UNIVERSITY OF CALIFORNIA SAN DIEGO

Investigation of Effects of Changing Length Scales of Uniformly Structured Rough Terrain on Hexapedal Locomotion using Simulation

A thesis submitted in partial satisfaction of the
requirements for the degree
Master of Science

in

Engineering Sciences (Mechanical Engineering)

by

Rahul Shirpurkar

Committee in charge:

Professor Nicholas G Gravish, Chair
Professor John T Hwang
Professor Michael T Tolley

2019

The thesis of Rahul Shirpurkar is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

Chair

University of California San Diego

2019

TABLE OF CONTENTS

Signature Page . . . iii

Table of Contents . . . iv

List of Figures . . . vi

List of Tables . . . viii

Acknowledgements . . . ix

Abstract of the Thesis . . . x

Chapter 1 Introduction . . . 1
1.1 Background . . . 1
1.2 Literature Review . . . 3
1.3 Motivation . . . 6

Chapter 2 Dynamic Simulation of Robots . . . 8
2.1 The Need of Simulation in Robotics . . . 8
2.2 Simulation Packages . . . 9
2.3 Physics Engines . . . 10
2.3.1 Overview of Physics Engines . . . 10
2.3.2 Comparison of Available Physics Engines . . . 14

Chapter 3 Experimental Setup and Methodology . . . 16
3.1 Hexapod Robot Design Specifications . . . 16
3.2 V-REP Model . . . 19
3.3 Kinematics . . . 20
3.3.1 Forward Kinematics . . . 20
3.3.2 Inverse Kinematics . . . 23
3.4 Gait . . . 25
3.5 Simulation of Motor Control . . . 27
3.6 Robot Control Script . . . 27
3.7 Substrate Generation . . . 31
3.8 Validation and Fine Tuning . . . 32
3.9 Methodology . . . 33

Chapter 4 Results and Discussion . . . 34
4.1 Distance Travelled . . . 34
4.2 Leg Sweep Angles . . . 39
4.3 Leg-tip Trajectories . . . 40
4.4 Discussion . . . 42

Chapter 5 Conclusion and Future Work 55
5.1 Conclusion 55
5.2 Future Work 55

Bibliography 58

LIST OF FIGURES

Figure 1.1: Little-Dog, a quadruped bio-inspired robot produced by Boston Dynamics Incorporated . . . 2
Figure 1.2: Examples of hexapod robots with bio-inspired legs . . . 3
Figure 1.3: Classification of robot based on leg morphology . . . 4
Figure 1.4: Examples of hexapods with non-zoomorphic legs . . . 4
Figure 1.5: Leg configurations in hexapods . . . 5
Figure 1.6: Different approaches for locomotion over rough terrain . . . 6

Figure 2.1: Screenshot of Industrial path planning software, Tecnomatix Process Simulate Robotics, distributed by Siemens . . . 9
Figure 2.2: Steps in a typical simulation loop . . . 11

Figure 3.1: Top View of HEXY with leg numbering . . . 17
Figure 3.2: HEXY Leg Assembly . . . 17
Figure 3.3: Example of conversion of mesh to combination of pure shapes (translucent purple cuboids) . . . 19
Figure 3.4: (a) Model of HEXY in V-REP (b) HEXY comprised of simple shapes for dynamic calculations. The orange cylinders are the dynamically enabled joints . . . 20
Figure 3.5: Denavit-Hartenberg parameters of leg . . . 20
Figure 3.6: X-Y view of the leg workspace . . . 22
Figure 3.7: X-Z view of leg workspace . . . 23
Figure 3.8: Schematic diagram of leg for inverse kinematics . . . 24
Figure 3.9: Schematic of alternating tripod gait . . . 25
Figure 3.10: Stride trajectories developed: a) Default b) Semi-Backward Swing c) Backward Swing d) Pseudo-sinusoidal . . . 26
Figure 3.11: Cycle timing of gait a) For default and backward swing trajectories b) For pseudo-sinusoidal trajectory . . . 26
Figure 3.12: Means of communication with V-REP . . . 28
Figure 3.13: schematic of synchronous mode operation . . . 29
Figure 3.14: Snippet of robot control script . . . 30
Figure 3.15: Sample heightfield surface with mesh visible . . . 31
Figure 3.16: Terrain samples compared to HEXY . . . 32
Figure 3.17: Ideal versus non-ideal handling of friction. Due to incorrect handling of friction, the robot tends to drift and travels less than the ideal case . . . 33

Figure 4.1: Example of data extracted from recorded positions. . . . 35
Figure 4.2: Distance travelled as a function of the substrate grid size . . . 36
Figure 4.3: Comparison of distance travelled in multiples of stride lengths with the substrate grid size. . . . 38

Figure 4.4: Comparison of the distance travelled on the 1 inch grid substrate using different leg sweep angles 39
Figure 4.5: Comparison of the distance travelled on the 1 inch grid substrate using different leg-tip trajectories 41
Figure 4.6: Comparison of distance travelled in half the time (7.5s) in multiples of stride lengths with the substrate grid size. 43
Figure 4.7: Arcs traced by leg-tips result in unequal distances swept in the sagittal and mediolateral directions 44
Figure 4.8: If the leg-tips land on the elevated portion of the surface and are unable to maintain a good grip, instead of pulling the body forward, they slip and fall back into the previous cavity 45
Figure 4.9: Probability of leg-tips taking-off and landing at different elevations 46
Figure 4.10: Model of hexapod being rotated about a fixed point to evaluate probability of changing foot-tip evaluation 47
Figure 4.11: Probability of leg-tips taking-off and landing in cavities 48
Figure 4.12: Probability of leg-tips taking-off and landing on top of surface structures . 48
Figure 4.13: Ideal leg angles (on flat ground) 49
Figure 4.14: The mid-limbs forced to very large angles on the 0.5 inch grid sized substrate 50
Figure 4.15: The number of times the legs are forced to large angles reduces on the 1 inch grid substrate 51
Figure 4.16: As the substrate grid size increases, the average returns to zero (ideal value) with fewer occurrences of large leg angles 52
Figure 4.17: Leg angles approach ideal values on the 10 inch grid substrate 53

Figure 5.1: Leg 6 (Front-Left) being forced to a large angle causing leg-body collisions 56

LIST OF TABLES

Table 3.1: Servo Specifications 18
Table 3.2: HEXY Dimensional Specifications 18
Table 3.3: Denavit-Hartenberg Table 21
Table 3.4: Leg motion parameters 31

ACKNOWLEDGEMENTS

I would like to thank Prof. Nicholas Gravish for giving me an opportunity to be a part of the Gravish Lab at UCSD. I am especially grateful to him for all his insightful guidance and motivation provided during this project.

My special thanks to Prof. John Hwang and Prof. Michael Tolley for serving on my dissertation committee and for the guidance they provided on improving my thesis.

I would also like to thank all the Gravish Lab members. Their questions and suggestions during lab meetings helped me immensely during the writing of this thesis.

ABSTRACT OF THE THESIS

Investigation of Effects of Changing Length Scales of Uniformly Structured Rough Terrain on Hexapedal Locomotion using Simulation

by

Rahul Shirpurkar

Master of Science in Engineering Sciences (Mechanical Engineering)

University of California San Diego, 2019

Professor Nicholas G Gravish, Chair

Despite of having higher complexity, in both morphology and control strategies, bio-inspired hexapod robots offer advantages such as higher mobility, omni-directional locomotion, ability to climb over obstacles, and greater adaptability to terrain and environment changes. While there have been numerous studies investigating the locomotion of hexapods over rough terrain utilizing exteroceptive sensing for terrain mapping and calculating optimum leg trajectories for unobstructed and fast locomotion, there have been none to study the locomotion over structured terrain, where structured terrain suggests terrain samples where surface features repeat in an alternating pattern in all directions giving rise to a checkerboard of surface structures. This study

investigated the locomotion of a hexapod implementing an open-loop alternating tripod gait using a rigid body simulation package – V-REP. Different parameters of the robot locomotion were recorded while it navigated the generated terrain samples with grid sizes ranging from 0.16 times the stride length to 3.33 times the stride length. The results indicate that when the stride length was more than the substrate feature size, the nature of leg-surface interactions governed the speed, while the rate of the leg-surface interactions influenced the locomotion when the stride length was smaller than the surface feature size.

Chapter 1

Introduction

1.1 Background

Terrestrial or land based mobile robots have to rely on three major strategies for locomotion- wheels, tracks and bio-inspired legs. While wheeled and tracked robots are capable of carrying lager payloads and efficiently navigating largely flat terrains or terrains with obstacle sizes limited to 50% of their wheel diameters, legged robots, although limited to lighter payloads, are capable of negotiating obstacles of sizes comparable to their leg lengths [1] . Further, it is estimated that more than 50% of the Earth's surface cannot be navigated using wheels or tracks alone [2] .

In general, legged robots consist of one or more appendages connected to a central frame or body. The appendages consist of multiple sections connected to one another using either revolute, prismatic or spherical joints giving the joint 1 to 3 degrees of freedom. The morphology of these legs is often based on structures occurring in biology, such as in reptilian and mammalian legs [3].

Due to their inherent static stability, most legged robots are designed to be quadrupeds (4 legs), hexapods (6 legs) or multi-pods (more than 6 legs). The minimum number of legs required to form a support triangle to maintain static balance is 3. As the number of legs increases, the

Figure 1.1: Little-Dog, a quadruped bio-inspired robot produced by Boston Dynamics Incorporated. [4]

static balance and stability can be improved by increasing the number of legs maintaining contact with the substrate. Due to the ease in maintaining static stability with the increase in the number of legs, the average speed of locomotion in hexapods has been observed to be 3 times as much as quadrupeds [5]. Additional legs can also be employed to maintain stable locomotion in case some are rendered inoperable or get damaged during operation. However, the optimum number of legs has been established to be 6 as beyond this number, the complexity and hardware costs outweigh the aforementioned advantages [6].

Despite of having higher complexity, in both morphology and control strategies, hexapod robots offer advantages such as higher mobility, omni-directional locomotion, ability to climb over obstacles, and greater adaptability to terrain and environment changes, all qualities vital for successful execution of exploration and SAR (Search and Rescue) operations. For these reasons, investigation of hexapod locomotion over unstructured terrain has been an area of great interest

(a) (b)

Figure 1.2: Examples of hexapod robots with bio-inspired legs: a) LAURON V, developed by FZI (Research Centre for Information Technology), Karlsruhe Germany [7] and b) RHex, developed by Boston Dynamics Incorporated [8]

to researchers in recent years [9].

1.2 Literature Review

For the literature review, the focus was on the study of hexapod locomotion over unstructured terrains and the control strategies employed to overcome the various challenges encountered during the navigation of such terrain. The literature review has been divided into 3 main sections- variation in hexapod design, experiments on unstructured terrain, and the generation of control algorithms and gait patterns for the successful navigation of such challenging terrains.

Different leg designs can arise from the requirements and constraints of the task at hand, for example, space and energy constraints or the payload carrying capacity. Broadly, the legs can be designed to mimic the morphology of animal legs and utilize animal-like gaits or be designed specifically to address terrain or application requirements. In case of bio-inspired hexapods, insects have been the source of inspiration because of their agility and capability of negotiating obstacles up to twice their size without detrimental effects on their speed [11]. While the mammalian configuration requires the lowest energy for maintaining upright orientation, it is not well suited for uneven terrain because of the challenges in maintaining stability caused

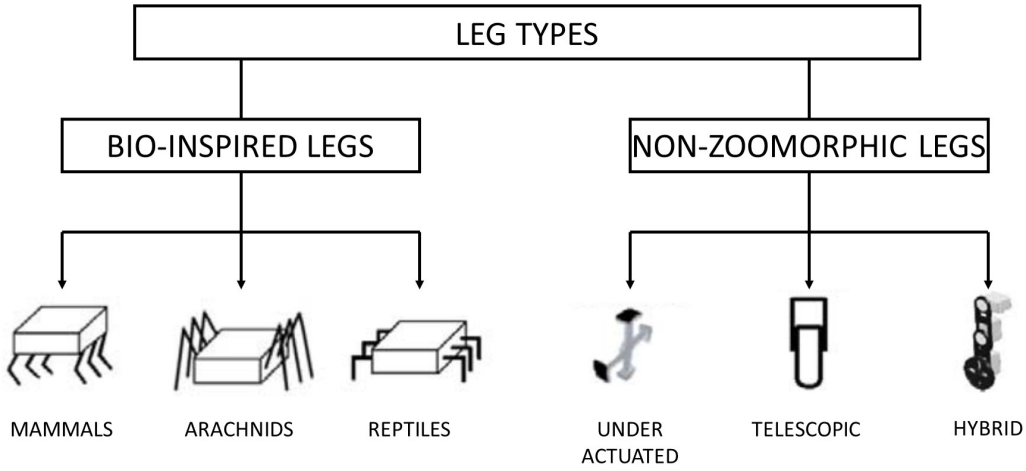

Figure 1.3: Classification of robot based on leg morphology [10]

by its higher center of gravity as compared to the arachnid and the reptilian configuration [10]. In case of non-zoomorphic legs, the legs can be under-actuated with only 1 degree of freedom in each leg and a simple controlling algorithm as in the case of Boston Dynamic's RHex, or be designed as parallel linkages in the case of Shanghai Jiao Tong University's more sophisticated Octopus 3 [12]. Hybrid legs have the advantage of being able to switch between the use of legs

(a) (b)

Figure 1.4: Examples of hexapods with non-zoomorphic legs: a) Octopus 3 developed by Shanghai Jiao Tong University [12] and b) NASA JPL's ATHLETE Rover [13]

for stepping over obstacles and wheels for efficient locomotion on flat terrain. The orientation of

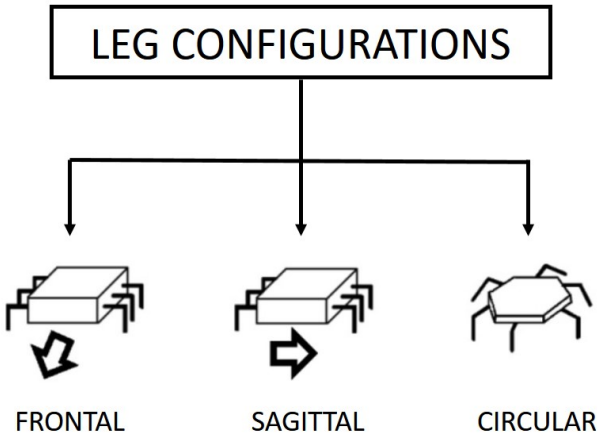

Figure 1.5: Leg configurations in hexapods [10]

the legs with respect to the body can also influence the direction of locomotion possible. The legs can be arranged in parallel on the sides of a rectangular body to give rise to faster locomotion in a preferred direction as shown in Figure 1.5 or be arranged in a circle around the body to achieve fast omni-directional locomotion.

In hexapods with 18 degrees of freedom (3 in each leg) or higher, alternating tripod gait (3 legs in stance) has been determined to be faster and more energy efficient than the amble (4 legs in stance) and wave (5 legs in stance) gait for traversing over hard unstructured terrain [14] while the wave gait is the fastest on smooth and slippery ground [15].

Other studies make use of sensors to gather and process exteroceptive or proprioceptive stimuli to deliberately step over obstacles and get strategic footholds. Eternal or exteroceptive stimuli is gathered from the environment by making use of stereo cameras, depth sensors and multi-directional strain gauges to build a picture of the environment and obstacles ahead [16, 17]. On the other hand, proprioceptive or internal stimuli in the form of current draw from servo motors is used to estimate the force at the leg tips and make appropriate foot placement decisions. [15, 18, 19]

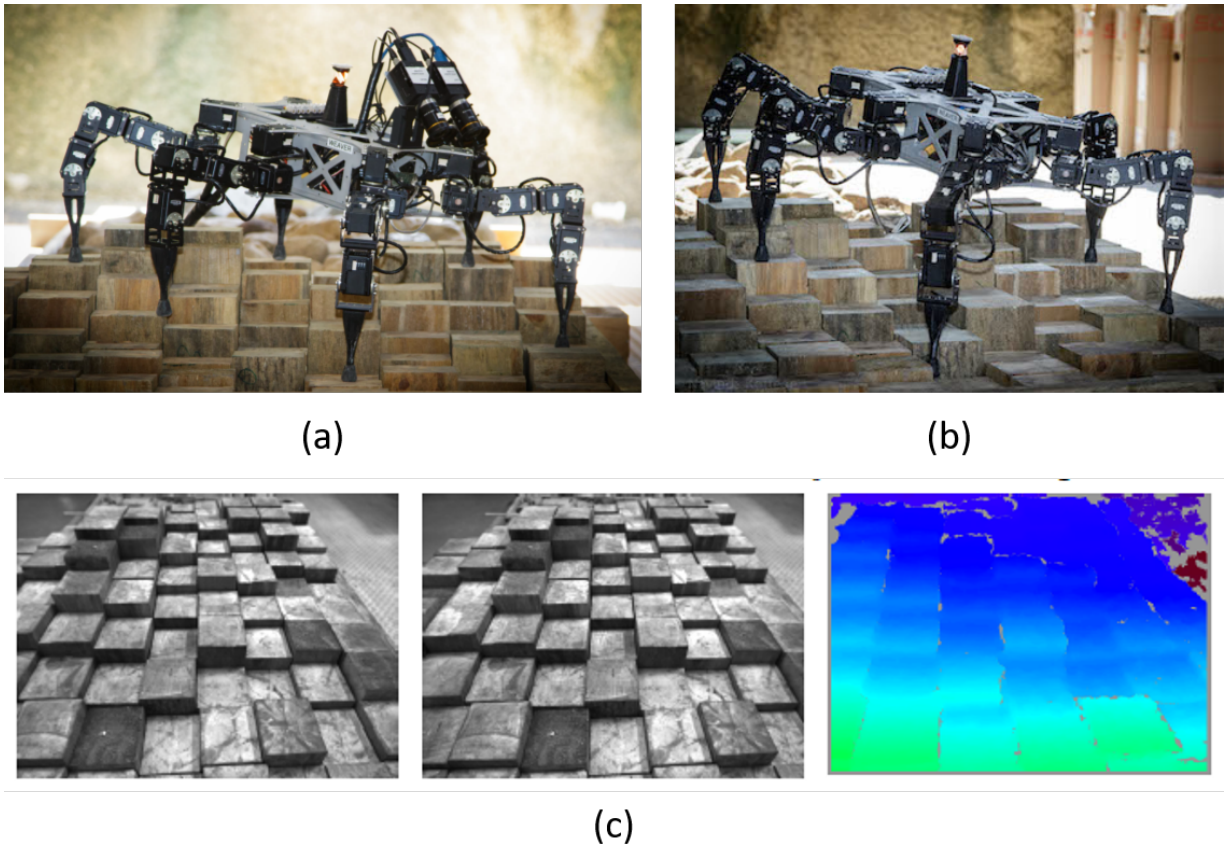

Figure 1.6: Different approaches for locomotion over rough terrain: a) Weaver using stereo camera system to sense the terrain [19], b) Weaver utilizing proprioceptive sensing to navigate rough terrain [19], c) Stereo camera outputs and disparity map [16]

1.3 Motivation

Although there have been many studies of hexapod locomotion over unstructured or rough terrain, both with and without utilizing sensors to gauge their environment, there have not been any pertaining to the study of locomotion over structured terrain. Here, structured terrain denotes a terrain sample characterized by surface features which repeat periodically in all directions, giving rise to a checkerboard pattern of alternating surface features. The checkerboard pattern is chosen over a completely random terrain because it represents a logical intermediate between completely random rough terrain and simple one step disturbance over otherwise smooth terrain.

