DESIGN, MODELING, AND GAIT CONTROL OF A ROLLING QUADRUPED (ROLL-U-PED)

by

Keith E. Dockstader Jr.

A thesis submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mechanical Engineering

The University of Utah

May 2018

The University of Utah Graduate School

STATEMENT OF DISSERTATION APPROVAL

The dissertation of **Keith E. Dockstader Jr.**
has been approved by the following supervisory committee members:

Kam K. Leang ,	Chair(s)	**02/07/2018** Date Approved
Sanford G. Meek ,	Member	**02/01/2018** Date Approved
Tucker Hermans ,	Member	**02/06/2018** Date Approved

by **Tim Ameel** , Chair of
the Department of **Mechanical Engineering**
and by **David B. Kieda** , Dean of The Graduate School.

ABSTRACT

This thesis describes the design, modeling, and gait control of a new bounding/rolling quadruped robot called the roll-U-ped. The robot has four uniquely-designed compliant legs for bounding gait locomotion, and the legs can reconfigure for passive and powered rolling. One of the main advantages of such a design is versatility as the robot can efficiently and quickly traverse over flat and downhill terrain via rolling and then transition to running for traveling over more complex terrain with a bounding gait. The contributions of this work are: (1) a detailed description of the robot design, (2) modeling and simulation of bounding motion, (3) investigation of bounding gait effectiveness using sinusoidal control inputs and inputs obtained from machine learning, and (4) prototype development and performance evaluation. Specifically, the prototype robot utilizes 3D-printed compliant legs for dynamic running and rolling, and the dual-purpose leg design minimizes the number of joints. Two functional prototypes are developed with on-board embedded electronics and a single-board computer running the Robot Operating System for motion control and evaluation. Simulations of the bounding gait locomotion are shown and compared to the performance of the prototype designs. Additionally, the robot's running motion is investigated for two types of inputs: a sinusoidal trajectory and a learned gait using the Q-learning technique, where results demonstrate effective running and rolling behavior. For example, using sinusoidal inputs, the robot can run with a bounding gait over a flat and stiff sandpaper-like surface at speeds of up to 0.21 m/s. On the other hand, over a flat and tacky-cushioned surface, the speed is measured at 0.14 m/s. Simulation results for Q-learning show gait speeds of 0.22 m/s for the tacky-cushioned surface, where experiments on the physical system yielded a gait speed of 0.15 m/s. For powered rolling, the robot was able to reach a speed of 0.53 m/s over a flat-smooth surface. The results demonstrate proof-of-concept of the design and feasibility of using machine learning to determine inputs for effective running locomotion. Finally, possible future improvements to the design, modeling, and motion control of the robot are discussed.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First, I would like to thank my advisor Dr. Kam K. Leang for his support and encouragement during this whole process. I also appreciate my other two committee members, Dr. Sanford Meek and Dr. Tucker Hermans, for providing input. A big thanks goes to my labmates for their help in setting up some of the experiments. Finally, I want to thank my family for their love and support as I completed this project.

CHAPTER 1

INTRODUCTION

This thesis describes the design, modeling, and gait control of a new rolling/running quadruped robot called the roll-U-ped. The robot has four uniquely designed compliant legs for bounding gait locomotion, and the legs can reconfigure for passive and powered rolling. One of the main advantages of such a design is versatility as the robot can efficiently and quickly traverse over flat and downhill terrain via rolling and then transition to running mode for traveling over more complex terrain with a bounding gait. The running motion of the robot is modeled using the bond graph technique for locomotion studies and gait control. Through simulations and experiments on a physical prototype, two types of running are investigated: (1) through experimentally derived sinusoidal parameterization and (2) by implementing Q-learning on the model. The robot's ability to roll is also examined by: (1) implementing motion tracking for closed-loop control and (2) remote user control causing the robot's limbs to extend/retract manually. In the following sections, the motivation for this research project, research goals, and contributions are described.

1.1 Motivation

Legged animals can traverse unstructured environments with ease. Nature has implemented and revised this highly effective method of locomotion through countless iterations by evolution to the point where quadrupedal animals are now capable of traversing such complex conditions as steep mountain slopes in the case of a goat or dashing across the Savannah at speeds approaching 61 mph in the case of the cheetah [7]. Many robots have been developed that are inspired by this effectiveness but have been largely unsuccessful on a variety of fronts. Expensive quadrupedal robot designs are often largely ineffective in unstructured environments. For example, BigDog is a robot developed for the military, with the ability to move at high speeds and traverse some variations of uneven

terrain, but it is a costly design [8]. At the opposite end of the spectrum, mechanisms such as those developed by Theo Jansen are inexpensive to make but have extremely limited controllability, where only a single actuator or a uni-directional mechanical energy source such as wind [9] is employed. Additionally, many of the robots that have been developed are largely inefficient or very limited in use. By incorporating the capacity for rolling into the roll-U-ped design, it is able to handle steep slopes, roll across difficult terrain like gravel, and traverse structures that could not easily be handled in running, such as stairs. An illustration of the real-world advantages of implementing hybrid-locomotion is shown in Fig. 1.1. It shows a prior work in which a quadcopter was designed with the ability to roll to enhance efficiency and increase versatility. Designs such as this inspired the development of roll-U-ped as described in this thesis.

1.2 Goal

The goal of this project is to create a novel quadrupedal robot capable of implementing both rolling and running as methods of hybrid-locomotion with the capability to independently transition between the two modes of locomotion. Figure 1.2 illustrates the concept of the robot. The reason for implementing hybrid mobility was to encompass the advantages of rolling and running. Rolling has increased efficiency across limited terrains with advantages across rough/loose surfaces such as gravel. Running has the capability for speed and dynamic motion up slopes and across uneven terrains. These methods of locomotion are also to be accomplished using the same mechanisms so the rolling, running, and transitioning all implement the same actuators. 3D printing was examined due to it being an attractive manufacturing process because of its potential uses in the field of mobile robots, enabling designers to incorporate rapid production and testing to make robots more cost effective. As a consequence, the mechanical design should incorporate 3D printing as much as possible for design and manufacturing flexibility and to examine its potential uses in the future. In the case of mass production, Injection Molding would likely be used in the place of 3D printing.

The envisioned roll-U-ped robot should be capable of running over a variety of surfaces without the need for human intervention. Two gait patterns are to be developed for this: the first using iterative experiments to construct a parameterized angle sequence, and

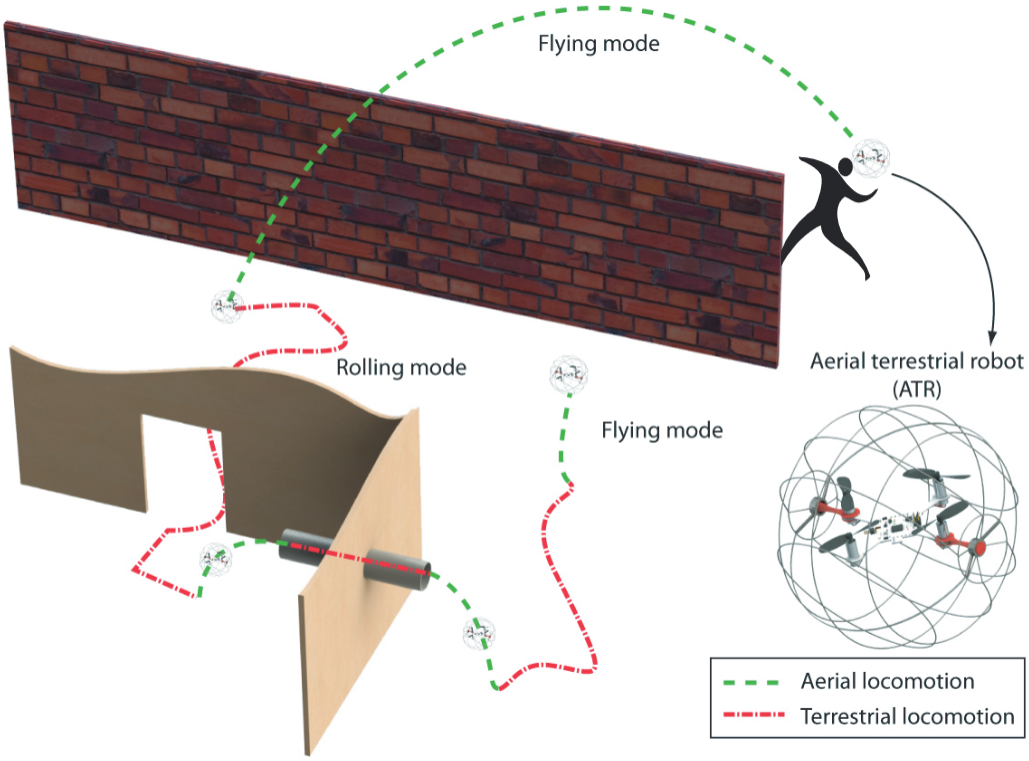

Figure 1.1. Illustration of a hybrid locomotion quadcopter traversing terrain by incorporating both rolling and flying for the purpose of most effectively handling environments. As illustrated, the rolling quadcopter can be tossed without fear of damage due to the external spherical shell, can easily traverse tight environments normally dangerous for quadcopters by implementing rolling, can fly for increased speed, and can fly to circumvent obstacles. Image used from [1] with permission ©2011 IEEE.

the second found with simulation and a gait optimization algorithm (such as machine learning). The robot should also be capable of rolling passively downhill and actively across a flat surface. Bounding/rolling controls are to be implemented using the motion tracking system VICON and used for waypoint setting of the quadruped in path tracking and locomotion control. A remote control system is also to be developed to allow for an easy human interface for locomotion control. At the end of this work, the objective is to demonstrate functional running and rolling and to describe methods for gait control that could be utilized in robot design and future extensions of the work.

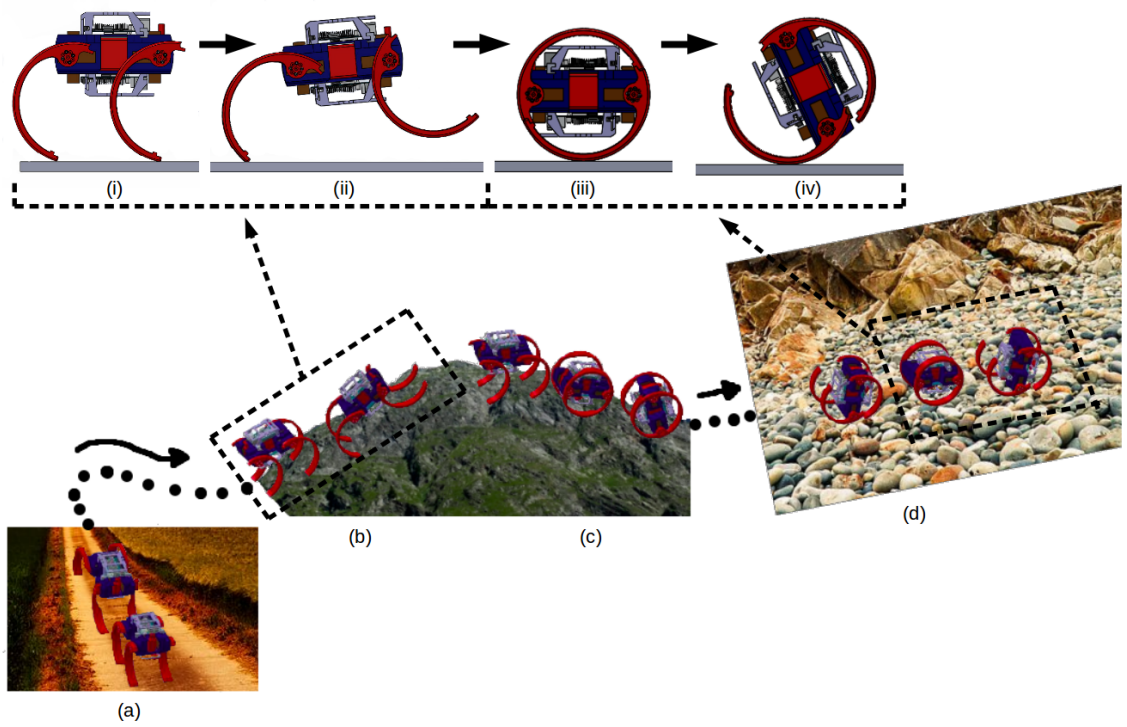

Figure 1.2. The modes of locomotion for the hybrid rolling/running quadruped, called roll-U-ped. Here roll-U-ped is shown traversing a path by implementing all of its mechanisms of locomotion as well as incorporating each stance in the process. These methods of locomotion are: (a) running along a flat surface, (b) climbing uphill, (c) passively rolling downhill, and (d) actively rolling along a flat surface. The stances are: (i) standing, (ii) running, (iii) curled, and (iv) powered rolling.

1.3 Contributions

The contributions of this thesis are: (1) a detailed description of the robot design, (2) modeling and simulation of bounding motion, (3) investigation of bounding gait effectiveness using sinusoidal control inputs and inputs obtained from machine learning, and (4) prototype development and performance evaluation. More specifically, the prototype robots utilize 3D-printed compliant legs for dynamic locomotion and to minimize the number of moving parts. A lumped-parameter model is presented and the bond graph technique is used to derive the equations of motion. Two functional prototypes are developed with on-board embedded electronics and a single-board computer running the Robot Operating System (ROS) for motion control and performance evaluation. Simulations of the bounding gait locomotion are shown and compared to the performance of the prototype designs (V1 and V2). Additionally, the robot's running motion is investigated

for two types of inputs: a sinusoidal trajectory and a learned gait using the Q-learning technique. Simulation and experimental results demonstrate effective running and rolling behavior. For example, using sinusoidal inputs, the robot can run with a bounding gait over a flat and stiff sandpaper-like surface at speeds up to 0.21 m/s. On the other hand, over a flat and tacky-cushioned surface, the speed is measured at 0.14 m/s. Simulation results for Q-learning show a gait speed of 0.22 m/s for the tacky-flexible surface, whereas experiments on the physical system yielded a gait speed of 0.15 m/s. During running, the endurance of the robot using one 2200 mAh lithium-polymer battery is approximately one hour. For powered rolling, the robot was able to reach a speed of 0.53 m/s over a flat smooth surface. The results demonstrate proof-of-concept of the design and feasibility of using machine learning to determine inputs for effective running locomotion. Finally, possible future improvements in the design, modeling, and motion control of the robot are discussed.

1.4 Organization

The thesis is organized as follows. First, in Chapter 2, the relevant material on running/rolling robots is examined along with the current methods of gait and simulation development. In Chapter 3, the design of roll-U-ped is outlined and the complete physical and control structures are detailed and specifications are given. Next, the complete modeling, including the lumped-parameter model and bond graph results, are given in Chapter 4. In Chapter 5, the methods of parameterization and Q-learning gait development are outlined. Next, the results of the parameterized trajectories and Q-learning along with the control structures implemented are given in Chapter 6. Finally, conclusions and future work are provided in Chapter 7.

CHAPTER 2

BACKGROUND: STATE-OF-THE-ART OF RUNNING/ROLLING ROBOTS

Mobile robots, and mobile mechatronic systems in general, have been a very expansive field due to the extensive transportation market and the current limitations of standard wheeled systems. Various methods of locomotion have been examined and improved on. These include flying, swimming, rolling, running, and driving via wheels, among others. Combining these into hybrid-locomotion systems provides significant advantages by allowing for traversing over additional terrains, increasing speed/stability, and/or adding efficiency. An examination of the current state for rolling and running robots is given in this chapter.

2.1 Legged Robots

As can be seen in nature, legged creatures are quite capable of handling difficult terrains [10]. As a consequence, legged robots have been conceived of for a long time. Mobile quadrupedal mechanisms have gone back as far as 1870 when P. L. Chebyshev implemented a four-bar mechanism in order to immitate natural gaits. Around 1940, researchers in the UK and the US began military-sponsored projects to study the potential of legged mechanisms for use as war machines [11]. Numerical systems that could function in real time to handle the actuation and sensor signals was a big step forward in quadruped development. Advancements in computational power and speed have made it possible to have on-board systems competent at taking in multiple sensor signals with inputs from the user and handle the necessary actuator signals by implementing feedback loops, all in real time. Research over the past two decades has exploded in this area because of the advantages legged robots have over the current best of wheeled and continuous-tread vehicles. The primary advantage is the ability to traverse much more of the earth's surface as less than half of land can be handled with wheeled vehicles [8]. Legged robots have

the advantages of increased versatility, including climbing up inclines and across granular surfaces such as deserts.

Some of the most commonly implemented legged robots have been quadrupeds for a variety of reasons, many being the same reasons quadrupeds have been so successful in nature. One is their relatively easy-to-control gaits compared to bipeds due to comparative ease of balance. Another is they often require fewer actuators than hexapedal, heptapedal, and robots with increasing numbers of legs, which reduces cost and size.

Quadrupedal robots have varied from highly complex systems with many redundant degrees of freedom to passive walkers that incorporate no actuators but are powered by external stimuli. Legged robots in general, however, tend to be of high complexity and cost. In addition, they require either gait planning of some kind or must be designed to allow the system to handle the gait actively or passively, as is the case with passive walkers.

There have been many studies specifically examining the design of various elements of a quadruped's base and legs. These include studies such as [3, 12–14]. These were examined for use in designing roll-U-ped. However, particular methods for designing the ratio of sizes to the leg and base length in studies, such as [3], were not implemented in roll-U-ped due to the design limitations based upon the rolling and running requirements. Several studies have also been developed examining the design of legs for directional compliance such as those done in Dr. Sanford Meek's lab at the University of Utah in [15].

Many of the current quadruped designs have an advantage in robust response to disturbances, extreme speeds, durability, etc. One of the biggest disadvantages of many of these designs is their runtime. Often these robots are only capable of running on the order of a half hour or require an external power source. The robot roll-U-ped was developed with hybrid mobility to levy many of the advantages of these running quadruped designs with an increased operating time by incorporating the capability of rolling. Specific examples of legged robots are examined below.

2.1.1 Robots

RHex is a hexapedal robot with six independently actuated flexible circular legs. It moves by continuously rotating the legs around in a method similar to those used for whegs or wheel/legs. This avoids many of the difficulties of running such as stubbing toes

and is passively stable despite the fast motion. However, as with standard gaits, a significant amount of leg motion is performed not propelling the leg forward, but rather rotating the leg in space above. An additional benefit is its ability to function forwards/backwards as well as upside down/right-side up. An illustration of XRHex, a variation on the RHex design, is given in Fig. 2.1. [2]

Scout II is a robot with four independently actuated flexible linear legs. It moves using a rotary gallop that was constructed using a leg state machine where the condition of the leg is divided into the states of flight, stance retraction, and stance break. This is illustrated in Fig. 2.2. The conditions for each transition were ground contact, leg contraction, and flight [3]. By implementing this, Scout II was able to run along uniform flat ground effectively for a cost of $\$25,000$ and at speeds approaching 1.3 m/s [16].

Ken is another quadrupedal robot, but, unlike Scout II, it incorporates twelve total muscles, with an additional degree of freedom (DOF) in a flexible joint on the base. Each leg has three actuators: (1) proximal joint's flexion/extension, (2) distal joint's flexion/extension, and (3) leg contraction/extension. It was able to be run stably using a Central Pattern Generator (CPG) with open-loop control [12, 17].

Big Dog is a quadrupedal robot approximately 3 ft in length developed by Boston Dynamics with funding from DARPA. It implements compliant mechanisms to absorb and redirect energy for increased efficiency, which is particularly important due to the purpose of functioning as a mechanism for carrying up to 240 lbs for soldiers across uneven terrain. The sensors implemented include those measuring ground contact force, base linear/rotational acceleration, and joint position/torque. Additionally, several sensors are used in monitoring the onboard mechatronic systems. A LIDAR and camera system are also used for mapping of the environment [8].

The actuation system is composed of an engine powered by gasoline driving a pump. The legs are actuated using hydraulic cylinders levered to generate high torque. Hydraulics are useful largely due to their high strength-to-weight ratio. This makes hydraulics a great option for motion generation in mobile robots in platforms large enough to contain the complete pump, hydraulic cylinder, and energy generation system, such as an engine, on board [18].

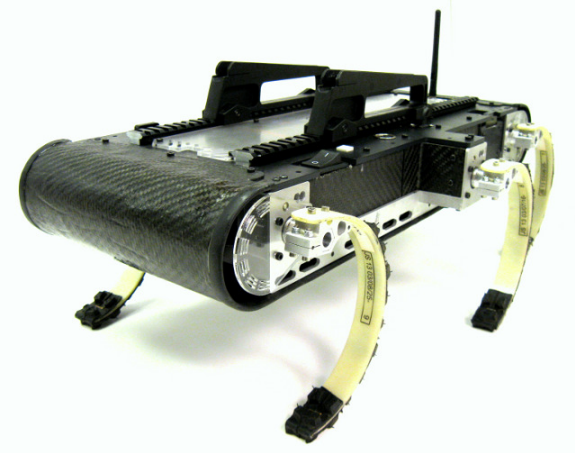

Figure 2.1. XRHex, an improved version of RHex. It is a hexapod designed for use as a research platform. Image used from [2] with permission.

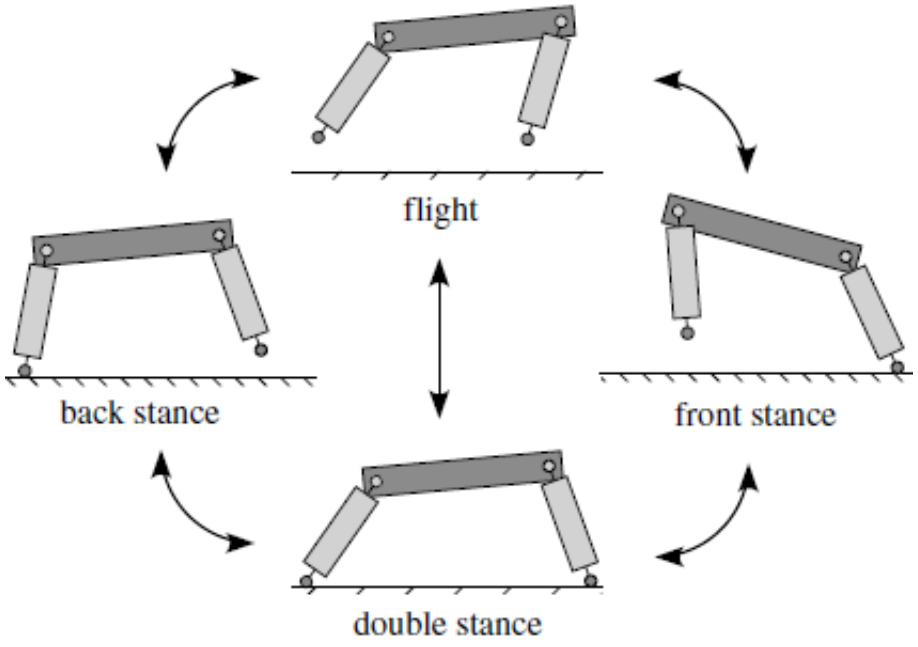

Figure 2.2. Scout II gait illustrated. Each stance has its own conditions and response. As shown, there are four states: (1) back stance, (2) flight, (3) front stance, and (4) double stance. Image used with permission from [3].

