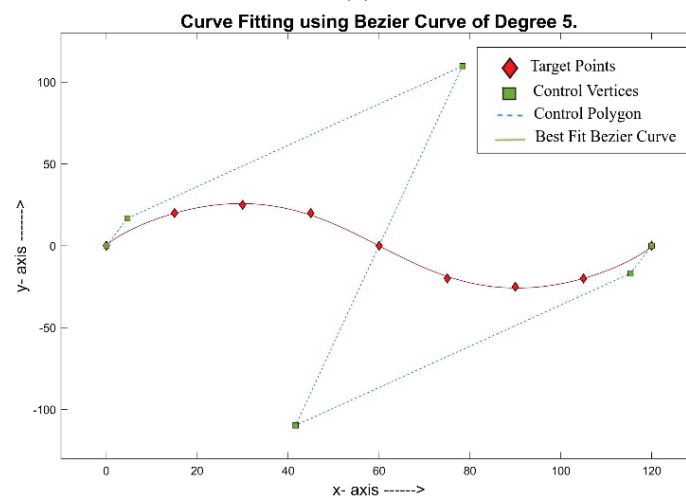
The considering a planar curve in x-y plane, the residual will also have components in the x and y directions. So taking the sum of squares of the residues in the x and y direction, the equation becomes:

$$R = \| [J][B] - [D] \| \quad \dots (6.23)$$

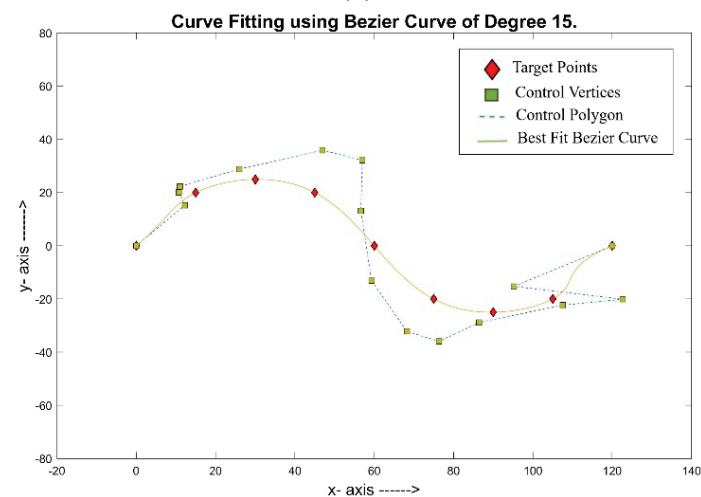
Minimizing R would result in finding the roots of the equation which is the required solution $[B]$, refer to Pastva [84]. Increasing the degree of the Bezier curve would allow greater flexibility and it is more likely that the residual at the target points would converge to zero. However, the shape of the curve may get distorted which is undesirable. It is up to the user to decide the value of n based on a trial and error basis, refer Fig. 6.1.



(a)



(b)



(c)

Fig. 6.1. Trade-off between Perfect Fit and Distortion.

(a) Curve fit using 3rd degree showing that it does not pass through all target points,

(b) Curve fir using 5th degree showing that it passes through all target points,

(c) Increasing degree of fitted curve results in localized distortion.

PART – II. Robot Guidance.

The objective is to obtain the history of joint angles that would allow the Hexcrawler to trace the designed path, refer Fig. 6.2. The most suitable method is the use of IK Solver stated in the previous chapter, however it required suitable modifications.

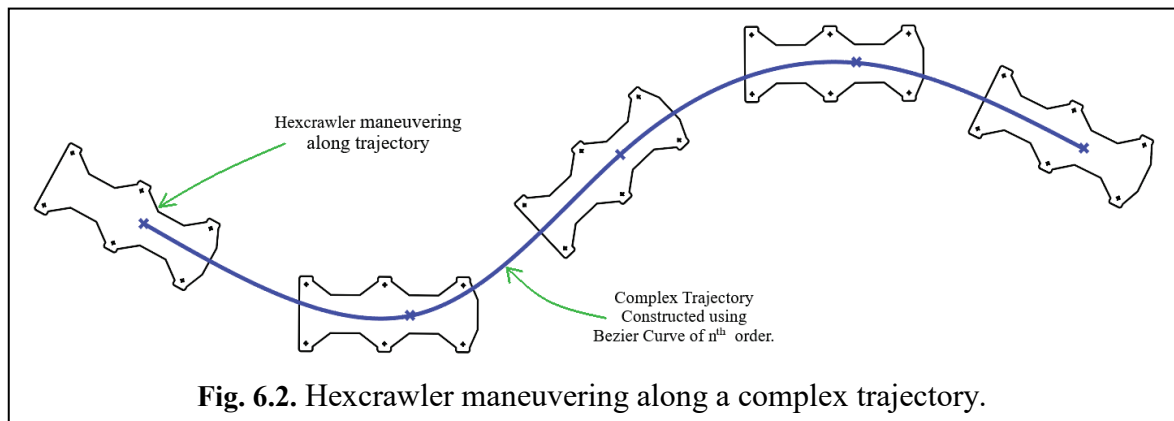


Fig. 6.2. Hexcrawler maneuvering along a complex trajectory.

6.3. IK Solver for Parametric Trajectories.

A trajectory defined by a parametric curve can be traversed in two ways-

- i. Using parametric intervals: if P_1 be the starting point at $u_1 = 0$, then the next point on the curve P_2 is located at $u_2 = u_1 + \Delta u$. Where u is the parameter and Δu is the parametric interval. Although choice of Δu is programmer specified, but choosing very small value of Δu could result in the IK Solver producing near zero actuations, while large values of Δu could result in a solution which is infeasible. This algorithm has been presented using this technique, and it works well for smooth trajectories only. Curves having local distortions cannot be planned using this method.
- ii. Using arc-length intervals: if P_1 be the starting point, then the next point on the curve P_2 is located at a distance d measured along the curve. Hence the successive points on the curve are located by computing the arc-length. This method is suitable for uneven curves having distortions and abrupt changes in localised curvatures.