The implementation of such a terrain helps gain understanding of how leg interactions with periodically distributed surface features affects different aspects of locomotion.

Through literature review, it has been established that alternating tripod gait results in energy efficient and fast locomotion over unstructured terrain. The aim of this study is to investigate the effects of changing length scales of the structured terrain on the characteristics of sensor-less, open-loop locomotion of a hexapod robot with 3-DOF legs utilizing alternating tripod gait and to investigate how foot lift-off and stride lengths affects locomotive performance. In order to obtain the largest possible data set in a relatively short period of time and to automate the data acquisition, the experimentation is carried out in a simulated environment created using a commercially available simulation package.

The thesis is structured as follows:

Chapter 2 gives a glimpse of the importance of simulation in robotics and details the steps involved in simulation of rigid body dynamics along with the strengths and weaknesses of each approach. It also describes the physics engine selection procedure.

Chapter 3 discusses the design considerations that went into modelling the experiment in the simulation environment and the experimental setup in detail.

Chapter 4 details the results of the experimentation and attempts to discuss possible reasons for the outcomes and their implications.

Chapter 5 summarizes the work and the results, and suggests possible refinements that may be done in the future.

Chapter 2

Dynamic Simulation of Robots

2.1 The Need of Simulation in Robotics

Simulating the kinematics and dynamics of robots is becoming increasingly popular in the field of robotics, especially as the robots increase in complexity and diversity. Simulation enables developers and researchers to predict, test and validate the performance of their robots before deployment and allows for faster and easier iterative improvement of their control strategies in a safe and cost effective way. The advancements in computation and simulation techniques have facilitated researchers to develop and train genetic algorithms requiring large number of iterations by simulating faster than real time. This increase in the simulating efficiency and speed also allows for the implementation of state estimation based predictive control models on robots, which has proved to be of critical importance for swarm based autonomous robots.

Simulation is gaining popularity in the field of industrial robotics for the purpose of planning collision free tool paths and optimization of operation paths using inverse kinematics. Simulation is especially advantageous in this field as it allows for complete offline code changes and training on new tasks thereby avoiding any interruption to ongoing operations and maintaining production rates.

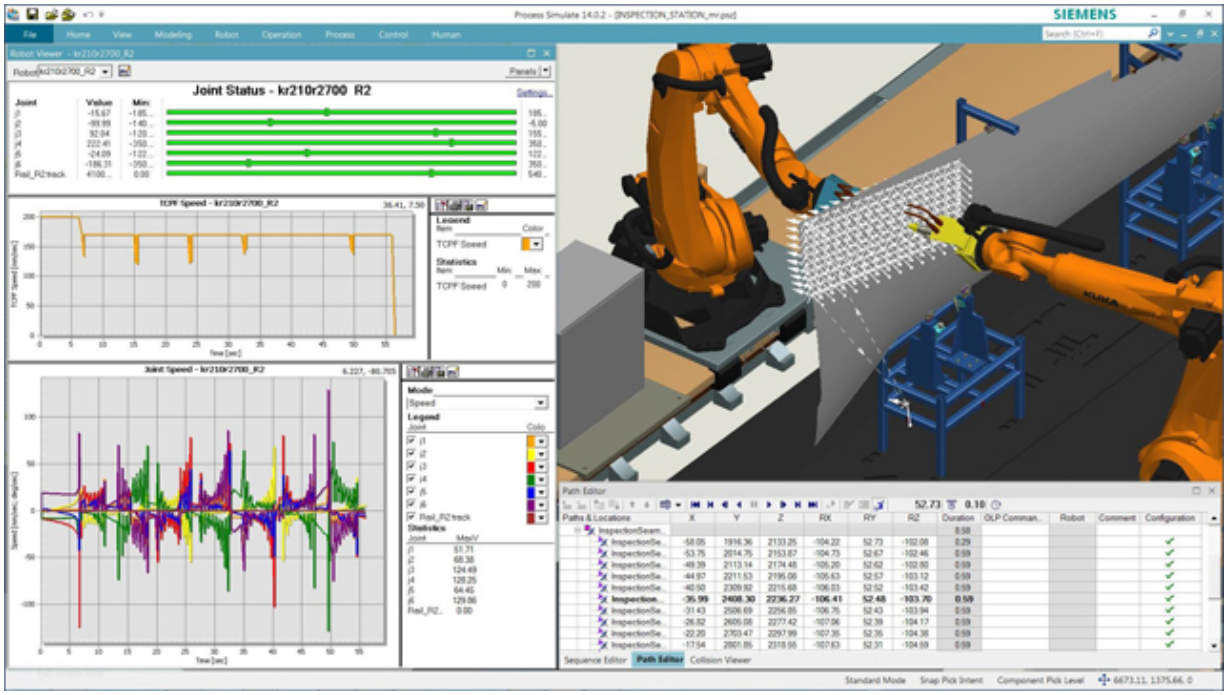

Figure 2.1: Screenshot of Industrial path planning software, Tecnomatix Process Simulate Robotics, distributed by Siemens [20]

2.2 Simulation Packages

Early work in simulation of robotic systems focused on the solving rigid body dynamics equations for robots with multiple joints without modelling contact and collision dynamics [21]. Parallel work in the animation industry on the simulation of realistic physics on rigid bodies led to the development of more sophisticated multi-body dynamics solvers that which were capable of real time rendering collisions and contact constraints. These solvers were also instrumental in the development of video game engines required for rendering realistic and believable real time interactions of on-screen elements. As these physics engines improved, it was realized that these simulators offer sufficient fidelity in modeling the real world dynamics of rigid body approximations of robots at the cost of computation costs and that it was no longer necessary to pursue the intractable goal of highly accurate simulation. Modern day commercial robotics

packages rely on these physics engines for delivering flexible and fast simulations at the required level of accuracy. Physics engines will be discussed in detail in the subsequent section.

Modern simulation packages allow users to easily model and program robots, their joints, sensors and environments fairly accurately and easily. The most popular open-source packages are Gazebo and ROS, while the most popular commercial packages are Webots, MATLAB with Simulink and its Robotics Toolbox and V-REP. The two most readily available packages, Gazebo and V-REP, both allow users to select from multiple built in physics engines to tailor the simulation as per their required level of accuracy and have additional features such as simulation of proximity and vision sensors using built-in image processors, custom user defined interfaces, and embedable scripts [22]. Between these packages, V-REP was found to be the most intuitive, user friendly, and the least CPU intensive [23] and was chosen for this investigation.

2.3 Physics Engines

2.3.1 Overview of Physics Engines

Physics engines are designed to solve the forward dynamics equations, that is, to determine the motion of a system, given the forces acting on it at that instant. In order to solve this problem, the simulation loop must locate the current position of the bodies, and solve a system of differential equations for the updated velocities given the constraints and then update the positions. A rigid body in a simulation has 6 degrees of freedom, 3 for translation and 3 for rotation along the three cardinal axes. The bodies are connected with joints, which are evaluated as constraints to motion. Finally, the system can be influenced by a number of forces. An example of external force can be the reaction force at the point of contact, or the force of gravity acting on the body. A typical simulation loop is shown in Figure 2.2.

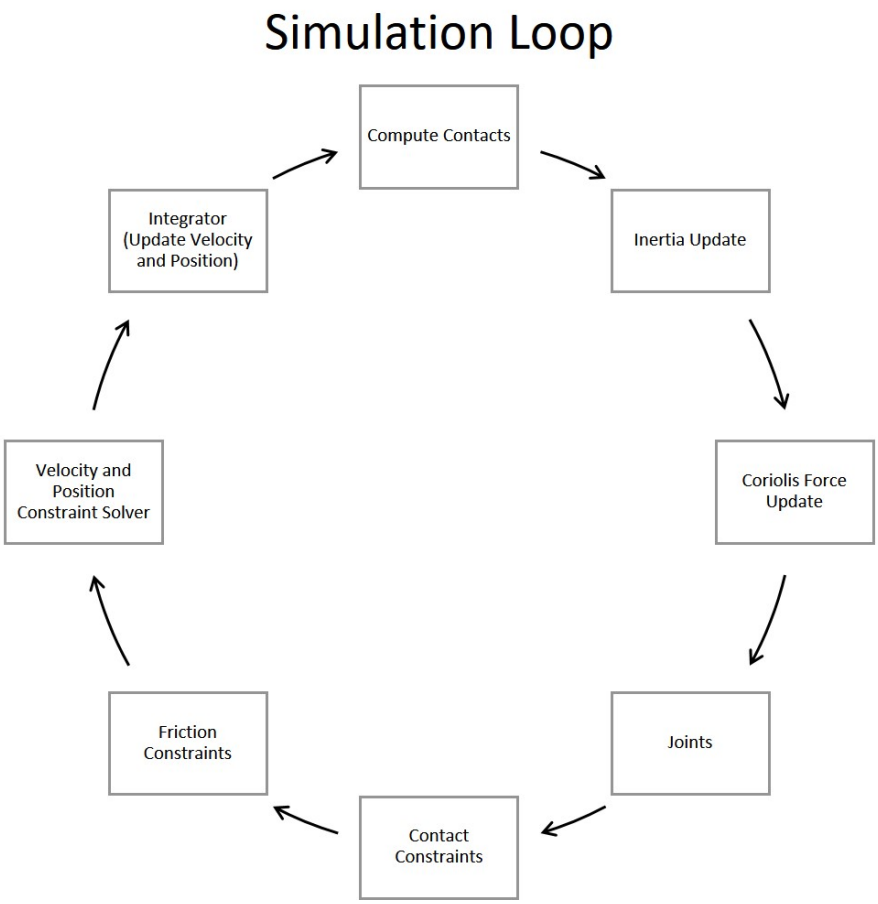

Figure 2.2: Steps in a typical simulation loop [24]

The equations solved to determine the motion of the bodies are:

Newton's Second Law:

$$F = M \cdot a \tag{2.1}$$

$$\tau = I \cdot \alpha + \omega \times I \cdot \omega \tag{2.2}$$

Equations of motion:

$$v_{t+\Delta t} = v_t + \alpha \Delta t = v_t + \frac{F_{ext}}{m} \Delta t \tag{2.3}$$

$$x_{t+\Delta t} = x_t + v_{t+\Delta t} \Delta t \tag{2.4}$$

Going through the steps carried out in a simulation loop in order-

1. Computing Contacts- To detect if particle are in contact, the distance between their centers is compared to the sum of their radii. If the distance is greater than the sum, the bodies are not in contact. If the distance is less or equal, then there is contact and a contact normal is assigned at the point of contact.

2. Inertia Update- The Inertia tensor of each of the body links is recalculated by multiplying the local inertia tensor with the updated global coordinate transform.

3. Coriolis Force Update- Coriolis force is a pseudo-force that is added to a rotating system for Newton's Laws to hold in that system.

4. Formulate the Constraints- Joint and contact constraints restrict the motion of the bodies in particular directions. The contact constraint prevents the bodies from intersecting at their boundaries. There are two ways of modelling contact – force methods and impulse methods. Penalty methods and analytic methods are examples of force methods. Penalty methods, the older approach, imagine a temporary spring between the objects in contact. The spring resists the penetration of the object boundaries and also pushes them back in case of inadvertent penetration. The spring gets destroyed once the objects start moving away from one another. The issue with this method is that assignment of spring constants is purely empirical and the spring force introduce additional force to the system. The analytical method models the non-penetration constraints as a linear complementarity problem (LCP) whose objective optimize the contact reaction forces and the relative normal acceleration at the point such that the acceleration and the reaction forces are positive (objects do not move into each other and the force can only move them apart) given the configuration of the system. Impulse methods do not have to model and meet explicit constraints but instead, model all forms of contact as a series of impulses in different directions. Modern physics engines use the sequential impulse methods because of their simplicity and robustness [25]. The friction is dependent on the coefficient of friction between the surface in contact and the normal force between the bodies in contact and the resultant of

all forces acting on the body must lie within the cone of friction when the bodies are sliding past one another. The constraints on friction are modeled as velocity constraints where the target velocity is zero. This condition is easy solve as an additional LCP constraint in 2D, but becomes conceptually and computationally arduous in 3D. This issue can be ameliorated to a certain extent by an approximation that allows for the constraint to be limited between two constants proportional to the object's weight [26].

5. Constraint Solver- The constraints solver solves the optimization problem in order to ensure that all constraints are satisfied. The sequential impulse method does not require the solving of a system of equations to satisfy the constraints, but rather solves for the impulses sequentially and updates the system with every time-step. Impulse method constraint modelling can also be solved as mixed linear complementarity problem (MLCP) as follows:

$$A\lambda + b \geq 0 \tag{2.5}$$

$$\lambda \geq 0 \tag{2.6}$$

$$\lambda(A\lambda + b) = 0 \tag{2.7}$$

Where A is the effective mass, defined as the change in velocity in the direction of the applied impulse per unit impulse.

$$A = \frac{\Delta u_{\text{in the direction of impulse}}}{|Impulse|} \tag{2.8}$$

There are several methods of evaluating the A matrix without the value of the Impulse known beforehand [24]. λ is the value of the impulse that is to be found, and b is the change in velocity. The first constraint limit the impulse to be greater than zero, so that the the objects bounce away, and the second constraint dictates that at a given instant, either the relative velocity is zero or the impulse is zero.

A popular iterative matrix method called the Projected Gauss-Seidel (PGS) method. The Projected Gauss-Seidel is similar to the regular Gauss-Seidel method which solves

$$Ax = b \tag{2.9}$$

for x, but, it has an additional part that limits the value of x to emulate the constraints.

6. Integrator- The position and velocity update is done by using semi-implicit Euler integrator. The integrator basically solves for the new values of velocity and position using the following equations:

$$v_t + \frac{F_{ext}}{m}\Delta t + \left[\frac{Impulse_{constraint}}{m}\right] \tag{2.10}$$

and

$$x_{t+\Delta t} = x_t + v_{t+\Delta t}\Delta t \tag{2.11}$$

The semi-implicit Euler method, like the standard explicit Euler method is a first order method, but has better stability (the ability of the method to converge to a solution given the conditions). This is because it uses $v_{t+\Delta t}$ instead of v_t to solve for $x_{t+\Delta t}$

2.3.2 Comparison of Available Physics Engines

As most physics engines prioritize different aspects of the simulation to optimize either speed or accuracy depending on the intended purpose of the engine, most simulation packages allow selection of physics engines to suit user preferences. The most common of these physics engines are PhysX, Newton, Bullet, and Open Dynamics Engine (ODE). In order to optimize one

area of the simulation over others(speed versus accuracy), the engines use different approximations for constraint and collision detection, and employ slight variation in the solver implementation leading to different levels of numerical accuracy and efficiency [27].

V-REP supports Bullet v2.78 and v2.83, ODE, Newton and Vortex Studio. Vortex Studio is a closed source physics engine that is used for industry level accurate simulations. Since it requires a separate license, and prioritizes reality and accuracy over speed, it was not used for this study. Comparing the integrator accuracy (cumulative error in position) between the available engines, Bullet produces the least error while Newton performed the worst. Bullet also modelled collision better than the others, while it performed worse than Newton in correctly handling friction. Another area where Newton excelled was constraint solving. It was able to accurately model a large number of constraints while keeping solver time comparable to Bullet. ODE on the other hand had the least error but consistently required 3 times the time required by the other engines [21, 27, 28]. In conclusion, while ODE gives very accurate results, it requires the most time to solve, disqualifying it from consideration for this study. Bullet engine was selected for this study because it provides an optimal trade-off between sufficient accuracy in modelling collisions and friction and time required.

Chapter 3

Experimental Setup and Methodology

3.1 Hexapod Robot Design Specifications

The hexapod model chosen for experimentation was based on HEXY, produced by Arcbotics, which was already being used on a parallel study. HEXY was chosen for its ease of repair, open source Arduino based programming and low cost. The hexapod design consists of hexagonal body with legs attached at each vertex. The legs themselves have 3 degrees of freedom each, bringing the total degrees of freedom for the robot to 18. The legs are actuated at the thoracic-coxal, coxal-femoral, and femoral-tibial joints using 9g servo motors whose specifications are listed in Table 3.1.

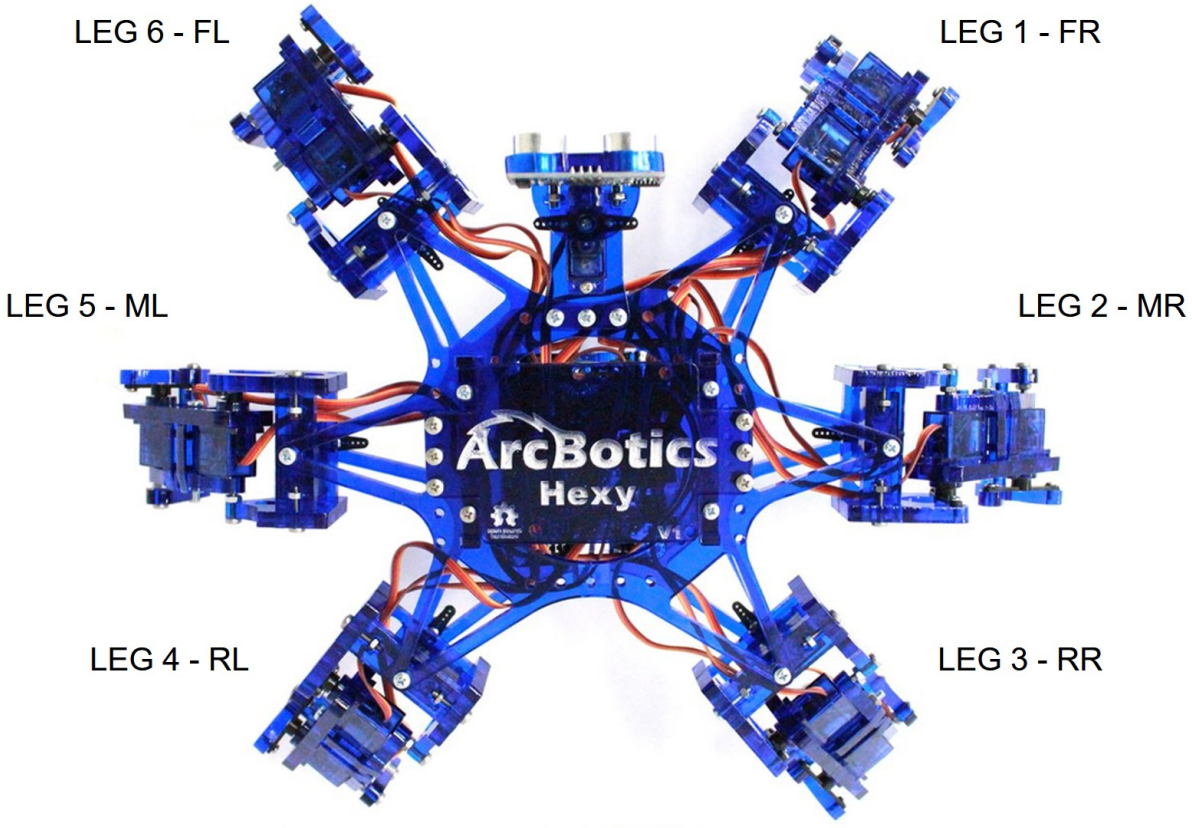

Figure 3.1: Top View of HEXY with leg numbering [29]

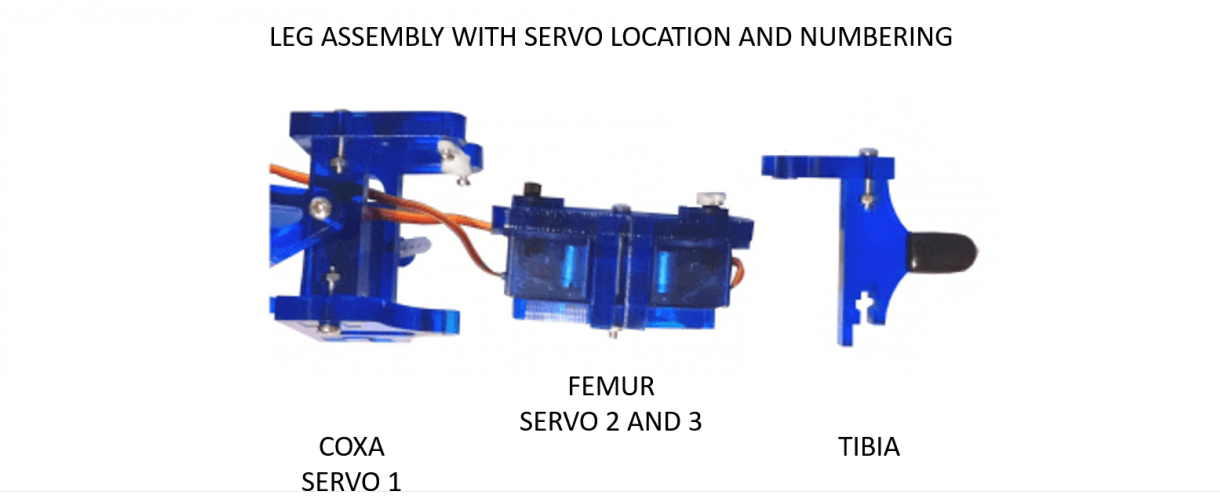

Figure 3.2: HEXY Leg Assembly [29]

Table 3.1: Servo Specifications

Model	TowerPro SG90
Torque at 4.8V [kg-cm]	1.8
Speed at 4.8V [sec/60°]	0.12
Weight [g]	9
Dimensions [mm]	23x12.2x29

The dimensional specifications of the robot are summarized in Table 3.2.

