Mobility Characterization and Control of a Spherical Multi-Legged Robot

Bryan Ellis Wagenknecht

CMU-RI-TR-09-46

Submitted in partial fulfillment of the
requirements for the degree of
Master of Science in Robotics.

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

December 18, 2009

Copyright © 2009 by Bryan Ellis Wagenknecht. All rights reserved.

Abstract

Mobility in rough terrain has long been a hindrance in ground-based robotic exploration. Historically, such tasks have been approached with wheeled platforms, which have difficulty surmounting obstacles of even moderate height, or with legged robots that can step over obstacles but require complex foot-placement planning and posture control. In this thesis, I present the latest Robotic All-Terrain Surveyor (RATS) prototype as a unique approach to locomotion in varied terrain through the ability to emulate multiple mobility modes using a single type of actuator. The novel robot has a spherical body roughly the size of a soccer ball with 12 legs symmetrically distributed around its surface. Each leg is a single-DOF pneumatic linear actuator, oriented normal to the spherical body. Thorough investigation of this prototype's mobility and actuation behavior has demonstrated the feasibility of achieving tipping, hopping, and prolonged rolling behavior without the need for controlling body posture. Here I summarize the experimental results of this characterization and present an understanding of the system's performance limitations in an effort to draw insight for controlling its movements. I also discuss the initial control strategies that have been developed to achieve multi-modal RATS locomotion, both in simulation and with the prototype. This paper concludes with a discussion of the numerous areas for future development to improve on the robot's mobility performance and implement more sophisticated control techniques for locomotion.

Acknowledgements

I would first like to thank my advisor Dr. Dimi Apostolopoulos. His constantly positive outlook, helpful guidance, and excellent insight were always available when I needed it, but he also encouraged me to push the RATS project in the direction I saw fit. For that granted freedom and the invaluable experience I gained as a result, I will always be grateful. His assistance in the shaping of this thesis is much appreciated as well.

I also want to thank the other members of the RATS team, past and present. The work of Mike Wagner, Stuart Heys, Allan Lüders, and others laid the foundation for the present course of RATS research. I am especially indebted to Mike and Allan for their help in getting me up to speed when I first joined the team at NREC. The efforts of James Teza and Benjamin Koopman have been essential in developing the latest prototype to its current untethered state, which enabled the successful results in this thesis.

I owe many thanks to the other members of my thesis committee, Dr. David Wettergreen and Amir Degani, for their willingness and guidance in helping me take care of the final requirements for this degree.

I thank Minas Tanielian and others at Boeing Corporation for the financial support that made the RATS program possible.

Finally, I thank my family and friends for their encouragement to pursue my dreams. This thesis and work that has gone into it would not have been possible without their unwavering support along the way.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The *Robotic All-Terrain Surveyor* (RATS) is a novel robot designed to navigate in varied terrain at high speeds. As a new approach to the hopping robot concept, RATS uses its unique spherical geometry and simple actuators to achieve diverse locomotive capabilities regardless of landing posture. It has 12 single-DOF pneumatic piston actuator legs that are evenly distributed around the surface of its spherical body and are oriented to point outward, normal to the sphere. Controlling the sequence, timing, and duration of leg actuation enables three distinct modes of locomotion for different terrain scenarios. The robot can execute discrete tipping/walking for high-precision maneuvers, hopping for obstacle avoidance, and high-speed rolling/running in open terrain. Additionally, the symmetry inherent in the RATS design means the robot is stable and equally capable of motion regardless of its initial orientation. Such locomotion is only possible with a thorough understanding of the actuation mechanism and the dynamic motion characteristics of the robot. This thesis details the mobility characterization of RATS and describes the development of control techniques for its multi-modal locomotion.

1.1 The RATS Project

The RATS project at the National Robotics Engineering Center at CMU was started with the help of Boeing Corporation to investigate potential mechanisms for pneumatic locomotion. The goal was to develop a robot that could hop over obstacles and handle rough terrain. Energy exchange techniques, such as using a catalyzed hydrogen peroxide reaction to produce high pressure steam, were investigated early in the project's history. Considerable time was also spent studying such issues as scale, geometry, and leg piston design. Since then, the project goals have expanded to include the development of a full robotic system capable of multi-modal locomotion. The project has studied the design and arrangement of pneumatic actuators and has explored the control problems involved with discrete leg actuation and spherical symmetry.