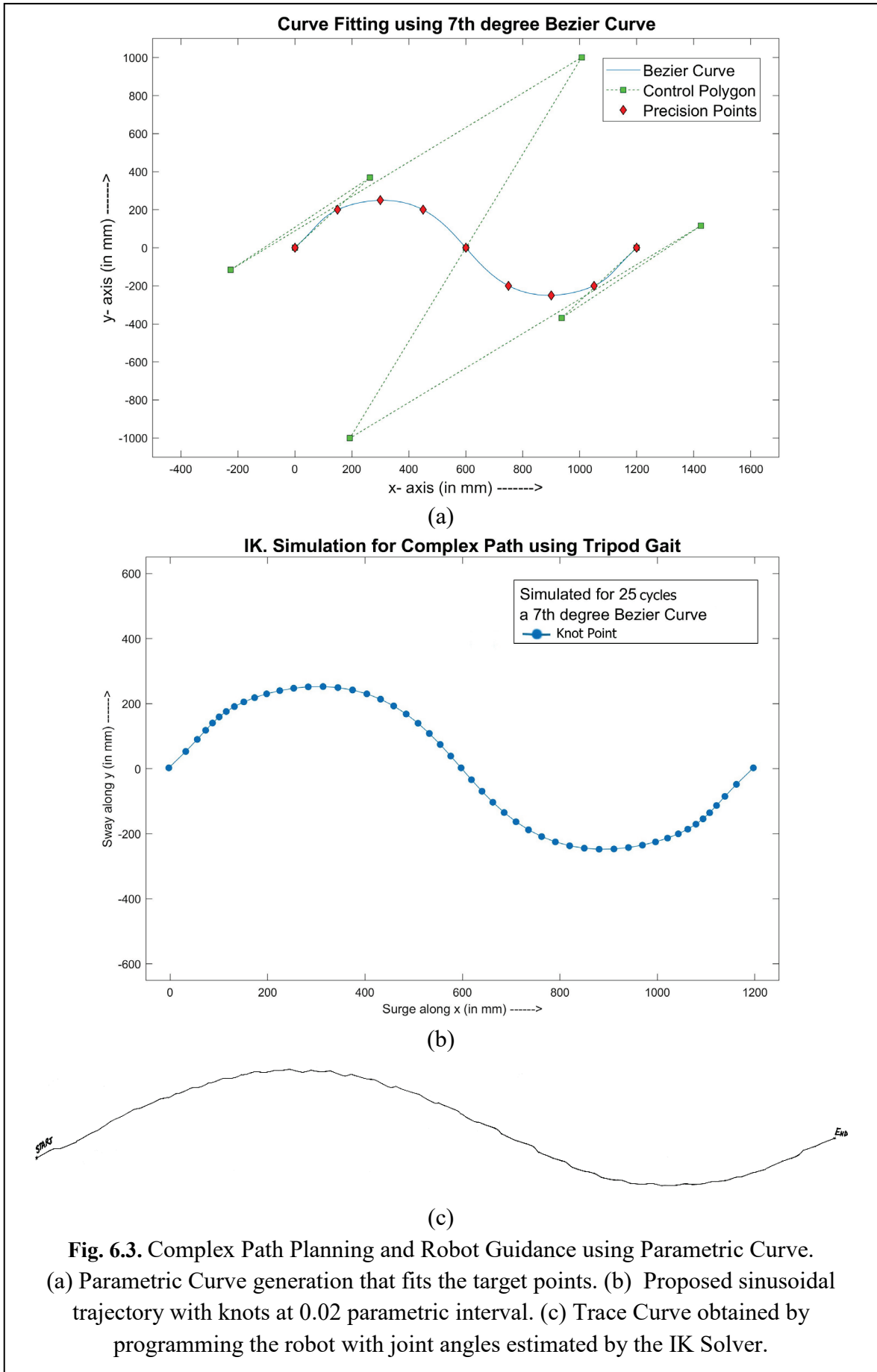
Arc-length interval method has not been used in this context because, considering the size of the Hexcrawler and the experimental workbench only smooth trajectories can be planned.

Proposed Algorithm:

- Step 1. Initialize the set of target points d_i and the degree of the Bezier curve n .
- Step 2. Use curve fitting to obtain the set of control vertices B .
- Step 3. Generate the Bezier curve and plot it graphically with the set of target points.
 Check whether the curve passes through d_i and the smoothness of the curve.
 If, the curve passes through d_i and is relatively smooth,
 Then, proceed to Step 4.
 Else, Change the value of n and go to Step 2.
- Step 4. Select interpolating step size, Δu also called the parametric interval.
- Step 5. Define the joint angles of the Hexcrawler at the starting position $u = 0$.
- Step 6. While $u \leq 1$.
- Set $u_A = u$ and $u_B = u + \Delta u$
 - Compute derivatives at u_A, u_B and obtain the yaw angles γ_A, γ_B .
 - Define the body frames $\{B\}_A$ and $\{B\}_B$.
 - The leg pivot and foot-tips $l_i|_A, l_i|_B$ and $f_i|_A$ is pre-defined.
 - Use IK Solver to obtain $f_i|_B$ and θ_{hi} using Stance phase.
 - Perform Swing and change of support legs as per Tripod Stepping Sequence.
 - Update $u = u + \Delta u$
- Step 7. The set of values for θ_{hi} is saved in excel file.

6.4. Results and Discussion

A complex trajectory has been constructed using Bezier curve fit that passes through a set of pre-defined target points as shown in Fig. 6.3 (a). The resultant curve is parametric, and knots are selected at parametric intervals of $\Delta u = 0.02$. The curve is sinusoidal in nature and uniform, hence use of parametric intervals is sufficient for selection of knots, as shown in Fig. 6.3 (b). Using the algorithm stated above the history of joint angles are obtained which would allow the Hexcrawler to trace the desired path. Programming the robot based on the solution obtained yields the trace curve shown in Fig. 6.3 (c). The solution set has not been provided since it is an exhaustive Table of 151 rows by 12 columns. The movement of the Hexcrawler is largely dependent on the surface friction, hence the trace curve will vary in dimensions for different surface conditions, but the overall shape and nature will be perfectly identical to the designed curve as shown in Fig. 6.3 (b) and Fig.6.3 (c).



PART - III. Trajectory Optimization for Jerk.

So far an n^{th} order Bezier curve has been constructed that passes through the target points. An algorithm that guides a mobile robot, such as the Hexcrawler has also been stated in Part – II. Now, it is desired to measure the time dependent kinematic parameters like velocity, acceleration, and jerk. For this the spatial curve must be linked to time frame. Further, the trajectory can be optimized to minimize a kinematic parameter such as jerk.

6.5. Need for Optimization

The trajectory planning and robot guidance can be optimized to obtain maximum benefits. Several literatures suggested on minimizing jerk, total travel time, energy consumption, maximizing payload capacity, etc. In the current thesis work, the jerk has been minimized because it has the following advantages:

- i. Generates a smooth motion, beneficial for material handling and transportation,
- ii. Reduces deviation from the designed path,
- iii. Reduces mechanical stresses on the actuators, drive system and structural components,
- iv. Reduces wear and tear, increasing the life span of the robot,
- v. Reduced chances of exciting the resonating frequency.

The trajectory optimization problem is not model specific, a general overview is presented. In the subsequent section some mathematical tools have been presented with the process of optimizing the trajectory.

6.6. Evaluating Time-Dependent Kinematic Parameters

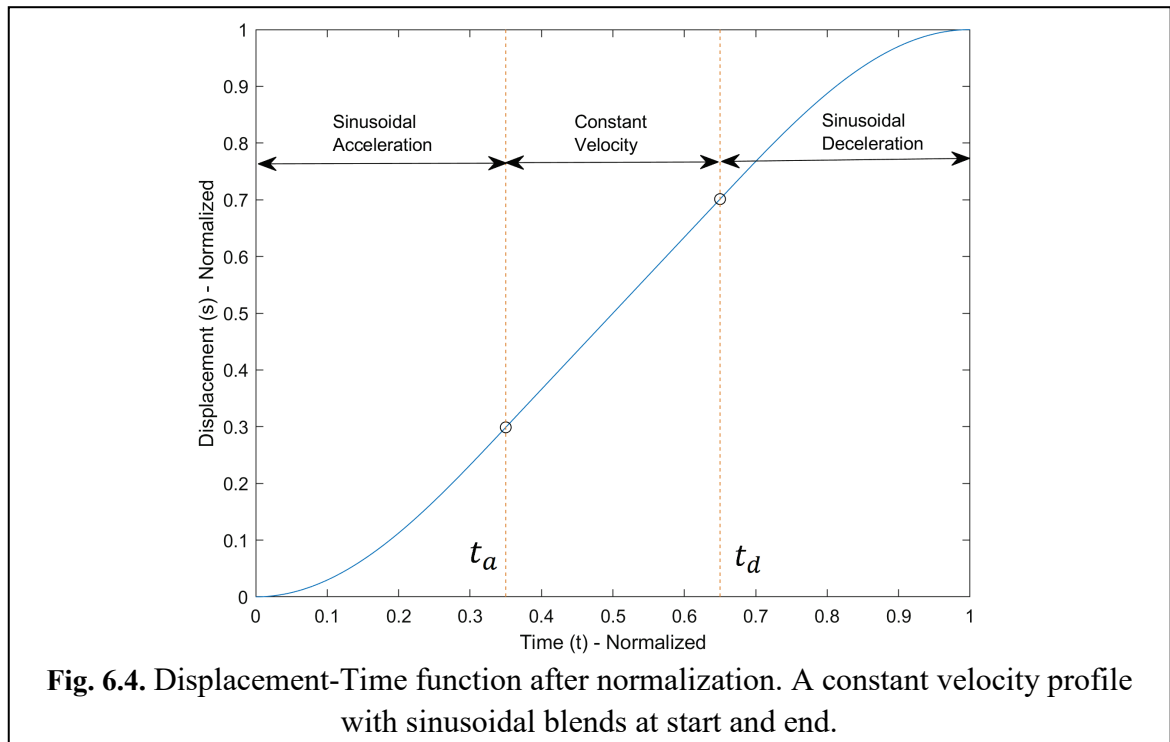
Part – I and II deals with the Spatial Planning of the Trajectory. However space planning alone is insufficient to describe time dependent kinematic parameters. So, the designed trajectory must be linked to a time frame. This can be achieved by defining a distance-time function as follows: $s = f(t)$, here s is the distance = arc length and t is the time. The ‘space-curve’ defines the path to be followed by an object, and the ‘displacement-time’ function relates time to the distance travelled. The arc-length is also a function the parametric variable u , denoted by $s(u)$. The distance along the curve is related to the parametric value by the Arc-Length Parameterization of the space-curve. It allows movement along the curve at a constant speed by evaluating the curve at equal arc-length intervals. Kinematic