2.2 Rolling Robots

Rolling robots have been another field of interest in mobile robot locomotion. Uses for rolling robots include entertainment, surveillance, traversing pipes, and handling extremely harsh environments such as extraterrestrial surfaces where all components must be enclosed.

There are several methods of producing powered rolling. One such method is using a flexible structure that bends around to generate a torque that propels the robot forward. An example of this is the robot GoQBot, which is modeled after a rolling caterpillar. It is a soft robot that propels itself by somersaulting forward into a spin. This process is examined in Fig. 2.3. Alternatively, the rotation can be produced by pushing out with attached limbs. This was the method implemented on roll-U-ped because incorporating additional actuators was undesirable.

There are many different actuation methods for generating the controlled roll of a sphere. These include wheel-based, two independent hemispheres, pendulum-based, relocation of center of gravity, wind powered, and deforming. Wheel-based incorporates motors to drive the sphere rotation for generating the desired motion. Two independent hemispheres is a robot design with independently moving hemispheres. By this, it can turn as well as move forward and back. In the case of relocation of COM, a mass is moved from inside the sphere around to generate a rotation defined by a vector from the contact point of the sphere with the ground to the COM and projected onto the ground (this is for the case of the robot being on flat ground). Deforming robots move by altering their body shape to one that is unstable with its COM out from above the section of the robot in contact with the ground. This generates continuous rolling if the robot is properly actuated [19].

2.3 Hybrid Running/Rolling Robots

Several hybrid-mobility running/rolling robots have been developed because of the advantage of rolling and running in being better able to handle different environments. For example, often when moving up a slope, running will be more effective, and when moving downhill, rolling will be most efficient.

Hex-A-Ball (HAB) is a spherical robot that extends out into a hexapod. It implements hybrid-mobility in both rolling and running locomotion. To roll, HAB curls into a ball and

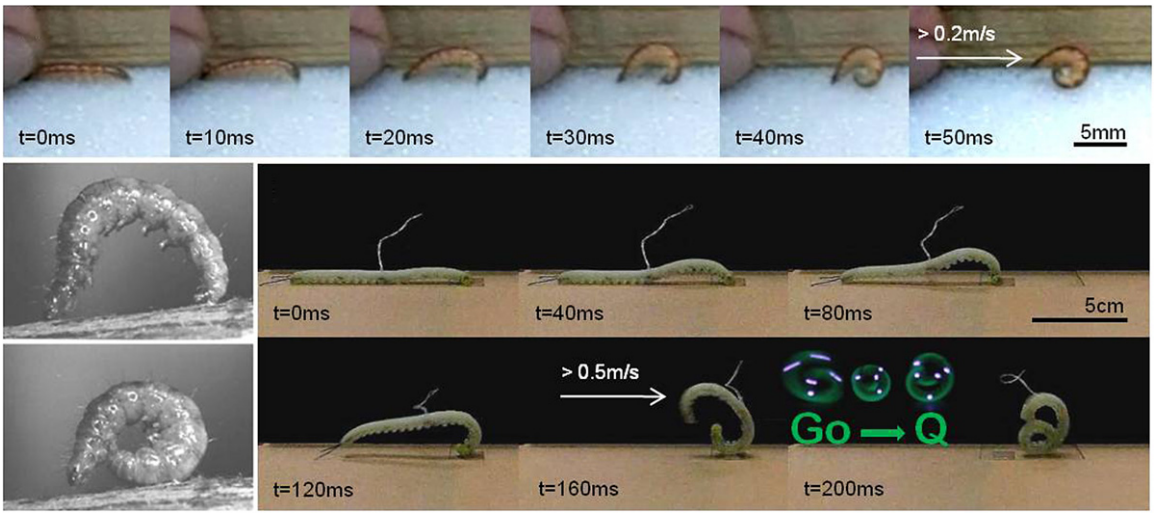

Figure 2.3. GoQBot is a mobile robot designed after the propelled rolling some caterpillars implement for rapid locomotion. Image used with permission from [4].

extends its legs out into a propelled roll. To translate, it implements a static hexapedal gait.

Scorpio is a hybrid locomotion running/rolling quadruped. A diagram of it transforming is given in Fig. 2.4. It was inspired by the huntsman spider, which performs rolling in a way similar to a cartwheel. The legs are set up in the spider layout (SL) shown in Fig. 2.5. Each leg contains three motors for each rotation in roll, pitch, and yaw of the leg. It implements a static quadrupedal run corresponding to SL. The rolling was caused by rotating the limbs back on itself to form a cylinder and propelling itself by extending the back legs out to push against the ground and continue rotating the robot to direct it forward. Scorpio was of particular inspiration in designing roll-U-ped as it implements the same mechanism of rolling [5].

QRoSS is a quadruped with a spherical enclosure. An illustration is given in Fig. 2.6. The shell is composed of thin wire spaced such that the legs can extend out and fold back into the enclosure. The robot is capable of running by using a passively stable quadrupedal gait, or walk [6]. The rolling is accomplished by folding two legs back into the enclosure and extending the other two out to push against the ground at the back of the robot, propelling it forward.

2.4 Modeling Dynamic Robots

Modeling dynamically moving robots is a complex task for which the method chosen is largely dependent on the conditions. There are several available general-purpose simulators such as Gazebo and VRep that allow you to make virtual robots constructed of stiff elements and rotational/linear actuators. Various quadrupeds have also implemented commercial software such as Working Model 2D in the case of Scout II [3]. However, the accuracy of such systems is highly dependent on the aspects of the robot dynamics incorporated in the running. Additionally, they are largely incapable of handling flexibility. As such, custom-made environments are often chosen in the case of modeling dynamic running. Matlab allows for 3D modeling with direct control of all of the system parameters to allow you to incorporate custom elements such as flexible legs. This is often the method of choice to construct a simulation for a complex robot.

The same result can be accomplished by developing the differential equations modeling a robot. A differential equation solver is then used to solve the differential equations

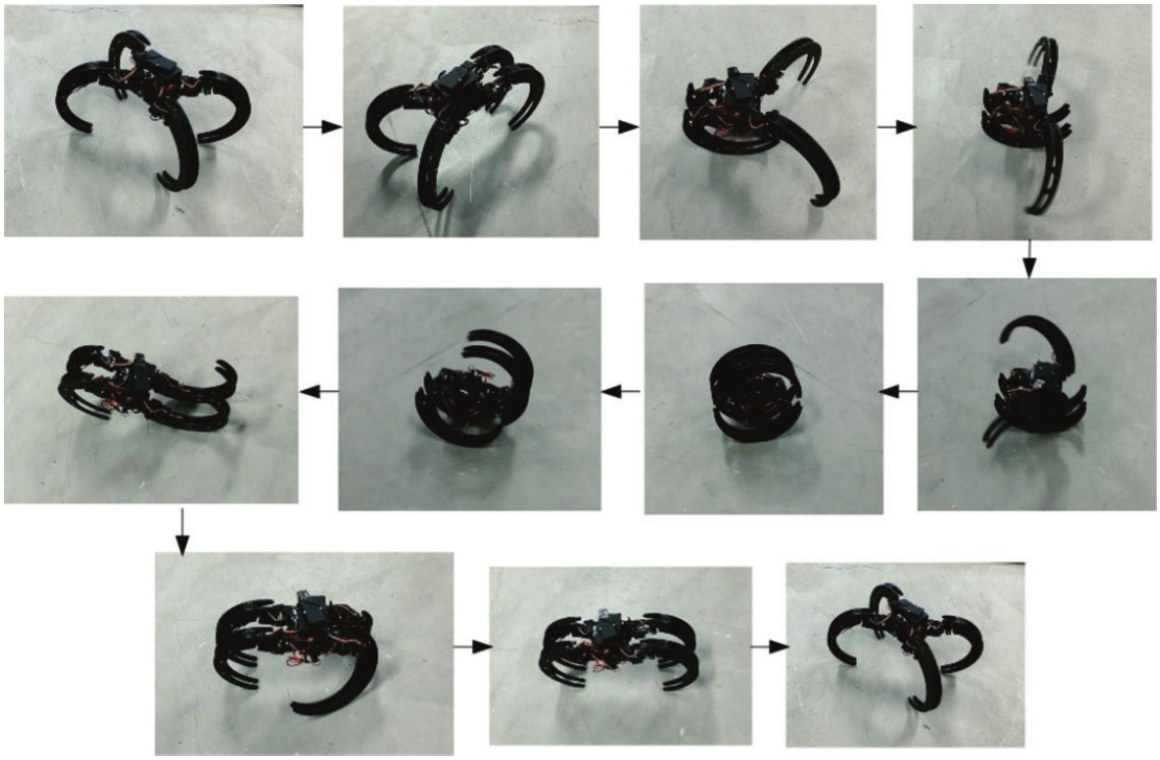

Figure 2.4. Image of the crawling/rolling robot Scorpio with the steps of transition from crawling to rolling and back. The roll-U-ped incorporates the same method of rolling as Scorpio. Image used with permission from [5].

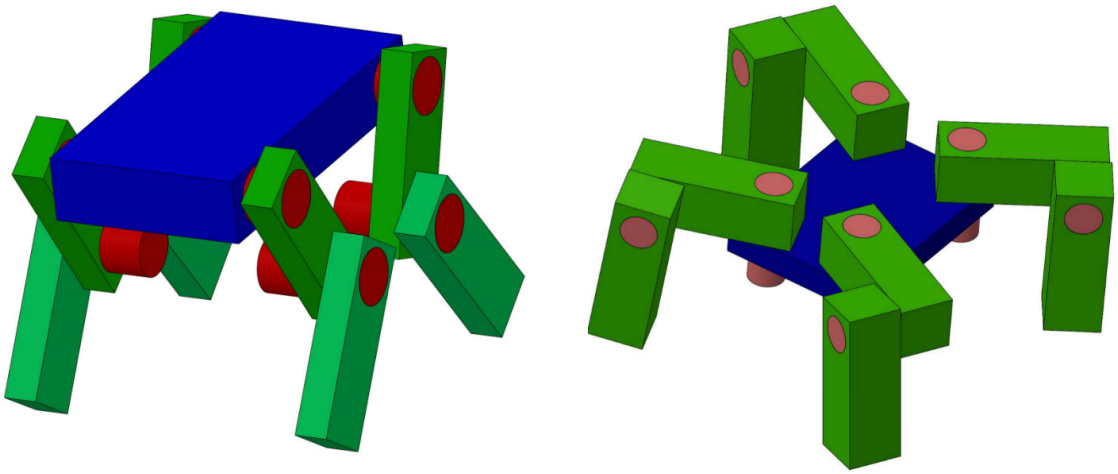

Figure 2.5. The two general structures of quadruped robots. To the left is the mammal layout (ML) that was implemented in roll-U-ped. To the right is the spider layout (SL) that is an alternative to ML.

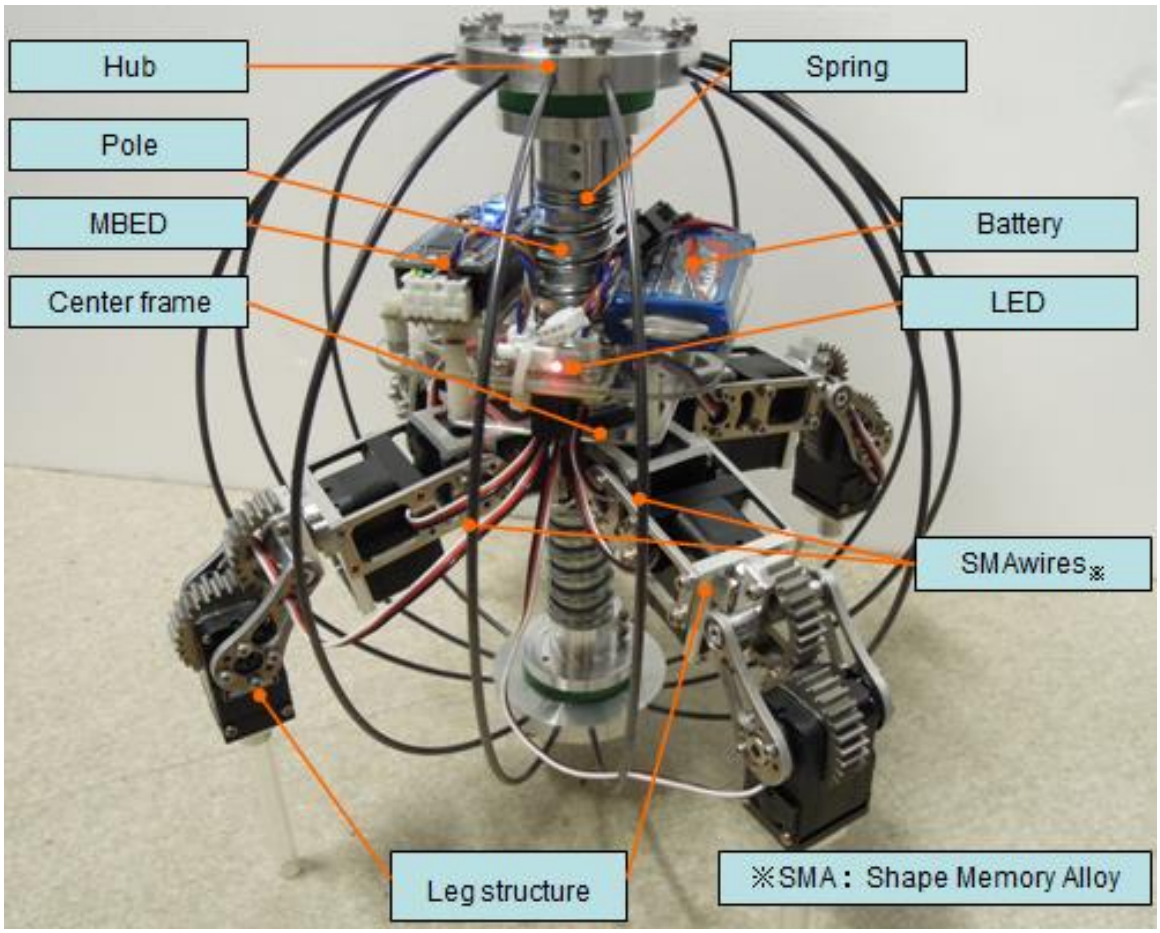

Figure 2.6. Image of the quadruped robot QRoSS standing with components listed. As is visible, the legs can be folded in toward the center frame to form a sphere and roll. Image used with permission from [6] © 2011 IEEE.

across the desired time. This is done by breaking down the system into basic elements such as capacitances, inductances, and resistances. The states of the system can then be resolved over time by generating differential equations from the elements and solving them. Methods such as bond graphs can derive the differential equations. This was the process used in this thesis.

2.5 Quadruped Gait Development

The development of quadruped gaits is a very complex task due to the large task space. A standard method implemented is using inverse-kinematics when using passively stable gaits such as crawling. This is done by controlling the trajectory of each leg so, across time, three to four legs are in contact with the ground with the COM above that supported region. The same effect can be produced using feet to provide the area of support. This method becomes useless, however, once the robot begins to move dynamically as the COM is not located in the supported region. As a consequence, this could not be implemented on roll-U-ped.

Dynamic gait development adds an additional complexity because the robot is not balanced throughout its motion. These are much more difficult to control, particularly in open-loop. Several different mechanisms of developing gaits are commonly used. However, they often require an informed user or a learning algorithm to experimentally find the gait parameters, such as the amplitude and frequency in a sinusoidal trajectory. For the case of someone experienced generating the gaits, an open-loop gait trajectory can be developed with varied effectiveness depending on the robot. This is useful primarily because it incorporates the informed perspective of the developer. However, it is very time consuming and is limited to open-loop and the set of parameterized trajectories examined, which is not guaranteed to have any level of effectiveness. A parameterized sinusoidal trajectory was developed in this thesis through experimental testing by an informed user with success.

Central Pattern Generators (CPGs) are another mechanism of gait development. They are based upon the general structure of the same neurons that animals use to generate their gaits. Implementing these allows for the capablity for developing rhythmic outputs without rhythmic inputs [20]. This, however, again requires an informed choice of the type

of CPG model and the number of oscillators to implement, so this method was not chosen.

Evolutionary and learning algorithms are other methods used to produce running quadruped gaits. The primary disadvantage of these is they usually require a high number of runs to get useful results. However, they have the advantage of not requiring a human to inform the design process past the original structure, which makes them perfect for use in simulation as the cost of high numbers of iterations is low. Q-learning was implemented in this thesis as a type of learning algorithm that, once developed, did not require an informed user for finding effective constants or redesigning the controller.

2.6 Summary

Much has been done in the field of hybrid locomotion due to the advantages in the variety of terrains it is capable of traversing, speed capabilities, and added efficiency. Running is very useful as a method of traversing terrains outside of the stiff smooth road surfaces used for wheeled vehicles. The primary disadvantage is the lower translational speed compared to wheeled systems. Rolling robots also have the advantage of added robustness due to resistance to being knocked over resulting in unusability. The robot roll-U-ped is designed to bring together many of these advantages by being able to run and roll in order to have the advantage of traversing terrains difficult for legged robots. These include loose gravel, down stairs, and down steep slopes. Quadruped robot capabilities include climbing inclines, stepping over small obstacles, and the novelty of use. A primary limitation of robots such as these is the upfront cost as seen in the case of robots like those developed by Boston Dynamics.

The advantage of such a system over the current running/rolling design is the scalability from the actuator layout chosen and its ability to run faster than many of the robots implementing static gaits like Spyder and Hex-A-Ball. The addition of an effective model of roll-U-ped running over a flexible tacky surface with automated running learning also allows for future alterations to design to improve results, largely due to the dynamic gait.

CHAPTER 3

DESIGN

Quadruped design incorporates the layout, base, leg components, and control system. The layout and components of the leg have particular influence on the ability of the robot to function while running and rolling. Various designs limit the gaits that can be used and so restrict the speed and surfaces to be traversed. The design of the base is primarily to house the necessary components, maintain structural integrity, and avoid restricting the necessary range of motion of the legs. The control system incorporates all the on-board computation needed for the robot to function. This chapter describes the design process to create the roll-U-ped robot. Below, two designs were created, design version 1 (V1) and design version 2 (V2).

3.1 Task Clarification and Specifications

The first step in roll-U+-ped design was formalizing the task and determining specifications for the robot. These restrict the workspace to something manageable while at the same time extracting the valuable aspects to be incorporated in the design.

3.1.1 Formulation

The reason for designing roll-U-ped was to develop a novel quadruped capable of bounding and powered rolling. Diagrams of the complete final designs of V1 and V2 are given in Fig. 3.1 and Fig. 3.2. Illustrations of roll-U-ped running and rolling are given in Figs. 3.3 and 3.4, respectively. This incorporates its lack of dependence on outside sources for power or actuation. Some off-board control could be implemented when using motion tracking, but other than that, roll-U-ped should be entirely self-reliant.

The primary uses of such a design are: entertainment, as a research platform, exploration, and search-and-rescue. As a consequence, the cost of the robot should be low, the reliability high, and the robot should not need an experienced user to be able to operate it.

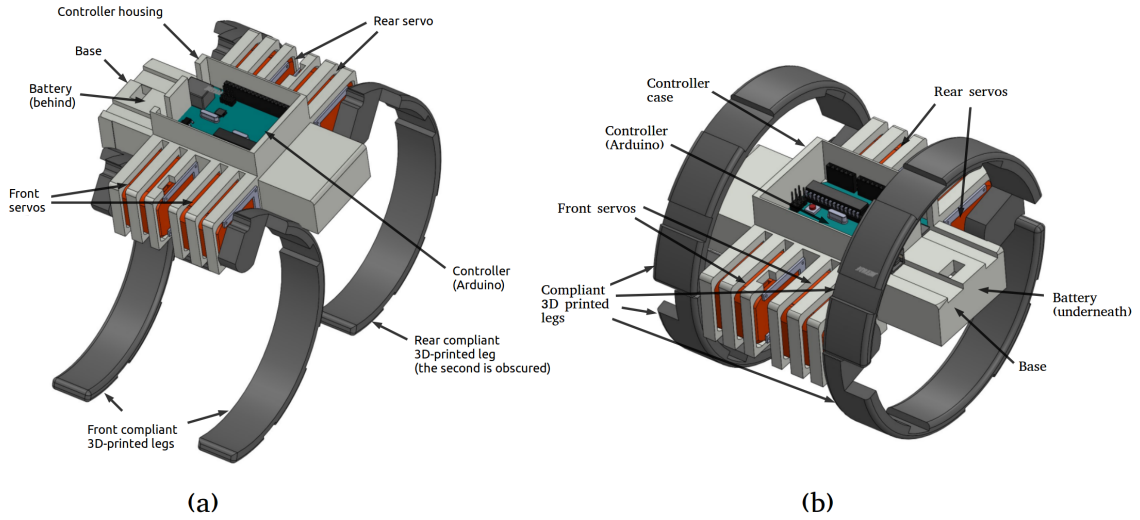

Figure 3.1. The original prototype of roll-U-ped, V1: (a) given from an orthogonal view with legs extended down and (b) given from an orthogonal view with legs curled in. The base and legs are both 3D printed. The remaining parts are commercial products.

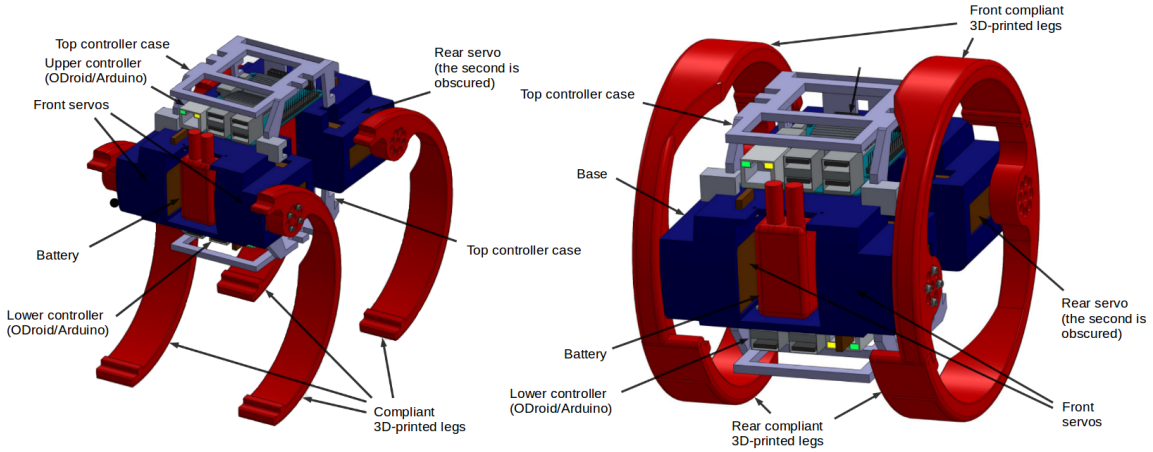

Figure 3.2. Prototype V2 of roll-U-ped, (a) given from an orthogonal view with legs extended down and (b) given from an orthogonal view with legs curled in. The base and legs are both 3D printed. The remaining parts are commercial products. The only alterations made to this design were replacing the ODroid container with another and the placement of several motion capture balls extended from the surface for motion tracking with VICON.