This thesis presents my work toward an understanding of how RATS can move and the

development of effective locomotion strategies. Given the initial fabricated prototype, it was crucial to investigate the mobility behavior of this robot to understand the limitations of its performance and draw insight for controlling its movements. The system behavior was characterized by studying how different combinations of leg actuation affected motion, and exploring how leg strength varied with such variables as gas pressure and valve opening duration. Details of the prototype's mechanical implementation, such as foot geometry and pneumatic control valves, were investigated and a few design improvements were introduced to enhance the controllability and mobility of the system. Throughout this characterization, I demonstrated the feasibility of achieving tipping, hopping, and prolonged rolling behavior with the prototype. I also revealed some key principles of actuator and system behavior that must be addressed for successful control of the robot.

Building on work from prior generations of RATS prototypes, I investigated techniques to generate locomotive behavior in the spherical, pneumatic version of RATS. A controller was developed to plan the proper firing sequence of legs for the robot to robustly follow a desired path using its tipping or walking mode of locomotion. Another controller was designed to generate high-speed rolling behavior in simulation that could be steered to follow desired headings or straight line paths. As on-board computing and sensing capabilities of the prototype were added, implementation of these controllers was begun on the real hardware. At the writing of this thesis, there are many opportunities for continued work in the implementation of sensing and control to achieve the full locomotion objectives of the project.

1.2 Thesis Overview

Chapter 2 of this paper discusses previous work in mobile robotics that relates to the present embodiment of the RATS concept. An overview of prior work in the RATS project is also included for context. A description of the spherical prototype follows in Chapter 3, including details on the robot's geometry, pneumatic actuation system, and electronics/control architecture. Next, the experiments used to characterize the physical system are summarized in Chapter 4 with a discussion of how the results contributed to design improvements and control insight. Chapter 5 presents a simulation framework that was established to allow development of controls. Some initial concepts for the control of the 12-legged prototype are given in Chapter 6. Results from the implementation of the controls on the prototype are discussed, along with results from the simulation work. The paper will conclude with an assessment of the lessons and observations that can be drawn from this work in Chapter 7 and a discussion of further areas of investigation for the future of the RATS project in Chapter 8.

Chapter 2

Related Work

The unique combination of geometry and actuator mechanism found in the RATS concept is intended to enable mobility in diverse and rugged terrain. Its spherical shape with 12 simple, pneumatic piston legs permits mobility that covers a wide range of robot locomotion, including rolling, walking, and hopping. To understand the advantages of the RATS mobility system, it is helpful to first review other examples of robot mobility. RATS displays similarities with the principles and mechanisms described here, but overcomes many of the locomotive drawbacks of these other systems by applying its unique blend of mobility modes.

2.1 Traditional Ground Robots

Planetary exploration and ground locomotion has traditionally been the domain of wheeled robots. But there are serious drawbacks when these robots must maneuver through rough terrain or encounter obstacles. Standard wheeled robots have significant difficulty climbing over obstacles and must rely on long-travel suspensions or highly articulated chassis to handle discontinuous terrain. Even with such modifications, wheeled locomotion often eliminates whole categories of terrain from the exploration space. This includes sites such as craters, boulder fields, rubble piles, and caves that tend to be of high value for the mission objective.

Mobile robots with articulated legs have been the next popular choice because they make discrete contacts with the ground and so can step over and through obstacles. The robustness of such robots to unexpected terrain features and issues with stability and balance still make legged locomotion a very difficult challenge. BigDog, developed by Boston Dynamics, employs sophisticated controllers to adapt to terrain changes by maintaining body posture relative to the ground while executing coordinated leg movements (Fig. 2.1) [1]. Of course, the benefits of such systems with many degrees of freedom come at the expense of high complexity due to the involvement of many actuators and the necessity for planning and control in high dimensional joint space.

Figure 2.1: Boston Dynamics' BigDog is a legged robot that handles rough terrain by reactively maintaining its body posture [1].

Different terrains call for different locomotion techniques. In flat terrain, wheeled locomotion is the most efficient method. Once the terrain becomes discontinuous, such as traveling through boulder fields or in craters, the ability to interact with the ground at discrete locations becomes valuable. Stepping or hopping over obstacles is much faster than driving around them. It is for these reasons that the RATS project has sought to develop a robot that can take advantage of rolling on the flats and use discrete leg/ground interactions when the terrain requires. Doing so without greatly increasing the control complexity further enables high-speed locomotion in diverse terrain. The following sections describe some alternative approaches to rolling or hopping that have similarities with the RATS concept.