parameters like velocity and acceleration can be computed by controlling the distance travelled in a given time interval. The displacement-time function must be monotonic and continuous in nature. From a real-world perspective a displacement-time function is constructed using a sinusoidal segment for acceleration, a linear segment for uniform velocity and a sinusoidal segment for deceleration. Such a function resembles a system that starts at rest, then accelerates up to a peak velocity and continues to travel at that constant velocity, finally decelerates to rest, refer Parent [85]. The plot of s vs t has been plotted and shown in Fig. 6.4. Sinusoidal functions are infinitely derivable leading to finite higher order derivatives like jerk and ping. The displacement-time function after normalization can be expressed as:

$$s(t) = \begin{cases} \frac{\frac{2t_a}{\pi}(\sin(\frac{\pi t}{2t_a} - \frac{\pi}{2}) + 1)}{f} & ; 0 \leq t \leq t_a \\ \frac{(\frac{2t_a}{\pi} + t - t_a)}{f} & ; t_a \leq t \leq t_d \\ \frac{(\frac{2t_a}{\pi} + t_d - t_a + \frac{2(1-t_d)}{\pi} \sin(\frac{\pi(t-t_d)}{2(1-t_d)}))}{f} & ; t_d \leq t \leq 1 \end{cases} \dots\dots (6.24)$$

Where, $f = \frac{2t_a}{\pi} + t_d - t_a + \frac{2(1-t_d)}{\pi}$

t_a and t_d are the normalized blending times for acceleration and deceleration respectively, $s(t)$ is the normalized arc-length.



Now it is possible to compute the kinematic parameters- velocity, acceleration and jerk using central-difference method.

If L_B^T represents the total arc-length of the designed trajectory, then,

True displacement is given by: $\bar{s}(t) = L_B^T \cdot s(t)$ (6.25a)

Velocity is given by: $v = \dot{\bar{s}} = \frac{\bar{s}(t+\Delta t) - \bar{s}(t-\Delta t)}{2\Delta t}$ (6.25b)

Acceleration is given by: $a = \ddot{\bar{s}} = \frac{\bar{s}(t+\Delta t) - 2\bar{s}(t) + \bar{s}(t-\Delta t)}{\Delta t^2}$ (6.25c)

Jerk is given by: $j = \ddot{\bar{s}} = \frac{\bar{s}(t+2\Delta t) - 2\bar{s}(t+\Delta t) + 2\bar{s}(t-\Delta t) - \bar{s}(t-2\Delta t)}{2\Delta t^2}$ (6.25d)

6.7. Minimizing Jerk

The designed trajectory is now completely defined in terms of both space and time. So, optimization is carried out with the objective of minimizing the jerk for a $(n - 1)$ degree Bezier curve following a predefined distance-time function. For such an optimization problem, we define the following:

- Objective function: minimize square of jerk j , since jerk value can be both positive and negative.
- Design Variable: blending time t_a and t_d . For simplification, $t_d = (1 - t_a)$ i.e. equal blend times for acceleration and deceleration.
- Constraints: Limiting values of velocity and acceleration denoted by v_{limit} , a_{limit} respectively.
- Constants: L_B^T (i.e. the designed trajectory is invariant of time), and t_{traj} (i.e. the total time taken to traverse the trajectory).

Therefore the optimization problem can stated as:

Minimize $j^2(t)$,
 Subject to: $v \leq v_{limit}$ and $a \leq a_{limit}$

Outcome of Optimization:

Optimum Blend Times t_a^{opt} and t_b^{opt} .

The optimization has been carried out using MATLAB™ Optimization Toolbox. The solver selected for the purpose is `fmincon()` which can minimize constrained, nonlinear, multivariable problems. The problem was solved using MATLAB™ version 2017b. The overall optimization process has been presented in the form of an algorithm.

Algorithm for Implementation:

Step 1. Define a set of points d , set knots points k , degree n , and t_{traj} for each segment, values of v_{limit} , a_{limit} based on kinematic model of the robot, and step size stp .

Step 2. Use curve-fit to construct the Bezier trajectory segments and join them at knot points using G^0 continuity.

Step 3. Define disp-time function for each segment based on the desired kinematic condition at the knot points.

Step 4. Calculate Arc Length L_B^i for each segment i .

For $i = 1$ to (no of segments)

For $t = 0$ to 1, (normalized time)

- calculate \bar{s} using algorithm in Section II.C
- calculate $\dot{\bar{s}}$, $\ddot{\bar{s}}$ and $\ddot{\bar{s}}$
- optimize using *fmincon* and obtain t_{ai}^{opt} , t_{bi}^{opt} .

Step 5. Use t_{ai}^{opt} , t_{bi}^{opt} in the respective disp-time function to obtain \bar{s} , $\dot{\bar{s}}$, $\ddot{\bar{s}}$ and $\ddot{\bar{s}}$ at each step.

6.8. Results and Discussion

The effectiveness of the algorithm stated above can be best described by implementing it in three different aspects, presented in the form of case studies:

Case Study 1.