Table 3.2: HEXY Dimensional Specifications

Body Size [mm]	200
Body Weight (without batteries) [g]	675.8
Coxa Length [mm]	26
Coxa Weight [g]	31.7
Femur Length [mm]	49
Femur Weight [g]	38.3
Tibia Length [mm]	52
Tibia Weight [g]	14.8
Leg Span (when standing) [mm]	304.8
Total Weight (without batteries) [g]	1184.6
Total Weight (with batteries) [g]	1300.2

3.2 V-REP Model

The CAD model of the robot can be imported directly into V-REP as a triangular meshed object. Importing the robot directly is quick and easy, but the higher resolution (larger number of triangular elements required to capture minute details) introduces extraneous elements that have to be tracked and operated upon during simulation causing immense slowdowns during various stages of the simulation loop. To ameliorate this issue, the triangle count is reduced by eliminating unnecessary details from the mesh. A way of doing this is by first removing all unnecessary details like screws and holes, then reducing the total number of triangles automatically using a built in function or replacing the shape with a convex approximation of that object. Another approach, which is preferred over the others, is to replace the shape by a combination of pure shapes (cube, sphere, and cylinder) which makes the simulation stable and faster [30].

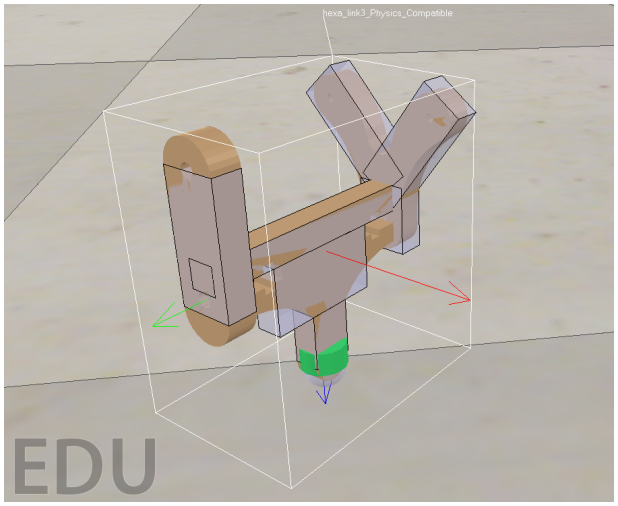

Figure 3.3: Example of conversion of mesh to combination of pure shapes (translucent purple cuboids)

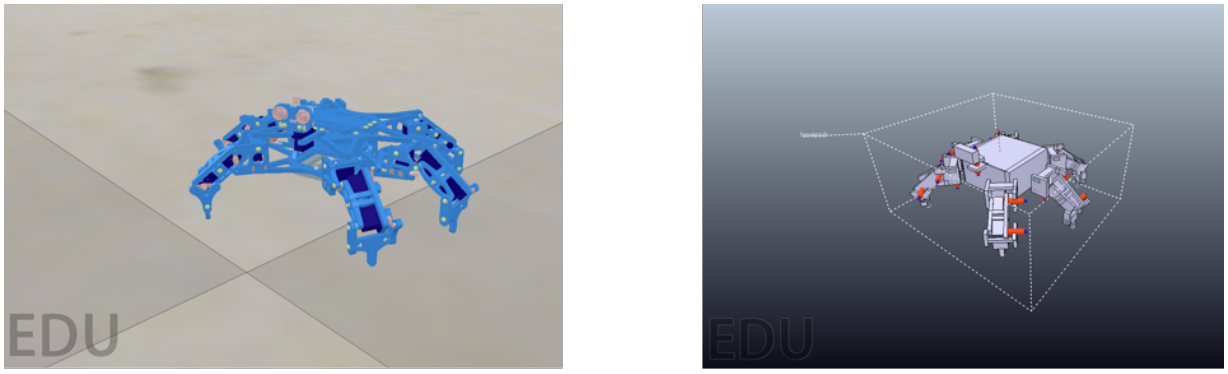

Figure 3.4: (a) Model of HEXY in V-REP (b) HEXY comprised of simple shapes for dynamic calculations. The orange cylinders are the dynamically enabled joints

3.3 Kinematics

3.3.1 Forward Kinematics

Forward kinematics is the procedure of determining end effector location in order to determine the workspace of the legs given the joint configurations. To be able to perform kinematics calculations, it is necessary to construct the Denavit–Hartenberg parameter table from the parameters.

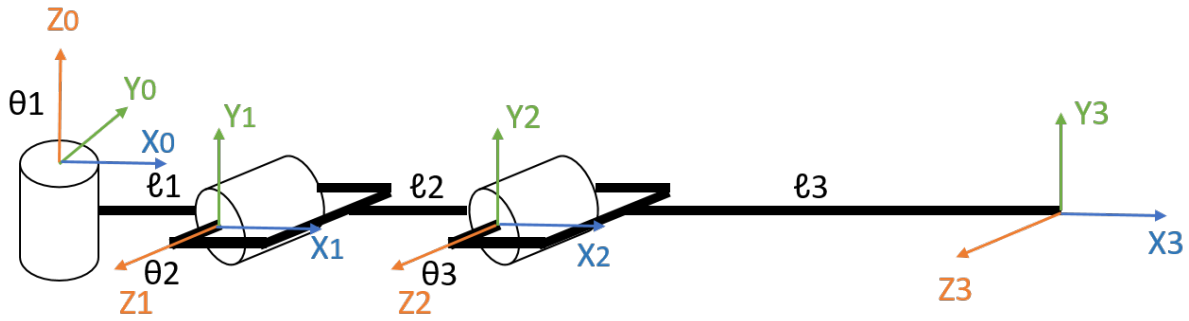

Figure 3.5: Denavit-Hartenberg parameters of leg

Table 3.3: Denavit-Hartenberg Table

Links	θ [degrees]	α [degrees]	a [mm]	d [mm]
Coxa	$\theta 1$	90	26	0
Femur	$\theta 2$	0	49	0
Tibia	$\theta 3$	0	52	0

The general transformation matrix is given by:

$$T_{i-1}^{i} = \begin{bmatrix} \cos\theta_{i-1} & -\sin\theta_{i-1}\cos\alpha_{i-1} & \sin\theta_{i-1}\sin\alpha_{i-1} & \cos\theta_{i-1} \\ \sin\theta_{i-1} & \cos\theta_{i-1}\cos\alpha_{i-1} & -\cos\theta_{i-1}\sin\alpha_{i-1} & \sin\theta_{i-1} \\ 0 & \sin\alpha_{i-1} & \cos\alpha_{i-1} & d_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

The transformation matrix for all the links are created using the table parameters. The transformation from the body (first actuator) to the end-effector (leg tip) is given by:

$$T_0^3 = T_0^1 \cdot T_1^2 \cdot T_2^3 \tag{3.2}$$

$$T_0^3 = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T1 \\ R_{21} & R_{22} & R_{23} & T2 \\ R_{31} & R_{32} & R_{33} & T3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

Where the R sub-matrix gives the rotation and the T sub-matrix gives the translation from the origin frame. Using the T sub-matrix, the workspace of the leg was found using a recursive algorithm.

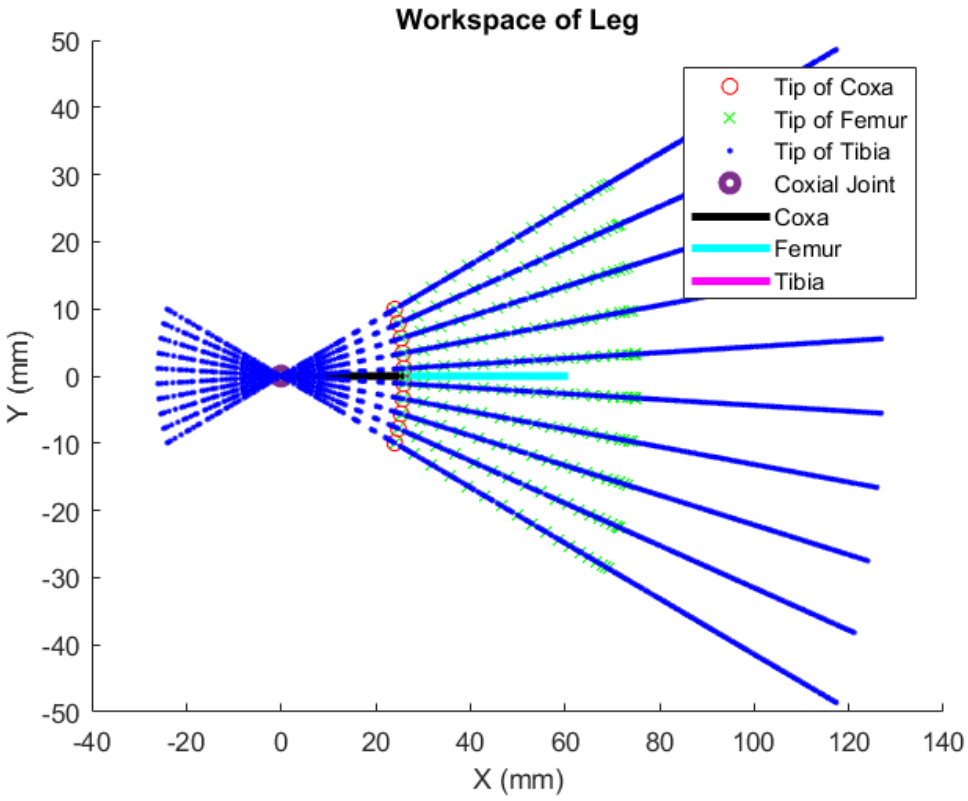

Figure 3.6: X-Y view of the leg workspace

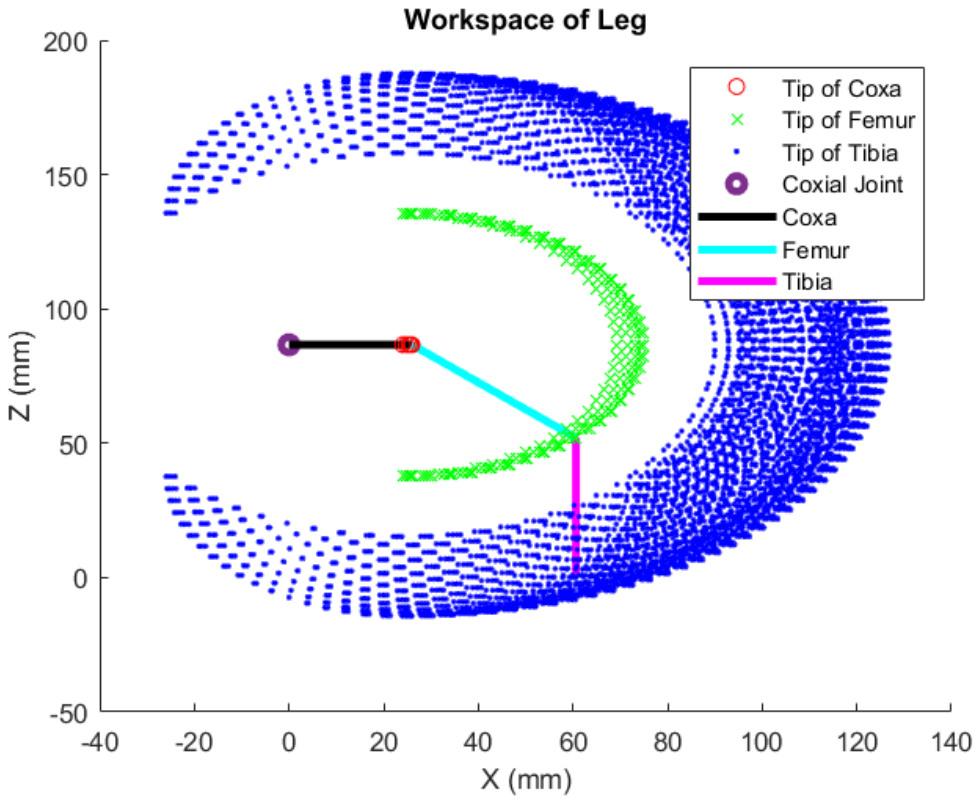

Figure 3.7: X-Z view of leg workspace

3.3.2 Inverse Kinematics

To be able to instruct the leg tip to go to a desired point in space, we need to know what angles the servo motors should go to. This task is achieved using inverse kinematics. Using geometry, we can find out the angles required given the position of the end-effector.

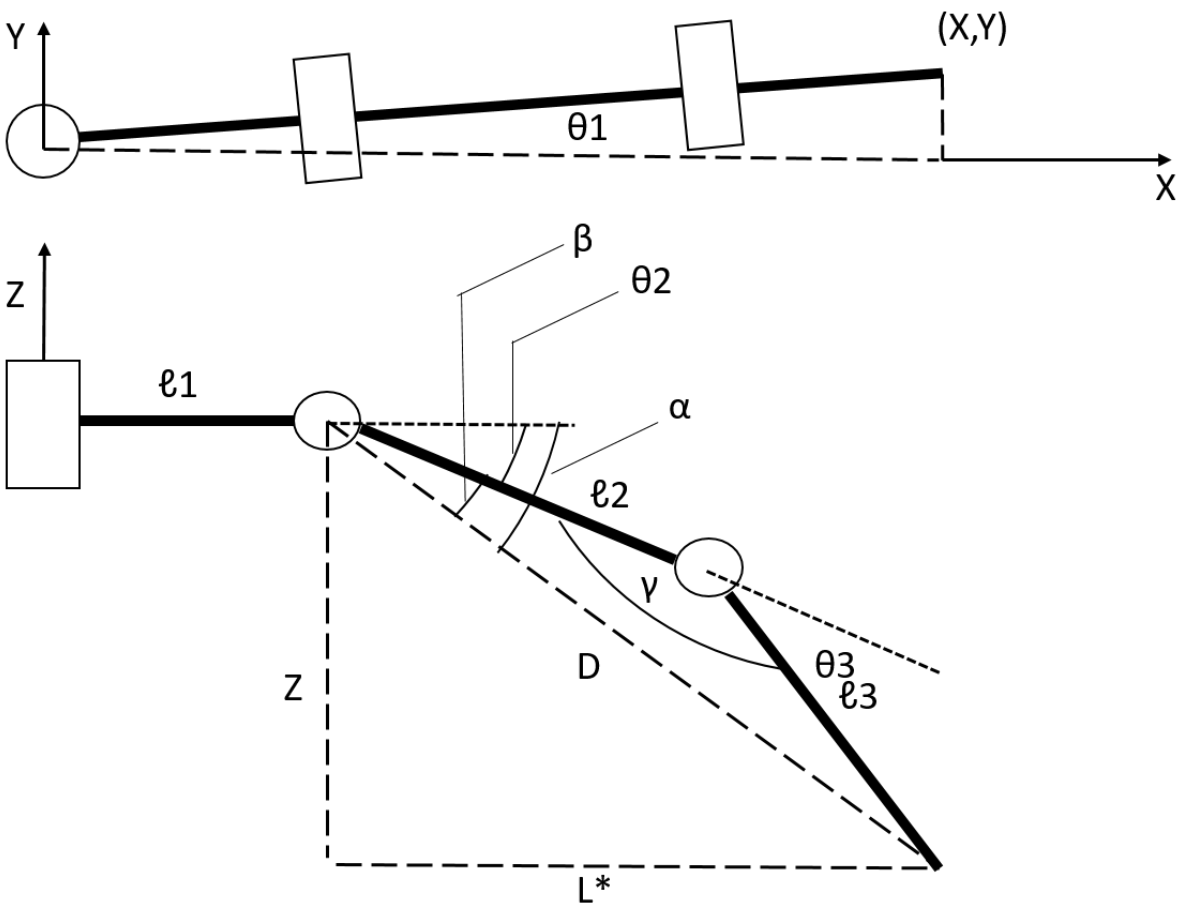

Figure 3.8: Schematic diagram of leg for inverse kinematics

$$L^* = \sqrt{X^2 + Y^2} - \ell 1 \tag{3.4}$$

$$D = \sqrt{L^{*2} + Z^2} \tag{3.5}$$

$$\theta 1 = \arctan\left(\frac{Y}{X}\right) \tag{3.6}$$

$$\beta = \arccos\left(\frac{\ell 2^2 + D^2 - \ell 3^2}{2 \cdot \ell 2 \cdot D}\right) \tag{3.7}$$

$$\gamma = \arccos\left(\frac{\ell 2^2 + \ell 3^2 - D^2}{2 \cdot \ell 2 \cdot \ell 3}\right) \tag{3.8}$$

$$\theta 2 = \alpha - \beta \tag{3.9}$$

$$\theta 3 = \pi - \gamma \tag{3.10}$$

3.4 Gait

The walking cycle or the gait chosen for this experiment was the alternating tripod gait in which 3 legs are in stance supporting the body at any given time with the center of mass of the insect inside the support triangle formed.