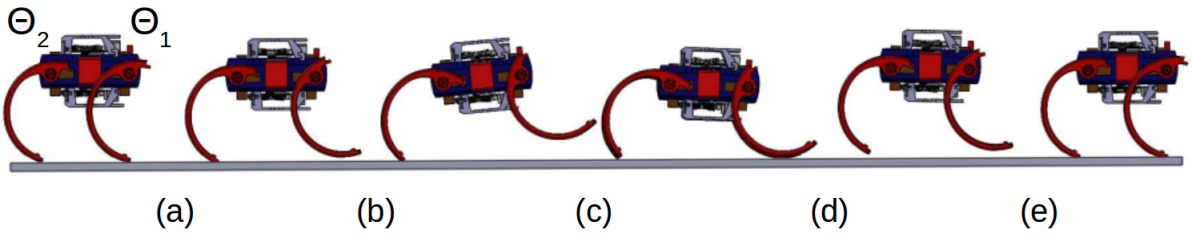

Figure 3.3. The bounding gait steps used in running roll-U-ped. The conditions are as follows: stand up, (a) begin extending both legs out away from the base to rotate the robot up some θ_a, (b) reach top of the trajectory and begin to rotate some $-\theta_b$ down, (c) hit the ground with the front legs, (d) begin to pull the legs in toward the center while the robot is propelled up into the air from contact with the ground, (e) reach a standing state again and repeat the process.

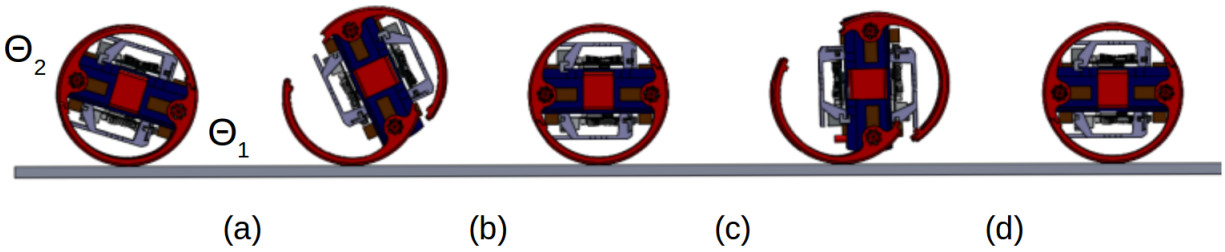

Figure 3.4. The steps in a rolling trajectory on roll-U-ped in the forward direction. The conditions are as follows: initially curl into a cylinder, (a) once the robot has reached -90 deg. $< \theta < 0$ deg. extend out the front legs to propel the robot forward, (b) fold the legs back in to form a cylinder, (c) once the robot has reached 90 deg. $< \theta < 180$ deg. extend the rear legs out to again roll the robot forward, (d) curl into a cylinder, which brings it back to the first step to repeat.

For increased reliability, the design should be made as modular as possible and allow for the addition of parts for motion capture, computation, etc., as necessary.

3.1.2 Specifications

A complete summary of the specifications in roll-U-ped design are given in Table 3.1. The specifications are designated as relative importance: 3 is absolutely necessary, 2 is very desirable but not essential, and 1 is beneficial but not necessary. These were developed from the task formulation and used in roll-U-ped design decisions.

3.2 Conceptual Design

The robot actuator layout was a fundamental decision made in developing roll-U-ped. There are several different configurations of quadruped actuators with the primary distinctions being mammal layout (ML) and spider layout (SL). Illustrations of these layouts are given in Fig. 3.5. There are variations to the mammal layout in which the knees and elbows are oriented different than the mammal design. The spider layout additionally has vaiations with the inclusion of more actuators of varied orientation. The advantages of the mammal layout allow for lower power consumption for body support [21] and fewer minimum actuators required. The primary advantage of SL is the lower COM.

A quadruped is in general composed of four legs and a single base. Due to the actuator layout chosen, two general sub-structures are capable of producing both rolling and running. These are illustrated in Fig. 3.5. The difference between each design is how the cylinder for rolling is formed. Design A incorporates the base into the curve whereas Design B folds the back legs up onto the base so the base is enclosed by the legs. Design A is a single curved base with two curved legs attached outside of each end of the base. The four legs form a cylinder from which the legs can extend to propel itself forward. The second structure is a single straight base with two curved legs attached outside of each end of the base. Both the front and back legs fold back against the base.

Both designs allow for customizable lengths of legs. The COM of each design can additionally be centered by adding or moving mass toward the center of the design. Finally, designs A and B both allow for variable numbers of actuators on each leg, different leg curve directions, and can accommodate the size range desired. Design B was chosen due

Table 3.1. A collection of the specifications outlining the design decisions of roll-U-ped. They are distinguised by the relative importance of: 3 being absolutely necessary, 2 being very desirable but not essential, and 1 being beneficial but not necessary.

Type	Element	Necessity
Rolling specs.	Capacity for a rolling mechanism of locomotion	3
	Robot can actively roll along a flat surface at 0.4 m/s	2
	Robot can passively roll downhill	3
	Robot can passively roll down 10 deg.	1
Running specs.	Capability for dynamic running	3
	Robot can run at 0.4 m/s	2
Design specs.	Minimalistic design	3
	Total robot cost $<$ \$300.00	2
	The structure and legs are 3D printed	2
	Size $<$ 12 inches in breadth	2
	Size $>$ 5 inches in breadth	2
	Total mass $<$ 2 kg total	1
Functional specs.	The robot can function running for 15 min	3
	The robot can function running for an hour	2
	Functions independent of external support	3
	Someone inexperienced with the robot can use it	2
	Replacement of battery and legs without disassembly	2

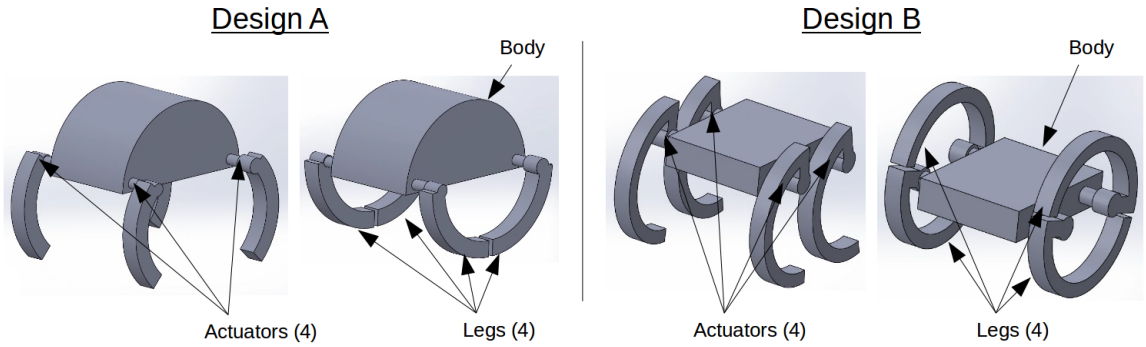

Figure 3.5. The two potential general designs of roll-U-ped conforming to the mammal layout (ML). Design A is structured with both legs facing inward and transforms into rolling by pulling both legs in to have the ends meet. Design B is structured with both legs having the inside curve facing backwards. It transforms into rolling by pulling the front and back legs into the base and kicking the legs out to spin the robot forward.

to the front and back legs being of equal length for symmetry in rolling in addition to minimizing the arc length the base composes of the cylinder formed when curled.

3.2.1 Actuation

Many kinds of actuators from hydraulic cylinders to high-torque direct-drive motors have been implemented in quadrupeds. Hydraulic and pneumatic actuators have a higher force-to-weight ratio than electric motors. However, they require full on-board tank(s) or engine(s) for pressure generation, which makes them ineffective at lower sizes. Additionally, hydraulic and pneumatic actuators are limited in their range of motion whereas electric motors are capable of continuous rotation, with some limits on systems like servos. Due to the previously outlined design requirements of roll-U-ped, the joints need to be able to handle high torque throughout their range of motion, at 90 deg. with a rotation of at least 270 deg. for the back legs to fold up. Incorporating pulleys and other mechanisms of altering the range of motion were examined, but due to the motion requirements, geared servos directly controlling the rotation were chosen. They allow for accurate positional control of the legs and with the high torque necessary. The specific servo was found by iterating through the torque-speed curves of the servos and the approximate weight of the base and legs expected to determine if the servo could rotate with enough torque and speed as necessary. The possible forward speed was approximated with the equation:

$$L\omega = v, \tag{3.1}$$

where L is the length of the leg, ω is the rotational velocity of the leg, and v is the approximated linear forward velocity of the robot.

Robotic servos were used in roll-U-ped as the mechanism of actuation. The exact specs and name of the servo chosen are given in Table 3.2. The primary reasons for this are: (1) they can produce high torque and speed, (2) they do not require large controllers, and (3) they can handle the range of motion necessary. The high torque and speed requirements are important in order to produce the bounding gait to be implemented in roll-U-ped as well as to rotate around fast enough to transition between bounding and rolling.

The compactness of control was necessary because it allowed for the scale and durability of design that is desirable. The range of motion was necessary due to the direct-drive

Table 3.2. RoboStar 7.4 high voltage Turnigy servo specifications.

Response for 12V	Servo units	SI units
Stall Torque	53.1 kg-cm	5.20911 N-m
No-load Speed	(0.14 sec) / (60 deg)	7.47998 rad/s

setup implemented, as determined above. Robotic servos are also quite low cost relative to the alternatives with easily implemented on-board power sources of LiPo batteries.

Quadrupedal gaits can be accomplished with as few as zero actuators across the complete structure, in the case of passive walkers, to more than half a dozen per leg. The actuator is one of the primary cost decisions in designing robots, particularly as the number implemented increases. Additionally, implementing increasing numbers of actuators requires increasing torque requirements as an additional torque of approximately $Lm_{actuator}g$ is experienced by the top actuator on the arm/leg. That is defined as L being the distance from the shoulder/hip joint, $m_{actuator}$ being the mass of the actuator, and g being the acceleration due to gravity (9.81 m/s^2). By examining available actuators, the decision was made to implement only a single actuator per leg. The final specs. for the servo chosen are given in Table 3.2.

The same result could be accomplished with three actuators by making either the front or back legs be controlled with only a single actuator. However, this was not done as it would reduce the control over turning particularly during rolling.

3.2.2 Base Design

The material chosen to construct the base was ABS plastic to be 3D printed to allow for customizability, lightweight, and rapid production. One of the primary goals behind roll-U-ped was to examine the uses of 3D printing in the field of mobile running/rolling robots. As a consequence, the base was designed to be 3D printed.

3.2.3 Leg Design

Quadruped robots with no flexure in their legs can function, even with only a single actuator per leg. However, adding flexure allows it to absorb impacts to reduce the maximum torque experienced by the actuators. As such, it was desirable to make the leg flexible while remaining strong enough to handle the impulse of hitting the ground hundreds of times. With these requirements, a stiff leg could be used with a linear/rotational joint incorporating a spring, or the leg could be constructed from a flexible metal or plastic. Again, because the legs could be 3D printed, they were, as this was desirable in outlined specifications. This provides the advantages of customizability and rapid production along with the general examination of the uses of 3D printing in mobile legged robots and

the use of the material flexure in design construction. The primary disadvantage of using 3D printing was the complexity of the stiffness and yield strength. As a consequence, the stiffness of each leg was experimentally found.

3.3 Control System

The required control system is of low complexity due to the use of servos as actuators. An Arduino was used to interface with the servos and run the open-loop sinusoidal trajectories. An ODroid was added to allow for interface with a remote control and VICON by implementing ROS and SSH. Details for V1 and V2 are given in the following subsections.

3.3.1 V1 Design

For V1 design, only an open-loop control system with sinusoidal inputs was implemented. Sinusoidal was chosen in place of another trajectory such as triangular or rectangular to avoid the high accelerations and smooth out the trajectory. The joint trajectories were experimentally varied sinusoidal trajectories given in the following equations:

$$\theta_1(t) = A_1 cos(f * (t + p_1)) + b_1 \tag{3.2}$$

where θ_1 is the current angle of the back legs, t is the time[s], and A_1, f_1, p_1, and b_1 are the respective parameters of the sinusoid.

$$\theta_2(t) = A_2 cos(f_2 * (t + p_2)) + b_2 \tag{3.3}$$

where θ_2 is the current angle of the front legs, t is the time[s], and A_2, f_2, p_2, and b_2 are the respective parameters of the sinusoid.

3.3.2 V2 Design

For V2 design, two control methods were developed: (1) using remote control and (2) using the motion capture system VICON. Both control systems incorporate an amplitude compensation and mechanism of locomotion control. Amplitude compensation is performed by altering the equations to the following:

$$\theta_{1L}(t) = (A_1 + A_L) cos(f * (t + p_1)) + b_1 \tag{3.4}$$

where θ_{1L} is the current angle of the back left leg, t is the time[s], and A_1, f_1, p_1, and b_1 are the respective parameters of the sinusoid. A_L is the amplitude compensation for the left side used to turn the robot.

$$\theta_{1R}(t) = (A_1 + A_R)cos(f * (t + p_1)) + b_1 \tag{3.5}$$

where θ_{1R} is the current angle of the back right leg, t is the time[s], and A_1, f_1, p_1, and b_1 are the respective parameters of the sinusoid. A_R is the amplitude compensation for the right side used to turn the robot.

$$\theta_{2L}(t) = (A_2 + A_L)cos(f_2 * (t + p_2)) + b_2 \tag{3.6}$$

where θ_{2L} is the current angle of the front left leg, t is the time[s], and A_2, f_2, p_2, and b_2 are the respective parameters of the sinusoid. A_L is the amplitude compensation for the left side used to turn the robot.

$$\theta_{2R}(t) = (A_2 + A_R)cos(f_2 * (t + p_2)) + b_2 \tag{3.7}$$

where θ_{2R} is the current angle of the front right leg, t is the time[s], and A_2, f_2, p_2, and b_2 are the respective parameters of the sinusoid. A_R is the amplitude compensation for the right side used to turn the robot.

The mechatronic system structure is illustrated in Fig. 3.6. With both remote control and motion tracking, a signal of (1) locomotion method, (2) locomotion method specifier (e.g., forwards/backwards), (3) amplitude, and (4) amplitude change were all sent from VICON/remote control over ROS to the ODroid and sent over Serial to the Arduino. The locomotion method was separated into (a) running, (b) rolling, (c) transition from running to rolling, and (d) transition from rolling to running. The locomotion method specifier was separated into forward/backwards for running and extending front/extending back/remaining stationary for rolling. No values were needed for second locomotion method during transformations to/from rolling from/to bounding. The complete logic implemented on the Arduino is given in Fig. 3.7.

3.3.2.1 Remote Control

Remote control via an XBox controller was incorporated as one interface for controlling roll-U-ped. This allowed for direct control over roll-U-ped's stance, amplitude compensa-

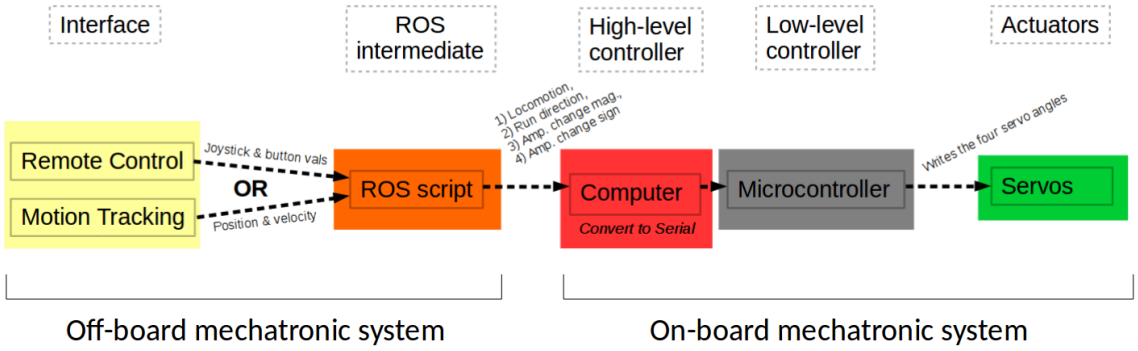

Figure 3.6. The mechatronic system structure of roll-U-ped. It is split into the interface (motion tracking with VICON or remote control with an XBox controller), ROS intermediate, high-level controller, low-level controller, and actuators. The structure is segmented to the off-board and on-board mechatronic systems. The off-board components are all located off of the robot. The on-board components are all located on the robot.

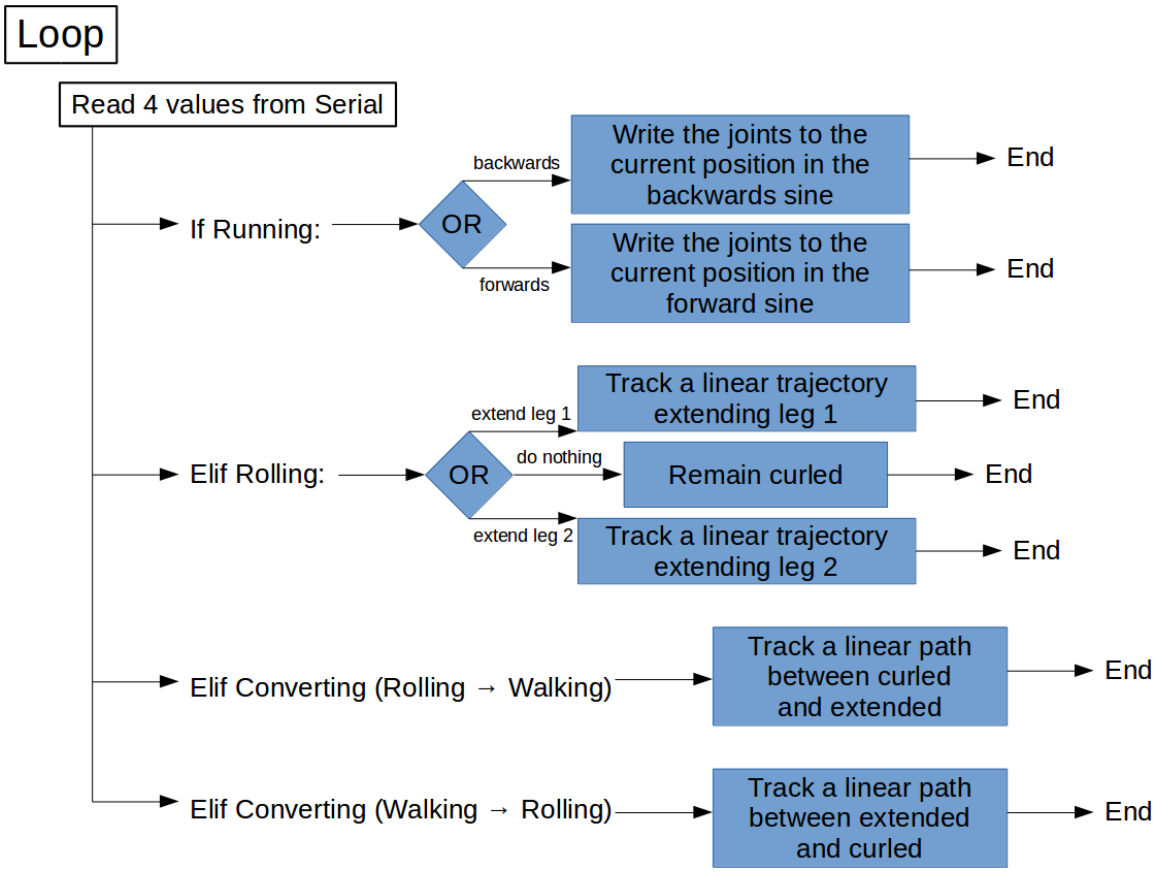

Figure 3.7. The logic implemented on Arduino to direct the actuation of each servo. The values read from serial are the ones sent to ODroid: (1) locomotion, (2) locomotion specifier (direction of motion), (3) amplitude change magnitude, (4) amplitude change sign. In order to do so, a lookup table was used in the ROS script to convert to a character to send over Serial and convert back in Arduino to the desired value.

tion, and rotational leg extension. The corresponding controller setup is given in Fig. 3.8. The control structure drove the servos to the front left, front right, back left, and back right angles given in the equations for the leg angles of V2 design.

In order to do so, the desired method of locomotion and the amplitude compensation were received from the remote and transferred to the robot. The complete process is: (1) receive the values for the robot locomotion from the remote, (2) read those values into the python ROS script, (3) publish over ROS the desired transmission to the ODroid, and (4) transmit from the ODroid over the serial the signal for the Arduino to read and convert to the current angle for each leg. In this case, the robot locomotion values are: (a) amplitude compensation from the Left-Right value on the joystick, (b) X, Y, A, and B buttons used for the method of locomotion, (c) LT and RT buttons used for expelling the front and back legs, respectively, during rolling.

3.3.2.2 Motion Tracking Control

Motion tracking control via VICON is the second control system developed for roll-U-ped. For this, a series of waypoints were implemented with a given method of locomotion (bounding or rolling) for each step of the path with interjections of transitions as needed. These were automatically handled with ROS and the VICON system. The same signal as the remote control was implemented, except the input was the position and orientation of the robot in space, which was then sent to the ROS script to extract the desired amplitude compensation to keep it moving along the path. A diagram of the features used in waypoint following is given in Fig. 3.9. A PID controller was used to track the path that incorporated the angle and distance from the path. The compensatory amplitude used to turn the robot to track along a path is given by the following equation:

$$A_{comp} = (D_{angle}\dot{\theta} + P_{angle}\theta + I_{angle}\int \theta) + (D_{dist}\dot{L} + P_{dist}L + I_{dist}\int L) \tag{3.8}$$

where, P_{angle}, I_{angle}, D_{angle} are the respective proportional, integral, and derivative constants for the θ term, P_{angle}, I_{angle}, D_{angle} are the respective proportional, integral, and derivative constants for the L term. The terms θ and L are defined in Fig. 3.9.

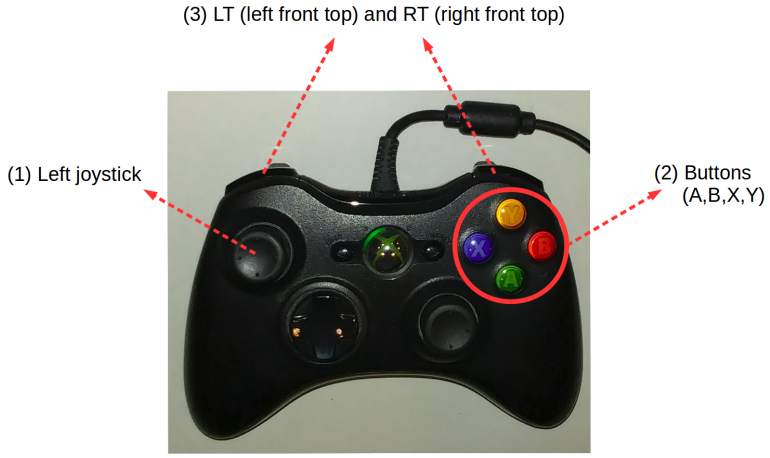

Figure 3.8. The button layout was customized to allow for locomotion and amplitude control. The joystick (1) is used to get the amplitude from the y-value and the amplitude compensation from the x-value. The A, B, X, and Y buttons (2) are used to switch between rolling and running. The RT and LT buttons (3) are used in expelling the front and back legs, respectively, when rolling.