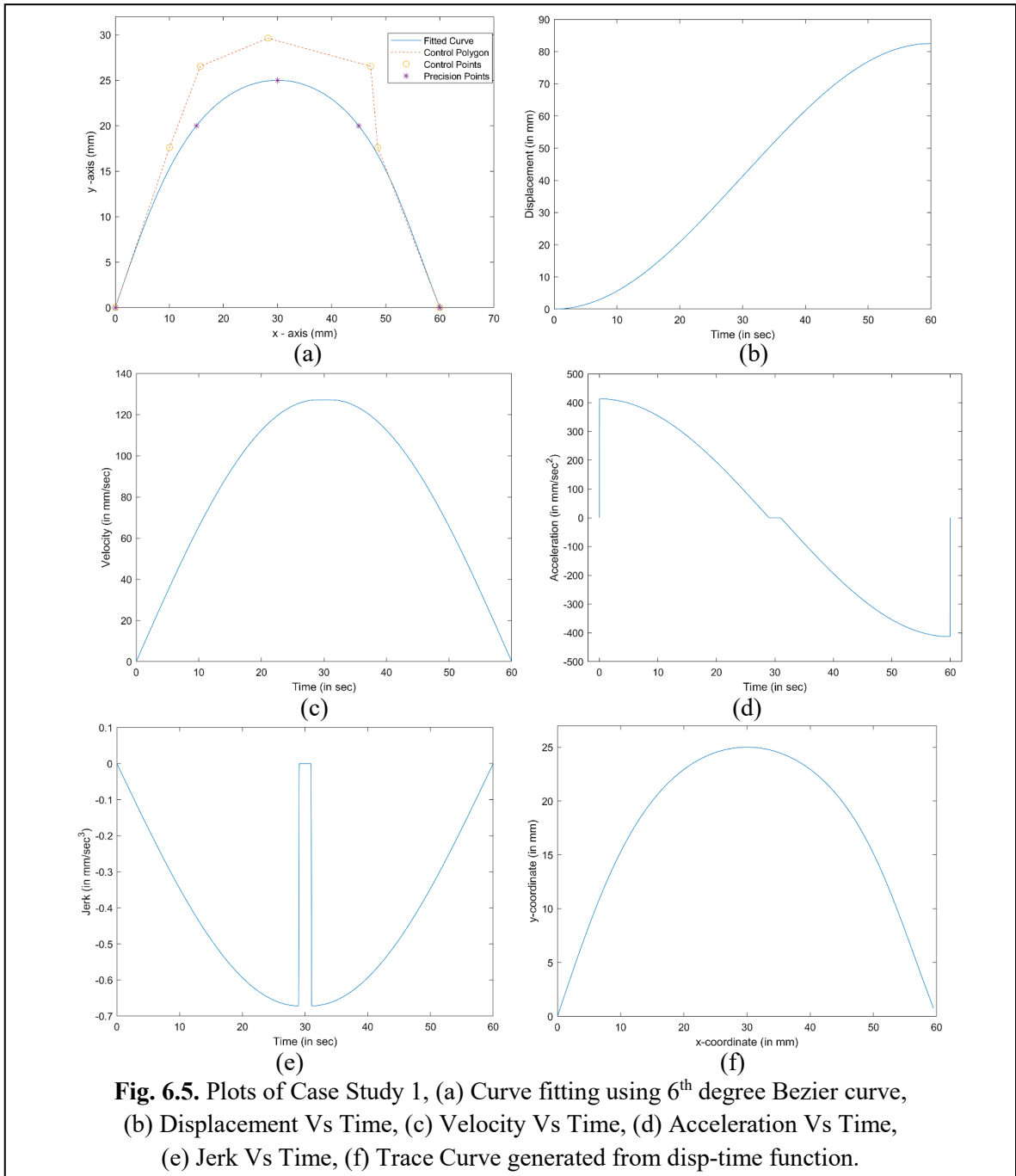
A simple planar trajectory has been constructed by joining 5 target points using a 6th degree Bezier curve segment. A displacement-time function of constant velocity with sinusoidal blends has been used. On optimizing with a specified boundary conditions, the optimum blend time is obtained and results are enlisted in Table 6.2. By varying time t from 0 to t_{traj} , the kinematic parameters have been evaluated at each time step. The variation of kinematic parameters with time has been plotted and shown in Fig. 6.5.

Boundary conditions:

- Travel time, $t_{traj} = 60$ sec,
- Kinematic Limits, $v_{limit} = 300$ mm/s, $a_{limit} = 600$ mm/s².
- Normalized step size, $stp = 0.0005$.

Table 6.2. Optimization Results for Case-1:

Parameter	Value
Opt. Blending Time	$t_a = 29.0017\text{sec}$
	$t_d = 30.9983\text{sec}$
Path Length L_B	82.5109 mm
Peak Velocity v_{max}	127.1918 mm/sec
Peak Acceleration a_{max}	413.3393 mm/sec ²
Peak Jerk j_{max}	0.67162 mm/sec ³



Case Study 2.

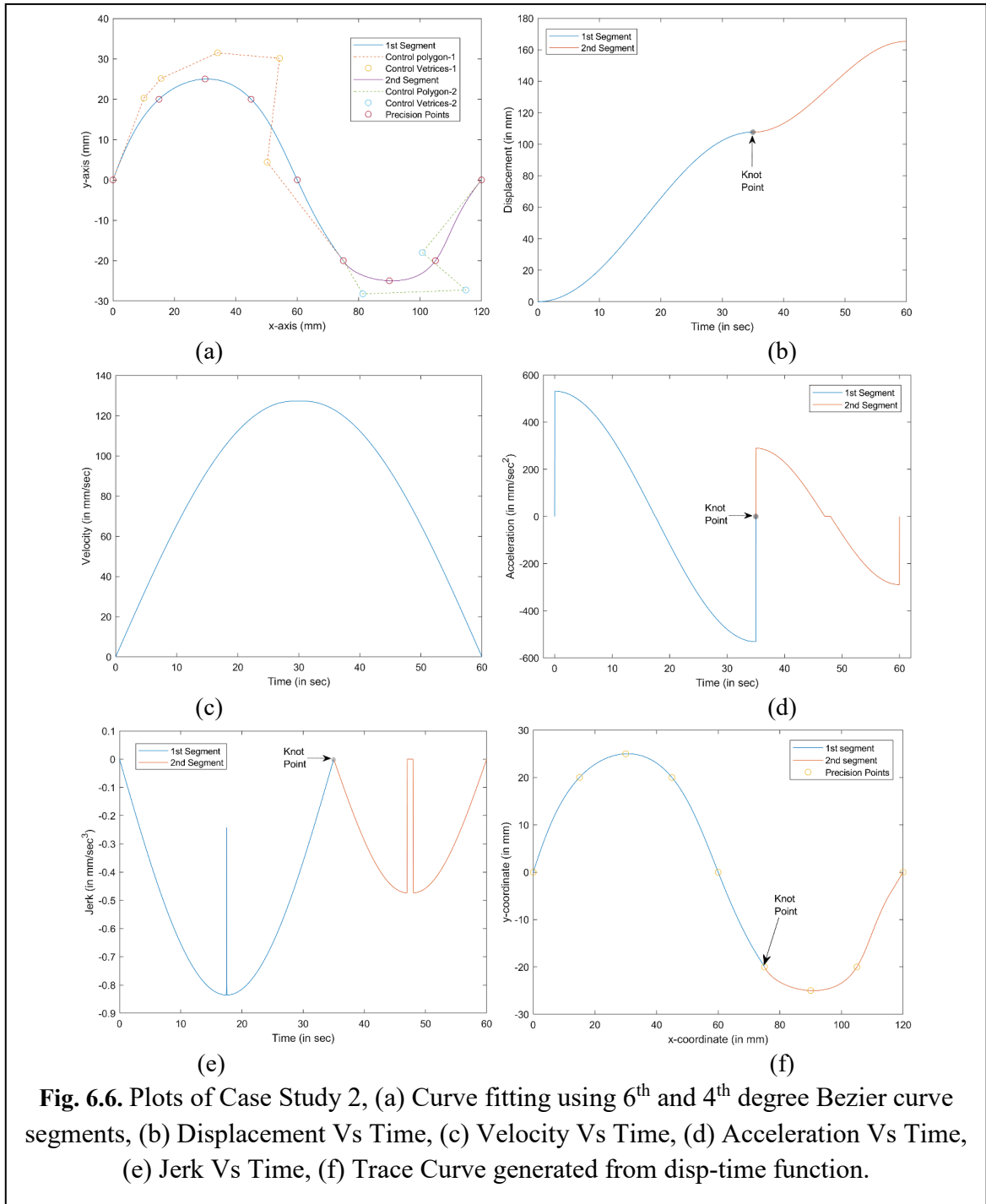
The planar trajectory constructed in this case comprises of two segments of varying degrees and arc length. 9 target points are considered with the 6th point as knot point. The degree of the 1st and 2nd segment is 6 and 4 respectively. The displacement-time function has been imposed and optimized with a specified boundary conditions. The optimum blend time is obtained and results are enlisted in Table 6.3. By varying time t from 0 to t_{traj} , the kinematic parameters have been evaluated at each time step. The variation of kinematic parameters with time has been plotted and shown in Fig. 6.6.