2.2 Spherical Rolling Robots

Instead of using wheels, spherical robots employ their entire body exterior as a rolling surface. In the past, they have been propelled by shifting the center of mass [2] or by manipulating the angular momentum of internal flywheels [3]. Such robots significantly increase their maneuverability over that of wheeled platforms while potentially reducing the footprint of the robot to allow navigation in tighter spaces. Spherical symmetry also enables effective locomotion without regard for orientation. Additionally, spherical robots experience a reduction in rolling resistance that is especially advantageous on downhill slopes and smooth surfaces. Unfortunately, they still face the same challenge of obstacle avoidance that impedes wheeled robots in rough terrains.

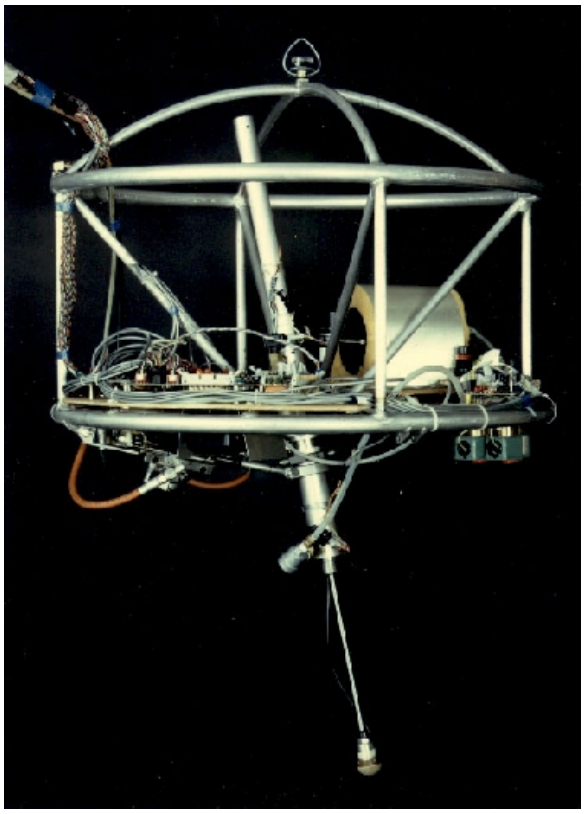


Figure 2.2: Three-dimensional hopping robot developed by Raibert, et al. uses a double-acting air cylinder for continuous hopping [5].

2.3 Single-Leg Hopping Robots

A large variety of single-legged hopping robots have been developed to employ locomotion other than rolling. Hopping can be an efficient mode of locomotion in rough terrains with obstacles and in microgravity environments where rolling traction is difficult to maintain. The actuation mechanisms involved in these robots varies, as does the nature of the motion they can achieve. Matsuoka's hopping robot used a high-force electric solenoid to achieve dynamically stabilized, continuous hopping [4]. Raibert used a double-acting air cylinder to achieve similar continuous hopping (Fig. 2.2) [5]. Other robots use discrete hops, requiring a pause between each hop to reset their propulsion mechanism. Sandia National Laboratory used a combustion-driven piston for hopping [6]. Others use energy stored elastically in a metal coil spring [7] or in a fiberglass leaf spring (Fig. 2.3) [8], which must be retracted after each hop.

A common theme in these robots is the importance of body orientation for directional control and efficiency of motion. Consequently, posture control during flight and self-righting mechanisms/procedures after landing are crucial for sustained locomotion in hoppers. It is this extra consideration that increases control complexity and may limit the speed at which the robots can travel.

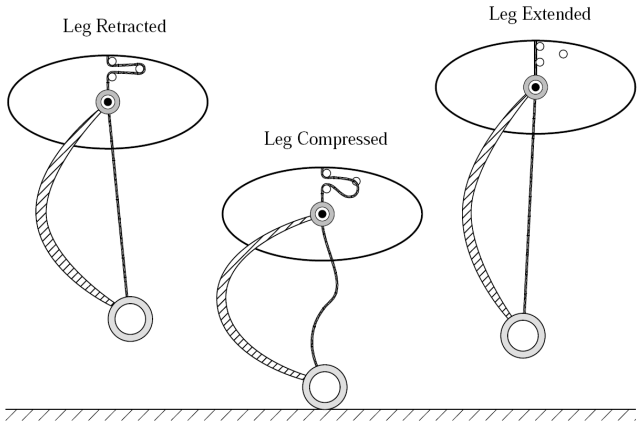


Figure 2.3: Bow leg hopping robot from CMU stores energy in a compressed leaf spring to achieve hopping [8].