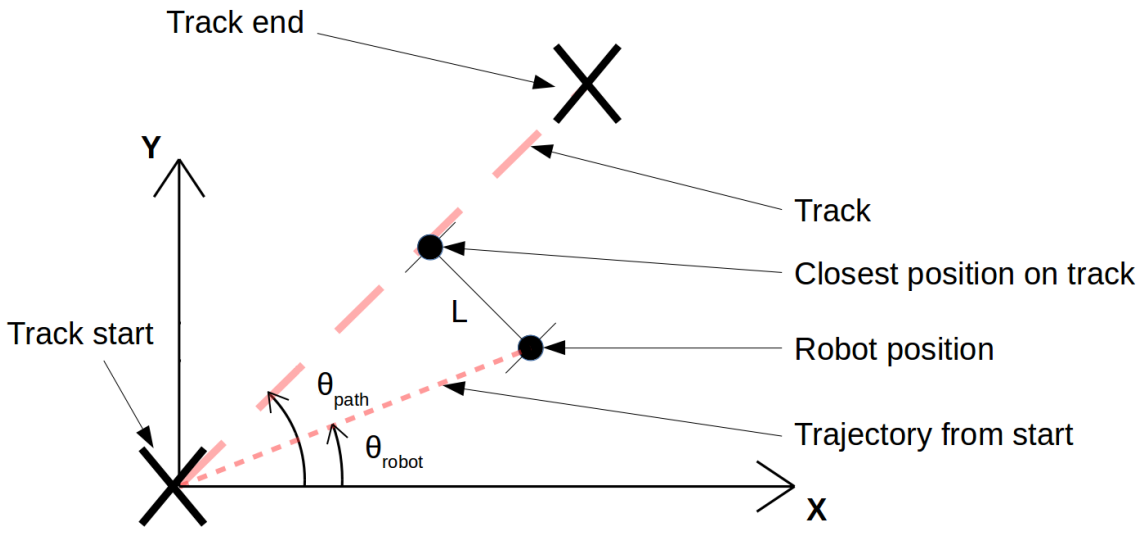

Figure 3.9. The virtual structure constructed for waypoint following with VICON. To begin, the start and end point of the desired motion are specified and VICON tracks the position and orientation of the robot. The process was broken down into the offset angle and displacement. The offset angle is: $\theta = (\theta_{path} - \theta_{robot}) + (\theta_{path} - \theta_{orient})$ where θ_{orient} is the current angle of the robot. The displacment L is the distance from the robot position to the closest position on the line along the path. The integrals and derivatives of these are also solved for use in the PID controller. This process can be expanded in several steps to follow a sequential set of linear steps.

3.4 Designs

3.4.1 V1 Design

V1 was the first prototype of roll-U-ped constructed. FEA models of the base and leg were used to reduce the risk of either breaking. However, this was only done with limited success due to both the varied force response acting on the leg in the set of trajectories examined as well as the material properties of 3D prints. As a consequence, the legs did break if the robot was put into a trajectory that produced a relatively extreme force response acting on the legs, such as by rocking the robot back and forth violently. The FEA and resulting breaks were used to inform V2. Both FEA analyses stress results are given in Fig. 3.10 and Fig. 3.11. These FEA analyses were successfully implemented to produce legs that did not break for V2. The complete bill of materials (BOM) of Design A and Design B are given in Tables 3.3 and 3.4, respectively.

3.4.2 V2 Design

The revisions to V1 are as follows. First, roll-U-ped V1 was incapable of running forward due to a joint limit built into the design, limiting the front leg to 5 deg. $< \theta_2 <$ 260 deg. and 10 deg. $< \theta_1 <$ -50 deg. It was revised to a minimum of -15 deg. $< \theta_2 <$ 260 deg. and 15 deg. $< \theta_1 <$ -50 deg. For each of these, θ_1 is the rear angle and θ_2 is the front angle. In order to resolve this, the battery was reoriented perpendicular to how it was and the base and legs were restructured to prevent that contact. The next revision made was on the legs, which experimentally could not handle the maximum forces exerted on them when running. To resolve this, the broken leg was reinforced by adding a brace around the broken region, and the system was rerun with the same inputs that broke the design until no more breaks occurred. The stress on each leg was approximated with FEA analysis. The redesign was accomplished by using the corresponding maximum stress on the FEA model of the across the non-reinforced regions as the limiting factor for the stress experienced by the new part. The resulting design was tapered rather than slotted for increased strength. This made the legs less flexible, however, so as little reinforcement as possible was used while still remaining within the stress limits. The final revision was the addition of the ODroid container. Due to the modular design of the base, however, this required no alterations, only strapping the container to the bottom of the base. A complete

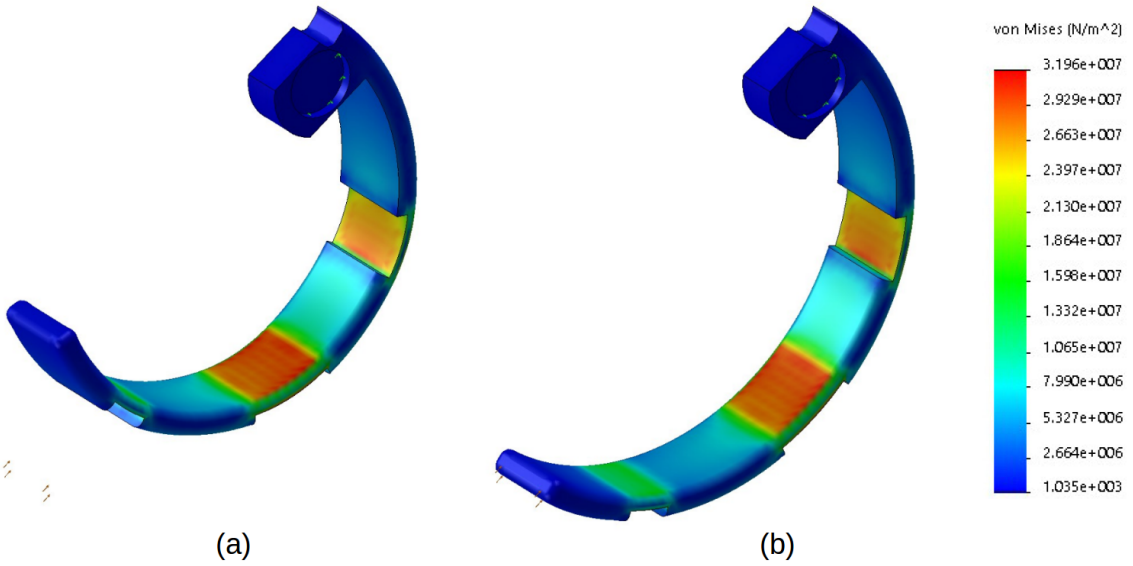

Figure 3.10. The deformed and undeformed FEA analyses von Mises stress results for the legs of roll-U-ped V1. To maximize the accuracy of this, the 3D print was made with 100% infill and the stress was used primarily as a comparison between both the leg designs for V1 and V2 rather than an appoximation of the stress across the design.

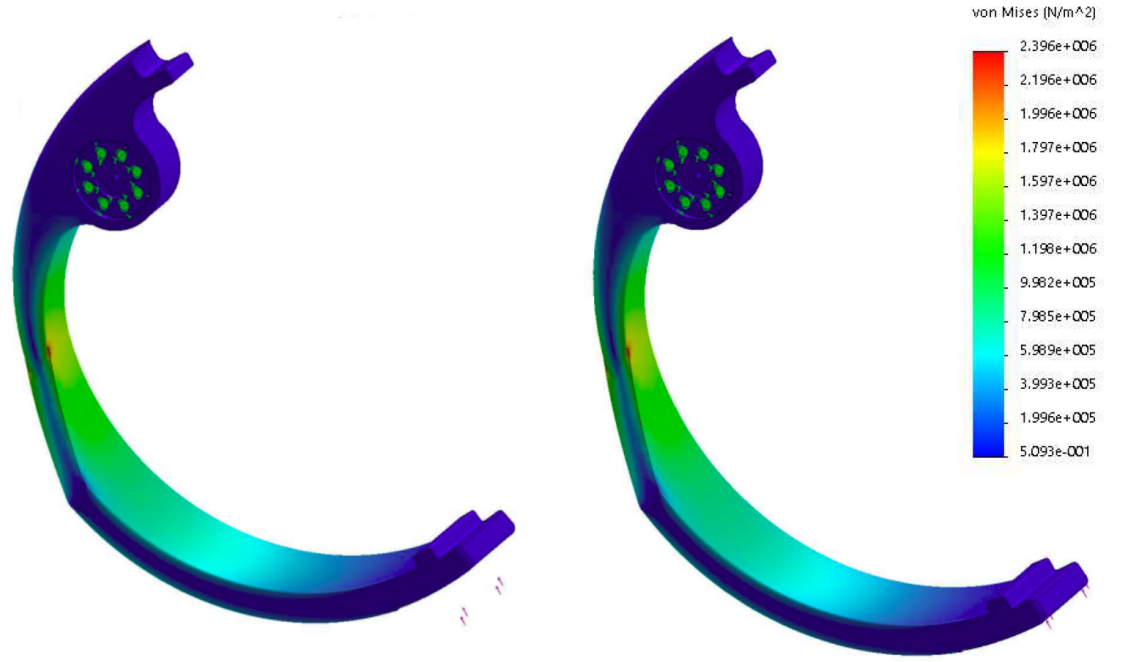

Figure 3.11. The deformed and undeformed FEA analyses von Mises stress results for the legs of roll-U-ped V2. To maximize the accuracy of this, the 3D print was made with 100% infill and the stress was used primarily as a comparison between both the leg designs for V1 and V2 rather than an appoximation of the stress across the design.

Table 3.3. Bill of materials (BOM) of the original design of roll-U-ped, V1. All component weights and prices are given.

Component	**Cost [US$]**	**Weight [kg]**
Base	$32.00	0.232
Arduino	$10.00	0.0280
Wiring and Electrical Components	$10.00	0.0233
Servos (4)	$158.40	0.298
Battery	$15.99	0.157
Screws (16)	$5.00	0.0100
Legs (4)	$32.00	0.0.132
Total	$263.39	0.880

Table 3.4. Bill of materials (BOM) of roll-U-ped V2. All component weights and prices are given. These were the values used in the mass and rotational inertias in modeling as well as to verify price specifications.

Component	**Cost [US]**	**Weight [kg]**
Base	$51.00	1.61
Arduino	$10.00	0.0280
ODroid	$50.00	0.0630
Wiring and Electrical Components	$10.00	0.0412
Servos (4)	$158.40	0.298
Battery	$15.99	0.203
Screws (16)	$5.00	0.01
Legs (4)	$40.76	0.160
Total	$341.15	2.41

list of components with costs and weights are given in Table. 3.4.

3.5 Summary

The design of roll-U-ped incorporates the structure layout, base and leg design, and control system. The mammal layout (ML) was chosen with only four rotational actuators for implementing dynamic motion and incorporating rolling and running with the same actuators. The base of roll-U-ped was designed as a housing for the electronic components, power source, actuators, and allows for the necessary leg angle ranges to produce forward and backward motion. This was accomplished by designing for 3D printing to be used as the manufacturing method of choice. With the use of only four actuators chosen, the legs were designed as curved, flexible, 3D printed elements for customizability, to allow for rolling, low weight, and limiting the maximum torque experienced by the joint actuators in the bounding gait. The control system chosen was a high-level controller to be accomplished by an ODroid for communication with outside directions, and a low-level controller of an Arduino to allow for the direct control of the servo actuators and track chosen OL and CL trajectories.

This design provides a platform for the development of a functional powered rolling and bounding gait, generation of a simulation, and verification of its capabilities with the production of the two prototypes outlined in the previous sections: V1 and V2. Each of these are examined and developed in the following chapters.

CHAPTER 4

MODELING

A model of roll-U-ped bounding is desirable because it allows for the offline development of gaits as well as the potential for then developing improvements to the design to change on the robot, such as altering the compliance/resistance of the legs or the length of the base/legs to generate more effective running. The reason for not having a rolling model is due to the lower complexity and relatively easy control system used in developing functional powered rolling. This was experimentally validated in the prototype design V2. The modeling process is described in detail in this chapter.

4.1 System Modeling

Many of the current mechanisms of modeling robots are designed for non-mobile systems such as robotic arms, meaning they do not implement robust contact models. There are, however, some generalized systems such as VRep and Gazebo that allow for the construction of a robot from stiff elements and position/torque, linear/rotational actuators to handle the robot motion. These are limited in the customizability of ground response, addition of non-stiff elements, and customizability of solvers for increased accuracy. As a consequence, alternative methods of more customizable systems are often developed for mobile robots.

Matlab is one such system that allows for higher customizability in constructing models using simulink with toolboxes such as Simscape for modeling physical systems with customizable relations from scripts and simulating schematics.

The most generalizable and customizable method is constructing differential equations exclusively for the system being modeled. This was the method chosen as it provides the most customizability and direct control over the system response. Additionally, it can be run on any platform with ODE solvers such as Matlab, C++, and Python.

4.2 Lumped Parameter Model

The first step in modeling roll-U-ped was determining the elements to compose the model. Primarily, it required examining the most important aspects of the robot to be included. The final decision was to model roll-U-ped as a 2D system with a single stiff base, two flexible legs on each side attached to the base, an inertia at the end of each leg, and rotational velocity inputs for the servos. The reason for each choice is examined below.

The robot roll-U-ped implements a bounding gait where the two front legs move in sync and the two back legs also move in sync. When moving in a straight line, the linear motion out of the sagittal plane and any roll and yaw are due to imperfections in the robot design, ground, and actuators, along with any external disturbances. The reason for the model is not to have an examination of turning control; rather, it is desirable to accurately model a linear gait for use in gait development. This is because the capability for turning was verified experimentally on prototype V1. As such, a 2D model in the sagittal plane was chosen.

There are several elements that can be used to model including the primary ones of inertias, capacitances, resistors, sources, and energy converters across domains.

The base is a very stiff element relative to the legs and ground flexion, so the flexure in its design was negligible and ignored. Consequently, it was modeled as an inertial element with an inertia in the x- and y-directions as well as a rotational inertia, which was approximated as a constant value and all elements were strapped tightly in on roll-U-ped to make this as accurate as possible. The flexure in the legs, however, is significant and has a non-negligible effect on the ground response. The legs are non-uniform, half circle 3D printed parts, so the flexure response is quite complex. To model this, a spring perpendicular and parallel to a line along its length were used and approximated as linear springs whose capacitances were experimentally found. Fields were also examined for use on the legs and to determine contact, but, due to unfamiliarity with this method, the initial method was chosen. To validite this method, several tests were compared in simulation to the physical system and the error was minimized. An illustration is given in Fig. 4.1.

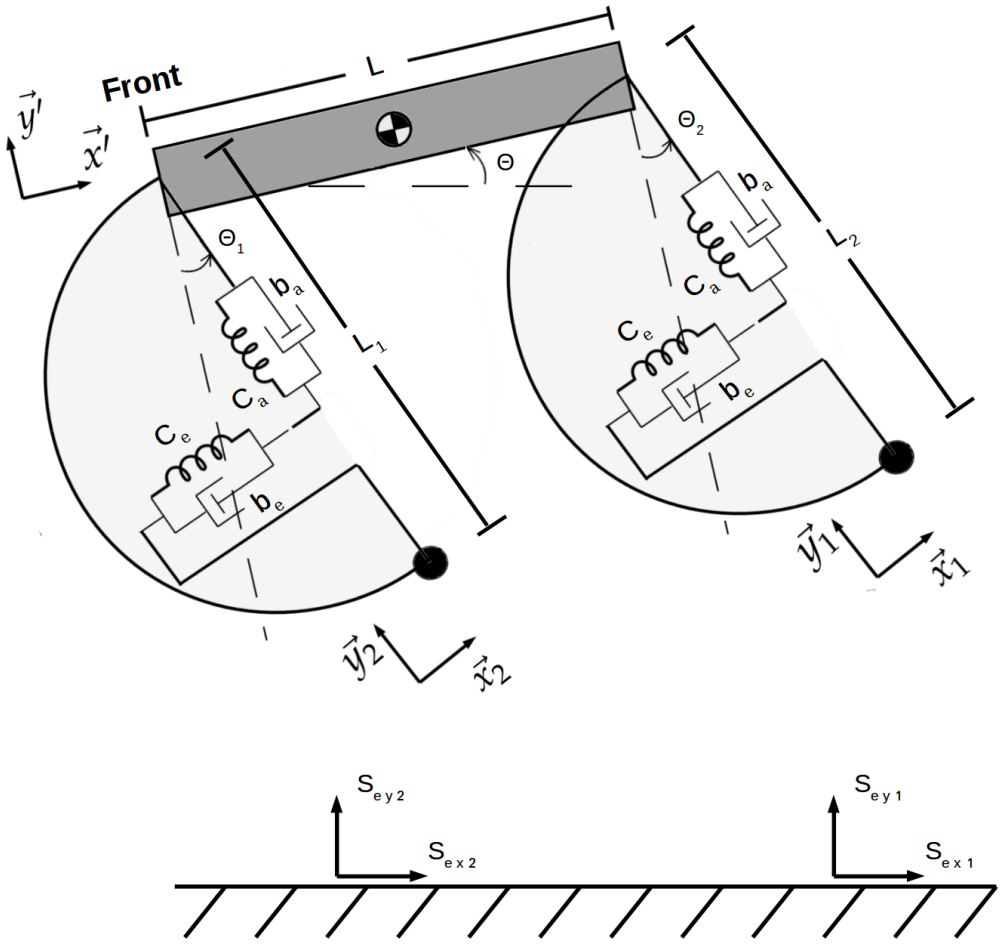

Figure 4.1. Lumped-Parameter model of the quadruped. This is composed of springs, dampers, and inertias. The ground response is given by the efforts S_{ey2}, S_{ex2}, S_{ey1}, S_{ex1}, being the normal force on the front leg, the frictional force on the front leg, the normal force on the rear leg, and the frictional force on the rear leg. k_e, b_e, k_a, and b_a are the capacitances and resistances with e being the perpendicular component and a being the parallel component.

4.3 Bond Graph Modeling

Bond graphs were used to extract the differential equations modeling the system. In addition, bond graphs make it easy to examine models for elements which are acting in differential causality. Differential causality is a redundant causality for the system. This was included in the model validation by examining all the desired states and ensuring they were in integral causality and so incorporated in the state equations. The final bond-graph is given in Fig. 4.2 with all causalities, transformers, and energy flow directions. An illustration of the bond graph segmented to illustrate how the components relate to the

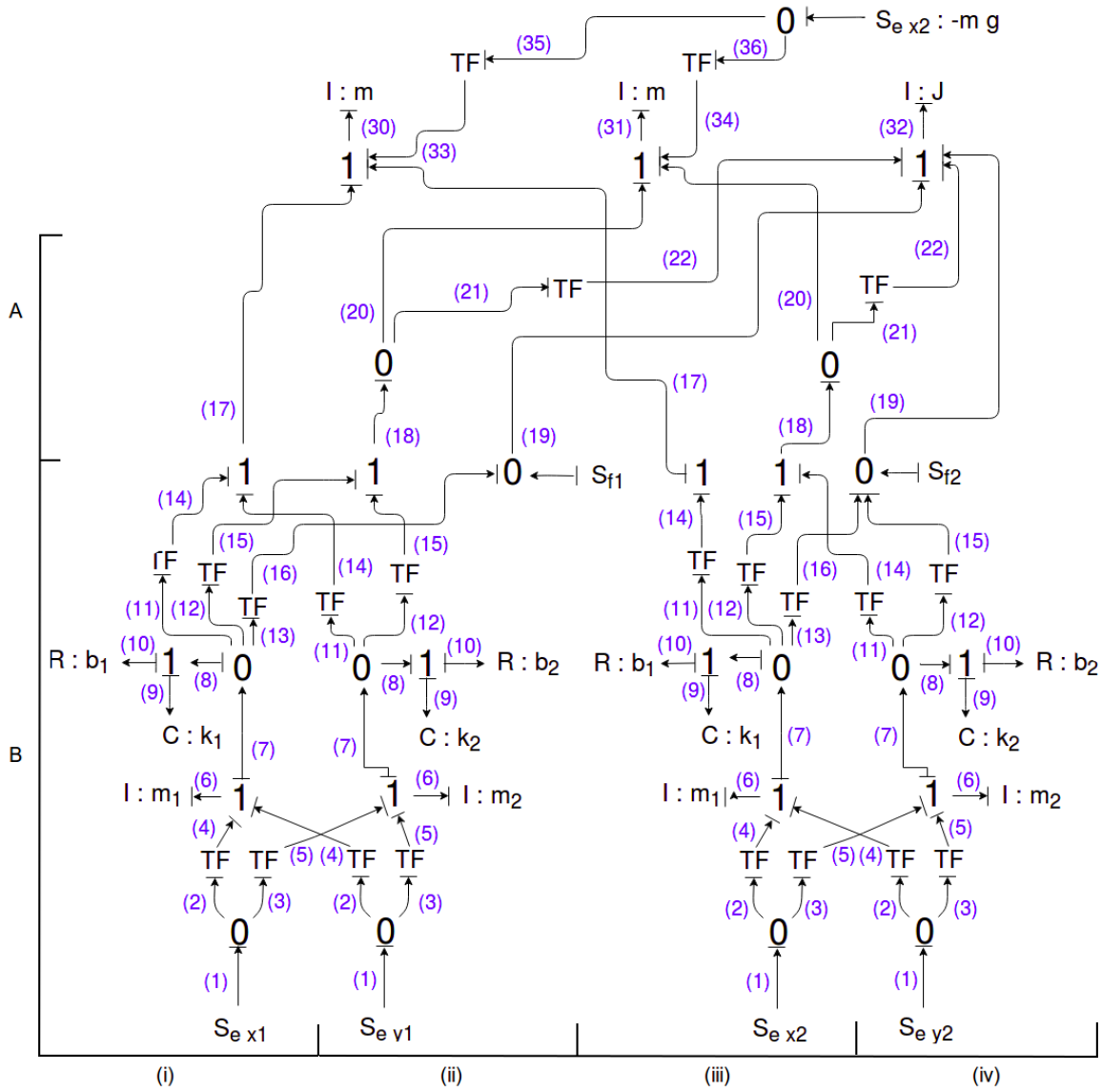

Figure 4.2. The complete bond graph used in modeling roll-U-ped. The values correspond to those in Table 4.2. The subscripts used in labeling are outlined in the sections as follows: (1) along B the sections are segmented into i, ii, iii, and iv, which correspond to the labels of i, ii, iii, and iv, (2) along A and the range of i-ii is labeled i, and (3) along B and the range of iii-iv is labeled ii.

phsical system is given in Fig. 4.3. Causalities in bond graphs determine the effort and flow directions in the system (used in solving the differential equations), transformers are mechanisms for transforming between domains (e.g., rotational and translational), and the direction of energy flow is decided by the signs determining where energies are entering and exiting the system. Causality is given by a bar at one end of an arc, transformers are defined by TF elements, and the direction of energy flow is given by the direction of the arrow on each arc.