Boundary Conditions:

- Travel time, $t_{traj}^{(1)} = 35$ sec, $t_{traj}^{(2)} = 25$ sec
- Kinematic Limits, $v_{limit} = 180$ mm/s, $a_{limit} = 800$ mm/s².
- Normalized step size, $stp = 0.0005$.
- $v = 0$, $a = 0$ at knot point.

Table 6.3. Optimization Results for Case-2:

Parameter	Segment - 1	Segment - 2
Opt. Blending Time	$t_a = 17.4861$ sec	$t_a = 12.0038$ sec
	$t_d = 17.5139$ sec	$t_d = 12.9962$ sec
Path Length L_B	107.6694 mm	57.5997 mm
Peak Velocity v_{max}	169.0501 mm/sec	88.4727 mm/sec
Peak Acceleration a_{max}	531.5076 mm/sec ²	289.4347 mm/sec ²
Peak Jerk j_{max}	0.83555 mm/sec ³	0.47343 mm/sec ³



Case Study 3.

In this case, the blend time t_a has been varied from 0 to $t_{traj}/2$. The variation of jerk with blend time has been shown in Table 6.4. It is seen that smaller blend time leads to higher acceleration and hence higher amount of jerk. The jerk decreases as the blend time approaches $t_{traj}/2$. It is interesting to note although the minimum jerk occurs

at $t = t_{traj}/2$, it is unobtainable from the algorithm as it would imply eliminating the constant velocity section of the disp-time function. The effect of changing v_{limit} and a_{limit} is studied. Initially, the magnitude of v_{limit} is reduced keeping a_{limit} constant. Then, the magnitude of a_{limit} is reduced keeping v_{limit} constant. The result is tabulated and shown in Table 6.5. It is observed that the optimum blend time t_a approaches $t_{traj}/2$, and jerk reduces simultaneously.

Table 6.4. Variation of Jerk with Blend Time:

Sl. No.	Travel Time (sec)	Actuator Limits		Blend Time (sec)		Calculated Values		Peak Jerk	Remarks
		v_{limit} (mm/sec)	a_{limit} (mm/sec ²)	t_a	t_d	v_{max} (mm/sec)	a_{max} (mm/sec ²)	j_{max} (mm/sec ³)	
1	60	150	800	5	55	87.83	1655.47	15.59	$a_{max} > a_{limit}$ Infeasible solution
2				10	50	93.88	884.81	4.16	
3				15	45	100.83	633.53	1.99	Feasible but not optimum
4				20	40	108.88	513.12	1.20	
5				25	35	118.34	446.16	0.84	
6				29.50	30.49	128.39	410.16	0.65	Obtained Optimum
7				30	30	129.60	407.17	0.63	Ideally Optimum

Table 6.5. Variation of Jerk with Actuator Limits:

Sl. No.	Travel Time (sec)	Actuator Limits		Opt. Blend Time (sec)		Calculated Values		Peak Jerk
		v_{limit} (mm/sec)	a_{limit} (mm/sec ²)	t_a	t_d	v_{max} (mm/sec)	a_{max} (mm/sec ²)	j_{max} (mm/sec ³)
1	60	120	800	25.6355	34.3645	119.6702	439.9618	0.80874
2		130		29.8426	30.1574	129.2208	408.0998	0.64442
3		140		29.028	30.972	127.2545	413.1673	0.67073
4				29.5016	30.4984	128.3903	410.1637	0.65516
5		790	29.5071	30.4929	128.4037	410.1294	0.65499	
6		150	780	29.0084	30.9916	127.2078	413.2953	0.67139
7		770	29.0103	30.9897	127.2123	413.2831	0.67133	
8		760	29.7521	30.2479	128.9994	408.6396	0.64723	

6.9. Conclusion

The results obtained from experiments and simulation show that the algorithm is successful. Using this technique, trajectories of any shape and complexity can be constructed and traced by the robot. For a robot traversing along a sinusoidal path, the yaw angle increases from zero to maximum, then goes on reducing to a minimum and again increases and stops at zero. Hence the choice of this curve is critical, as it proves that robot can take up both right and left sided curvatures as per the situation. So, it can be fairly concluded that the algorithm can solve any complex trajectories which is defined by parametric curves.

The trajectory planner can be implemented by combining it with the state-space model of a mobile robot. The use of n^{th} degree Bezier curves makes modeling of complex trajectory easier. Blending time is a critical parameter for the robot controller, hence optimizing it is beneficial. The use of knots allows the user to enforce local kinematic states such as using an intermediate stop. The algorithm and model is fast and converges to a solution rapidly. The novelty of the work is thus generation of n^{th} degree Bezier curves with knots and optimization of blending time that may help in designing trajectories suitable for using mobile manipulators in rough terrains.

Chapter 7.

Conclusion and Future Scope.

7.1. Conclusion

In the current thesis work, the detailed kinematic analysis of the Hexcrawler robot was performed. The Hexcrawler robot is not a perfect testing model, however suitable changes have been made to the robot to suit the purpose. This includes changing the control architecture and installing a tracing attachment. The leg which is a 2 d.o.f linkage is analysed using DH-method. The workspace is generated comprises of all feasible positions of the foot-tip calculated w.r.t the leg pivot and shown on a 3D plot. Further, considering all six legs, it has been shown that the workspaces are non-intersecting within a prescribed actuator limit. The primary gaits have been studied and programmed on the Hexcrawler by tracing a straight line path.

To accurately guide the robot along a prescribed path and to predict its motion, the forward and inverse kinematic models were developed. Considering the physical limitations of the robot, the conditions of planar maneuvering has been imposed on the kinematic model. Different trajectories have been simulated using the kinematic model and the results were validated using experimentally obtained trace curves. The simulated and experimental curves were found to be of the same nature. However, dimensionally it has not been compared since the effects of friction and dynamics has not been considered. The inverse kinematics solver has also been tested for an S- shaped curve modelled using Bezier curve. The nature of the experimentally obtained trace curve was found to be satisfactory.

The idea of optimizing the parametric trajectory is generalized and can be implemented on any mobile robot – wheeled or legged. The trajectory planner can be implemented by combining it with the state-space model of a mobile robot. The use of n^{th} degree Bezier

curves provides greater flexibility to shape and interpolate trajectories. The optimum blending time is used by the robot controller to provide smoother control signals to the servo actuators.

7.2. Future Scope of Work

Although, the works presented in this thesis is according to aim and objectives stated in Chapter-1. However, there is more that could have been done. The kinematic analysis was limited to space planning only. Due to the limited time available, the time based analysis was omitted. So, calculating the Jacobian to estimate the velocity, acceleration and jerk could be considered for future work. Once, the state space model is obtained, it could be integrated with the trajectory optimization model. This way the kinematic model could be perfected. Further, there is scope of developing a dynamic model using the Lagrangian equations for formulating the equation of motion for the six-legged robot.