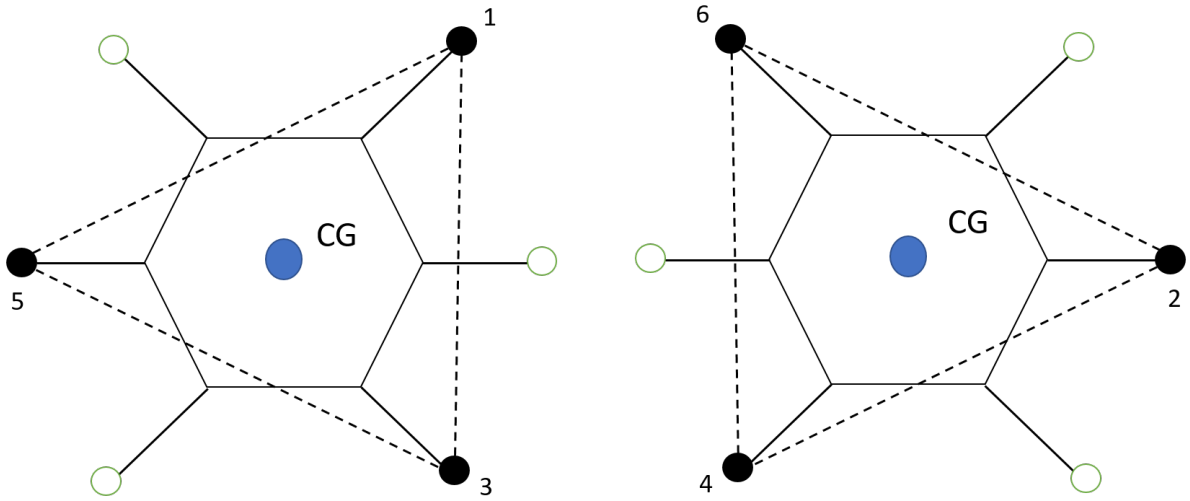

Figure 3.9: Schematic of alternating tripod gait

The leg tip trajectories used for the gaits are shown in Figure 3.10. All the trajectories have equal height and stride lengths.

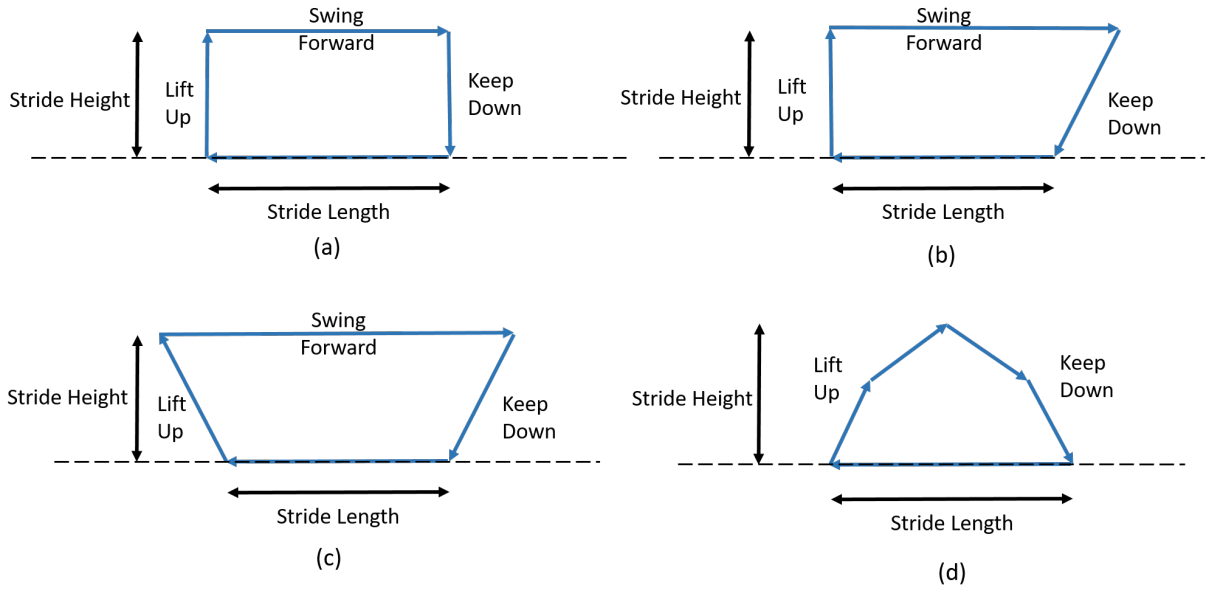

Figure 3.10: Stride trajectories developed: a) Default b) Semi-Backward Swing c) Backward Swing d) Pseudo-sinusoidal

The timings of the phases in one cycle of the tripod gait are detailed in Figure 3.11.

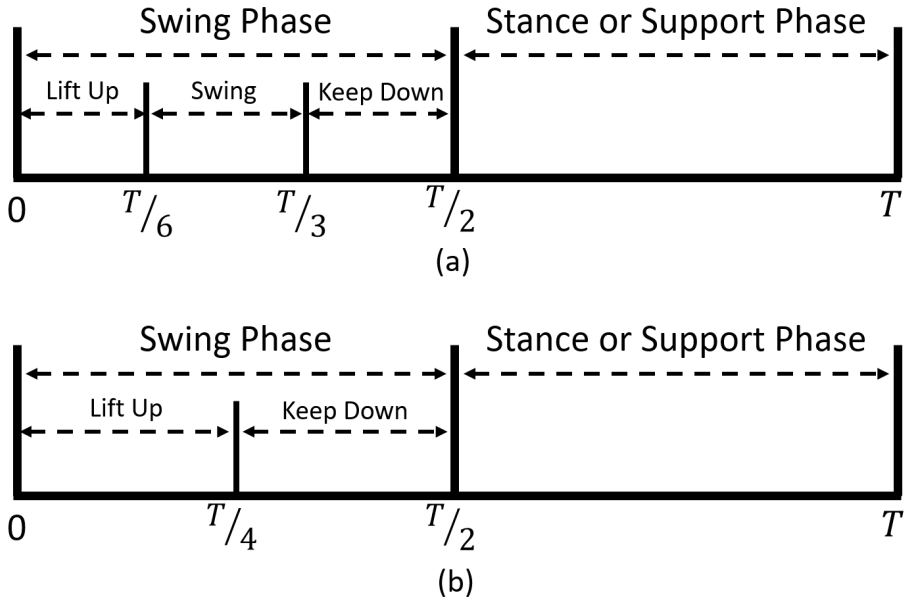

Figure 3.11: Cycle timing of gait a) For default and backward swing trajectories b) For pseudo-sinusoidal trajectory

3.5 Simulation of Motor Control

The motors are simulated as revolute joints controlled using V-REP's "Force Mode" option to enable position control. Unlike a physical motor, the velocity and torques are not coupled and are not influenced by the motor voltage and current levels, instead the built-in PID controller modulates the velocity while maintaining a peak torque [31, 32]. Once the assigned peak velocity is attained, the motor will continue to move at peak velocity until the PID control instructs it to do otherwise. This issue is resolved by assigning a very high peak velocity that cannot be attained during the duration of motor actuation. The downside of this, however, is that it is very difficult to assign specific target velocities at will during the simulation. The integral (Ki) and derivative (Kd) gains set to 0, while the proportional gain (Kp) set to 1 to imitate the PID control implemented in the servo motors.

3.6 Robot Control Script

The robot control scripts were designed to allow flexibility in parameters such as stepping patterns, stride lengths, step heights, and stepping or motor actuation frequencies. In each time-step, the appropriate legs are selected and are instructed to go to a the chosen position. The script also monitors parameters such as center of gravity location and orientation, joint positions, and ground reaction forces. These parameters are logged to an appropriately named text file for later analysis.

A governing script keeps track of the number of runs and resets the position of the robot at the end of each run. The script is capable of randomizing the starting location and orientation, allowing for large variation between runs.

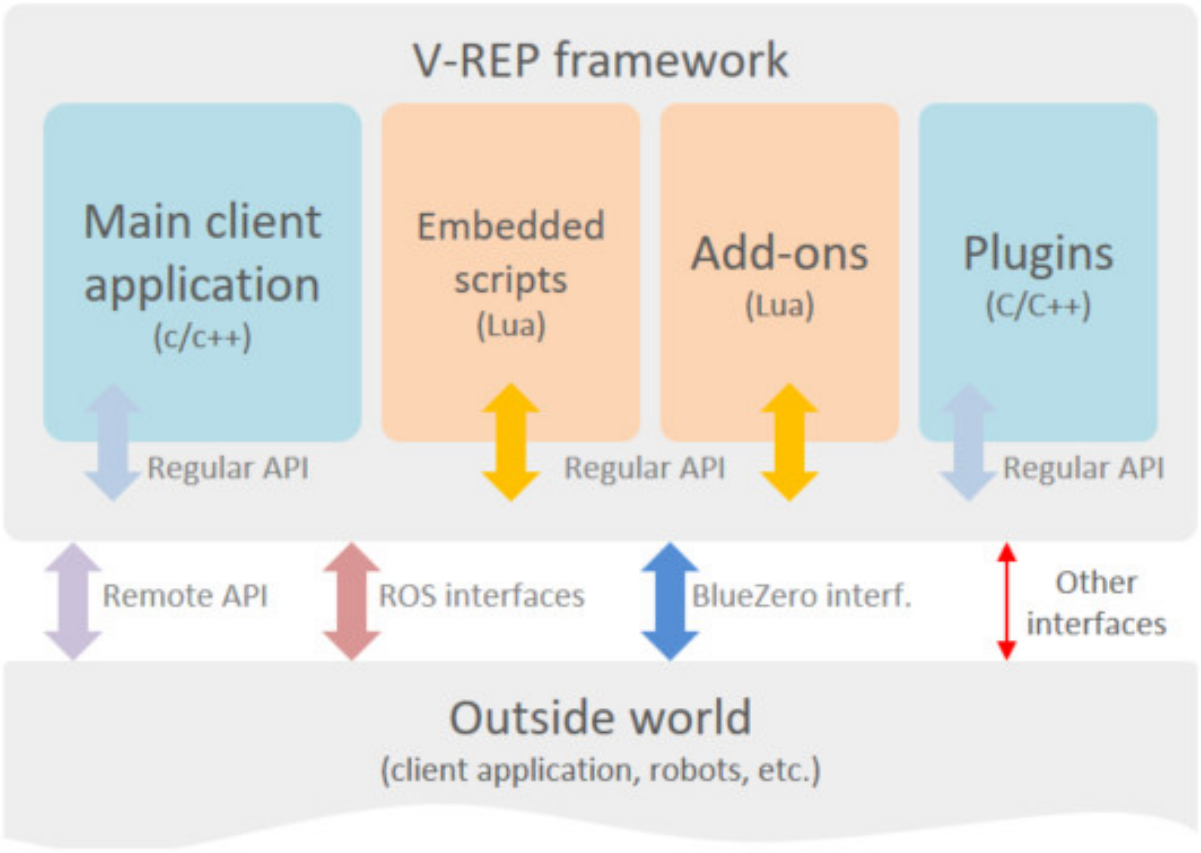

Figure 3.12: Means of communication with V-REP [33]

V-REP allows the simulation to be controlled from both, internal (V-REP regular API) and external (remote API) sources. When using the remote API option, the user has the option of operating it in the synchronous mode and the asynchronous mode.

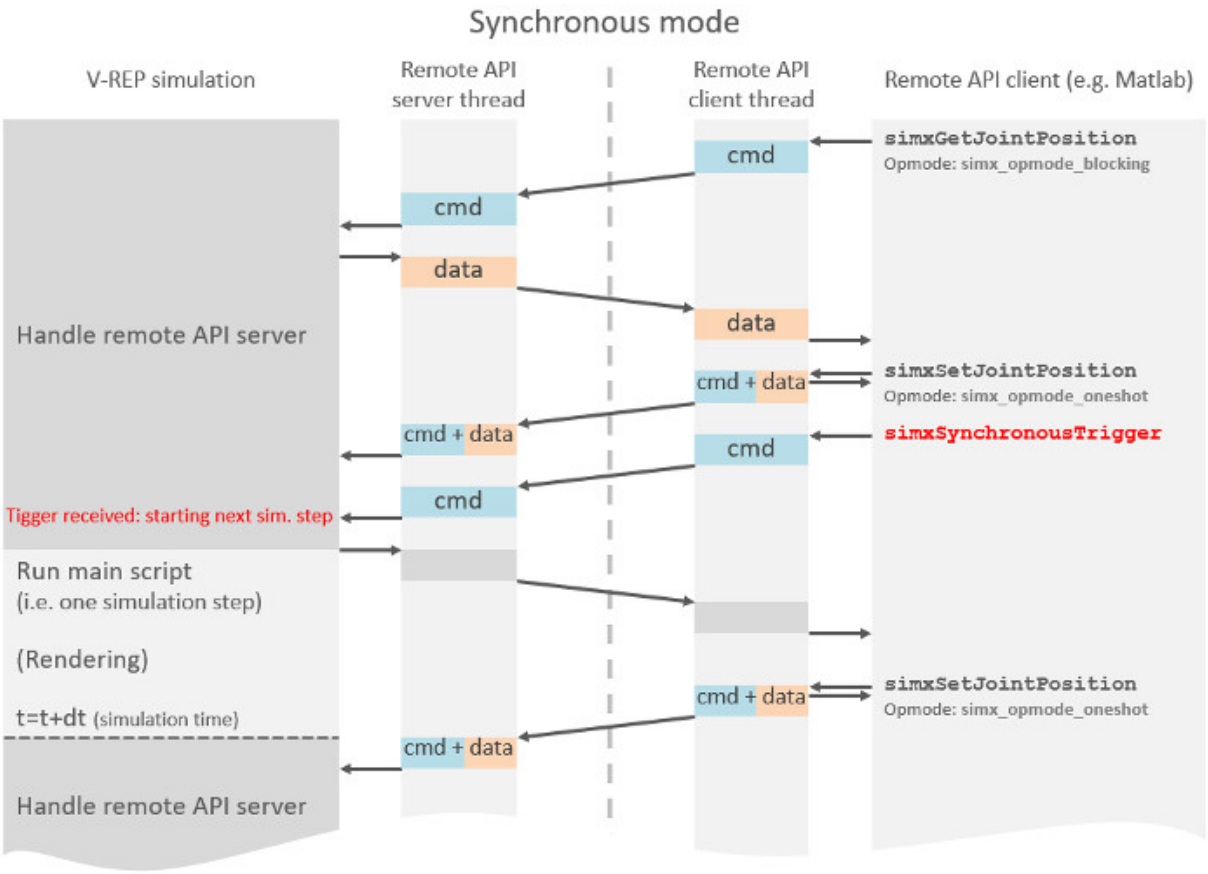

Figure 3.13: Schematic of synchronous mode operation [34]

The synchronous mode is used so that the controlling script and the simulation engine advance at the same rate ensuring that the data is scent and received without any losses due to de-synchronization. On the other hand, the asynchronous mode, where the controlling script and the simulation advance at separate rates, is used when time based control is not required. Since this project required continuous data logging, it was crucial that the controlling script and the simulation progressed at the same rate.

The disadvantage of using remote API for simulation control is that a significant slowdown is incurred because of the additional time required for the communication between the two programs (the V-REP server and in this case, the Python client). To mitigate this problem , the control script was rewritten using the built-in LUA API, which resulted in almost 400% increase

in simulation speed. A simulated second takes 0.5 real world seconds to simulate using the LUA API, while it took 2.33 seconds when using the Python API.

A code snippet is shown in Figure 3.14 below.

Figure 3.14: Snippet of robot control script

Table 3.4: Leg motion parameters

Leg sweep angle [degrees]	40
Stride length [mm]	76
Leg lift angle [degrees]	65
Stride height [mm]	40.15

3.7 Substrate Generation

V-REP handles environments as heightfield objects. The heightfield is generated as a csv file consisting of 3000 rows and 3000 columns to form a 3000x3000 grid. Each element in the csv file represents the height value at that coordinate. When it is imported into V-REP, each element represents a square cell of dimension 1mm x 1mm, thus forming a 3m x 3m meshed area.

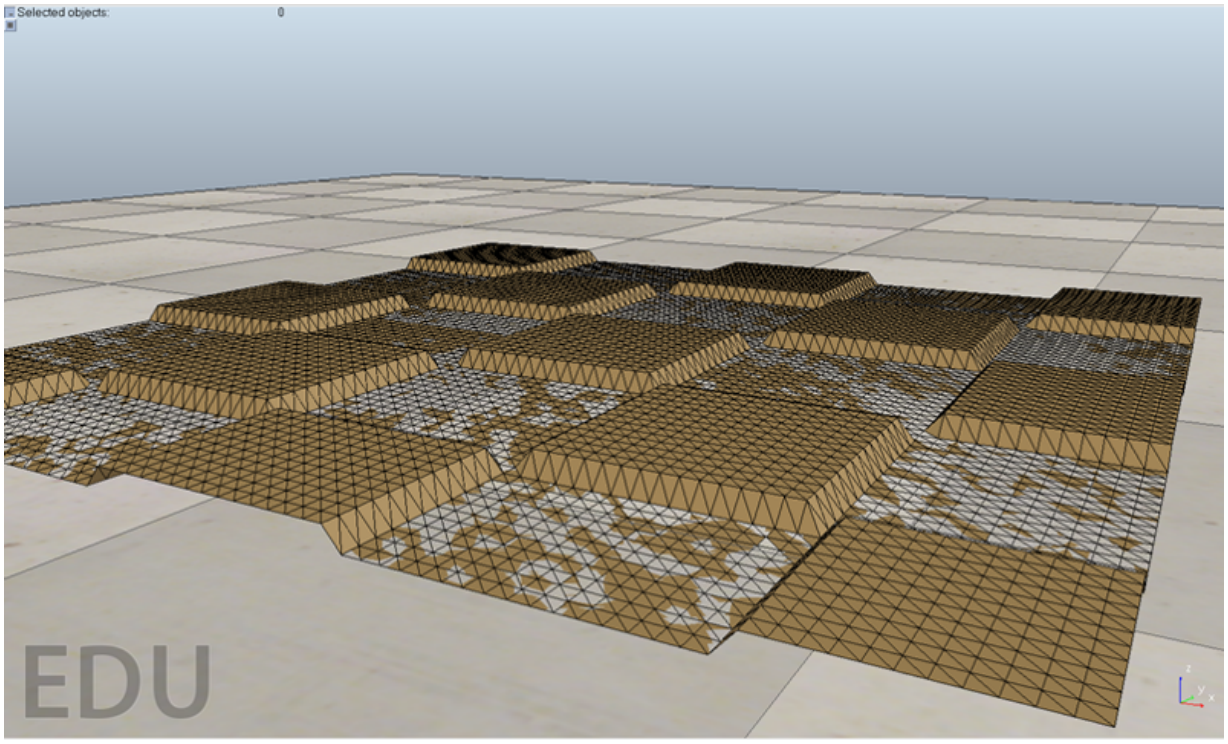

Figure 3.15: Sample heightfield surface with mesh visible

The heightfields were created with grid size ranging from 0in (flat ground) to 10in in steps of 0.5in and block height of 1in (approximately a third of the robot height).