From the bond graph, the differential equations were solved. The final states were the displacements of each leg in the parallel and perpendicular directions, the momentums of the masses at the end of each leg, the linear momentums of the base, and the rotational momentum of the base. The output states were the displacements of the base. These are all given in Table 4.1. The values corresponding to each variable are given in Table 4.2. The capacitance values listed are the values for each leg, so the effective constants used in the leg were half of those in the table and the friction were twice those in the table.

The complete set of state equations are:

$$\dot{q}_{9i} = \frac{p_{6i}}{m_1} - \frac{p_{30}}{mTF_{11i}} - \frac{\frac{p_{31}}{m} + \frac{p_{32}}{JTF_{21i}}}{TF_{12i}} - \frac{\frac{p_{32}}{J} - S_{f1}}{TF_{13i}}, \tag{4.1}$$

$$\dot{q}_{9ii} = \frac{p_{6ii}}{m_1} - \frac{p_{30}}{mTF_{11ii}} - \frac{\frac{p_{31}}{m} + \frac{p_{32}}{JTF_{21i}}}{TF_{12ii}}, \tag{4.2}$$

$$\dot{q}_{9iii} = \frac{p_{6iii}}{m_2} - \frac{p_{30}}{mTF_{11iii}} - \frac{\frac{p_{31}}{m} + \frac{p_{32}}{JTF_{21ii}}}{TF_{12iii}} - \frac{\frac{p_{32}}{J} - S_{f2}}{TF_{13iii}}, \tag{4.3}$$

$$\dot{q}_{9iv} = \frac{p_{6iv}}{m_2} - \frac{p_{30}}{mTF_{11iv}} - \frac{\frac{p_{31}}{m} + \frac{p_{32}}{JTF_{21ii}}}{TF_{12iv}}, \tag{4.4}$$

$$\dot{p}_{6i} = \frac{S_{ex1}}{TF_{2i}} + \frac{S_{ey1}}{TF_{2ii}} - \dot{q}_{9i}b_1 + k_1 q_{9i}, \tag{4.5}$$

$$\dot{p}_{6ii} = \frac{S_{ex1}}{TF_{3i}} + \frac{S_{ey1}}{TF_{3ii}} - \dot{q}_{9ii}b_2 + k_2 q_{9ii}, \tag{4.6}$$

$$\dot{p}_{6iii} = \frac{S_{ex2}}{TF_{2iii}} + \frac{S_{ey2}}{TF_{2iv}} - \dot{q}_{9iii}b_1 + k_1 q_{9iii}, \tag{4.7}$$

$$\dot{p}_{6iv} = \frac{S_{ex2}}{TF_{3iii}} + \frac{S_{ey2}}{TF_{3iv}} - \dot{q}_{9iv}b_2 + k_2 q_{9iv}. \tag{4.8}$$

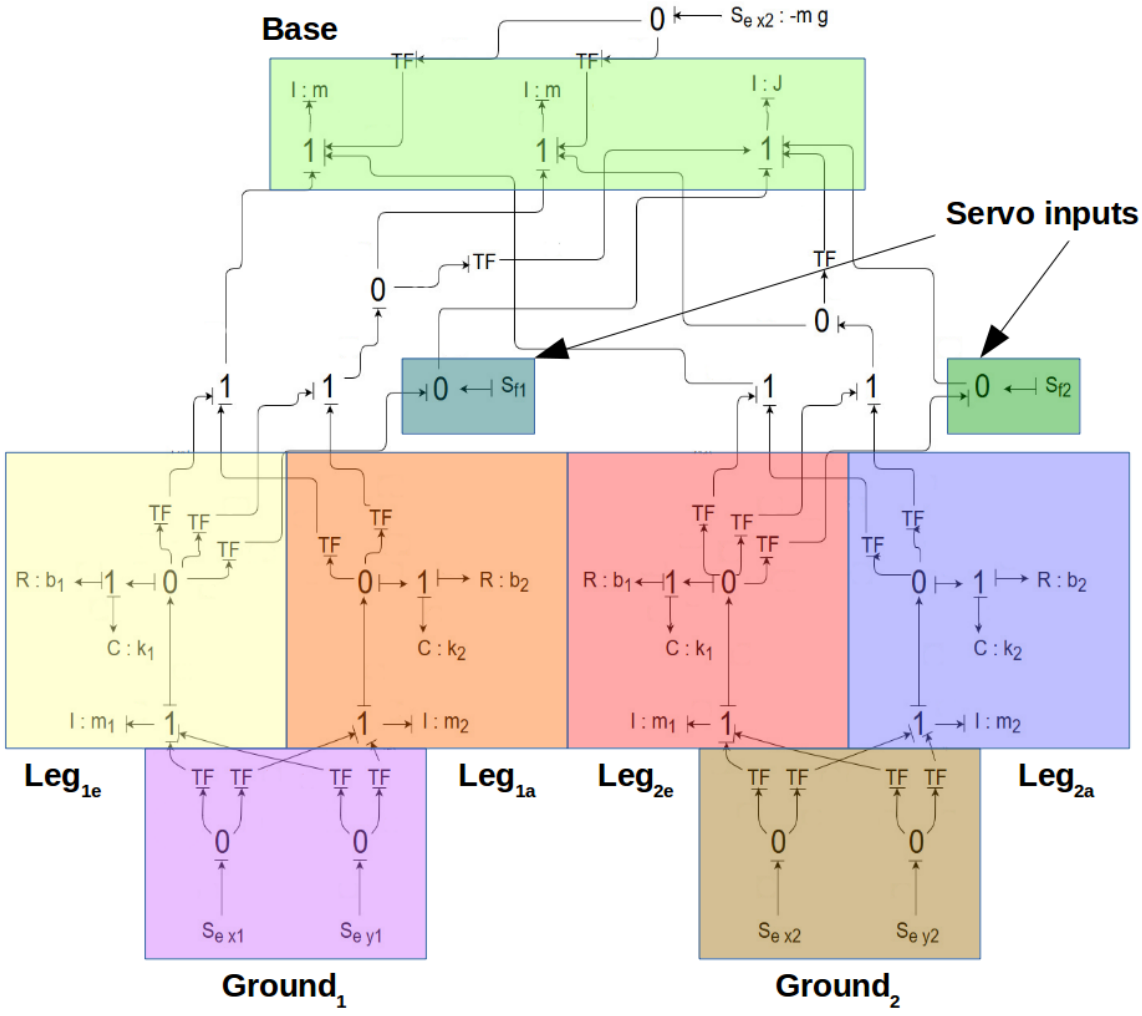

Figure 4.3. The complete bond graph used in modeling roll-U-ped segmented into the regions correlating to the physical robot. The components are the: Base, Servo inputs, Legs, and Ground responses.

Table 4.1. Table of all states and ouput states with their corresponding real-world meaning.

Variable	**Meaning**
States	
q_{9i}	Linear displacement of the leg in the $\vec{x_1}$ direction.
q_{9ii}	Linear displacement of the leg in the $\vec{y_1}$ direction.
q_{9iii}	Linear displacement of the leg in the $\vec{x_2}$ direction.
q_{9iv}	Linear displacement of the leg in the y_2 direction.
p_{6i}	Linear momentum of the leg in the $\vec{x_1}$ direction.
p_{6ii}	Linear momentum of the leg in the $\vec{y_1}$ direction.
p_{6iii}	Linear momentum of the leg in the $\vec{x_2}$ direction.
p_{6iv}	Linear momentum of the leg in the $\vec{y_2}$ direction.
p_{30}	Linear momentum of the base in the $\vec{x'}$ direction.
p_{31}	Linear momentum of the leg in the $\vec{y'}$ direction.
p_{32}	Angular momentum of the leg in-plane.
Output States	
q_x	Linear displacement of the base in the $\vec{x}$ direction.
q_y	Linear displacement of the base in the $\vec{y}$ direction.
q_θ	Angular displacement of the base in-plane. The roll angle.

Table 4.2. Table of the values of each variable in the bond graph. μ_{C1}, μ_{C2}, μ_{V1}, μ_{V1}, and $max_{coulomb}$ are the respective frictional constants where (a) $_C$ is coulomb friction, (b) $_V$ is viscous friction, (c) $_1$ is for $\theta_1/\theta_2 > 0$, and (d) $_2$ is for $\theta_1/\theta_2 < 0$. The respective capacitances and resistances of k_e, k_a, b_e, b_a, k_{ground}, and b_{ground} are also given. Finally, the dimensional and global values are given in the linear and rotational inertia of the base, m and J, the lengths of L_{base} and L_{leg}, the gravitational constant of g, and the Robot diameter.

Variable	Value	Transformer (TF)	Value
m_1	0.0536 kg	TF_{2i}	$\frac{1}{cos(\theta+\theta_1)}$
m_2	0.0536 kg	TF_{3i}	$-\frac{1}{sin(\theta+\theta_1)}$
k_e	8250 N/m	TF_{11i}	$\frac{1}{cos(\theta_1)}$
k_a	5559 N/m	TF_{12i}	$\frac{1}{sin(\theta_1)}$
b_e	69 Ns/m	TF_{13i}	$\frac{1}{L_{leg}}$
b_a	55 Ns/m		
k_{ground}	2750 N/m	TF_{2ii}	$\frac{1}{sin(\theta+\theta_1)}$
b_{ground}	44 Ns/m	TF_{3ii}	$\frac{1}{cos(\theta+\theta_1)}$
μ_{C1}	0.78	TF_{11ii}	$-\frac{1}{sin(\theta_1)}$
μ_{C2}	0.76	TF_{12ii}	$\frac{1}{cos(\theta_1)}$
μ_{V1}	0.37		
μ_{V_2}	0.37	TF_{2iii}	$\frac{1}{cos(\theta+\theta_2)}$
m	0.933 kg	TF_{3iii}	$-\frac{1}{sin(\theta+\theta_2)}$
J	0.0685 kg m^2	TF_{11iii}	$\frac{1}{cos(\theta_2)}$
L_{base}	4.58 in	TF_{12iii}	$\frac{1}{sin(\theta_2)}$
L_{leg}	5.03 in	TF_{13iii}	$\frac{1}{L_{leg}}$
g	$9.81\frac{m}{s^2}$		
Robot diameter	5.9 in	TF_{2iv}	$\frac{1}{sin(\theta+\theta_2)}$
$max_{coulomb}$	55 N	TF_{3iv}	$\frac{1}{cos(\theta+\theta_2)}$
		TF_{11iv}	$-\frac{1}{sin(\theta_2)}$
		TF_{12iv}	$\frac{1}{cos(\theta_2)}$
		TF_{21i}	$\frac{1}{-L/2}$
		TF_{21ii}	$\frac{1}{L/2}$
		TF_{33i}	$\frac{1}{sin(\theta)}$
		TF_{34i}	$\frac{1}{cos(\theta)}$

$$\dot{p}_{30} = \frac{\dot{q}_{9i}b_1 + k_1 q_{9i}}{TF_{11i}} + \frac{\dot{q}_{9ii}b_2 + k_2 q_{9ii}}{TF_{11ii}} + \frac{\dot{q}_{9iii}b_1 + k_1 q_{9iii}}{TF_{11iii}} + \frac{\dot{q}_{9iv}b_2 + k_2 q_{9iv}}{TF_{11iv}} + \frac{-mg}{TF_{33}}, \quad (4.9)$$

$$\dot{p}_{31} = \frac{\dot{q}_{9i}b_1 + k_1 q_{9i}}{TF_{12i}} + \frac{\dot{q}_{9ii}b_2 + k_2 q_{9ii}}{TF_{12ii}} + \frac{\dot{q}_{9iii}b_1 + k_1 q_{9iii}}{TF_{12iii}} + \frac{\dot{q}_{9iv}b_2 + k_2 q_{9iv}}{TF_{12iv}} + \frac{-mg}{TF_{34}}, \quad (4.10)$$

$$\dot{p}_{32} = \frac{\frac{\dot{q}_{9i}b_1 + k_1 q_{9i}}{TF_{12i}} + \frac{\dot{q}_{9ii}b_2 + k_2 q_{9ii}}{TF_{12ii}}}{TF_{21i}} + \frac{\dot{q}_{9i}b_1 + k_1 q_{9i}}{TF_{13i}} + \frac{\frac{\dot{q}_{9iii}b_1 + k_1 q_{9iii}}{TF_{12iii}} + \frac{\dot{q}_{9iv}b_2 + k_2 q_{9iv}}{TF_{12iv}}}{TF_{21ii}} + \frac{\dot{q}_{9iii}b_1 + k_1 q_{9iii}}{TF_{13iii}}. \quad (4.11)$$

The additional equations used for the added states needed to extract the position and orientation of the base:

$$\dot{q}_x = p_{30}cos(\theta)/m - p_{31}sin(\theta)/m \quad (4.12)$$

$$\dot{q}_y = p_{30}sin(\theta)/m + p_{31}cos(\theta)/m \quad (4.13)$$

$$\dot{q}_\theta = p_{32}/J \quad (4.14)$$

4.3.1 Actuator Inputs

The only inputs to roll-U-ped are the four servos, reduced to two for the 2D model. Servos are motors that are usually modeled as voltage inputs with a resistance and inductance. However, the input in this case was defined as a function of the position, velocity for bond graphs, which the servo is tracking. This was implemented by applying positions to the physical system and the derivative to the simulation. However, the servo is not a perfect velocity input as, though it has accurate position tracking, it has an upper torque limit. To resolve this, the equations were solved once in the ODE solver and if the corresponding torque on the legs, given as e_{19i} for θ_1 and e_{19ii} for θ_2 below, was higher than the torque that can be produced by the servo, then the state equations were re-solved for the maximimum flow it could produce. These are given in the equations below. The maximum speed for a given torque was determined from the torque-speed curve of the servos. The servo specs are given by Table 3.2.

$$\tau_1 = e_{19i} = \frac{e_{8i}}{TF_{18i}}, \quad (4.15)$$

$$\tau_2 = e_{19ii} = \frac{e_{8iii}}{TF_{18iii}}. \quad (4.16)$$

where τ_1 and τ_2 are the respective torques on the rear and forward legs as well as e_{19i}, e_{19ii}, e_{8i}, e_{8iii}, TF_{18i}, and TF_{18iii} being the respective efforts and transformers from the bond graph.

4.3.2 Ground Response

The final thing to model is the ground response. The ground response was split into two components: the normal force and the frictional force. Coulomb and viscous friction was used as the frictional model and the normal force was approximated by a spring damper. The equations modeling the force response from the leg being in contact with the ground are:

$$F_y = k_{ground}|y| - b_{ground}\dot{y}, \tag{4.17}$$

$$F_x = F_{coulomb} - b_{friction}\dot{x}. \tag{4.18}$$

$$F_{coulomb} = min(\mu F_y sign(\dot{x}), max_{coulomb}) \tag{4.19}$$

where F_y is the normal force, F_x is the frictional force, k_{ground} and b_{ground} are the respective ground stiffness and damping, y is the height from the ground, μ is the coulomb coefficient, $b_{friction}$ is the viscous frictional coefficient, and $max_{coulomb}$ is the maximum coulomb friction.

The ground response parameters were unknown, so these constants were derived experimentally, which is examined in the next chapter.

4.4 Simulation

A list of the final results for various inputs compared between the physical system and the model is given in Table 4.3. The average % error in step length as found by the table is 33% for a % error modeled after engineering strain. The model does break down, however, if the quadruped falls over, if the force on the leg is high enough that the linear approximation of the leg flexure causes significant error, or if the robot is run on a surface other than the flexible-tacky surface for which the ground parameters were set. All the

Table 4.3. Table of compared results of the physical system with the simulation given three sets of inputs. These inputs were all varied sinusoidal trajectories given in Table 6.1 in the respective order top-to-bottom for that table, 1-3 in this table. The table contains the mean and standard deviations of step length for both the robot and simulation in all three tests. The % error of the simulation is found using the equation $\frac{||S_{max}|-|P_{max}||}{|P_{max}|}|$ where P is the physical robot and S is the simulation.

	1_{mean}	1_{std}	2_{mean}	2_{std}	3_{mean}	3_{std}
Simulation step dist [m]	0.08	0.0152	0.100	0.000	0.085	0.032
Robot step dist [m]	0.056	0.0326	0.0667	0.0125	0.08	0.0158
% error	42.9	53.17	50.0	100.	6.25	102.

values of each state are output from the simulation in addition to an animation of the robot running for visual purposes. A figure of the schematic produced in roll-U-ped simulations is given in Fig. 4.4. An illustration of the comparative physical system with the simulation run for the same inputs is given in Fig. 4.5.

4.5 Summary

This chapter goes over the development of a model for roll-U-ped traversing a flexible-tacky surface, implemented as a yoga mat. In order to test the validity of this model, gaits were constructed for the robot to run over the simulated surface. The ground parameters were then iterated through to minimize the error in the velocity of the robot between the simulation and the physical system. This model was then used to construct a Q-learned gait. The primary limitation of the model is it being only accurate for simulating a single ground contact surface.

There are several inaccuracies in the model, primary of which is the linear approximation of the flexure of the legs. Due to the curve of the leg, the flexure of the leg in the perpendicular and parallel directions is not decoupled. In order to more accurately model the legs, modulation of the current deflection of the leg for the stiffnesses of the legs could be implemented. This would require more testing on the stiffness and damping of the legs for the physical system. A corresponding limitation for the use of this simulation is in using it to perform revisions to the design as this is a current knowledge gap in 3D printing.

Two more inaccuracies in the model are the approximation of the robot inertia and the ground response. The robot rotational inertia is not a constant due to the legs being in motion. This could be resolved by modulating the angles of each leg to the robot base rotational inertia J. The ground response could be better approximated by testing the capacitance and damping for various speed and force inputs and implementing interpolation to find the nearest approximation.

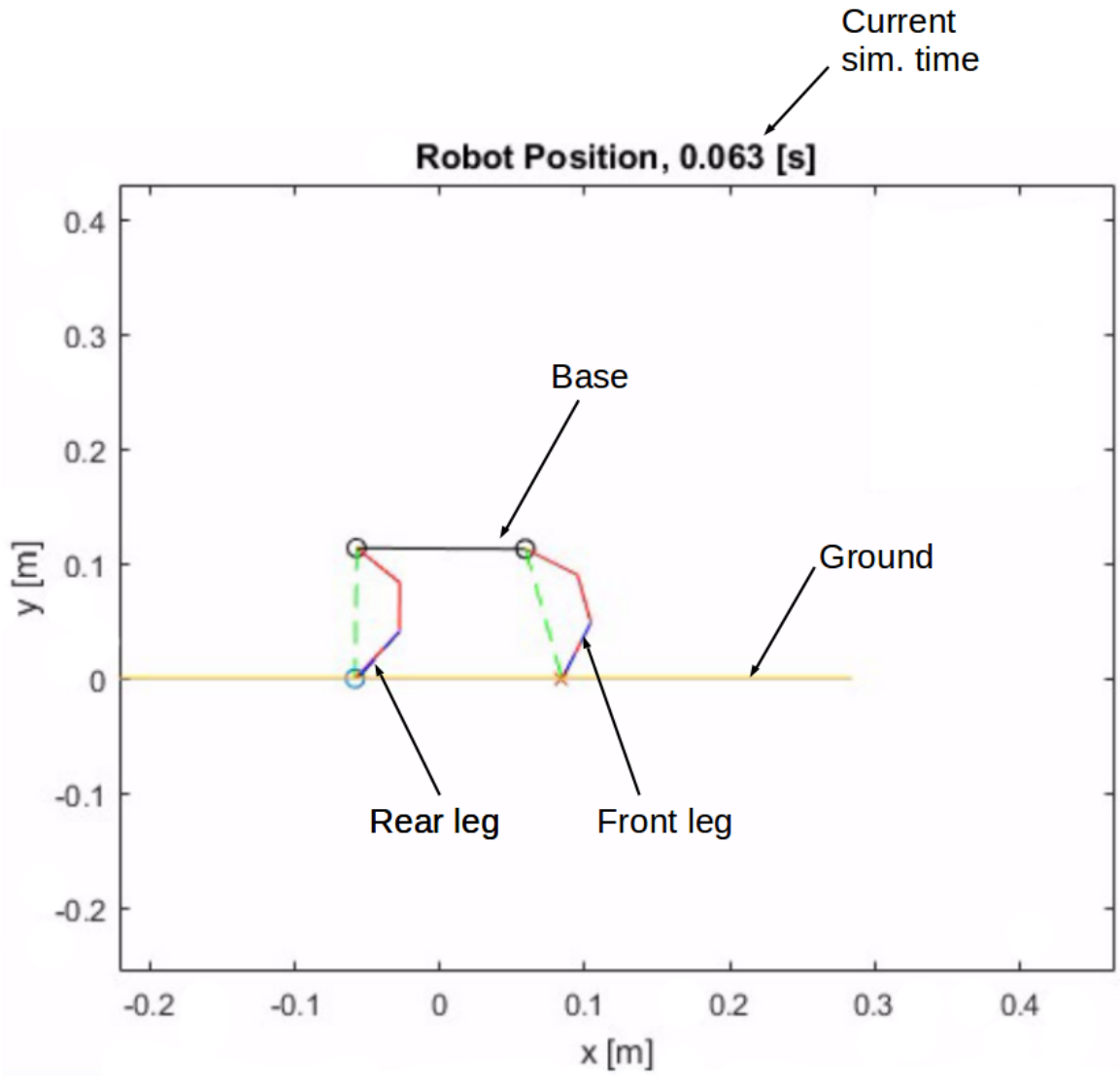

Figure 4.4. An illustration of the details of the simulation animation output. This was used to visually illustrate roll-U-ped moving. A 3-linkage element was used in place of the semi-circular leg to illustrate the direction of the bend.