2.4 Combined Mobility Robots

Several robot systems have already been developed to use multi-modal locomotion. Some are wheeled robots with discrete hopping ability to jump over obstacles in otherwise flat terrain. One example is the University of Minnesota's Scout robot (Fig. 2.4), which uses a spring wrapped around its body for jumping and relies on its cylindrical shape to ensure it lands on its wheels [9]. The 3rd generation hopping robot at JPL/Caltech (Fig. 2.5) uses an active linkage to return to hopping posture after landing [10]. Both of these robots use their wheels to reorient and hop in the desired direction.

The Jollbot, shown in Fig. 2.6, is a spherical rolling robot that can also hop [11]. Like other spherical robots, it uses a round external cage to roll and reorient itself by shifting its center of mass. Rather than employing a rigid outer surface, however, Jollbot can execute hops through the sudden release of the energy stored by elastically deforming its outer cage.

Other robots use combinations of wheels and legs in rough terrain. The IMPASS robot at Virginia Tech uses rimless wheels with linearly extending spokes to handle discontinuous terrain (Fig. 2.7) [12]. JPL's ATHLETE platform has 6 articulated legs for stepping over obstacles (Fig. 2.8), but it also has motorized wheels at the ends of these legs for smooth and efficient rolling on the flats [13].

These examples of combined mobility illustrate an improvement in rough terrain locomotion without sacrificing efficiency on smooth surfaces. They also achieve such capability with relatively simple combinations of mechanisms, some of which are used for multiple purposes while locomoting.

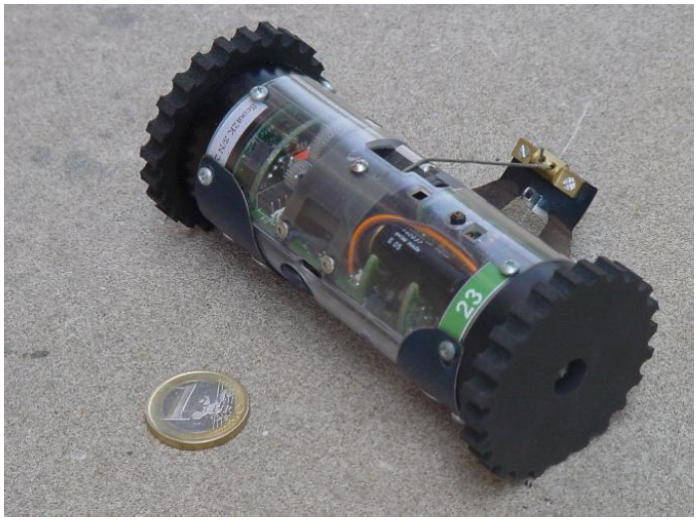


Figure 2.4: Scout robot from University of Minnesota has a cylindrical body with two wheels for efficient rolling and a spring-loaded foot for jumping [9].

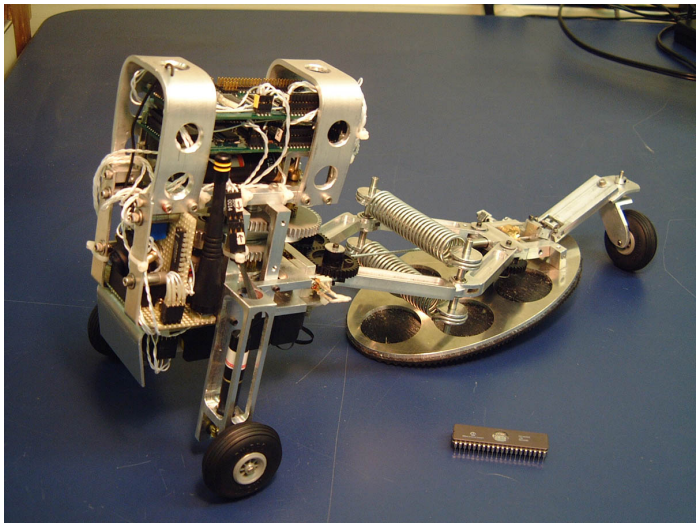


Figure 2.5: 3rd-generation hopping robot from JPL/Caltech uses wheels to move in flat terrain and discrete hopping ability to clear obstacles [10].

Figure 2.6: Jollbot from University of Bath elastically deforms its rolling cage to store energy for hopping [11].