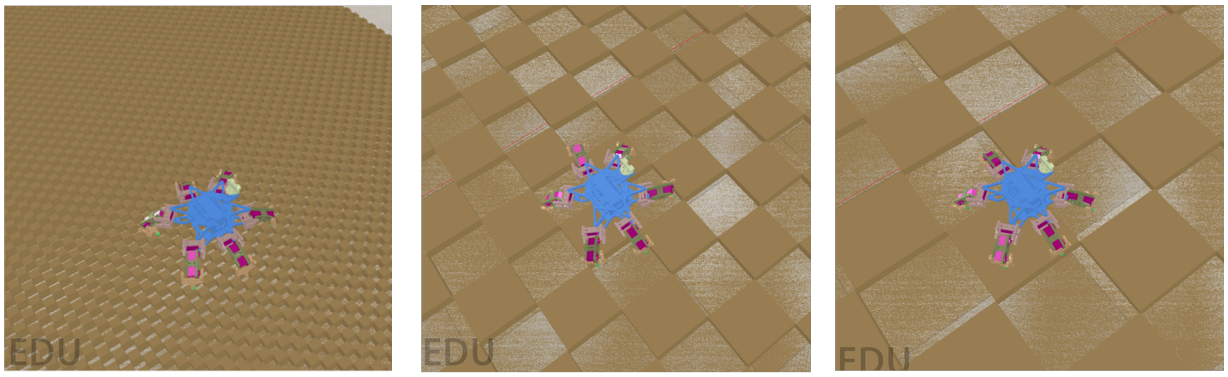

Figure 3.16: Terrain samples compared to HEXY a) 1in grid size, b) 6in grid size, c) 10in grid size

3.8 Validation and Fine Tuning

To ensure that the simulation returns correct parameters and results, the output parameters should be validated with corresponding real world parameters. The parameters chosen for verification were ground reaction forces measured by the force sensors on the foot-tips and the total displacement in a straight line on flat terrain. The sum of the forces measured by the sensors, if equal to the weight of the robot, would help in validation of the dynamics of the simulation. On the other hand, the kinematics of the simulation is verified by comparing the product of stride length and the number of strides taken during the simulation to the total displacement of the robot in that time. Ideally, both quantities should be equal.

However, due to lack of additional real world data for validating the dynamics aspect of the simulation, it was ultimately decided to study only the kinematics of the robot locomotion.

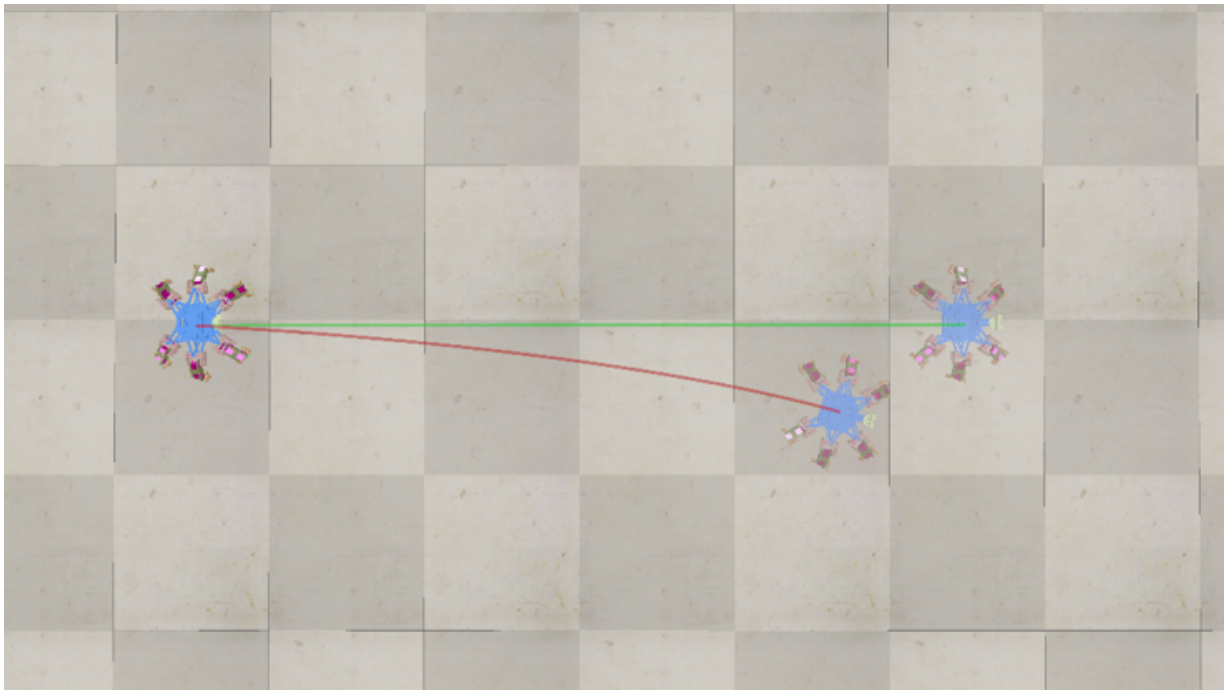

Figure 3.17: Ideal (green) versus non-ideal (red) handling of friction. Due to incorrect handling of friction, the robot tends to drift and travels less than the ideal case

Initially, the default engine, Bullet v2.78 was used and it was discovered that due to the incorrect handling of friction, the kinematics validation test failed. The newer Bullet v2.83 handled friction and collisions better and was able to solve the issue. In order to improve the accuracy of the simulation further, it was decided to decrease the timestep form the default value of 50ms to 25ms.

3.9 Methodology

The simulation environment was first set up by loading the heightfield into the scene, setting the joint angle ranges to fix the stride length and height values, and setting the number of repetitions required to 300. Thus 300 repetitions were obtained on each of the substrate and the data for each was stored in a suitably named folder for further analysis.

Chapter 4

Results and Discussion

4.1 Distance Travelled

By examining the recorded positions over each substrate for each of the iterations, it is possible to extract data such as the minimum, maximum, average and median distance travelled for a particular substrate. An example of this data is visualized in Figure 4.1.

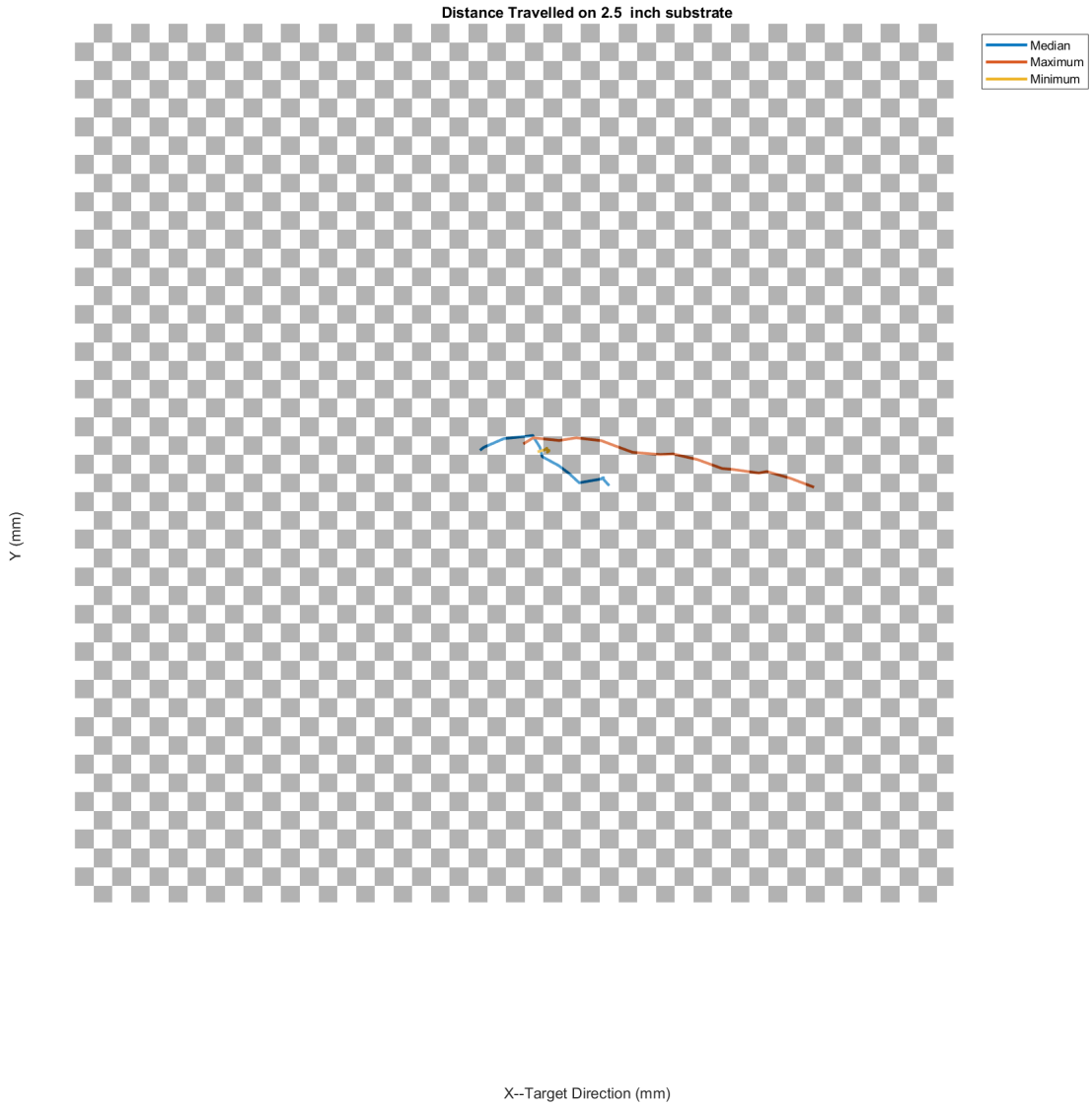

Figure 4.1: Example of data extracted from recorded positions.

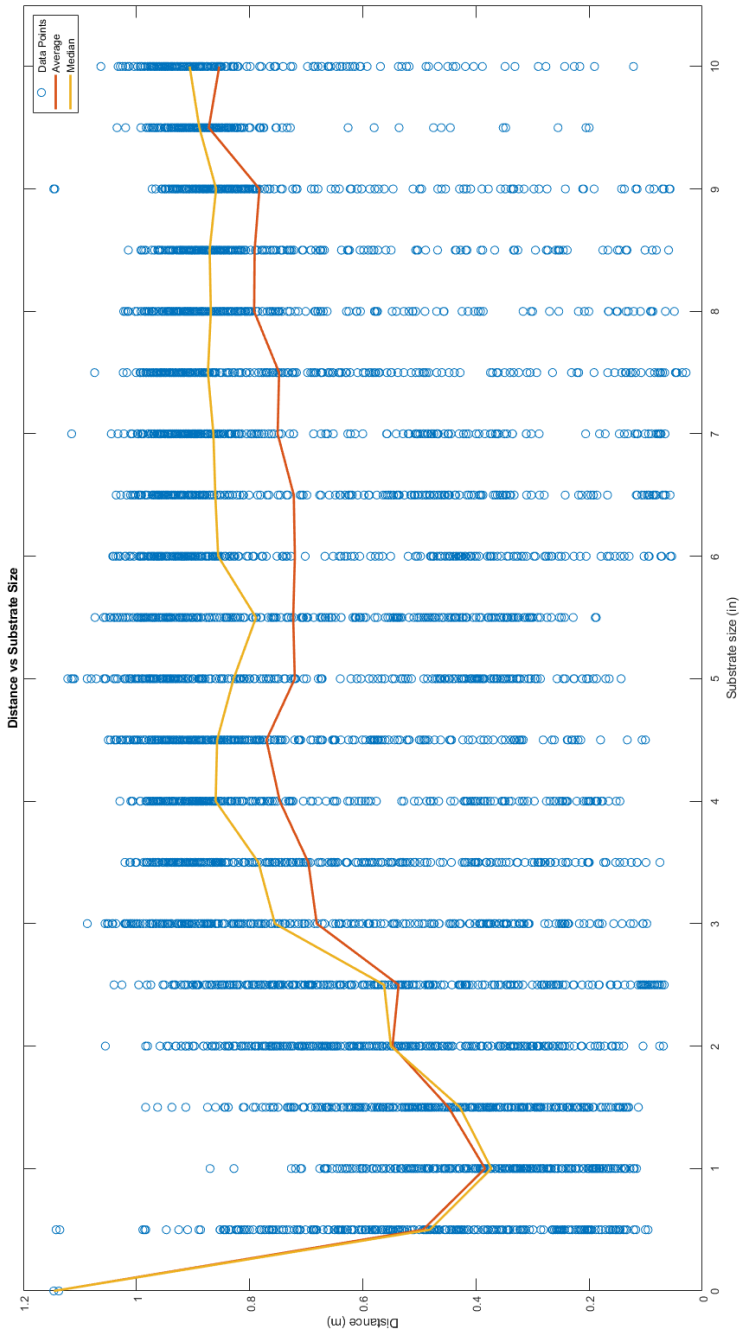

Figure 4.2: Distance travelled as a function of the substrate grid size

Plotting the distances travelled in all 300 iterations on a particular substrate for each of the grid sizes, as shown in Figure 4.2, allows a direct evaluation of performance. By converting all lengths to multiples of the stride length (7.6cm), we get Figure 4.3.

The plot can be divided into two regions, first, where there is a sudden drop from the flat substrate to the 1 inch grid substrate (0.33 times the stride length), which is followed by a monotonic rise in the distance till the 4.5 inch grid substrate (1.5 times the stride length), and second, where there is a very small, but gradual increase in the distance travelled until the 10 inch grid substrate (3.3 times the stride length).

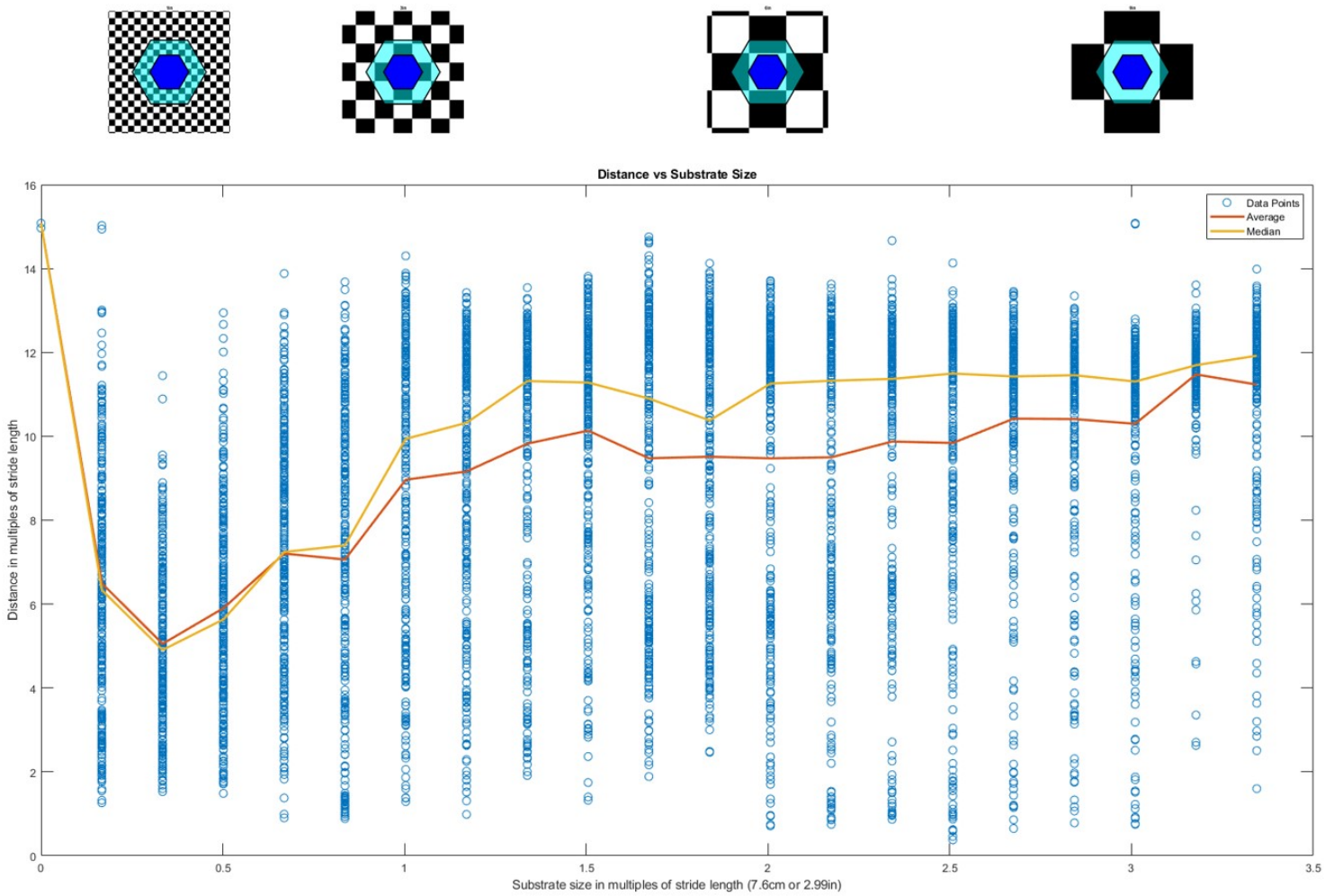

Figure 4.3: Comparison of distance travelled in multiples of stride lengths with the substrate grid size.

4.2 Leg Sweep Angles

In order to identify the cause of the abysmal performance on the 1 inch grid substrate, different leg sweep angles (angles the thoracic-coxal joint sweeps through during the stride and stance phase) were investigated. Since any change in the leg sweep corresponds to a similar change in the stride length, it was expected that the performance would improve with the increase in stride lengths and worsen with any decrease. The upper limit on the total sweep was set by the maximum allowable angle that did not result in a collision of the robot's legs. This angle, as determined using forward kinematics, was found to be 50 degrees. Since the original or the default value was close to this maximum, a major increase in the stride length was not expected, nonetheless, a drop in the performance for smaller strides could still be expected.