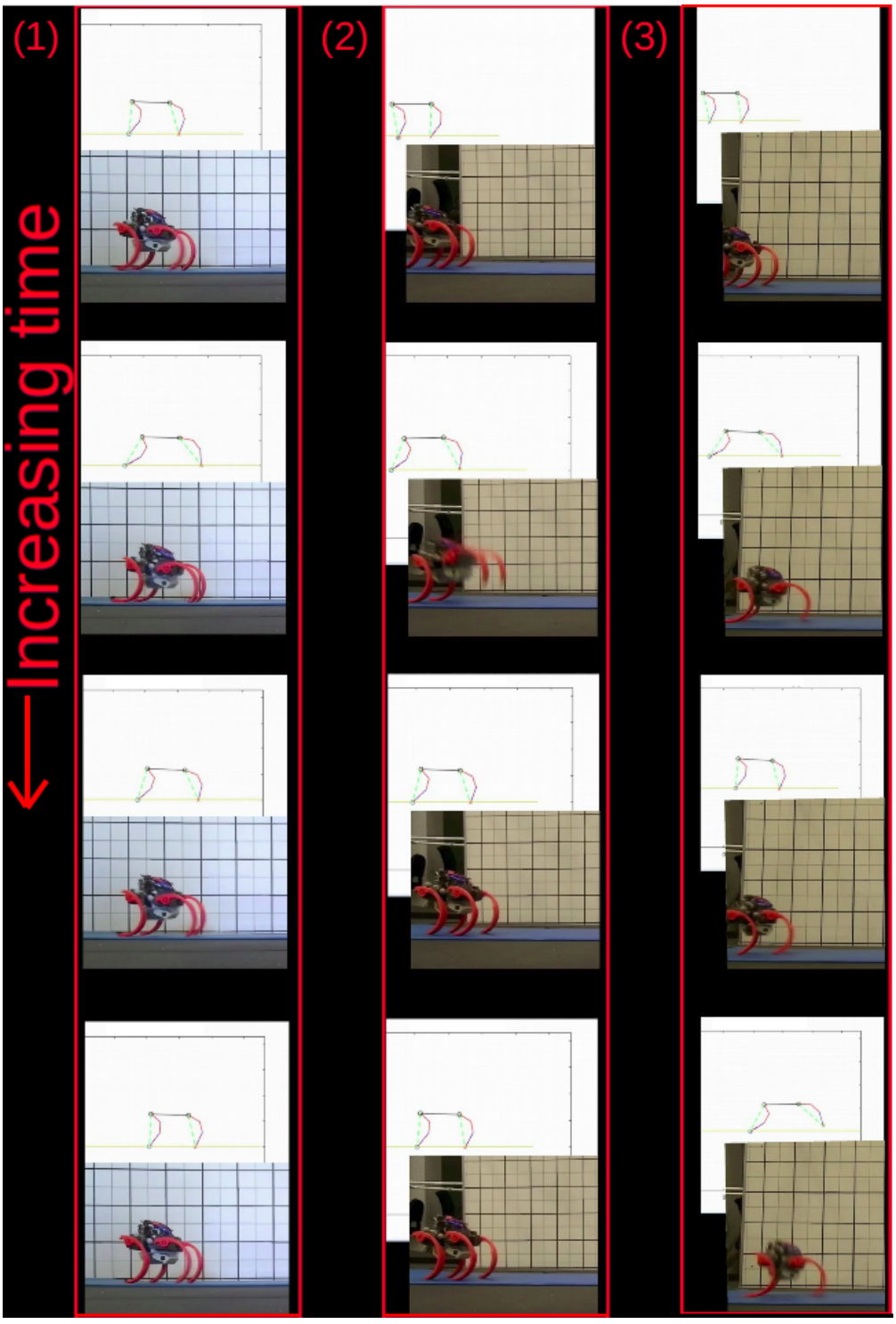

Figure 4.5. Comparison of three runs (1), (2), and (3) of the simulation used in refining the ground values. The frames were compared at the same time in simulation and on the physical system. These are images of the responses from Table 4.3.

CHAPTER 5

GAIT DEVELOPMENT

This chapter focuses on the gait development of the roll-U-ped robot. First, bounding motion is studied using sinusoidal inputs and inputs obtained using machine learning. Second, both powered and unpowered rolling motions are studied, as illustrated in Fig. 5.1.

5.1 Running

The roll-U-ped gait is illustrated in Fig. 3.3. As is shown, the gait incorporates a jump up when moving both legs in. It is extremely difficult, however, to create an open-loop trajectory that pefectly follows that exact motion sequence repeated across the run. This is particularly true when traversing variable surfaces. As a consequence, there is often a stumble introduced where the robot is either in the air when it is propelling itself forward or is on the ground when it is propelling itself backwards. This could be simplified by always taking a single step, halting until the robot's linear and angular velocities have reached zero, and continuing. However, this significantly slows down the robot, so the running was constructed without the halt.

5.1.1 Bounding - Parameterized Gait

As discussed, traditional robot control methods are difficult to implement on dynamic robots due to the lack of a robust kinetic model to examine their complex motion. As such, the first trajectories of the robot legs were generated according to the following sinusoidal trajectory as defined by the equations:

$$\theta_1(t) = A_1 cos(f * (t + p_1)) + b_1, \tag{5.1}$$

$$\theta_2(t) = A_2 cos(f * (t + p_2)) + b_2, \tag{5.2}$$

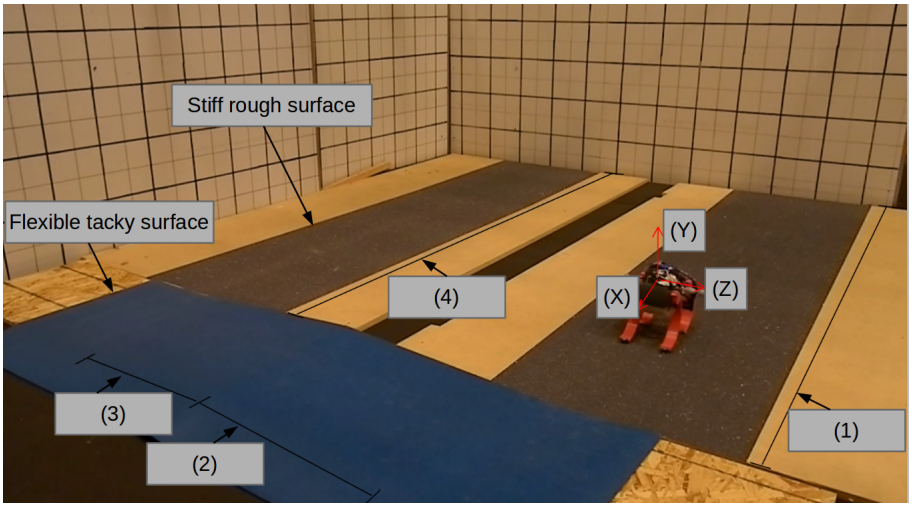

Figure 5.1. Elements of a sample track for roll-U-ped implemented for testing in running/rolling capabilities with remote control. The methods of locomotion implemented were: (1) running/powered rolling, (2) running uphill, (3) running/passive rolling downhill, and (4) flat linear path. The terrains examined were: (1) flat linear path, (2) inclined linear path, (3) declined linear path, and (4) flat linear path. Motion in both directions was tested across this and similar paths. The stiff and rough as well as flexible and tacky surfaces used in testing are both shown here.

where θ_2 is the current angle of the front legs, θ_1 of the back, and t is the time in seconds. A_1, p_1, f, and b_1 are the respective amplitude, phase offset, frequency, and vertical shift of the trajectory of θ_1. A_2, p_2, f, and b_2 are the respective amplitude, phase offset, frequency, and vertical shift of the trajectory of θ_2.

The purpose of choosing a sinusoidal trajectory was to produce the oscillation as desired, while not introducing the high accelerations in triangular and rectangular trajectories to avoid overheating the servos.

5.1.2 Bounding - Machine Learning

The second method of bounding gait development was designed as an automated process that could be performed on various designs of the robot for revisions in the future. It was not implemented for this purpose here but it was successful in learning a gait that could be applied to the physical robot, which was the secondary goal. Turning was not examined in machine learning; rather, the same technique of sinusoidal trajectories was used by increasing the amplitude of the angle of the left legs while reducing the amplitude of the angle on the right legs to turn right and the reverse to turn left. This

was experimentally found to be successful.

Q-learning is a reinforcement learning algorithm that implements a domain with specified states and actions that define the workspace. The purpose of Q-learning is to determine the best action to take in any given state. This is done by developing a structure containing the Q-value for each state-action pair. Discrete states and actions were implemented and to do so, the current state and action was always approximated as the nearest of the districtized states. The Q-learning states and actions, with corresponding sample values in a test and a description, are given in Table 5.1. All parameters were defined in the global state-space. Local state-space was developed as well, and results were collected. The parameter limits were not set as the limits or resolutions of the states used in simulation, they were only used in extracting the current state-action pair for revising and extracting Q-values.

Q-learning uses those Q-values to determine the action to take in a given state by applying: action = $max_r(Q(s, a_r))$ where s is the current state and a_r are the possible actions to take. The goal for implementing Q-learning was to construct an effective open-loop trajectory that could be used. In order to do so, Q-learning was run in simulation to generate trajectories for the front and back legs. Those trajectories were then separated into the initial and rhythmic movement primitives. To implement this on roll-U-ped, it was initialized with the same conditions of the simulation and the initial trajectories were used to direct the legs until they were completed and the rhythmic trajectories were then run in a continuous loop.

An update equation is used in Q-learning to revise the states over time. The update equation used here is:

$$Q(s_t, a_t) \leftarrow (1 - \alpha) * Q(s_t, a_t) + \alpha * (r_t + \gamma * max_a(Q(s_{t+1}, a))), \tag{5.3}$$

where $Q(s_t, a_t)$ is the current Q-value for the corresponding state-action pair, α is the learning rate, and γ is the discount factor. α determines how the old values are weighted with the new ones. γ defines how the future predicted reward is weighted with the current reward. The limits for each are: $0 < \alpha < 1$ and $0 < \gamma < 1$. The final values used for the Q-learning were: $\alpha = 0.9$ and $\gamma = 0.9$.

Table 5.1. A table of the Q-learning states and actions used with a description of each. As can be seen, not all of the states incorporated in the model were used in the Q-learning states. This is because limiting the workspace reduces runtime and simplifies the problem. Other states considered were: the forward velocity, height, and $\dot{\theta}$.

State	Description	Values used
θ [deg.]	The current roll angle of the robot.	-15 to 15 in increments of 4
θ_1 [deg.]	The current angle of the rear servos	45 to 10 in increments of 7
θ_2 [deg.]	The current angle of the front servos	-10 to 45 in increments of 7
Action	Description	Values used
S_{f1} [rad/s]	The rotational velocity of the rear servo	[-4.26, -2.1318, 2.1318, 4.26]
S_{f2} [rad/s]	The rotational velocity of the front servo	[-4.26, -2.1318, 2.1318, 4.26]

The reward was picked as the desirable value Q-learning learns to take the actions to increase. Q-learning maximizes the expected long-term reward, or the Q-values of every state-action pair. The desirable outcome of this Q-learning was a fast cyclic motion forward. As a consequence, the reward was made the forward or backward velocity of the robot, depending on the direction of motion desired. The base angle and height were also considered because of the modeling limits and to speed up the simulation by stopping exploration of the robot once it has fallen over. Instead, the simulation was ended if the robot began to fall over and the Q-learned value discarded for that step. This was primarily because the simulation was not valid at that point. If the model were altered to be valid for the robot falling over and in making contact between the base and ground, a cost, or negative reward, would instead be applied in the Q-value update for making the robot fall over rather than stopping the simulation.

There are many additional elements that can be incorporated with Q-learning to improve results. Several of these were implemented in this thesis. The first is exploration. Epsilon-greedy is a method in which the optimal action was picked some percentage of the time and otherwise, a random action was chosen. A variation of this was implemented for which rather than a random action being chosen, the least visited action was taken 70% of the time and the one with the highest Q-value taken otherwise. This was used to introduce exploration. Secondly, angle limits were put on the directly controlled states of θ_1 and θ_2 from -40 to 10 deg. and 10 to -40 deg., respectively. This was to prevent the robot legs from continuously rotating out of the range that they would normally travel during bounding. This was only necessary due to how the states were extracted by approximating them as the closest state in the array of states. When this was not implemented, valid learning could not be accomplished as the simulation would become invalid as contact between the ground and the legs or base was ignored at any point other than the end of the leg in modeling.

5.2 Rolling

Open-loop rolling is not an effective method of propulsion as small discrepancies in the surfaces being rolled over causes extreme variances in the rolling response. In addition, closed-loop control is relatively easy to implement due to both position/angle tracking

with VICON and with a remote control. In VICON, rolling can be controlled by expelling the respective limbs at a specified angle where the time of the expelling is assigned to a button in the case of the remote control. In future work, the current roll velocity could also be included in the roll angle at which to propel out. In the field, this could be accomplished by adding contact sensors to the surface of each leg to determine when it has reached the desired angle to extend its legs out.

5.3 Transition

Transitions between rolling and running are very simple. From running to rolling the legs are folded in to form the circle using a linear trajectory from the current extended position to the desired folded position. Rolling to running is slightly more difficult because, depending on the angle at which the robot is transformed, it could end up oriented upside down. This was remedied in VICON by forcing a delay from the point the robot is desired to extend out to standing until it is within the acceptable roll angle range of 110 deg. $< \theta <$ 160 deg. For remote control, it is up to the user to handle the condition of when to expel the legs to avoid becoming upside down. This is, however, easily remedied by allowing the user to choose the standing orientation.

5.4 Turning

Several mechanisms of rotating the robot were examined. Slightly reducing the amplitude of the joint trajectories for one side of the robot caused a curved running path. However, it is also desirable to be able to rotate in place. To do so, two methods were tried: (1) holding the right legs extended while the left legs track the joint trajectory of the robot running and (2) holding the back legs extended while the front legs track a sinusoidal trajectory. Both methods were successful with the respective motions generated being: (a) rotating the robot about a small circle using low frequency sinusoids and (b) incorporating high frequency oscillations for a tight curve.

5.5 Summary

The control of quadruped gaits can be accomplished with open-loop or closed-loop control. Closed-loop control allows for the robot to respond to the measured states, which requires it to have the necessary sensors to read those states. As a result, open-loop control

with a parameterized gait was chosen for the running to alleviate the need for any sensors. Parameterized trajectories can function well for open loop trajectories, but they require an experienced user to take the time to choose a trajectory and find effective parameter constants to use. This was very time consuming, so an additional method of constructing gaits was produced in which the model was used to run Q-learning on and construct a new trajectory that was then used on the physical system.

Open-loop control for powered rolling was also considered, but, in order to produce fast and consistent motion, the current roll angle of the robot was used as feedback to determine the time at which to extend the front or back legs out. The need for external sensors could be eliminated, however, by adding touch sensors on roll-U-ped's legs to approximate the current roll angle.

CHAPTER 6

RESULTS AND DISCUSSION

This chapter encompasses the results and discussion of the robot prototypes V1 and V2. In order to examine roll-U-ped's capabilities, it was tested across various surfaces with alterations to sinusoidal trajectories to examine the response of the robot to assorted inputs and external environments. Both designs are capable of running both right-side up and upside down. However, the response is equivalent due to a symmetrical design. So, in order to preserve time, tests were only conducted right-side up. In later work, this could be incorporated into a final version of the remote and motion tracking control in order to handle the transition from rolling to running smoother.

6.1 V1 Design

V1 of roll-U-ped was a verification of design and used in determining the limitations in the design to inform revisions for V2. In order to do this, both powered rolling and powered running were examined. Powered rolling was not fully developed as, due to incorrect dimensions of the servos from the manufacturer, the legs did not align correctly and thus could not roll smoothly. Furthermore, it was unable to run forward due to a joint limit on the robot preventing the leg from extending out at least 5 deg. from beneath the base. This prevented V1 from producing a forward-moving trajectory. Rolling was tested to determine the capability of the motors to handle the propulsion. The design was also capable of running upside down, which was kept for V2.

6.1.1 Results

An illustration of a gait developed on roll-U-ped V1 implemented on V2 for completing a run along a carpeted surface is given in Fig. 6.1. Tests were run to ensure the legs were able to act fast enough and provide enough torque in order to handle the propelled rolling. Completed powered rolling was not constructed, but the robot was tested in

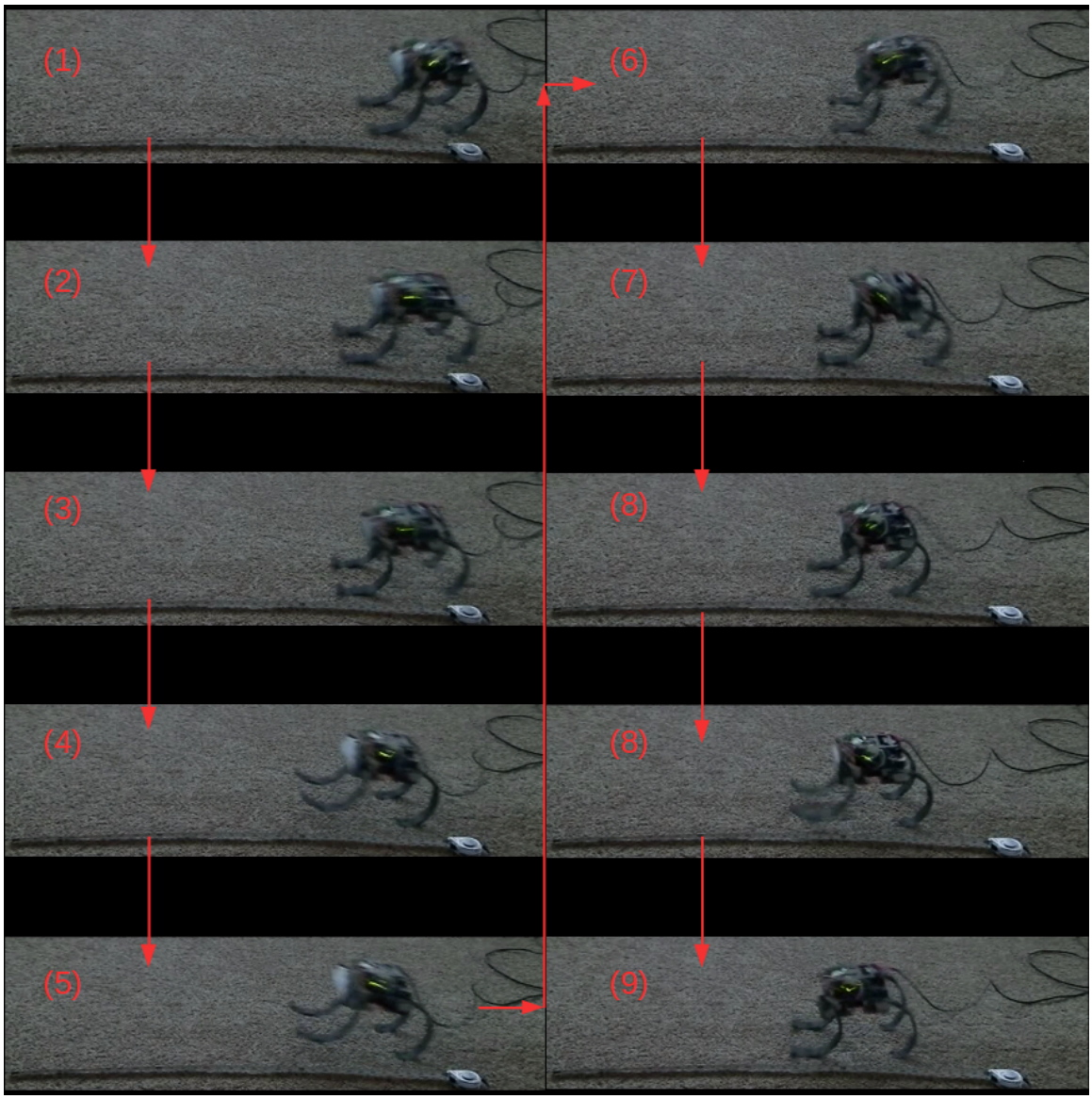

Figure 6.1. The robot roll-U-ped V2 traversing across carpet with the gait developed for V1. It received a parameterized sinusoidal input with no closed-loop control. The same gait is unable to produce effective motions on other surfaces, which is a primary downside of the parameterized trajectories in general.

open-loop to ensure the actuators were capable of the torques of speeds need to roll. Running backwards was successful at speeds approaching 0.39 m/s across a single carpet surface.

6.2 V2 Design

V2 of roll-U-ped was the final version constructed and was the result of revisions to the prototype V1. More comprehensive tests were done on the robot's capabilities over two different surfaces in addition to the carpet used in V1. Forward open-loop control gaits were constructed that can be generalized for right-side up and upside down. With slight alterations, the backwards gait developed for V1 over carpet can be run on V2 and produce a velocity approaching 0.41 m/s. The primary differences were due to the increased stiffness of the leg and the increased mass and rotational inertia of the base due to several additions. Powered rolling was also constructed using closed-loop control from remote control and a motion tracking via VICON.

6.2.1 Results

For the design of roll-U-ped, the capability for running forward and backwards was verified. Running was conducted across a variety of surfaces and the results were collected for: (1) a rough-stiff surface and (2) a flexible-tacky surface. Additionally, the capability for powered rolling was verified and the results for powered rolling implementing CL directional compensation with VICON were collected.

6.2.1.1 Bounding - Sinusoidal trajectory

For the stiff rough surface, an Ice and Water shield used in roofing was used in tests. Running on the stiff rough surface, stumbles were largely avoided because there was significant slippage. This did, however, cause it to lose speed because it slipped both when it was propelling itself forward and in the stumble. The results for running at high and low frequencies on the stiff rough surface are, respectively, given in Fig. 6.2 and Fig. 6.3. As shown in those figures, the maximum speed achieved was 0.21 m/s.

A yoga mat was used as the flexible-tacky surface. That surface had significantly less slippage, which meant arbitrary sinusoidal trajectories that could be implemented on the stiff-rough surface would knock the robot over. As a consequence, fewer responses were

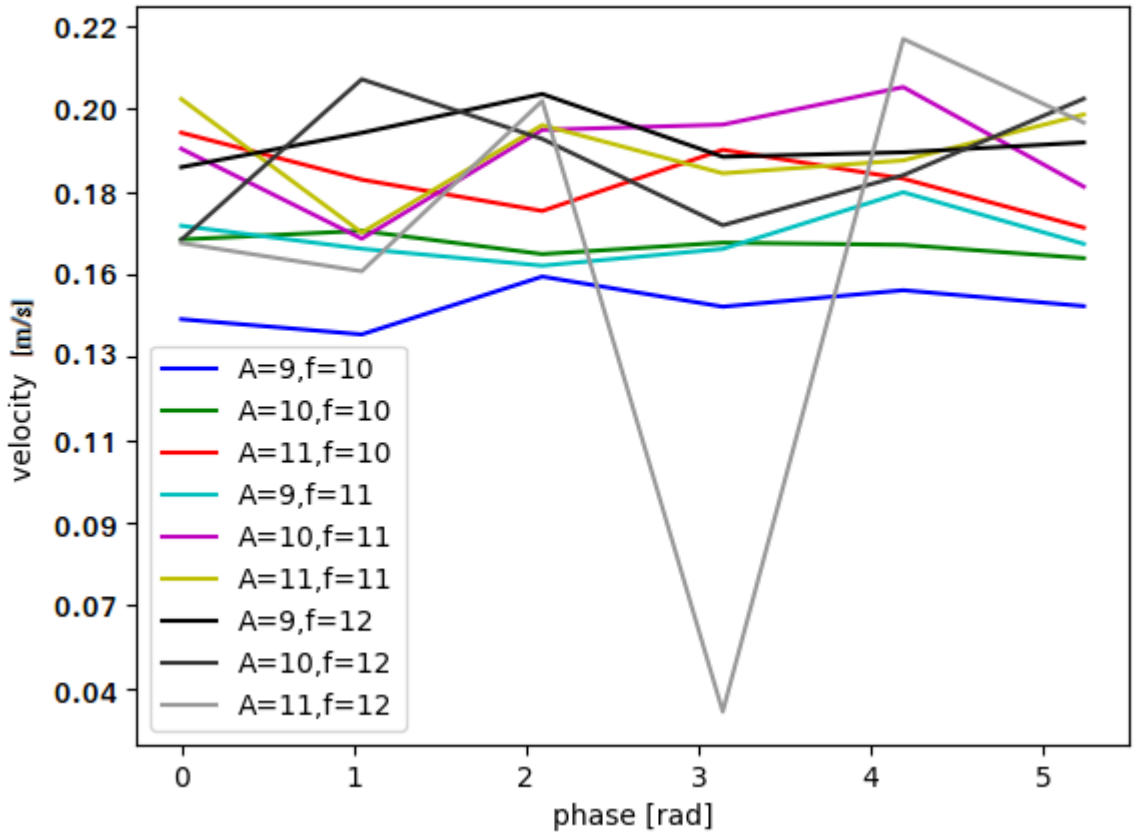

Figure 6.2. The results of running roll-U-ped V2 backwards on a flat, stiff, and rough surface. The plot shows the summary of the average velocity of a given parameter set over a period of approximately 8 seconds. The variables used in each run are specified as the phase, or p_2 while p_1 was always 0, A as A_1 and A_2, and f as f_1 and f_2 in the sinusoidal trajectory equations. The domain is greater than 2D so a set of lines with two dimensions are given.