References

- [1] R.B. McGhee, “Some Finite State Aspects of Legged Locomotion”, *Mathematical Biosciences* 2, 67-84 (1968) by American Elsevier Publishing Company, Inc.
- [2] Y. Zhu, T. Guo, Q. Liu, Q. Li, R. Yan, A study of arbitrary gait pattern generation for turning of a bio-inspired hexapod robot, *Robotics and Autonomous Systems* (2017), <http://dx.doi.org/10.1016/j.robot.2017.08.012>.
- [3] Z.-Y. Wang, X.-L. Ding and A. Rovetta, “Analysis of typical locomotion of a symmetric hexapod robot”, *Robotica* (2010) Vol. 28, pp. 893-907, Cambridge University Press, doi: 10.1017/S0263574709990725.
- [4] Ervin Burkus, Peter Odry, “Autonomous Hexapod Walker Robot Szabad(ka)”, *Acta Polytechnica Hungarica*, Vol. 5, No. 1, 2008, pp. 69-85.
- [5] Zennir Youcef and Couturier Pierre, “control of the Trajectory of a Hexapod Robot based on Distributed Q-learning”, doi: 0-7803-8304-4/04, 2004 IEEE.
- [6] Zhi-Yuan Yang and Chia-Feng Juang, “Evolutionary Locomotion Control of a Hexapod Robot using Particle Swarm Optimized Fuzzy Controller”, 2014 IEEE International Conference on Systems, Man and Cybernetics, October 5-8,2014, San-Diego, CA, USA.
- [7] Ricardo Campos, Vitor Matos, Miguel Oliveira, Cristina Santos, “Gait Generation for a Simulated Hexapod Robot: a nonlinear dynamical system approach”, Industrial Electronics Department, University of Minho, Guimaraes, Portugal.
- [8] Shinkichi Inagaki, Hideo Yuasa, Tankori Suzuki, Tamio Arai, “Wave CPG model for autonomous decentralized multi-legged robot: Gait generation and walking control”, *Robotics and Autonomous Systems* 54 (2006) pp. 118-126, Elsevier.
- [9] M. Anthony Lewis, Andrew H. Fagg and George A. Bekey, “Genetic Algorithms for Gait Synthesis in a Hexapod Robot”, *Recent Trend in Mobile Robots*, World Scientific.
- [10] Servet Soyguder, Hasan Alli, “Kinematic and dynamic analysis of a Hexapod walking-running-bounding gaits robot and control actions”, *Computer and Electrical Engineering* 38 (2012), pp. 444-458, Elsevier.
- [11] Giuseppe Carbone and Marco Ceccarelli, “Legged Robotic Systems”, *Cutting edge Robotics*, ISBN 3-86611-038-3, pp. 784, Germany, July 2005.
- [12] Xilun Ding, Zhiying Wang, Alberto Rovetta and J.M. Zhu (2010). *Locomotion Analysis of Hexapod Robot, Climbing and Walking Robots*, Behnam Miripour (Ed.), ISBN: 978-953-307-030-8, InTech, Available from: www.intechopen.com/books/climbing-and-walking-robots/locomotion-analysis-of-hexapod-robot

-
- [13] Gary B. Parker and Jonathan W. Mills, “Metachronal Wave Gait Generator for Hexapod robots”, Department of Computer Science, Indiana University, Bloomington.
- [14] Zongquan Deng, Yiqun Liu, Liang Ding, Haibo Gao, Haitao Yu and Zhen Liu, “Motion planning and simulation verification of a hydraulic hexapod robot based on reducing energy/flow consumption”, *Journal of Mechanical Science and Technology* 29 (10) (2015) 4427~ 4436, doi:10.1007/s12206-015-0941-0.
- [15] Hua Deng, Guiyang Xin, Guoliang Zhong , Michael Mistry, “Object carrying of hexapod robots with integrated mechanism of leg and arm”, *Robotics and Computer-Integrated Manufacturing*, Elsevier 2017. <https://doi.org/10.1016/j.rcim.2017.11.014>.
- [16] Bo Jin, Cheng Chen, Wei Li, “Power Consumption Optimization for Hexapod Walking Robot”, *Journal of intelligent Robot Systems* (2013) 71:195-2019, doi: 10.1007/s10846-012-9771-9, Springer 2012.
- [17] J. M. Porta, E. Celaya, “Reactive free-gait generation to follow arbitrary trajectories with a hexapod robot”, *Robotics and Autonomous Systems* 47 (2004) 187-201, doi:10.1016/j.robot.2004.04.001, Elsevier.
- [18] E. Z. Moore, D. Campbell, F. Grimminger, and M. Buehler, “Reliable Stair Climbing in the Simple Hexapod RHex”, *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*, Washington DC, May 2002, doi: 0-7803-7272-7102.
- [19] Uluc Saranlı, Martin Buehler, Daniel E. Koditschek, “Rhex: A Simple and Highly Mobile Hexapod Robot”, *The International Journal of Robotics Research*, Vol. 20, No. 7, July 2001, pp. 616-631, Sage Publications.
- [20] István Kecskés, Ervin Burkus, Peter Odry, “Swarm-Based Optimizations in Hexapod Robot Walking”, 9th IEEE International Symposium on Applied Computational Intelligence and Informatics, May 15-17, Timisoara, Romania, doi: 978-1-4799-4694-5/14.
- [21] Shuaishuai Zhang, Yibin Li, Rui Song, Xuewen Rong and Bin Li, “A Free Gait Planning Method Based on the Foothold Search Strategy for Quadruped Robot”, school of Control Science and Engineering, Shandong University, Jinan, China.
- [22] Prabir K. Pal, Vivek Mahadev, and K. Jayarajan, “Gait Generation for a six-legged Walking Machine through Graph Search”, doi: 1050-4729/9, IEEE.
- [23] Prabir K. Pal and K. Jayarajan, “Generation of Free Gait – A Graph Search Approach”, *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 3, June 1991, pp. 299-305, doi: 1042-296X/91/0600-0299.
-

-
- [24] Xingji Duan, Weihai Chen, Shouqian Yu, and Jingmeng Liu, "Tripod Gaits Planning and Kinematics Analysis of a Hexapod Robot", 2009 IEEE International Conference on Control and Automation, Christchurch, New Zealand, December 9-11, 2009, doi: 978-1-4244-4707-7/09, IEEE.
- [25] Bill Goodwine and Joel Burdick, "Trajectory Generation for Kinematic Legged Robots", Proceedings of the 1997 IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico – April 1997, doi: 0-7803-3612-7-4/97.
- [26] Sorin Mănoiu-Olaru, Mircea Nițulescu, "Basic Walking Simulations and Gravitational Stability Analysis for a Hexapod Robot Using MatLab", Department of Automation, Electronics and Mechatronics, University of Craiova, Romania.
- [27] Jun Yang and Z. Jason Geng, "Closed Form Forward Kinematics Solution to a Class of Hexapod Robots", IEEE Transactions on Robotics and Automation, Vol. 14, No. 3, June 1998, doi: 1042–296X/98.
- [28] Manuel F. Silva, J.A. Tenreiro Machado, Ramiro S. Barbosa, "Complex-order dynamics in hexapod locomotion", Signal Processing 86 (2006) 2785–2793, 2006 Elsevier, doi:10.1016/j.sigpro.2006.02.024.
- [29] Manuel F. Silva, Ramiro S. Barbosa, J. A. Tenreiro Machado, "Development of a Genetic Algorithm for the Optimization of Hexapod Robot Parameters", Proceedings of the IASTED International Conference Applied Simulation and Modelling (ASM 2009) September 7 - 9, 2009 Palma de Mallorca, Spain.
- [30] Jin Bo , Chen Cheng, Li Wei, Li Xiangyun, "Design and configuration of a hexapod walking robot", 2011 Third International Conference on Measuring Technology and Mechatronics Automation, 978-0-7695-4296-6/11, 2011 IEEE, doi: 10.1109/ICMTMA.2011.216.
- [31] Franco Tedeschi and Giuseppe Carbone, "Design Issues for Hexapod Walking Robots", *Robotics* 2014, 3, 181-206; doi: 10.3390/robotics3020181, ISSN 2218-6581, www.mdpi.com/journal/robotics.
- [32] Guoliang Zhong, Hua Deng, Guiyang Xin, Hengsheng Wang, "Dynamic Hybrid Control of a Hexapod Walking Robot: Experimental Verification", Doi: 10.1109/TIE.2016.2551679, IEEE Transactions on Industrial Electronics.
- [33] A. Schneider and U. Schmucker (2006). Force Sensing for Multi-Legged Walking Robots: Theory and Experiments Part 1: Overview and Force Sensing, Mobile Robotics, Moving Intelligence, Jonas Buchli Ed. ISBN: 3-86611-284-X, InTech.
- [34] A. Schneider and U. Schmucker (2006). Force Sensing for Multi-Legged Walking Robots: Theory and Experiments Part 2: Force Control of Legged Vehicles, Mobile Robotics, Moving Intelligence, Jonas Buchli Ed. ISBN: 3-86611-284-X, InTech.
-