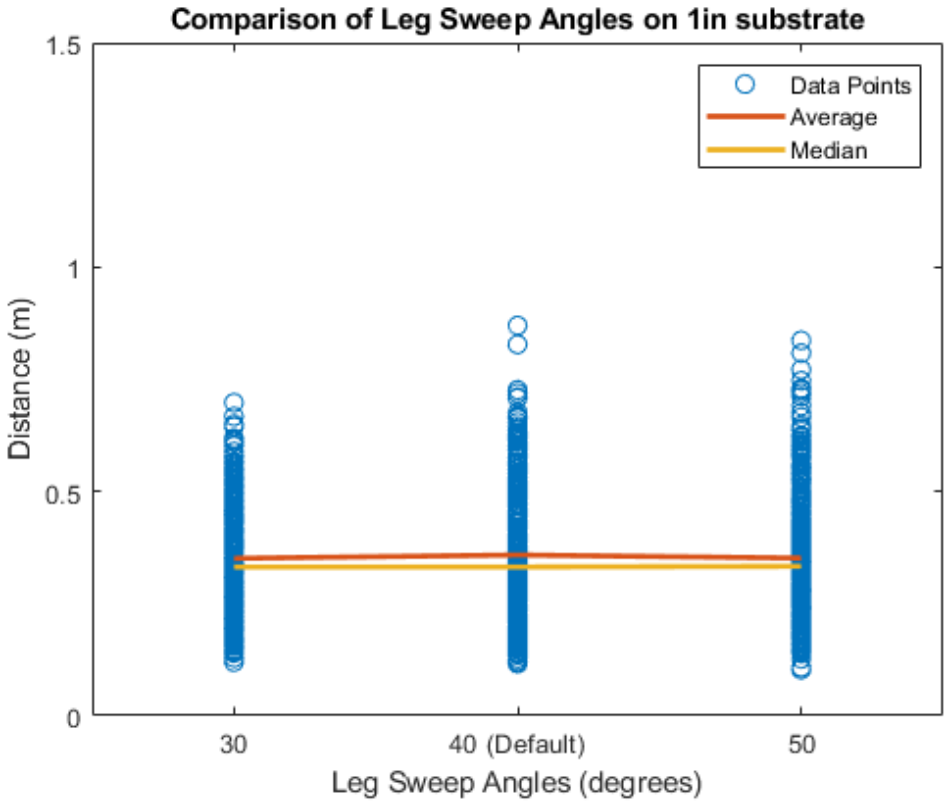

Figure 4.4: Comparison of the distance travelled on the 1 inch grid substrate using different leg sweep angles

4.3 Leg-tip Trajectories

Additionally, to investigate how different leg-tip trajectories could affect the performance, the two additional trajectories described in section 3.4 were implemented. Here also, the maximum total leg sweep was limited to 50 degrees, limiting the angle for lift-off and touch-down to nearly 14 degrees for the backward swing trajectory. It was hypothesized that for the pseudo-sinusoidal tip trajectory, the higher probability of the leg-tip colliding with the surface structures would substantially hinder the rate of advancement. On the other hand, the backward swing trajectory would aid in not only covering more distance during the swing phase, but also help in hooking the legs in crevices and effectively help in getting better foot-holds and improving the rate of advancement. The semi-backward swing trajectory was expected to perform somewhere in between the default and backward trajectory.

As expected, the results showed that the pseudo-sinusoidal trajectory results in a 24.61% decrease in the distance travelled. Surprisingly, the backward swing trajectory did not perform according to expectation and instead resulted in a 6.65% decrease in the distance, while the semi-backward swing trajectory resulted in a 6.98% decrease in speed compared to the default trajectory.

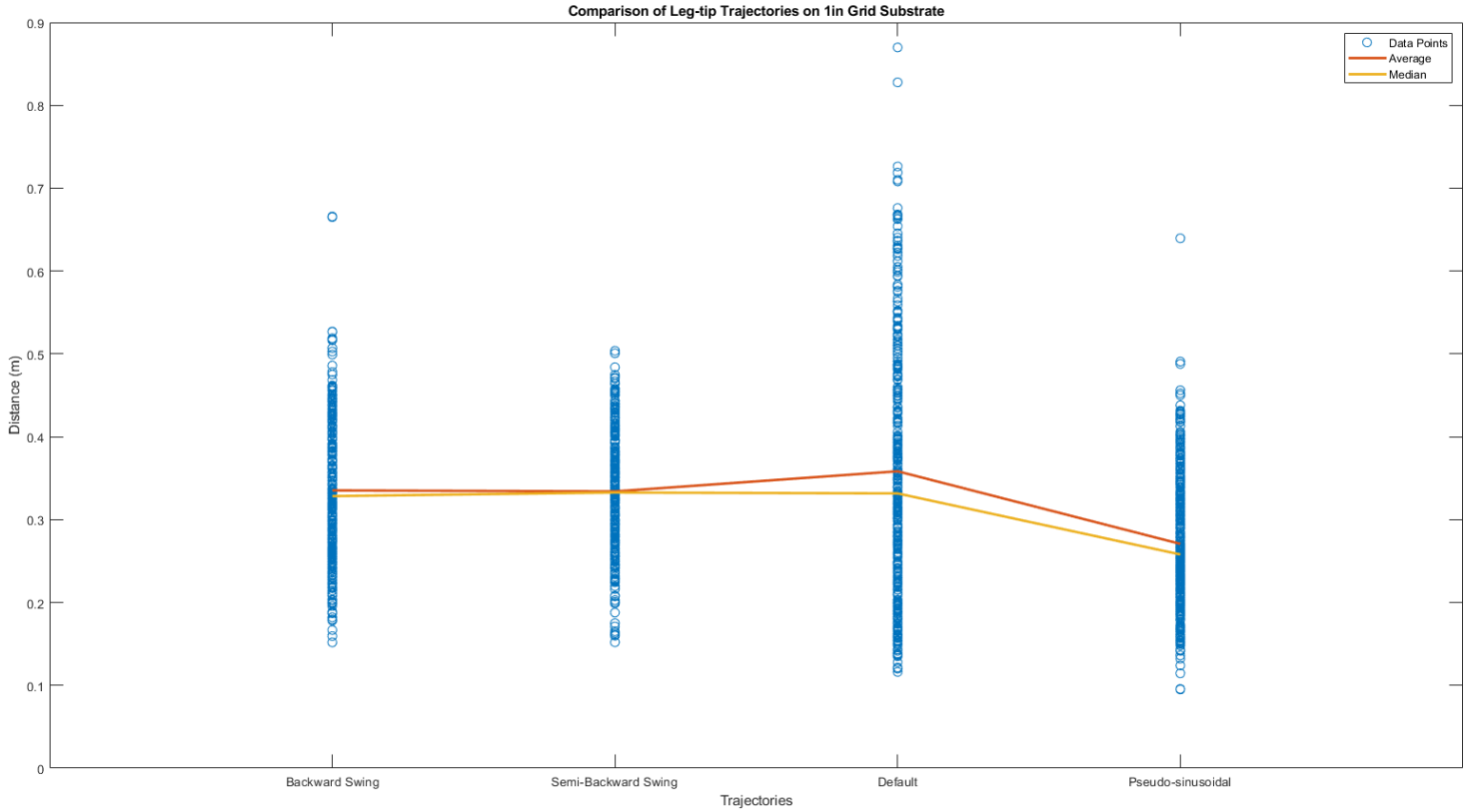

Figure 4.5: Comparison of the distance travelled on the 1 inch grid substrate using different leg-tip trajectories

4.4 Discussion

The initial portion of the distance versus substrate size plot suggests that the locomotion is strongly affected by the substrate grid size when the stride length is larger than the substrate grid size, while the effect is smaller once the grid size exceeds the stride length. It can be argued that if the distance travelled is a function of the substrate grid size and if the time is halved, the slope of the adjusted distance travelled versus the substrate grid size should remain the same.

Curiously, as evident from Figure 4.6, this holds true only in region 1 (substrate grid size is between 0.33 and 1.5 times stride length) where the percent error between the predicted and actual rate of increase in the distance only 4.05%, while it is over 50% in region 2 (substrate grid size is between 1.5 and 3.3 times stride length). This suggests that the nature of leg-surface structure interactions (slippage, collisions, and immobilization of legs) strongly affects the distance travelled and thus the speed, while factors other than the nature of interaction and rather the rate of leg-ground interactions govern the speed in region 2.

The behavior in region 1 is attributed to the limited space available for the leg-tips to move, which is further exacerbated by the limitations of the gait controlling script. The open-loop tripod gait implemented cannot maintain a no-slip conditions at the leg-ground interface. This is because all the thoracic-coxal joints are programmed to sweep through equal angles, resulting in equal arcs traced by the leg-tips when in stance phase. Additionally, during the entire stance phase, the coxal-femoral and the femoral-tibial joints are not actuated further and maintain a fixed position. Since the legs are centered at the vertices of a regular hexagon, the arcs traced by the leg-tips are also oriented in different directions. This results in different lengths covered by the leg-tips in the sagittal and mediolateral directions as shown in Figure4.7.

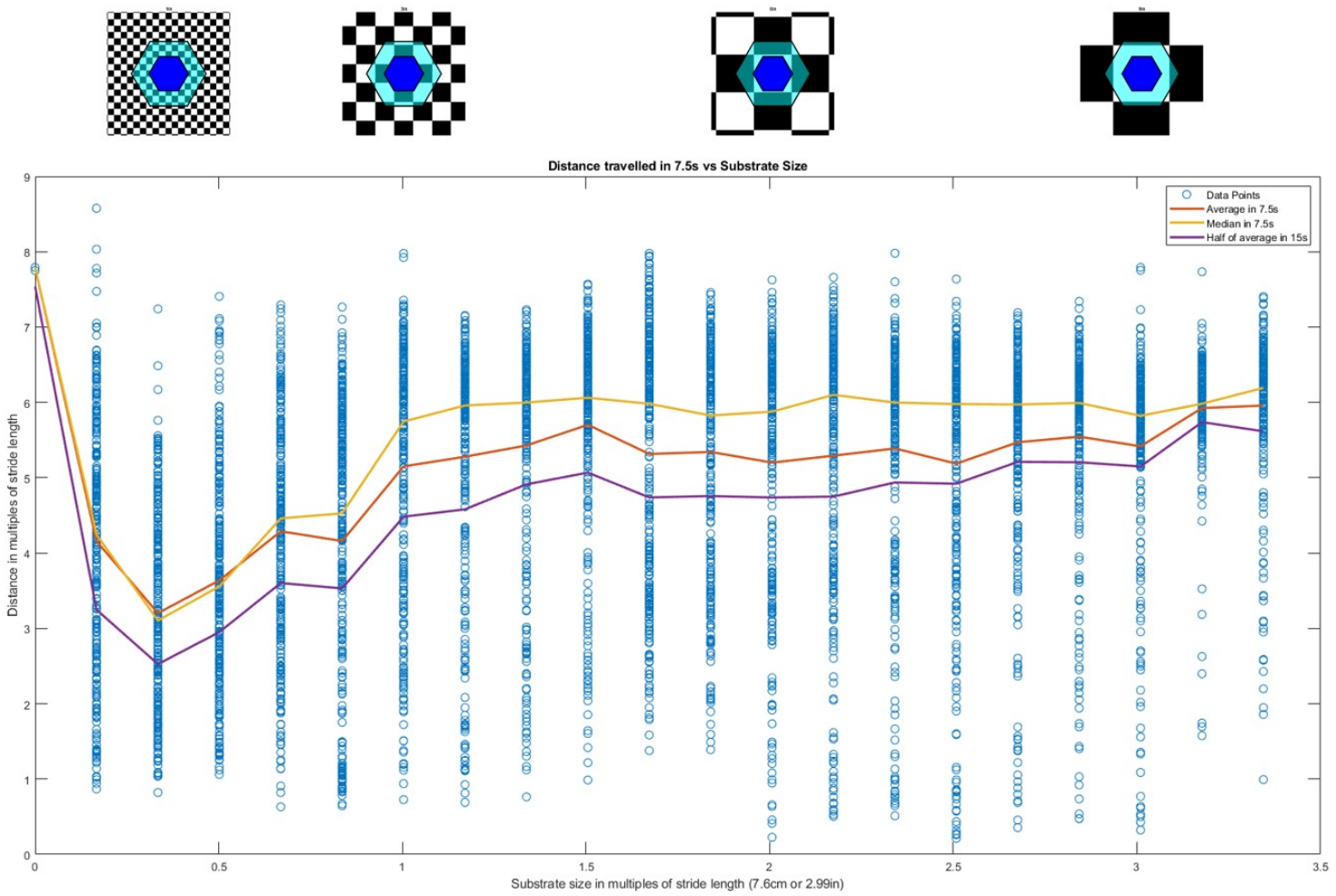

Figure 4.6: Comparison of distance travelled in half the time (7.5s) in multiples of stride lengths with the substrate grid size.

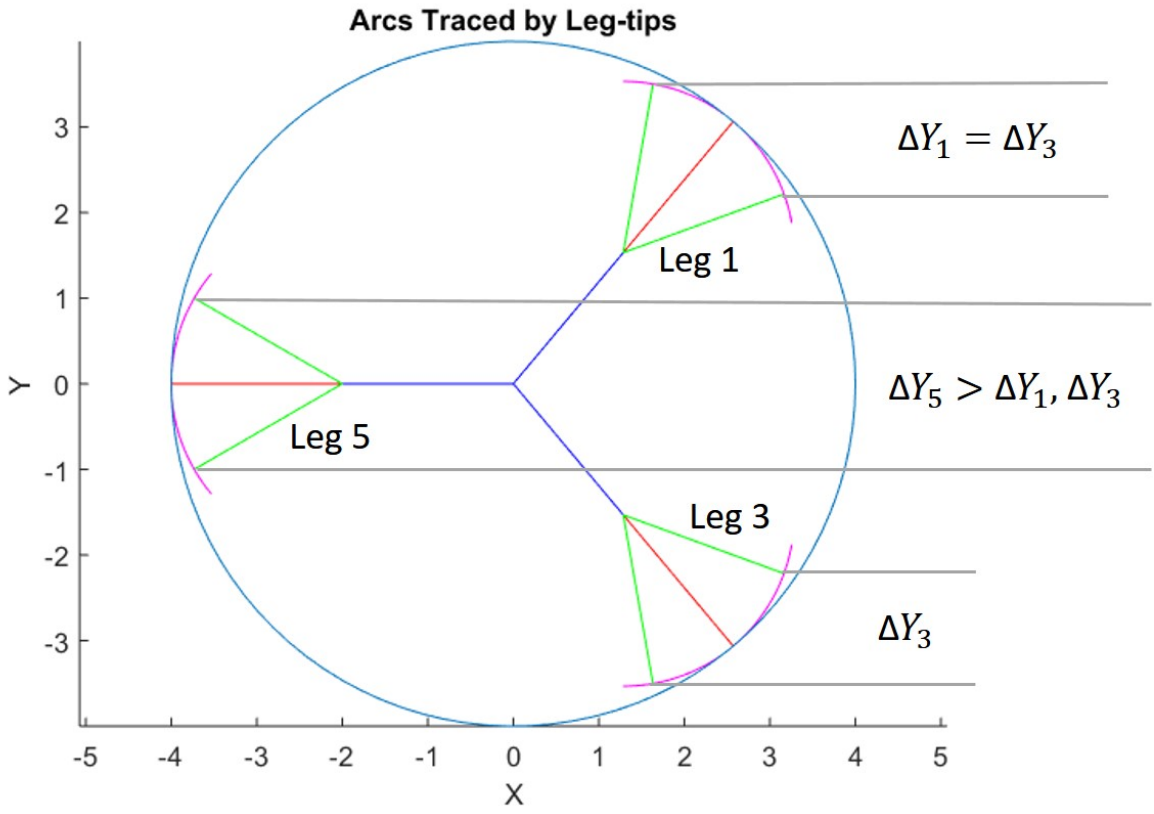

Figure 4.7: Arcs traced by leg-tips result in unequal distances swept in the sagittal and medio-lateral directions

The result of this difference in travelled distance in the intended direction of locomotion results in leg slippage. The issue of slippage becomes particularly problematic when the leg-tips touch-down on the raised part of the surface structures and slip and fall into the cavities during the backward leg motion initiated during the stance phase as shown in Figure 4.8. Instead of raising and pushing the body forward as intended, the legs slip back into the cavities which were occupied during the previous cycle while keeping the body in the same position and result in a repetition of the previous cycle, thus hampering the rate of advancement.

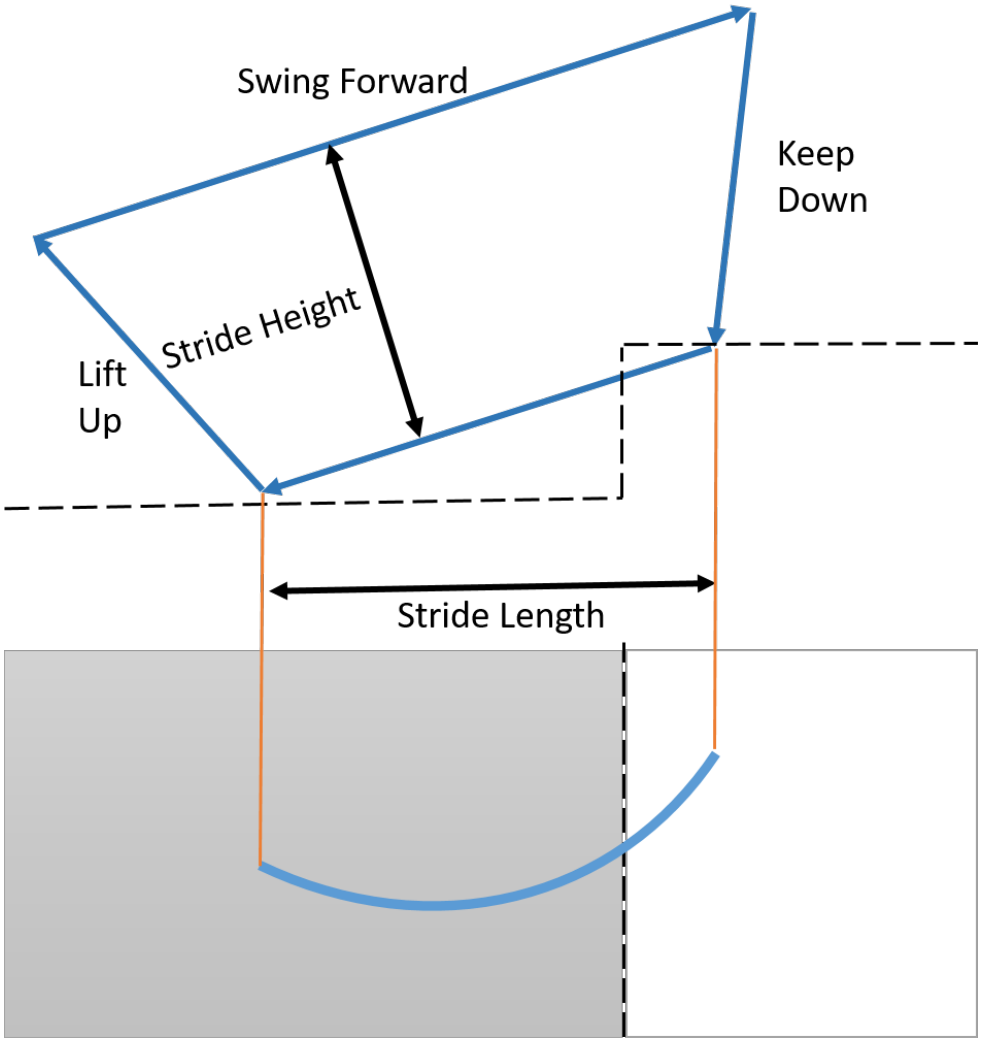

Figure 4.8: If the leg-tips land on the elevated portion of the surface and are unable to maintain a good grip, instead of pulling the body forward, they slip and fall back into the previous cavity

Another problem that can be caused because of this is that the legs trapped in a cavity not large enough to accommodate this slippage occasionally force the body to rotate. This rotation may allow the robot to get better foot-holds and eventually free the trapped leg. If the rotation does not result in this scenario, there are chances that the free legs will be forced to occupy positions other than those intended. If the legs land on top of the surface structures and there is enough space to accommodate the new positions, the robot continues operation as usual. On the other hand, if the legs land in a cavity and if there is not enough space to accommodate the new

leg positions (the leg-tips are surrounded by the walls formed by the surface structures), these legs then get pushed against the walls formed by the surface structure and force their thoracic-coxal joints in angles larger than intended. This could further aggravate the problem by causing leg-leg collisions, further restricting the space available for motion.