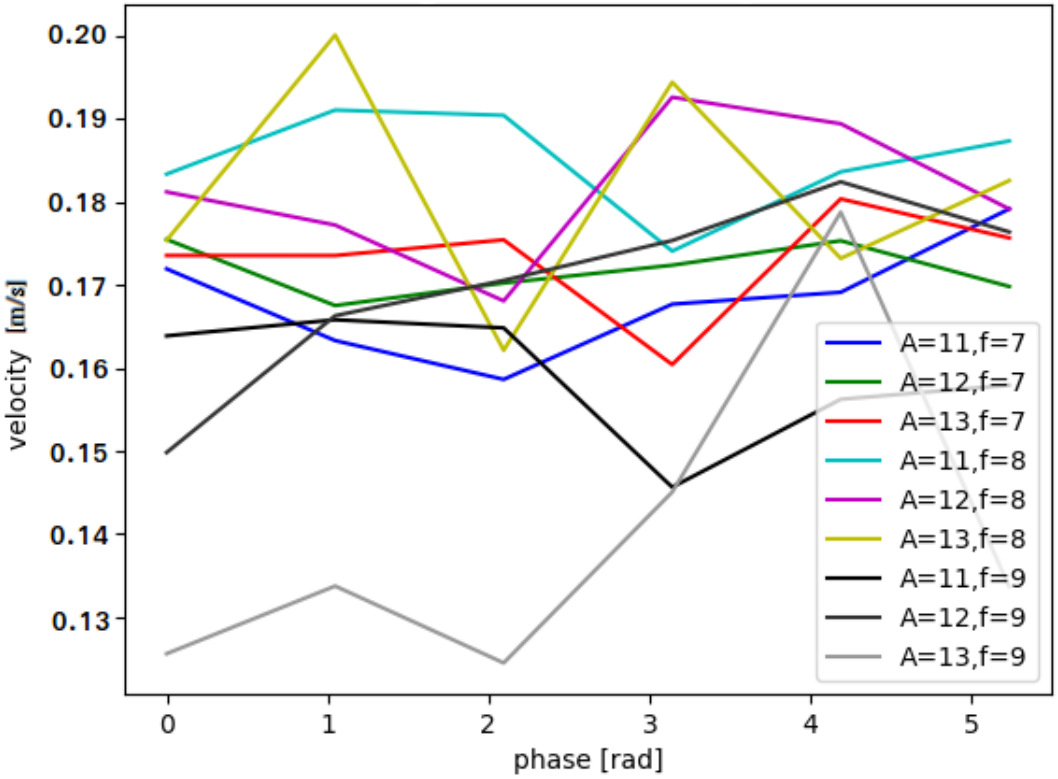

Figure 6.3. The results of running roll-U-ped V2 backwards on a horizontal, stiff, and rough surface. The plot shows the summary of the average velocity of a given parameter set over a period of approximately 8 seconds. The variables used in each run are specified as the phase, or p_2 while p_1 was always 0, A as A_1 and A_2, and f as f_1 and f_2 in the sinusoidal trajectory equations. The domain is greater than 2D so a set of lines with two dimensions are given.

collected. The results for three trajectories are given in Table 6.1. Images featuring several frames of the roll-U-ped running along a flexible tacky surface are given in Fig. 6.4 and Fig. 6.5. The maximum speed achieved using a sinusoidal trajectory along the flexible-tacky surface was 0.136 m/s.

6.2.1.2 Bounding - Q-learning

Initially, a complete gait speed of 0.1 m/s was developed with states of $\theta, \theta_1, \theta_2$ with resolutions such that the total number of states used was 45. With increasing the state resolution, Q-learning was capable of producing a gait with an initial velocity of 0.22 m/s after performing 24 runs of 2.1 seconds taking a total of 2419.5 seconds to run with a total number of states of 396. The corresponding output states, actions, and sample frames of the bounding motion in simulation are given in Figs. 6.6, 6.7, and 6.8. Several frames of roll-U-ped V2 implementing this trajectory are given in Fig. 6.9.

A revised Q-learning was developed with the addition of a cost, or a negative reward, for the velocity of the servos. The result for this was a Q-learned trajectory constructed with a revised reward of $20\dot{x}$ - $5|S_{f1} - S_{f1old}|$ - $5|S_{f2} - S_{f2old}|$. The same states and actions were chosen, again an epsilon of 0.7, alpha of 0.9, and gamma of 0.9. This was done by doing a 2.2 second run 16 times in the Q-learning. The resulting states and actions of running this in simulation are, respectively, given in Figs. 6.10 and 6.11. As is shown, the cost was successful at preventing the robot from always taking the fastest action while still generating a speed of 0.134 m/s.

Finally, Q-learning was constructed with states in the body frame, using the local variables. The chosen variables were: (1) the rotational momentum of the base, (2) the momentum of the mass at the end of the rear leg in the perpendicular direction, and (3) the momentum of the mass at the end of the front leg in the perpendicular direction. The corresponding bond graph variables are: (1) p_θ, (2) p_i, and (3) p_{iii} being the rotational momentum of the base and the rear and back legs for a total number of 252 unique states. The actions to take were six evenly spaced from -4.49 to 4.49 for both the front and back legs. The reward was again: $20\dot{x}$ - $5|S_{f1} - S_{f1old}|$ - $5|S_{f2} - S_{f2old}|$ as defined previously. Finally, the same alpha, gamma, and % chance of taking the least visited action were used as in all the previous results. The complete set of global states, the same states used

Table 6.1. The results of three variable sinusoidal trajectories, being Sine 1, Sine 2, and Sine 3, on the robot. The responses both on the physical system and in the simulation are listed as, respectively, the resulting physical setup velocity, Vel. P, and the simulated velocity, Vel. S. A_1, A_2, f, p_1, p_1, b_1, and b_2 are the parameters used in the sinusoidal trajectory given previously. Here f is the frequency being both f_1 and f_2.

	Sine 1	Sine 2	Sine 3
A_1 [deg.]	$\frac{-25}{2}$	$\frac{-40}{2}$	$\frac{-35}{2}$
A_2 [deg.]	$\frac{21}{2}$	$\frac{37}{2}$	$\frac{34}{2}$
f [1/s]	3.4	1.8	1.78
p_1 [deg.]	$\frac{180}{f}$	$\frac{180}{f}$	$\frac{180}{f}$
p_2 [deg.]	$\frac{180}{f}$	$\frac{190}{f}$	$\frac{180}{f}$
b_1 [deg.]	$-A_1+2$	$-A_1+8$	$-A_1+10$
b_2 [deg.]	A_2+13	A_2-2	A_2+6
Vel. P [m/s]	0.136	0.11	0.124
Vel. S [m/s]	0.172	0.147	0.13

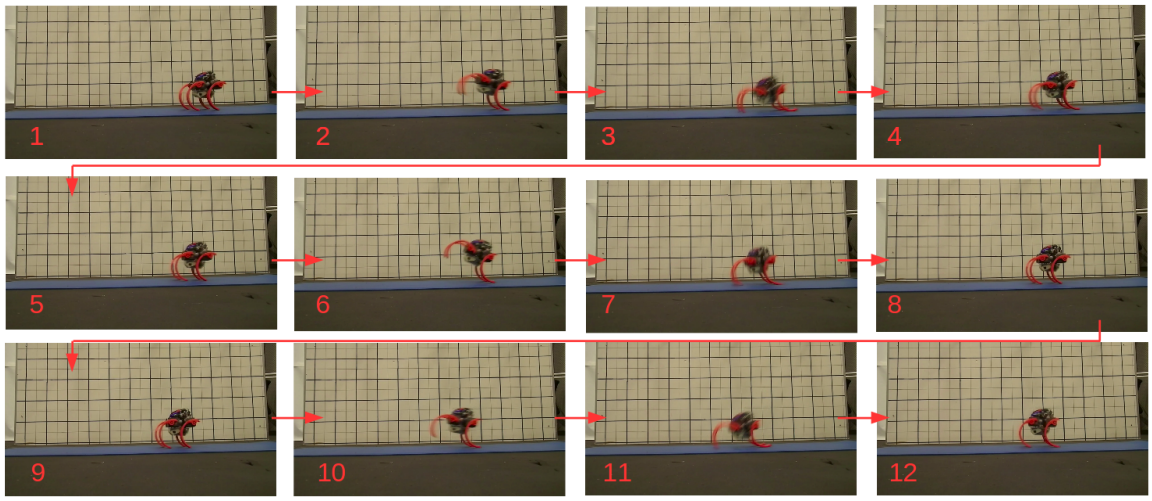

Figure 6.4. The robot roll-U-ped V2 traversing a flexible tacky surface by running. It receives a sinusoidal input with no closed-loop control except an amplitude compensation for turning toward the center of the track.

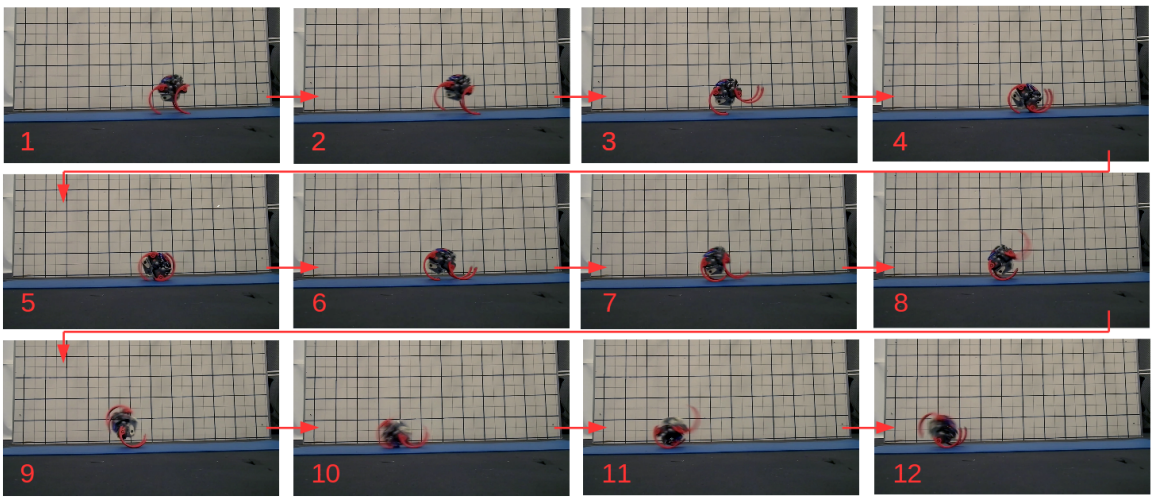

Figure 6.5. The robot roll-U-ped V2 converting from a standing state from the completion of a running path and beginning propelling through rolling. It receives a linear input from the initial standing joint angles to the final curled joint angles. Finally, the robot begins a propelled roll by extendeding the leg on the ground, curling back up, and repeating. There is closed-loop control for timing of the leg extension in addition to amplitude compensation to track a linear path.

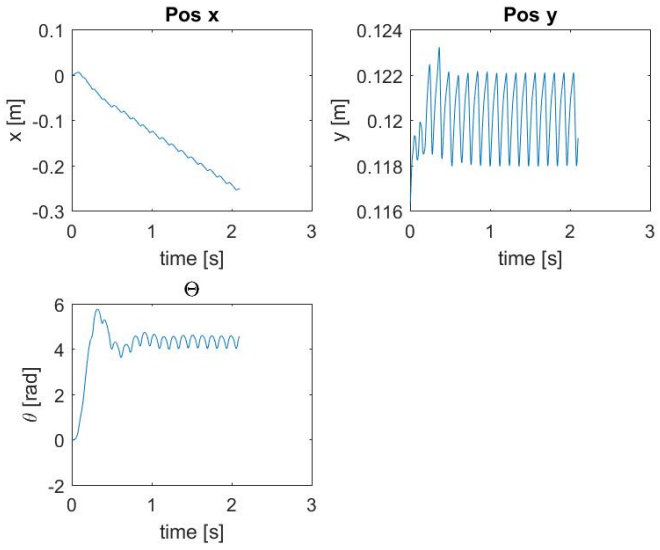

Figure 6.6. The resulting output states of position in x and y and the robot angle over time for the gait developed in Q-learning with 396 distinct states from θ, θ_1, and θ_2.

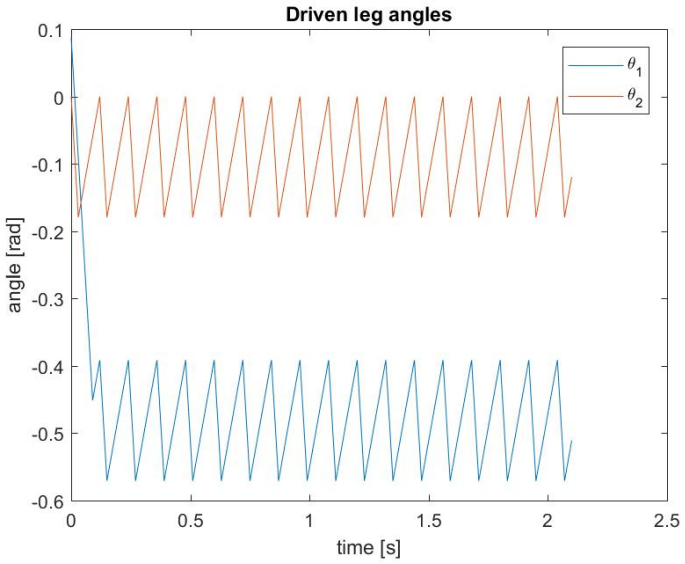

Figure 6.7. The resulting output leg angle trajectories of Q-learning for 396 distinct states from θ, θ_1, and θ_2. These were then implemented on the physical system by segmenting into initial and rhythmic movement primitives.

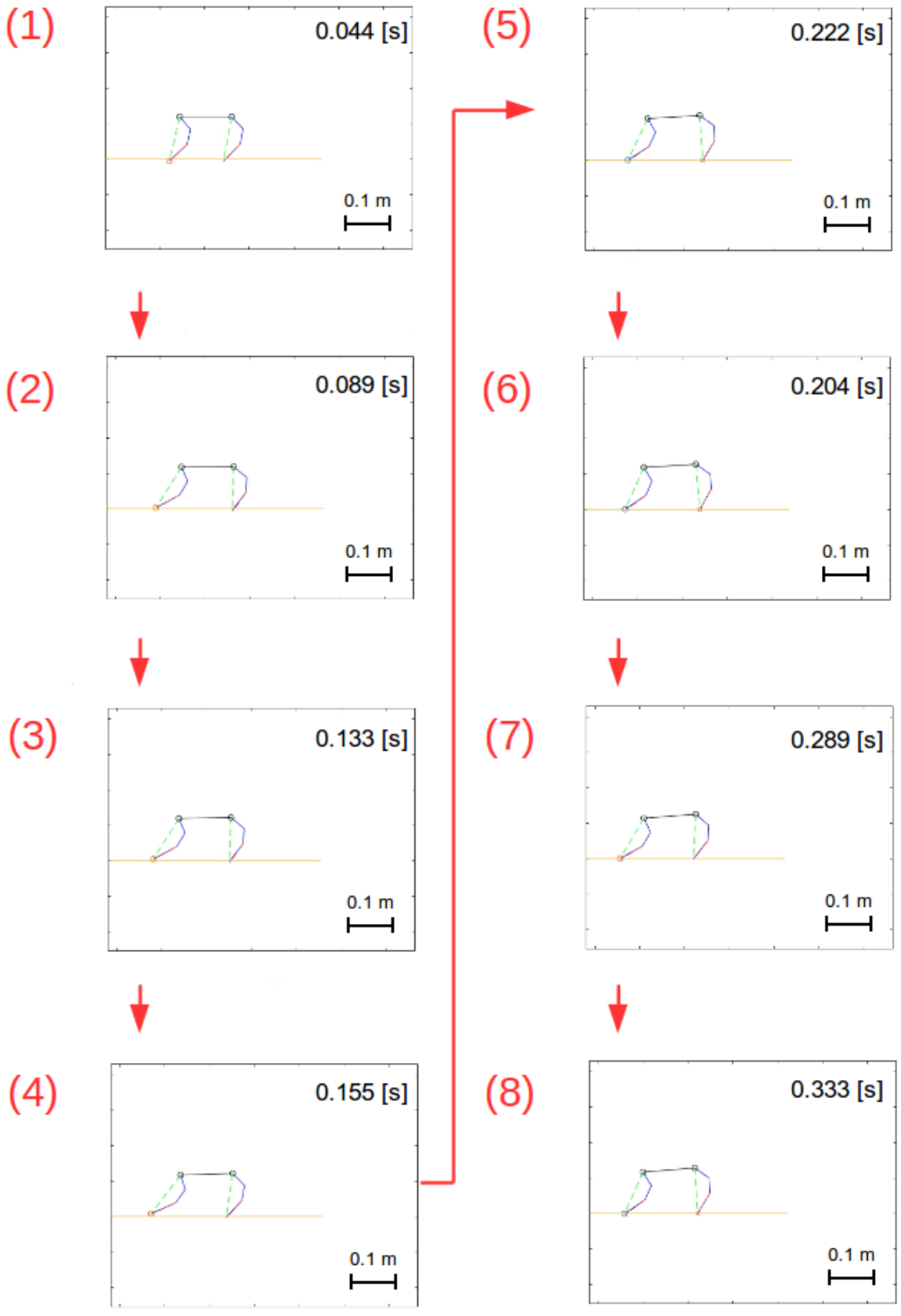

Figure 6.8. An illustration of the motion produced with snapshots of the simulation result of construcing a bounding gait in simulation. This corresponds to the same inputs and outputs of Figs. 6.6 and 6.7 with 396 distinct states of θ, θ_1, and θ_2. This same result implemented on the physical system was provided in Fig. 6.9.

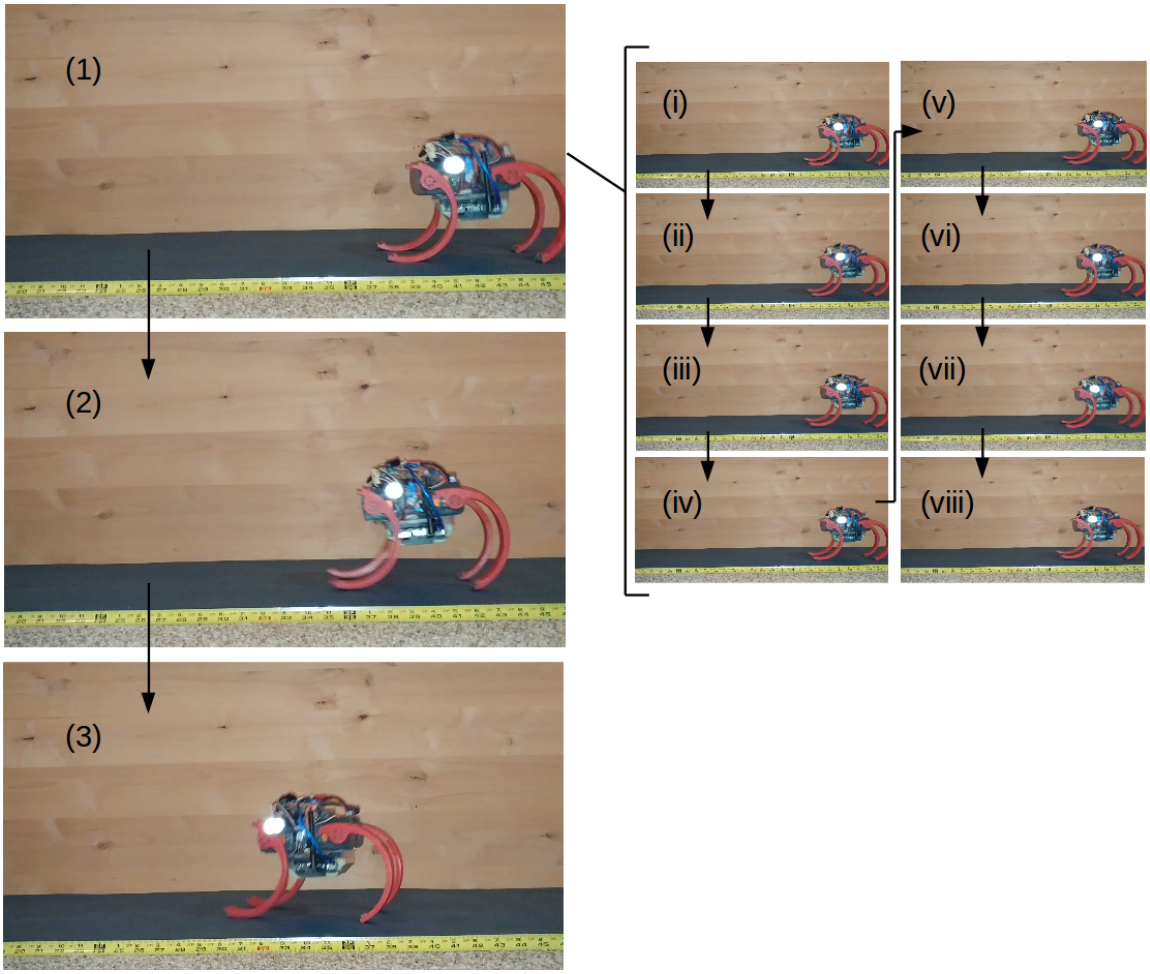

Figure 6.9. Several frames of roll-U-ped implementing the gait developed in the simulation with Q-learning. (1)-(3) illustrate the resulting output displacement. (i)-(viii) illustrate the repetitive trajectory implemented after the initial trajectory was tracked with higher resolution.