-
- [35] Salvador M. c. Netto, Alexandra Evsukoff, and Max Suell Dutra, “Fuzzy Systems to Solve Inverse Kinematics Problems in Robots Control: Application to a Hexapod Robot’s Leg”, 1522-4899/00, 2010 IEEE.
- [36] Aiste Skaburskyte, Mindaugas Luneckas, Tomas Luneckas, Jonas Kriauciunas, Dainius Udris, “Hexapod Robot Gait Stability Investigation”, 978-1-5090-4473-3/16, 2016 IEEE.
- [37] Garcia Lopez, M.C. Gorrostieta-Hurtado, E. Vargas-Soto, E. Ramos-Arreguín, J. M. Sotomayor-Olmedo, A. Moya Morales, “Kinematic analysis for trajectory generation in one leg of a hexapod robot”, The 2012 Iberoamerican Conference on Electronics Engineering and Computer Science, doi:10.1016/j.protcy.2012.03.037, 2012 Elsevier.
- [38] D. E. Orin, R. B. McGhee, M. Vukobratovic, G. Hartoch, “Kinematic and Kinetic Analysis of Open-Chain Linkages Utilizing Newton-Euler Methods”, *Mathematical Biosciences* 43, pp. 107-130, doi: 0025-5564/79/010107, Elsevier.
- [39] P. Gonzalez de Santos, E. Garcia, R. Ponticelli & M. Armada (2009), “Minimizing Energy Consumption in Hexapod Robots”, *Advanced Robotics*, 23:6, 681-704, doi: 10.1163/156855309X431677.
- [40] Abhijit Mahapatra, Shibendu S. Roy, Dilip K. Pratihari,” Modelling and Simulation of Wave Gait of a Hexapod Walking Robot: A CAD/CAE Approach”, *International Journal of Robotics and Automation (IJRA)* Vol. 2, No. 3, September 2013, pp. 104~111, ISSN: 2089-4856 doi: 10.11591/ijra.v2i3.2016.
- [41] Ya-guang Zhu, Bo Jin, Wei Li, Shi-tong Li, “Optimal Design of Hexapod Walking Robot Leg Structure based on Energy Consumption and Workspace”, *Transactions of the Canadian Society of Mechanical Engineering*, Vol. 38, No. 4, 2014. No. 13-CSME-125, E.I.C. Accession Number 3583.
- [42] C. Mahfoudi, K. Djouani, S. Rechak, M. Bouaziz, “Optimal Force Distribution for the legs of a Hexapod Robot”, doi: 0-7803-7729-X/03, 2003 IEEE.
- [43] Yudi Isvara, Syawaludin Rachmatullah, Kusprasapta Mutijarsa, Dinara Enggar Prabakti, Wiharsa Pragitatama, “Terrain Adaptation Gait Algorithm in a Hexapod Walking Robot”, 2014 13th International Conference on Control, Automation, Robotics & Vision, Marina Bay Sands, Singapore, 10-12th December 2014 (ICARCV 2014), 978-1-4799-5199-4/14, 2014 IEEE.
- [44] Shin-Min Song and Byoung Soo Choi, “A study on Continuous Follow-The Leader (FTL) Gaits: and effective Walking Algorithm over Rough Terrain”, *Mathematical Biosciences* 97, pp. 199-233, 1989 Elsevier, doi: 0025-5565/89.
- [45] R. B. McGhee and G. I. Iswandhi, “Adaptive Locomotion of a Multi-Legged Robot over Rough Terrain”, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. Smc-9, No. 4, April 1979, pp. 176-182, doi: 0018-9472/79/0400-0716.
-

-
- [46] D. Belter, “Adaptive Foothold Selection for a Hexapod Robot Walking on Rough Terrain”, Institute of Control and Information Engineering, Poznan University of Technology, Poznan, Poland.
- [47] K. Hauser, T. Bretl, J.C. Latombe, K. Harada, “Motion Planning for Legged Robots on Varied Terrain”, *The International Journal of Robotics Research* Vol. 27, No. 11–12, Nov/Dec 2008, pp. 1325–1349, doi: 10.1177/0278364908098447, SAGE Publications 2008.
- [48] Vijay R. Kumar and Kenneth J. Waldron, “Adaptive Gait Control for a Walking Robot”, *Journal of Robotic Systems*, 6(1), pp. 49-76, 1989 John Wiley & Sons, doi: CCC 0471-2223/89/010049-28.
- [49] Mrinal Kalakrishnan, Jonas Buchli, Peter Pastor, Michael Mistry, and Stefan Schaal, “Fast, Robust Quadruped Locomotion over Challenging Terrain”, 2010 IEEE International Conference on Robotics and Automation Anchorage Convention District, May 3-8, 2010, Anchorage, Alaska, USA, doi: 978-1-4244-5040-4/10, 2010 IEEE.
- [50] Maureen N. Rojas Medrano, Novel Certad, José Cappelletto, Juan Carlos Grieco, “Foothold planning and gait generation for a hexapod robot traversing irregular terrain”, *Mechatronics Research Group. Universidad Simon Bolívar, Caracas, Venezuela.*
- [51] Maureen Rojas, Novel Certad, José Cappelletto, Juan Carlos Grieco, “Foothold planning and gait generation for a hexapod robot traversing terrains with forbidden zones”, 2015 12th Latin American Robotics Symposium and 2015 Third Brazilian Symposium on Robotics, 978-1-4673-7129-2/15, 2015 IEEE, doi: 10.1109/LARS-SBR.2015.7049.
- [52] Hua Deng, Guiyang Xin, Guoliang Zhong, Michael Mistry, “Gait and trajectory rolling planning and control of hexapod robots for disaster rescue applications”, *Robotics and Autonomous Systems*, <http://dx.doi.org/10.1016/j.robot.2017.05.007>, 0921-8890, 2017 Elsevier.
- [53] Aaron M. Hoover, Samuel Burden, Xiao-Yu Fu, S. Shankar Sastry, and R. S. Fearing, “Bio-inspired design and dynamic manoeuvrability of a minimally actuated six-legged robot”, *Proceedings of the 2010 3rd IEEE RAS & EMBS International Conference on Biomedical Robotics and Bio-mechatronics*, The University of Tokyo, Tokyo, Japan, September 26-29, 2010. 978-1-4244-7709-8/10, 2010 IEEE.
- [54] N. Plamondon and M. Nahon, “A trajectory tracking controller for an underwater hexapod vehicle”, *Bioinspiration & Biomimetics*, 4 (2009) 036005 (13pp), IOP Publishing, doi:10.1088/1748-3182/4/3/036005.