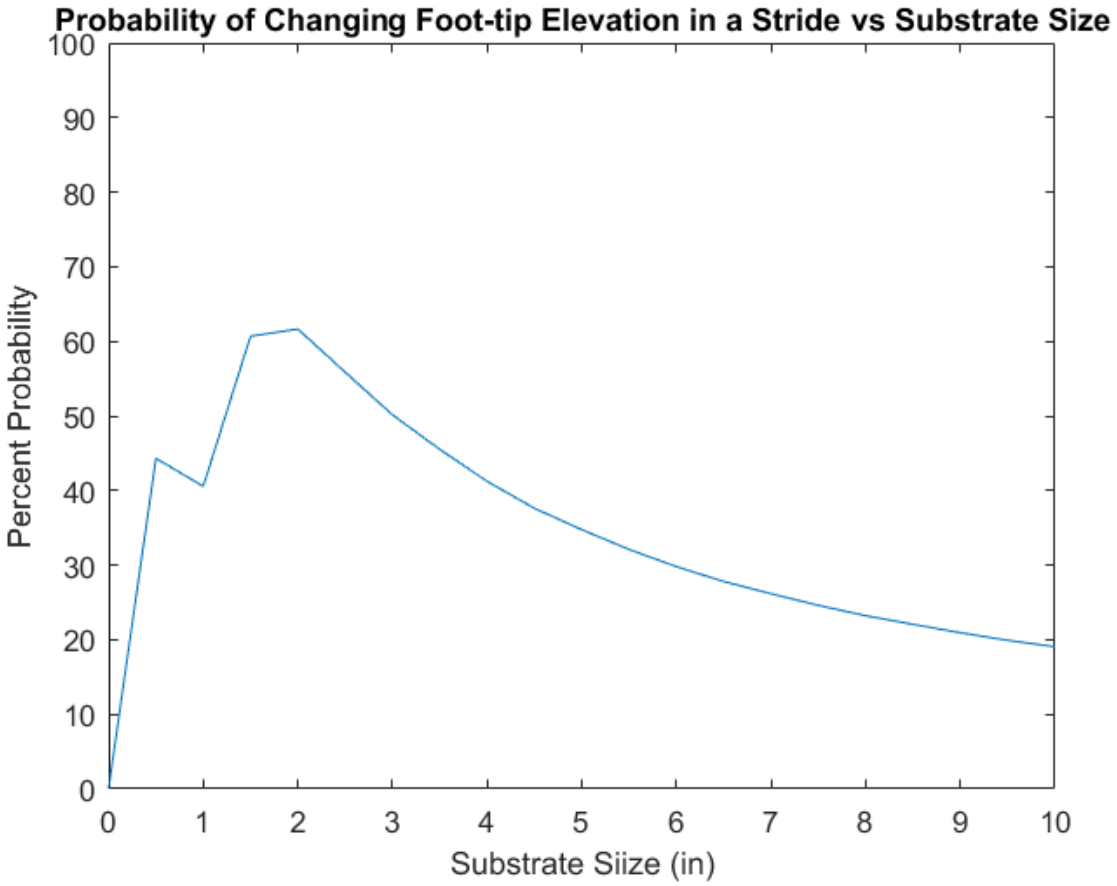

Figure 4.9: Probability of leg-tips taking-off and landing at different elevations

To investigate the statistical probability of the occurrence of these events as a function of the substrate grid size, Monte Carlo method was employed. The method involved discretizing the substrate surface structures into 1 mm^2 square areas. A simplified model of the hexapod (with 3 legs in stance phase) leg-tip positions was then iteratively swept over the possible positions and orientations and the elevation of the foot-tips at the beginning and end of the stance phase was recorded as shown in Figure 4.10. The probability of changing foot-tip elevation was evaluated

by counting the total possible foot placements and the number of placements that resulted in the occurrences of this change.

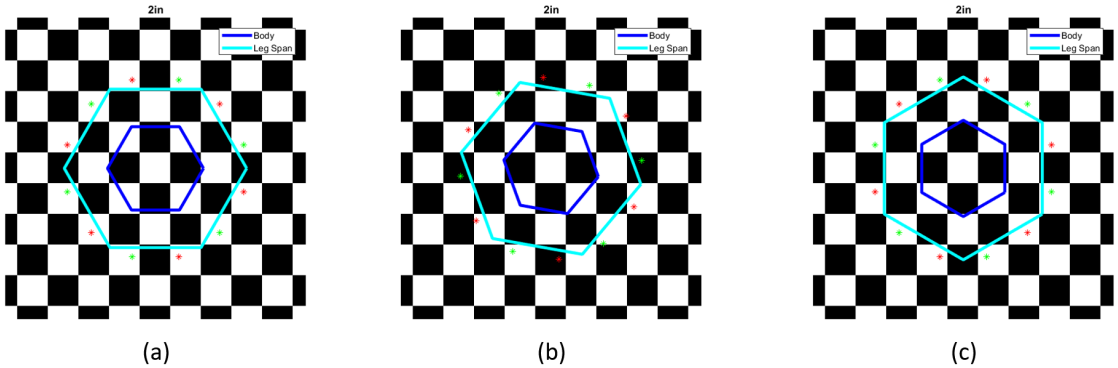

Figure 4.10: Model of hexapod being rotated about a fixed point to evaluate probability of changing foot-tip evaluation: a) At 0°, the blue hexagon represents the body while the corners of the cyan hexagon represents the leg-tips. The red and green points represent possible foot touch-down locations b) At 10° orientation c) At 30° orientation

As expected, Figure 4.9 shows that the probability of changing foot-tip elevation decreases gradually as the substrate size increases, as more time is spent traversing over individual surface structures or the cavities between them when the substrate size is larger than the stride length (approximately 2.99in). When the stride length is smaller, however, the probability, instead of increasing gradually from the 0.5in grid size to the 2in size, dips suddenly on the 1in grid size. The probability of changing elevations is maximum on the 2in grid sized substrate at nearly 60% and drops to nearly 50% on the 3in substrate.

While Figure 4.9 depicts the probability of changing leg-tip elevation, Figures 4.11 and 4.12 represent the probability of not changing the foot-tip elevation and maintaining a fixed elevation, either by taking-off and landing in different or the same cavity or by taking-off and landing on top of different or the same surface structure.

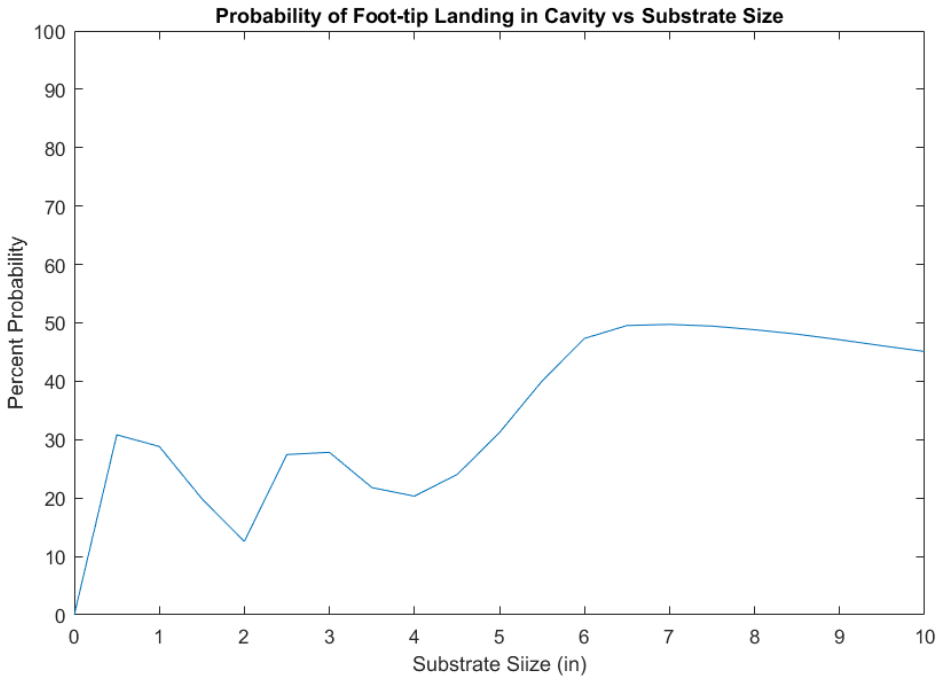

Figure 4.11: Probability of leg-tips taking-off and landing in cavities

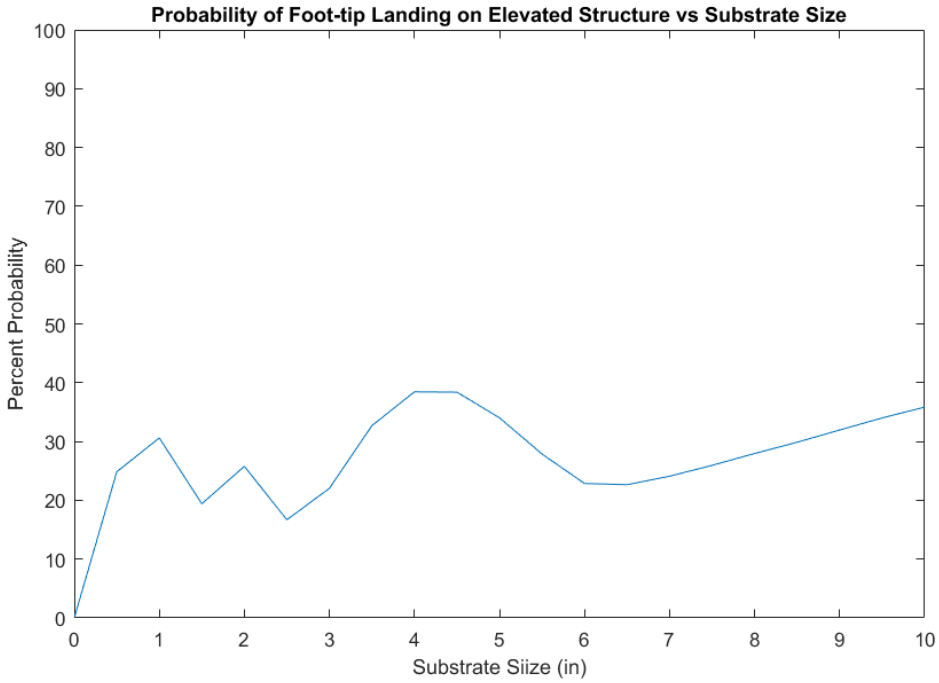

Figure 4.12: Probability of leg-tips taking-off and landing on top of surface structures

From Figure 4.9, it is evident that the probability of maintaining the same foot-tip elevation is higher between 0.5in and 1.5in substrate. Comparing the probabilities, there is a slightly higher chance of the foot-tip landing in a cavity on the 0.5in grid substrate, while the chances of landing on top of the surface structure are higher for the 1in substrate. On the 1.5in substrate size, the probability of landing in cavities and on top of surface structures are equal. On the 2in substrate, the probability of changing the foot-tip elevation is the highest, and coincidentally, the probability of landing in cavities is the lowest. As the substrate size increases to 3in, the probability of maintaining the same elevation lower increases. As the substrate grid size increases beyond that to 4in, the probability of the legs maintaining the higher elevation increases, after which, the probability of maintaining a lower level increases.

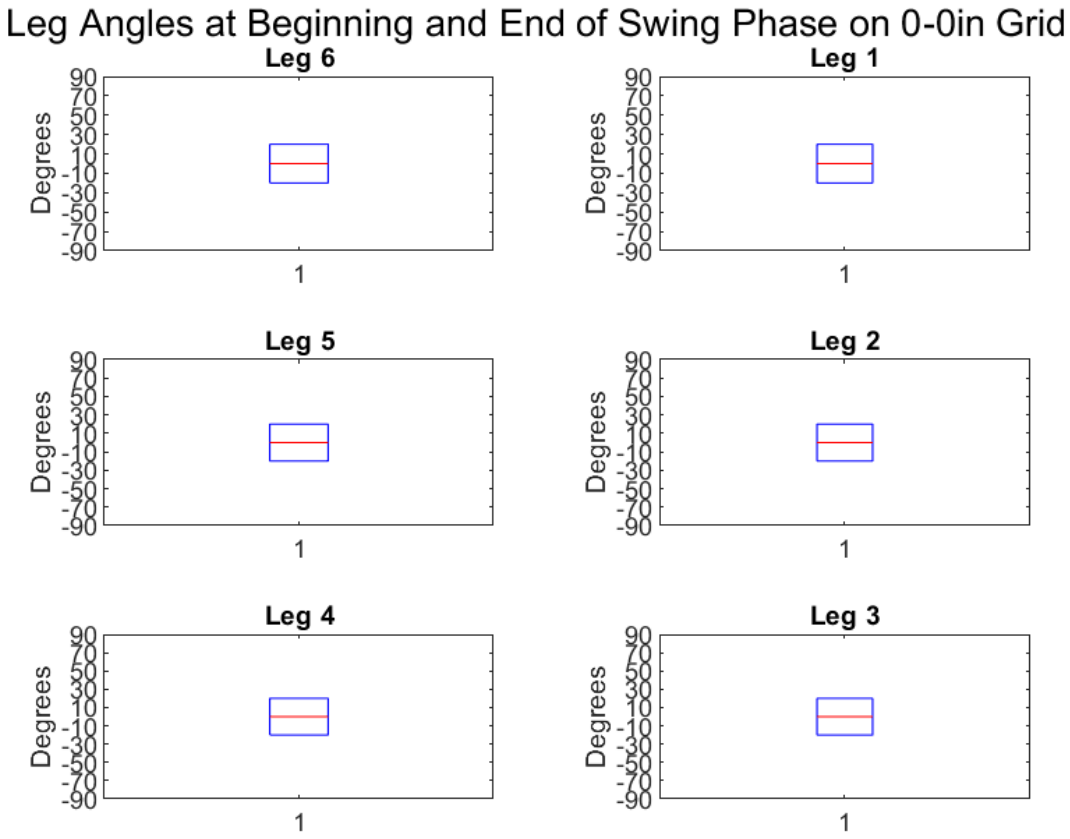

Figure 4.13: Ideal leg angles (on flat ground)

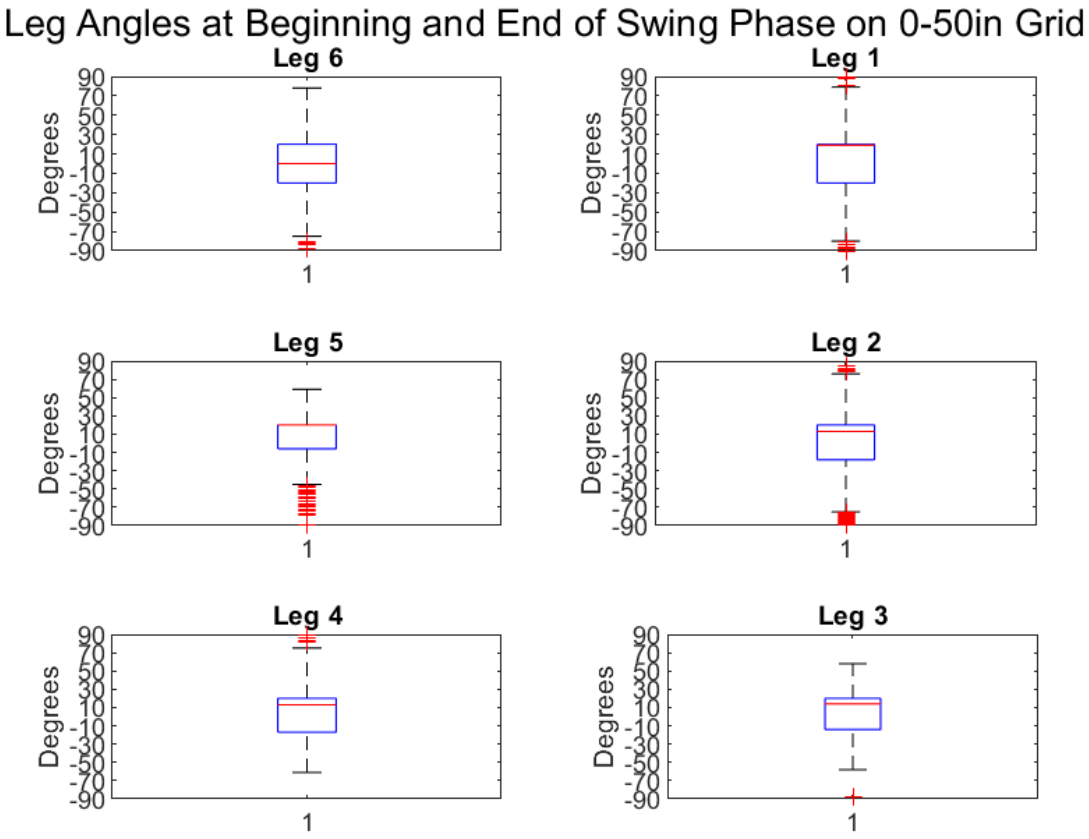

Figure 4.14: The mid-limbs forced to very large angles on the 0.5 inch grid sized substrate

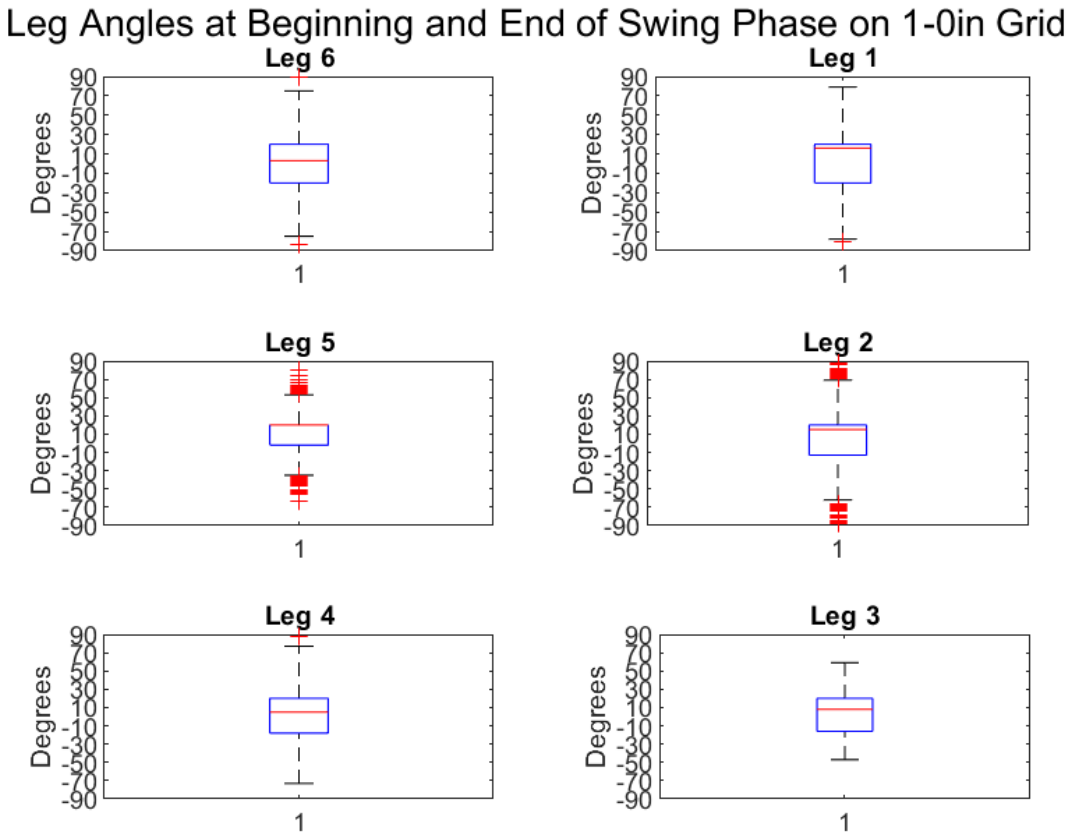

Figure 4.15: The number of times the legs are forced to large angles reduces on the 1 inch grid substrate