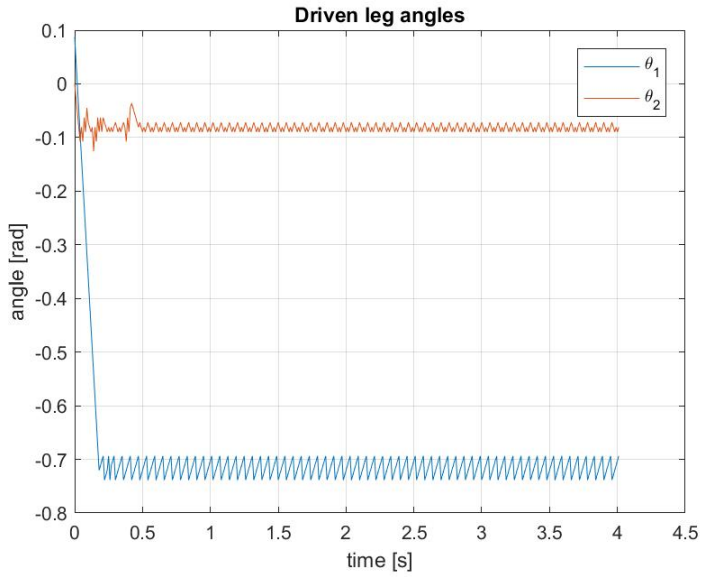

Figure 6.10. The resulting output leg angle trajectories of Q-learning for 396 distinct states from θ, θ_1, and θ_2. These can then be implemented on the physical system by segmenting into initial and rhythmic movement primitives as other Q-learned trajectories were. This trajectory was constructed with the revised reward function.

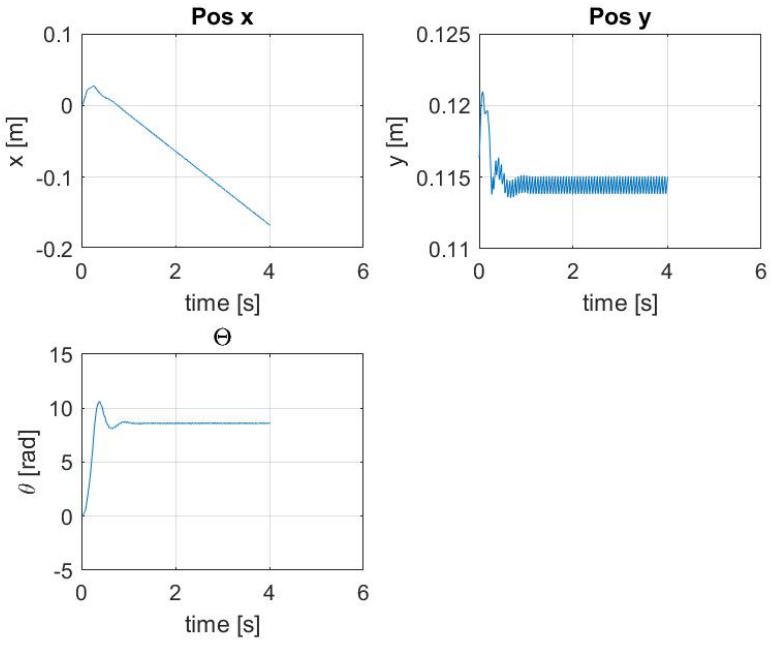

Figure 6.11. The resulting output states of position in x and y and the robot angle over time for the gait developed in Q-learning with 396 distinct states from θ, θ_1, and θ_2. This trajectory was constructed with the revised reward function.

in previous Q-learning, and the resulting actions to take are given in, respectively, Figs. 6.12 and 6.13.

6.2.1.3 Rolling

Results for rolling were all collected for the flexible tacky surface. The robot was found capable of unpowered rolling down slopes of less than -18 deg. It was additionally able to produce powered rolling at speeds approaching 0.53 m/s using VICON motion tracking. Speeds closer to 0.3 m/s were generated with remote control, which varied greatly depending on the user. Rolling was also successfully produced along the very stiff and flexible surfaces of, respectively, foam and a concrete floor.

6.2.1.4 Turning

Turning during running was accomplished by reducing the amplitude for the side of the robot into which it is desirable to turn and increasing the amplitude on the other side. This produces a turning run. A largely stationary rotation is also possible by quickly oscillating the robot's legs to induce a bounce that favors one side or the other by again producing an amplitude variance on both sides. This does induce some uncontrolled translation, but it is able to generate a turn in less linear distance than a turning run would require. Varying amplitudes for turning of approximately 15 deg. (depending on the surface) was capable of producing a turning radius of approximately 0.9 m. Powered rolling along the flexible tacky surface produced speeds approaching 0.56 m/s using CL control with the VICON motion capture system. The turning radius during rolling is approximately at 3 m, though this becomes increasingly small with lower surface stiffness. In fact, a turning radius of less than 6 in was achieved by altering the same linear trajectory from Q-learning slightly to proportionally increasing the amplitude on one side by a factor of 1.5.

6.3 Discussion

6.3.1 Design

In developing roll-U-ped, it was desirable for the robot to be 3D printed. This did, however, introduce running limitations as the legs had low friction with nearly all surfaces tested. Consequently, a thin rubberized material, Plymounth Rubber - Plywrap 20, was

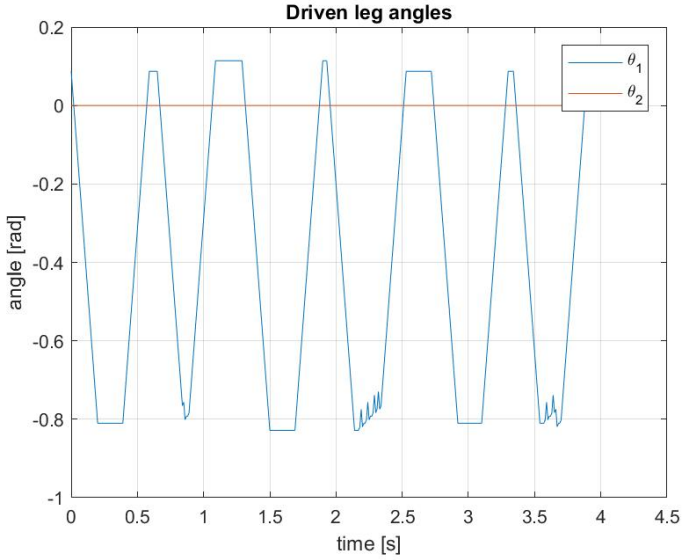

Figure 6.12. The resulting output leg angle trajectories of Q-learning for 252 distinct states from $\dot{p_\theta}$, $\dot{p_i}$, and $\dot{p_{iii}}$. These can then be implemented on the physical system by segmenting into initial and rhythmic movement primitives as other Q-learned trajectories were. This trajectory was constructed with the revised reward function.

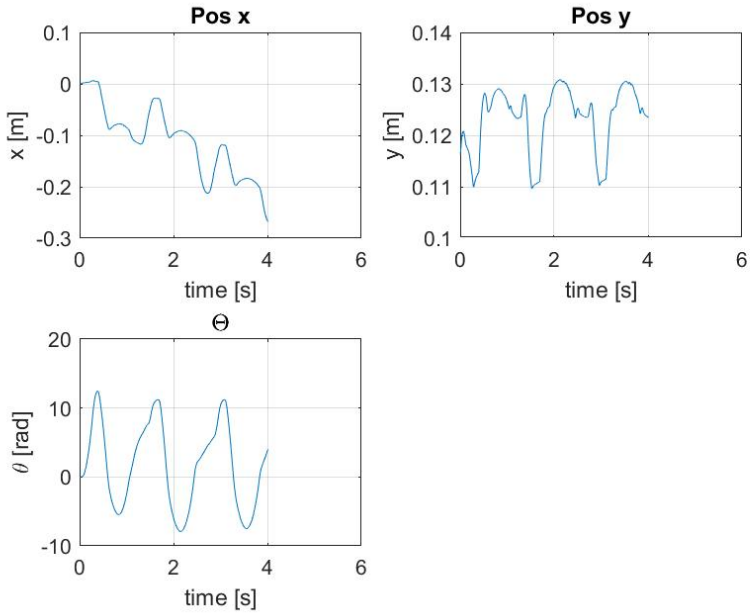

Figure 6.13. The resulting output states of position in x and y and the robot angle over time for the gait developed in Q-learning with 252 distinct states from $\dot{p_\theta}$, $\dot{p_i}$, and $\dot{p_{iii}}$. This trajectory was constructed with the revised reward function.

added to reduce slip and wear on the 3D printed leg. This is something that would have to be studied further in the future to determine an effective general-purpose durable material to place over the surface of the legs to provide an increased coefficient of friction. Additions to the surface of the leg in the 3D print, such as a serrated or bumpy surface, could also be implemented for use across materials like dirt, grass, and gravel. However, the disadvantage of this is it could damage some surfaces it runs over.

6.3.2 Locomotion

The complete set of locomotions are given in Table 6.2. As shown, powered rolling along a flat surface with CL control from a motion tracking system such as VICON is the fastest along flat ground. The disadvantage of this is that it requires the tracking system to function. This would, however, be easily remedied by adding touch sensors onto the surface of each leg to determine when to expel the legs. The rolling can also be done by a user with remote control, but it is difficult for inexperienced users to produce as high of a forward velocity. On the other hand, running forward/backwards is slower but can currently be run without the need for any external outputs and allows for the capability of moving uphill. However, a limitation of roll-U-ped is its inability to go up slopes of much more than approximately 20 deg.

Table 6.2. Methods of locomotion roll-U-ped is capable of producing with the corresponding output speed over the tested surfaces. The methods of locomotion are running forward/backwards being right-side up(RSU)/upside down(USD) and rolling forward. The type of trajectory is either a parameterized sinusoid, developed in Q-learning, or using CL leg extension control. The surfaces are either a stiff rough surface, a carpet, or a flexible tacky surface.

Method of locomotion	Gait type	Surface	Linear speed m/s
RSU running forward	Sinusoidal	Stiff rough surface	0.210
	Sinusoidal	Carpet	0.400
RSU running backward	Sinusoidal	Flexible tacky surface	0.136
	Q-learning	Flexible tacky surface	0.145
USD running forward	Sinusoidal	Flexible tacky surface	0.136
	Q-learning	Flexible tacky surface	0.145
USD running backward	Sinusoidal	Stiff rough surface	0.210
	Sinusoidal	Carpet	0.400
Powered rolling forward	CL	Flexible tacky surface	0.530

6.3.3 Gait Development

The Froude number is a dimensionless number used in mobile robots to compare across platforms. It is defined here as the following:

$$Fr = V^2/(gL) \tag{6.1}$$

where V is the velocity of the robot, g is the gravitational constant (9.81 m/s^2), and L is the length of the robot.

The Froude number for roll-U-ped in running is 0.14 and 0.246 for rolling. As a comparison, the Froude numbers for several other robots are as follows: the Cheetah Cub reaches a running trot with a maximum Froude number of 1.3 [22]; the Froude number for Scout II varies between 0.78 and 2.4 [23]; iSprawl is a hexapedal robot of length 0.153 m with a Froude number of 4.164 [24]; finally, HyQ is a hydraulically powered quadruped capable of speeds approaching 1.67 m/s with a length of 1 m resulting in a Froude number of approximately 0.283 [25] [26].

6.3.3.1 Sinusoidal Trajectory

The limitations of a parameterized sinusoidal trajectory are: (1) limited action space, (2) limited resolution, (3) time for development, and (4) the need for an informed user. The limitations of the discrete Q-learning implemented are: (1) limited action space, (2) limited resolution, and (3) robust measurements are required of the robot states. With the advantage of having a model, the requirement for robust measurements of the state space is waived, which corresponds to every advantage being on the side of Q-learning over a parameterized gait. There are additional possibilities, however. Some that could be examined in the future include evolutionary algorithms, training the model to run like functional quadrupedal robots/animals, and implementing a central pattern generator. Evolutionary algorithms have many of the advantages of Q-learning and could be implemented directly on the model in the same way. Training the model off of current quadruped robots and animals, however, would require extracting the motion made by these systems and constructing a neural net to teach roll-U-ped to learn off of them. A central-pattern-generator is a common method of gait development on quadrupeds, and it would be advantageous to have a system for comparing it with the other methods

implemented.

6.3.3.2 Q-learning

The Q-learning gait developed implemented high-frequency, low-amplitude oscillations to generate a fast trajectory. Although this can be successful, as shown by the effective results of running the trajectory on the physical robot, the reason quadrupedal robots and animals typically do not implement such a trajectory is the high torque/speed requirement. A trajectory more like what would appear naturally could be developed by including the torque and velocity of the actuators as a cost, or negative reward, in the Q-learning. That way it would learn to limit the effort made while maximizing the velocity.

There was a difference between the results in simulation and on the physical system. The reasons for the errors include the limitations of the model, discussed in Chapter 4, in addition to some that particularly effect this trajectory. The faster contact with the ground compared to those over which the grounds response was linearized provides a source of error. Additionally, the servos only have a resolution of approximately 1 deg. This has little effect on a trajectory with an amplitude near 40 deg. It does, however, have a much larger effect on a trajectory with an amplitude closer to 10 deg. Something else that could be incorporated to improve results is adding some randomness to the ground response to account for imperfections on the surface and leg. Having this would allow the Q-learning from simulation to better function on the physical system.

6.3.4 Rolling

For practical purposes, transitioning to rolling for a short length of time is ineffective. The primary reasons for implementing running is to climb inclines or traverse flat and even ground. The primary reason for using rolling is to passively roll down slopes of less than -18 deg. or to actively roll along low inclined tumultuous surfaces such as gravel or foam. Combined, these allow for the traversal of most surfaces that would be encountered in use of roll-U-ped as a surveillance, entertainment, research platform, or search and rescue device. The primary limitations are the inability to climb inclines of greater than about 20 deg. (depending on the surface), climb a slick incline, or traverse gaps of much more than four inches. An additional advantage the rolling aspect of roll-U-ped has over many of the current quadrupedal robots is the ability for it to be tossed in a roll by a user. This

allows it to be put in a terrain where another robot may have to be transported in and set on the ground in a specific orientation to function. Finally, it is proposed that roll-U-ped should be capable of rolling backwards by using rapid dynamics to produce a reverse spin of the robot by sliding the end of its leg along the ground rather than pushing off with the legs outside surface. This has not been tested and is an avenue to be further researched.

6.3.5 Drawback of Current Design

A limitation of roll-U-ped is its inability to recover if it has fallen on its side. A simple revision to the design of attaching a triangular end to each side of the robot would allow the robot to stand back up when it would have fallen permanently onto its side with the current design. The problem with this is it introduces additional weight onto the base and reduces the minimum space the robot can fit through. This is an addition that could be made in the future but the drawbacks were determined to out-weigh the upsides for the purposes of the prototypes constructed.

An additional limitation is the method of running used. Bounding can be effective but its implementation here with only a single actuator for each leg restricts the robot's ability to maneuver. As a consequence, roll-U-ped is not able to specify foot placement to traverse dangerous terrain, twist in place without altering foot placement, or handle objects.

CHAPTER 7

CONCLUSIONS AND FUTURE RESEARCH

This thesis focused on the design, modeling, and gait control of a new rolling/running quadruped robot called the roll-U-ped. The robot is a hybrid-mobility quadruped capable of locomotion via running forward, running backwards, unpowered rolling bidirectionally, and powered rolling forward. The design process was outlined and two prototypes were constructed, V1 and V2. A model was constructed using a lumped parameter model and a bond graph to develop differential equations to solve with a stiff differential equation solver. The simulation was verified for the following conditions: robot does not fall over, the forces on the legs do not extend past linearization, and the robot is running over the flexible tacky surface it was developed for. Experimentally derived trajectories were found for roll-U-ped across three surfaces: a stiff rough surface, a flexible tacky surface, and carpet. A parameterized trajectory for getting the robot to run at 0.43 m/s was constructed for roll-U-ped V2 across carpet. A parameterized trajectory was also developed for running across a flexible tacky surface allowing it to run at a speed of 0.14 m/s. Additionally, for running across a stiff rough surface, a second trajectory was constructed capable of producing a speed of 0.21 m/s. Using the model, Q-learning was implemented to develop an additional trajectory capable of running at 0.22 m/s in simulation and 0.15 m/s on the physical system. Rolling using CL control with the VICON motion capture system was able to generate speeds approaching 0.53 m/s with some variance depending on the accuracy of the pitch angle measured, while an experienced user with the remote control would produce a slower speed. Additionally, roll-U-ped is also capable of running for over an hour with a 2200 mAh battery. This gives roll-U-ped several advantages over other designs as it is able to run for such an extensive period, handle a wider range of circumstances, and reduce energy costs because of its rolling capacity.

Further research can be done to implement the roll-U-ped model and Q-learning algo-

rithm for the development of a new design by iterating through the design parameters of length and inertia of the base in addition to the stiffness and damping of the legs. These would allow for the informed design in generating a new roll-U-ped prototype that is more effective than the current design.

REFERENCES

[1] C. J. Dudley, A. C. Woods, and K. K. Leang, "A micro spherical rolling and flying robot," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5863–5869, September 2015. [Online]. Available: http://doi.org/10.1109/IROS.2015.7354210

[2] K. C. Galloway, G. Haynes, A. M. Johnson, R. Knopf, G. Lynch, and B. Plotnick, "X-rhex: A highly mobile hexapedal robot for sensorimotor tasks," November 2010. [Online]. Available: https://kodlab.seas.upenn.edu/uploads/Main/xrhextechreport.pdf

[3] P. Chatzakos and E. Papadopoulos, "Bio-inspired design of electrically-driven bounding quadrupeds via parametric analysis," *Mechanism and Machine Theory*, vol. 44, no. 3, pp. 559–579, March 2009. [Online]. Available: https://doi.org/10.1016/j.mechmachtheory.2008.08.007

[4] H.-T. Lin, G. G Leisk, and B. Trimmer, "Goqbot: A caterpillar-inspired soft-bodied rolling robot," *Bioinspiration and Biomimetics*, vol. 6, no. 2, June 2011. [Online]. Available: http://doi.org/10.1088/1748-3182%2F6%2F2%2F026007

[5] N. Tan, R. E. Mohan, and K. Elangovan, "Scorpio: A biomimetic reconfigurable rolling-crawling robot," *International Journal of Advanced Robotic Systems*, vol. 13, no. 5, pp. 1–16, Sage Publishing, September-October 2016. [Online]. Available: https://doi.org/10.1177/1729881416658180

[6] T. Aoki, S. Ito, and Y. Sei, "Development of quadruped walking robot with spherical shell-mechanical design for rotational locomotion," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5706–5711, September 2015. [Online]. Available: https://doi.org/10.1109/IROS.2015.7354187

[7] R. Kinugasa and Y. Usami, "How fast can a human run? - bipedal vs. quadrupedal running," *Frontiers in Bioengineering and Biotechnology*, June 2016. [Online]. Available: https://doi.org/10.3389/fbioe.2016.00056

[8] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, "Bigdog, the rough-terrain quadruped robot," *The International Federation of Automatic Control (IFAC)*, vol. 41, no. 2, pp. 10 822–10 825, July 2008. [Online]. Available: https://doi.org/10.3182/20080706-5-KR-1001.01833

[9] S. Patnaik, "Analysis of theo jansen mechanism (strandbeest) and its comparative advantages over wheel based mine escavation system," *IOSR Journal of Engineering (IOSRJEN)*, vol. 05, no. 07, pp. 2278–8719, July 2015.

[10] N. B. Ignell, N. Rasmusson, and J. Matsson, "An overview of legged and wheeled robotic locomotion," 2012.

[11] P. G. de Santos, E. Garcia, and J. Estremera, *Quadrupedal Locomotion: An Introduction to the Control of Four-legged Robots*. Springer-Verlag London, 2006. [Online]. Available: https://doi.org/10.1007/1-84628-307-8

[12] K. Narioka, A. Rosendo, and A. Sproewitz, "Development of a minimalistic pneumatic quadruped robot for fast locomotion," *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 307–311, December 2012.

[13] S. Oak and V. Narwane, "Design, analysis and fabrication of quadruped robot with four bar chain leg mechanism," *International Journal of Innovative Science, Engineering and Technology (IJISET)*, vol. 1, no. 6, August 2014.

[14] A. Sproewitz, M. Fremerey, and K. Karakasiliotis, "Compliant leg design for a quadruped robot," *Proceedings of Dynamic Walking 2009*, January 2009.

[15] D. V. Lee and S. G. Meek, "Directionally compliant legs influence the intrinsic pitch behaviour of a trotting quadruped," *Proceedings. Biological Sciences*, pp. 567–572, March 2005. [Online]. Available: https://doi.org/10.1098/rspb.2004.3014

[16] I. Poulakakis, J. A. Smith, and M. Buehler, "Modeling and experiments of untethered quadrupedal running with a bounding gait: The scout ii robot," *International Journal of Robotics Research*, vol. 24, no. 4, pp. 239–256, April 2005. [Online]. Available: https://doi.org/10.1177/0278364904050917

[17] R. Akagi, Y. Nakagawa, and K. Narioka, "Realization of various gaits by pneumatic driven quadruped robot," *The Proceedings of JSME annual Conference on Robotics and Mechatronics (Robomec)*, pp. 1–4, May 2015. [Online]. Available: https://doi.org/10.1299/jsmermd.2015._2A2-O03_1

[18] Y. Tanaka, S. Skama, and K. Nakano, "Comparative study on dynamic characteristics of hydraulic, pneumatic and electric motors," *Proceedings of the ASME/BATH 2013 Symposium on Fluid Power and Motion Control (FPMC 2013)*, October 2013. [Online]. Available: https://doi.org/10.1115/FPMC2013-4459

[19] V. A. Crossley, "A literature review on the design of spherical rolling robots," pp. 1–6, January 2006.

[20] J. Zhang, F. Gao, and X. Han, "Trot gait design and cpg method for a quadruped robot," *Journal of Bionic Engineering*, vol. 11, no. 1, pp. 18–25, January 2014. [Online]. Available: https://doi.org/10.1016/S1672-6529(14)60016-0

[21] F. Tedeschi and G. Carbone, "Design issues for hexapod walking robots," *Robotics*, vol. 3, no. 2, pp. 181–206, June 2014. [Online]. Available: https://doi.org/10.3390/robotics3020181

[22] A. Sprowitz, A. Tuleu, and M. Vespignani, "Towards dynamic trot gait locomotion: Design, control, and experiments with cheetah-cub, a compliant quadruped robot," *International Journal of Robotics Research (IJRR)*, vol. 32, no. 8, pp. 932–950, June 2013. [Online]. Available: https://doi.org/10.1177/0278364913489205

[23] P. Chatzakos and E. Papadopoulos, "Self-stabilising quadrupedal running by mechanical design," *Applied Bionics and Biomechanics*, vol. 6, no. 1, pp. 73–85, May 2009. [Online]. Available: https://doi.org/10.1080/11762320902863908

[24] S. Kim, J. E. Clark, and M. R. Cutkosky, "isprawl: Design and tuning for high-speed autonomous open-loop running," *The International Journal of Robotics Research (IJRR)*, vol. 25, no. 9, pp. 903–912, September 2006. [Online]. Available: https://doi.org/10.1177/0278364906069150

[25] E. Guizzo, "Hyq quadruped robot from italy can trot, kick," *IEEE Spectrum*, October 2011. [Online]. Available: https://spectrum.ieee.org/automaton/robotics/industrial-robots/hyq-quadruped-robot

[26] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell, "Design of hyq - a hydraulically and electrically actuated quadruped robot," *Proceedings of the Insitution of Mechanical Engineers Part I: Journal of Systems and Control Engineering*, vol. 225, no. 6, pp. 831–849, October 2011. [Online]. Available: https://doi.org/10.1177/0959651811402275