-
- [55] Eleni Aggelopoulou, Georgios Rekleitis, and Evangelos Papadopoulos, “Optimal Leg-Sequence Selection for an Underwater Hexapod Robot in the Presence of Slopes and External Forces”, 25th Mediterranean Conference on Control and Automation (MED) July 3-6, 2017. Valletta, Malta. 978-1-5090-4532-7/17, IEEE.
- [56] Ch. Grand, F. BenAmar, F. Plumet and Ph. Bidaud, “Decoupled control of Posture and Trajectory of the hybrid Wheel-Legged robot Hylos”, Proceedings of the 2004 IEEE International Conference on Robotics & Automation, New Orleans, LA, April 2004. 0-7803-8232-3/04, 2004 IEEE.
- [57] Bong-Huan Jun, Hyungwon Shim, Banghyun Kim, Jin-Yeong Park, Hyuk Baek, Seongyeol Yoo, Pan-Mook Lee, “Development of Seabed Walking Robot CR200”, 978-1-4799-0002-2/13, 2013 IEEE.
- [58] Bong-Huan Jun, Hyungwon Shim, Banghyun Kim, Jin-Yeong Park, Hyuk Baek, Seongyeol Yoo, Hango Kang, Gyeong-mok Lee, Pan-Mook Lee, “First Field-Test of Seabed Walking Robot CR200”, 978-0-933957-40-4 MTS.
- [59] K. Autumn, M. Buehler, M. Cutkosky, R. Fearing, R. J. Full, D. Goldman, R. Groff, W. Provancher, A. A. Rizzi, U. Saranli, A. Saunders, D.E. Koditschek, “Robotics In Scansorial Environments”, Unmanned Ground Vehicle Technology VII, edited by Grant R. Gerhart, Charles M. Shoemaker, Douglas W. Gage, Proceedings of SPIE Vol. 5804 (SPIE, Bellingham, WA, 2005), 0277-786X/05, doi: 10.1117/12.606157.
- [60] Yaguang Zhu, Bo Jin, Yongsheng Wu, Tong Guo and Xiangmo Zhao, “Trajectory Correction and Locomotion Analysis of a Hexapod Walking Robot with Semi-Round Rigid Feet”, Sensors 2016, 16, 1392; doi:10.3390/s16091392, MDPI.
- [61] Filipp Seljanko, “Towards Omnidirectional Locomotion Strategy for Hexapod Walking Robot”, Proceedings of the 2011 IEEE International Symposium on Safety, Security and Rescue Robotics Kyoto, Japan, Nov. 1-5 2011, 978-1-61284-769-6/11, IEEE.
- [62] Dong Han, Hong Nie, Jinbao Chen, Meng Chen, “Dynamic obstacle avoidance for manipulators using distance calculation and discrete detection”, Robotics and Computer-Integrated Manufacturing, <http://dx.doi.org/10.1016/j.rcim.2017.05.013>, 0736-5845, 2017 Elsevier.
- [63] S. Skaff, G. Kantor, D. Maiwand, A. A. Rizzi, “ Inertial Navigation and Visual Line following for a Dynamical Hexapod Robot”, Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, Nevada, Oct-2003. 0-7803-7806-1/03, IEEE.
- [64] Y. Zhao, X. Chai, F. Gao, C. Qi, Obstacle avoidance and motion planning scheme for a hexapod robot Octopus-III, Robotics and Autonomous Systems (2018), <https://doi.org/10.1016/j.robot.2018.01.007>.
-

-
- [65] Jung-Min Yang, “Omnidirectional walking of legged robots with a failed leg”, *Mathematical and Computer Modelling* 47 (2008) 1372–1388, doi: 10.1016/j.mcm.2007.08.006, Elsevier.
- [66] A. Piazzoli and A. Visioli, “An Interval Algorithm for Minimum-Jerk Trajectory Planning of Robot Manipulators”, *Proc. of the 36th Conference on Decision & Control*, IEEE press, Dec. 1997, pp. 1924-27, doi: 0-7803-3970-8/97.
- [67] S. Saramago, “Optimization of the Trajectory Planning of Robot Manipulators taking into account the Dynamics of the System”, Elsevier Science, *Mech. Mach. Theory*, Vol. 33, No. 7, pp. 883-94, 1998
- [68] A. Gasparetto and V. Zanotto, “A technique for time-jerk optimal planning of robot trajectories”, Elsevier Science, *Robotics and Computer-integrated Manufacturing* 24(2008), doi:10.1016/j.rcim.2007.04.001.
- [69] M. H. Korayem, M. Nazemizadeh, and H. Rahimi, “Dynamic Optimal Payload path planning of mobile manipulators among moving obstacles”, *Advanced Robotics*, 2014, Vol. 28, No. 20, pp. 1389-1402, Taylor & Francis, doi:10.1080/01691864.2014.939105.
- [70] D. Simon, C. Isik, “Optimal trigonometric robot joint trajectories”, *Robotica* (1991) volume 9, pp. 379-86, doi:10.1017/S0263574700000552.
- [71] B. Cao, G. I. Dodds, and G. W. Irwin, “Constrained time-efficient and smooth cubic spline trajectory generation for industrial robots”, *IEE Proc.- Control Theory Appl.*, Vol 144, No. 5, Sept. 1997.
- [72] C. S. Lin, P. R. Chang, and J. Y. S. Luh, “Formulation and optimization of cubic polynomial joint trajectories for industrial robots”, *IEEE Transactions on Automatic Control*, Vol. 28, No. 12, Dec 1983, pp. 1066-74, doi:0018-9286/83/1200-1066.
- [73] A. Gasparetto and V. Zanotto, “A new method for smooth trajectory planning of robot manipulators”, *Mechanism and Machine Theory* 42 (2007) 455–471, doi:10.1016/j.mechmachtheory.2006.04.002, Elsevier.
- [74] M. Boryga and A. Grabos, “Planning of manipulator motion trajectory with higher-degree polynomial use”, *Mechanism and Machine Theory* 44 (2009), pp. 1400-19, doi:10.1016/j.mechmachtheory. 2008.11.003.
- [75] H. I. Lin, “A fast and unified method to find a minimum-jerk robot joint trajectory using particle swarm optimization”, Springer Science, *J Intell Robot Syst.*, Oct-2013, doi: 10.1007/s10846-013-9982-8.
- [76] J. Huang, P. Hu, Kaiyuan, and M. Zeng, “Optimal time-jerk trajectory planning for industrial robots”, *Mechanism and Machine Theory* 121(2018), pp. 530-44, doi:10.1016/j.mechmachtheory.2017.11.006.
-

- [77] S. A. Alshahrani, H. Diken and A. A. N. Aljawi, "Optimum trajectory function for minimum energy requirements of a spherical robot", 6th Saudi Engineering Conference, KFUPM, Dhahran, Dec. 2002, Vol 4., pp. 613-25
- [78] G. J. Yang and B. W. Choi, "Smooth trajectory planning along Bezier curve for mobile robots with velocity constraints", International Journal of Control and Automation, Vol. 6, No. 2, April, 2013, pp. 225-34.
- [79] J. H. Hwang, R. C. Arkin, D. S. Kwon, "Mobile robots at your fingertip-Bezier curve on-line trajectory generation for supervisory control", Proc. Of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robot and Systems, Nevada, Oct. 2003, pp. 1444-49, doi:0-7803-7860-1/03.
- [80] Ashitava Ghosal, "Robotics: Fundamental Concepts and Analysis", Oxford.
- [81] D. F. Rogers, "An Introduction to NURBS with historical perspective", Morgan-Kaufmann Publishers, pp. 34-35.
- [82] J. Gravesen, "Adaptive subdivision and the length and energy of Bezier curves", Computational Geometry 8 (1997), pp. 13-31, doi: 0925-7721/97.
- [83] B. Guenter, "Computing the Arc Length of Parametric Curves", IEEE Computer Graphics & Applications, 1990, pp. 72-78, doi: 0272-17-16/90/0500-0072.
- [84] T.A. Pastva, "Bezier Curve Fitting", Master's Thesis, Naval Postgraduate School, Monterey, California, Sept. 1998.
- [85] R. Parent, "Computer Animation Algorithms and Techniques", Morgan-Kaufmann Publishers, pp. 69-84.