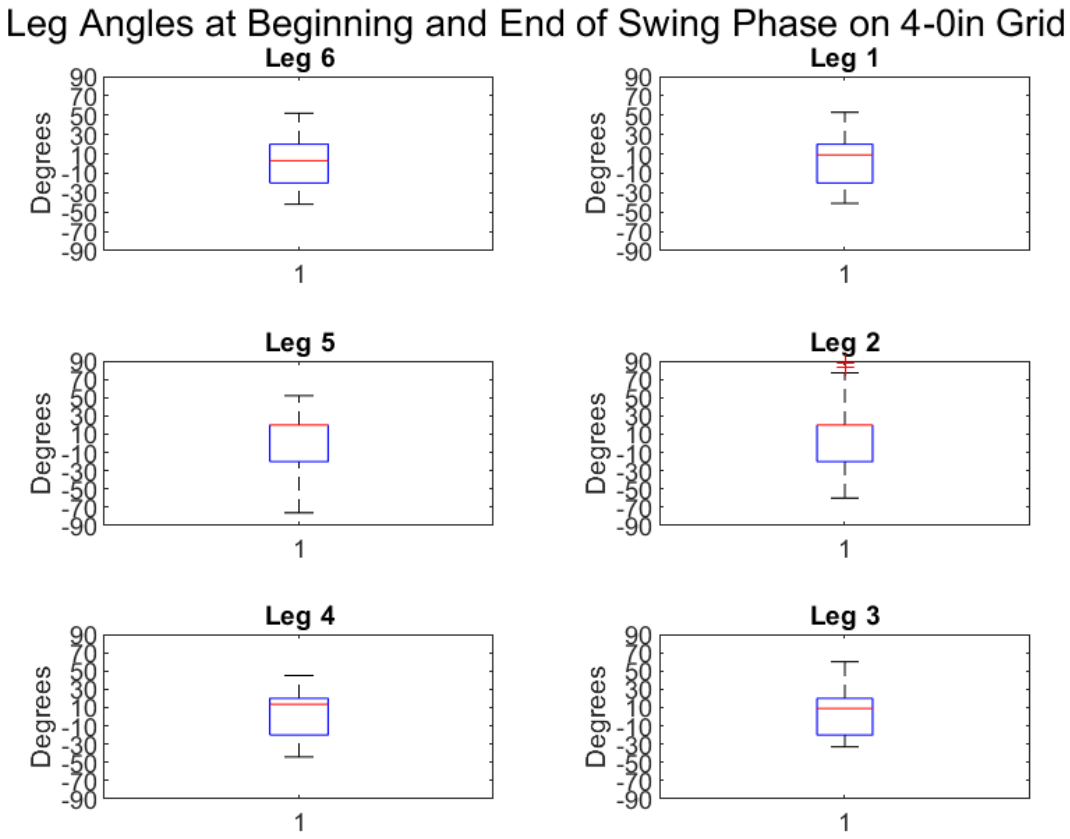

Figure 4.16: As the substrate grid size increases, the average returns to zero (ideal value) with fewer occurrences of large leg angles

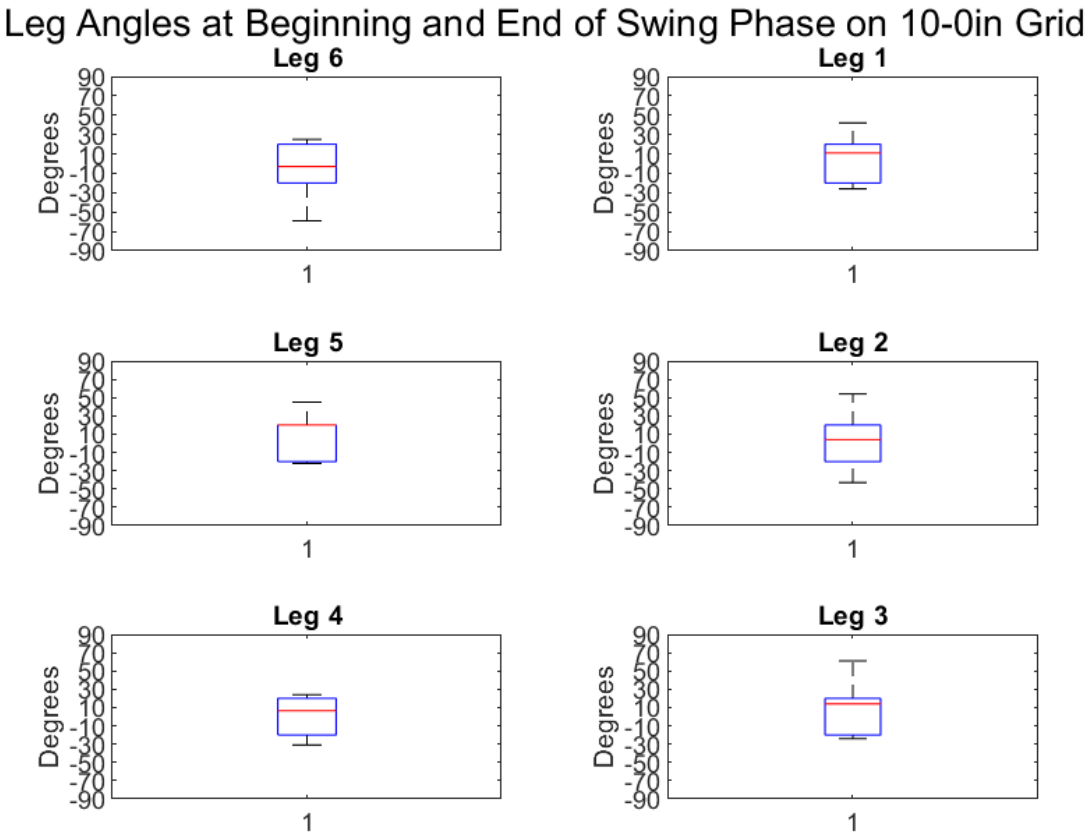

Figure 4.17: Leg angles approach ideal values on the 10 inch grid substrate

As evident from Figures 4.13 to 4.17 which depict the leg angles at the beginning and end of the swing phase on different substrates, the legs experience higher frequency of being forced and higher magnitudes of angles on the smaller substrates. As the substrate grid size increases, both of these parameters gradually approach the ideal values (on flat ground, where there is no obstruction).

Based on this evidence, it is hypothesized that the problem of repeated cycles in minimized on the smallest grid size (0.5 inches) and the primary cause for the slow-down is the problem of legs being forced into larger angles than intended. As the substrates grid size increases, it is speculated that the contribution of the leg slippage (leading to repeated cycles) to the loss in speed increases while the problem of legs being forced into large angles diminishes until the grid

size starts to exceed the stride length after which leg slippage and the frequency of interaction with the surface structure edges become the major reasons for the decrease in speed.

The variation in leg sweep did not result in significant change in the distance travelled because the small increments in the sweep angle only attribute to 1 to 1.5cm increase in the stride length.

The reason for the poorer than expected performance of the backward and the semi-backward swing trajectories can also be attributed to both the issue of limited space for leg motion – causing collisions with the surface structures during backward swing - and the issue of inadequate grip and slippage with the substrate during the touch-down because of the backward motion. This suggests that the default trajectory remains ideal.

Ensuring the no-slip condition at the leg tips is a possible solution that is expected to result in improvement in performance by eliminating the problem of slippage and hence the problem of legs being forced into large angles. While the current gait control script is incapable of varying motor speed at will as mentioned in section 3.5 and hence incapable of actuating the leg motors to ensure no-slip, it is possible to design such a gait for physical robots using inverse kinematics.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

The locomotion of a hexapod robot over structured terrain was investigated using simulation package V-REP. The robot was modelled on HEXY, a open source hexapod produced by Arcbotics. The effects of varying stride lengths and leg-tip trajectories while using open-loop alternating tripod gait were studied. The results showed that when the stride length was more than the substrate feature size, the nature of leg-surface interactions influenced the speed of locomotion over those substrates. When the stride length was smaller than the substrate feature size, it was the rate of the leg-surface interactions that influenced speed the most. An improvement is suggested in the form of implementation of leg-tip trajectory that ensures no-slip condition at the leg-surface interface.

5.2 Future Work

There is a tremendous scope for improvement in the simulation. An inherent issue with the current setup is that if the leg angles exceed a certain value, it results in leg-leg or

leg-body collisions. In the case that the physics engine is unable to model the collisions and surface contacts, given the constraints, the simulation becomes unstable which ultimately leads to inaccurate simulation of the physics.

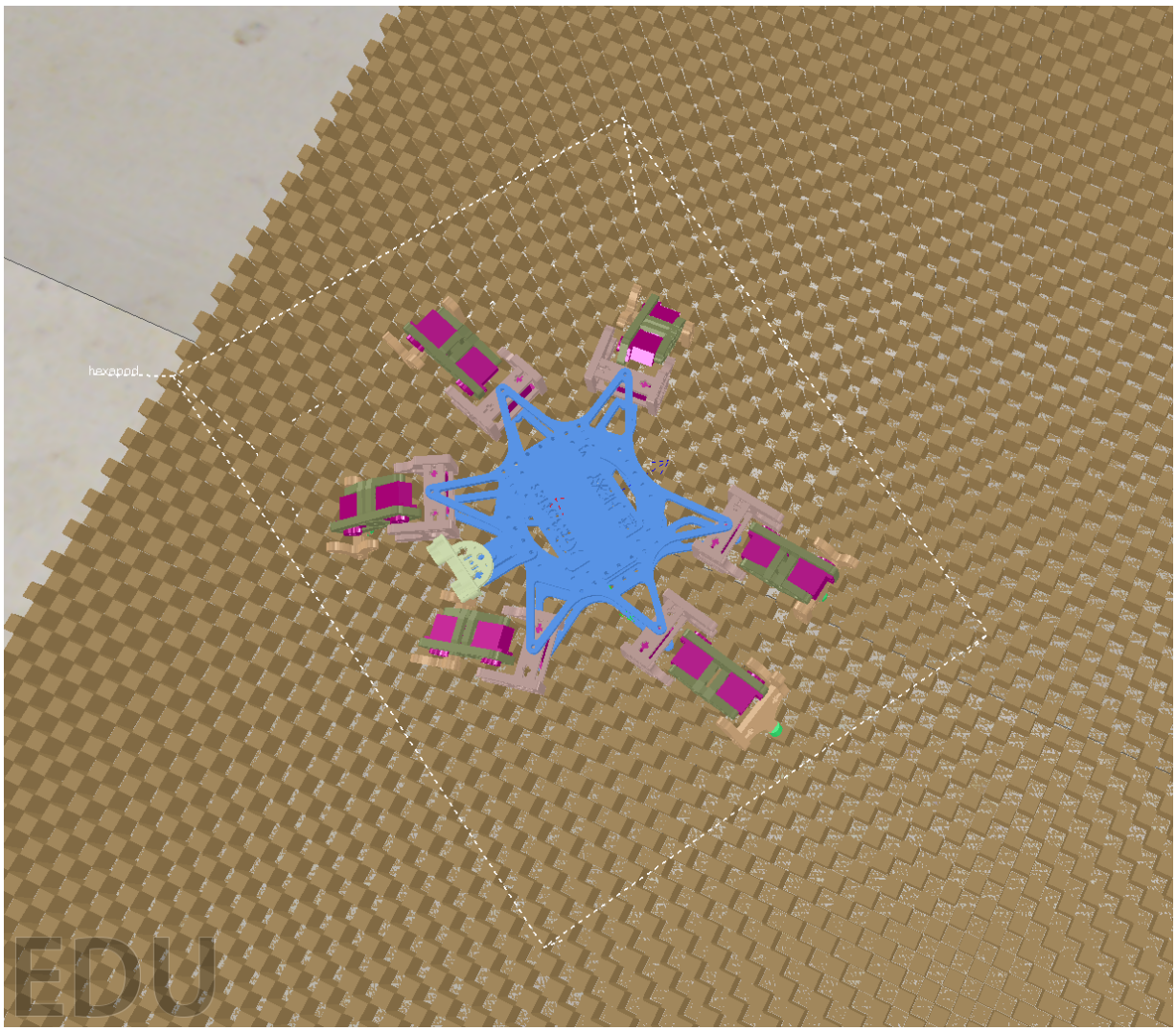

Figure 5.1: Leg 6 (Front-Left) being forced to a large angle causing leg-body collisions

As mentioned in section 2.3.2, Vortex Studio gives the highest accuracy, but requires the most time to solve. In order to get the most accurate results, it would be desirable to invest time to conduct a similar study using this tool.

Future work might lead to implementation of a central pattern generator or using force

sensors on the leg tips to implement reflex based locomotion. As seen in Figure 5.1, in case the leg placement is fortuitous, the distance travelled on all substrates is nearly equal. A study may be conducted to identify those foot placements and use them to implement a dictionary or a map to influence foot placement decision of a hexapod using a reinforcement learning algorithm.

Bibliography

[1] G. Carbone and M. Ceccarelli, "Legged robotic systems, kordic, v.; lazinica, a.; merdan, m.(editors), cutting edge robotics, ars international vienna," 2005.

[2] M. Nandhini, V. Krithika, and K. Chittal, "Design of four pedal quadruped robot," in *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, pp. 2548–2552, IEEE, 2017.

[3] D. Pongas, M. Mistry, and S. Schaal, "A robust quadruped walking gait for traversing rough terrain," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 1474–1479, IEEE, 2007.

[4] M. P. Murphy, A. Saunders, C. Moreira, A. A. Rizzi, and M. Raibert, "The littledog robot," *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 145–149, 2011.

[5] X. Y. Sandoval-Castro, M. Garcia-Murillo, L. A. Perez-Resendiz, and E. Castillo-Castañeda, "Kinematics of hex-piderix-a six-legged robot-using screw theory," *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, p. 19, 2013.

[6] S. Kajita and B. Espiau, "Legged robots," *Springer handbook of robotics*, pp. 361–389, 2008.

[7] FZI, "Lauron," 2019. Last accessed 16 September 2019.

[8] B. Dynamics, "Rhex," 2019. Last accessed 16 September 2019.

[9] J. De León, M. Garzón, D. Garzón-Ramos, and A. Barrientos, "Study of gait patterns for an hexapod robot in search and rescue tasks," in *Iberian Robotics conference*, pp. 731–742, Springer, 2017.

[10] F. Tedeschi and G. Carbone, "Design issues for hexapod walking robots," *Robotics*, vol. 3, no. 2, pp. 181–206, 2014.

[11] J. T. Watson, R. E. Ritzmann, S. N. Zill, and A. J. Pollack, "Control of obstacle climbing in the cockroach, blaberus discoidalis. i. kinematics," *Journal of Comparative Physiology A*, vol. 188, no. 1, pp. 39–53, 2002.

[12] X. Chai, F. Gao, Y. Pan, and Y.-l. Xu, "Autonomous gait planning for a hexapod robot in unstructured environments based on 3d terrain perception," in *The 14th International Federation for the Promotion of Mechanism and Machine Science World Congress (IFToMM), Taipei, Taiwan*, 2015.

[13] R. Volpe and K. Kawasaki, "Systems - image gallery - the athlete rover," 2019. Last accessed 16 September 2019.

[14] J. Faigl and P. Čížek, "Adaptive locomotion control of hexapod walking robot for traversing rough terrains with position feedback only," *Robotics and Autonomous Systems*, vol. 116, pp. 136–147, 2019.

[15] N. Kottege, C. Parkinson, P. Moghadam, A. Elfes, and S. P. Singh, "Energetics-informed hexapod gait transitions across terrains," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5140–5147, IEEE, 2015.

[16] T. Homberger, M. Bjelonic, N. Kottege, and P. V. Borges, "Terrain-dependant control of hexapod robots using vision," in *International Symposium on Experimental Robotics*, pp. 92–102, Springer, 2016.

[17] P. Čížek, D. Masri, and J. Faigl, "Foothold placement planning with a hexapod crawling robot," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4096–4101, IEEE, 2017.

[18] Y. Xu, F. Gao, Y. Pan, and X. Chai, "Hexapod adaptive gait inspired by human behavior for six-legged robot without force sensor," *Journal of Intelligent & Robotic Systems*, vol. 88, no. 1, pp. 19–35, 2017.

[19] M. Bjelonic, N. Kottege, and P. Beckerle, "Proprioceptive control of an over-actuated hexapod robot in unstructured terrain," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2042–2049, IEEE, 2016.

[20] S. P. L. M. S. Inc., "Engineer automated production systems using robotics and automation simulation," 2019. Last accessed 16 September 2019.

[21] T. Erez, Y. Tassa, and E. Todorov, "Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx," in *2015 IEEE international conference on robotics and automation (ICRA)*, pp. 4397–4404, IEEE, 2015.

[22] C. Robotics, "Coppelia robotics features," 2019. Last accessed 16 September 2019.

[23] L. Nogueira, "Comparative analysis between gazebo and v-rep robotic simulators," *Seminario Interno de Cognicao Artificial-SICA*, vol. 2014, p. 5, 2014.

[24] E. Coumans, "Exploring mlcp solvers and featherstone," in *Game Developers Conf*, pp. 17–21, 2014.

[25] B. V. Mirtich, *Impulse-based dynamic simulation of rigid body systems*. University of California, Berkeley, 1996.

[26] N. Souto, "Video game physics tutorial - part iii: Constrained rigid body simulation," 2019. Last accessed 16 September 2019.

[27] A. Boeing and T. Bräunl, "Evaluation of real-time physics simulation systems.," in *Graphite*, vol. 7, pp. 281–288, 2007.

[28] A. Rönnau, F. Sutter, G. Heppner, J. Oberländer, and R. Dillmann, "Evaluation of physics engines for robotic simulations with a special focus on the dynamics of walking robots," in *2013 16th International Conference on Advanced Robotics (ICAR)*, pp. 1–7, IEEE, 2013.

[29] ArcBotics, "Building your hexy the hexapod," 2019. Last accessed 16 September 2019.

[30] C. Robotics, "Building a clean model tutorial," 2019. Last accessed 16 September 2019.

[31] C. Robotics, "Joint types and operation," 2019. Last accessed 16 September 2019.

[32] "V-rep forum questions/answers around v-rep," 2019. Last accessed 16 September 2019.

[33] C. Robotics, "V-rep api framework," 2019. Last accessed 16 September 2019.

[34] C. Robotics, "Remote api modus operandi," 2019. Last accessed 16 September 2019.