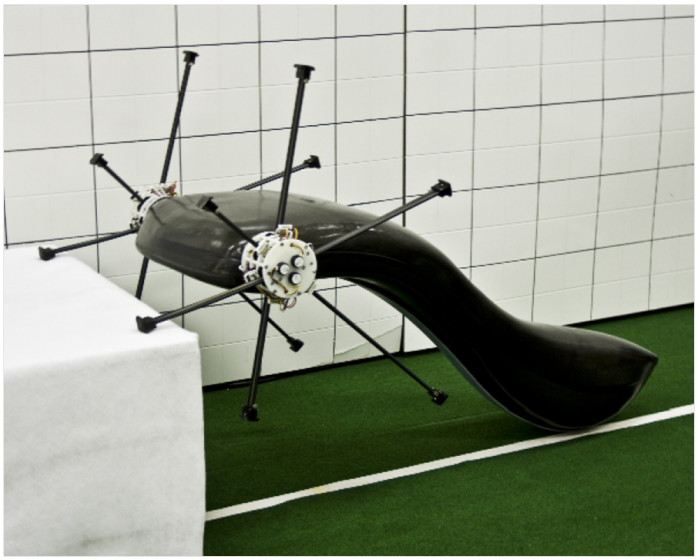


Figure 2.7: IMPASS from Virginia Tech uses leg wheels with linearly extending spokes to handle terrain with discontinuities [12].

Figure 2.8: NASA's ATHLETE robot has wheels on the ends of its 6 articulated legs for efficient rolling and stepping over obstacles [13].

2.5 Relationship to RATS

The 12-legged pneumatic RATS prototype takes combined mobility to the next level as it incorporates features from spherical, legged, and hopping robots. It combines the external linearly extending leg of hopping mechanisms with the rolling behavior used by spherical robots and the sequencing of leg actuations like a legged robot. Its round shape is beneficial for high speed motion and rolling down inclines. Its discrete hopping ability allows for obstacle avoidance. Controlled, precision positioning or navigation is possible using its tipping or walking mode. The elegance of the RATS design lies in its ability to achieve varied locomotion through the use of 12 identical actuators that are simple and robust. The symmetry arising from the geometric arrangement of its legs also eliminates the need for pose control or a self-righting mechanism, providing true omni-orientational mobility. This fusion of concepts gives RATS unique locomotive abilities in highly diverse terrains.

2.6 Previous RATS Prototypes

The current embodiment of RATS is the continuation of five years of previous research and development. Two prototype robots have come before the 12-legged pneumatic prototype that is discussed in this paper.

2.6.1 5-Legged Pneumatic Prototype

The first mechanical prototype in the RATS project was a 5-legged planar robot with 1-DOF pneumatic piston legs (Figure 2.9). The legs were spring-returned, single-acting air cylinders, arranged pointing radially outward to form a wheel that could spin freely on

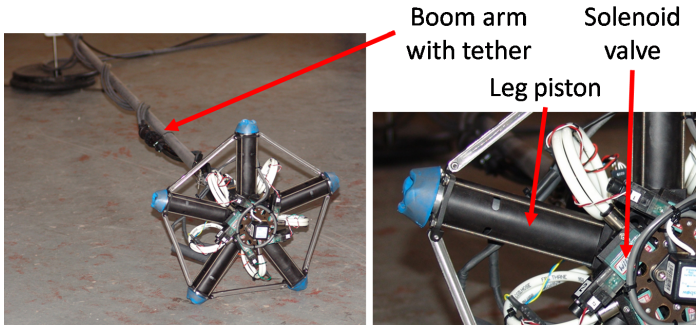


Figure 2.9: 5-legged pneumatic RATS prototype was constrained to motion in a circular pseudo plane. This prototype was used to explore pneumatic actuation and develop initial strategies for rolling control [14].

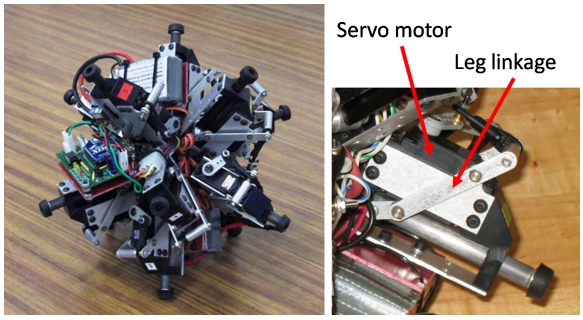


Figure 2.10: 12-legged servo RATS prototype demonstrated the tipping ability made possible by its symmetric leg arrangement.

the end of a pivoting boom arm. The boom arm restricted the robot to motion within a circular pseudo-plane, which simplified the control task. Compressed air was supplied from a tether hose and the pressure in each cylinder was controlled by a solenoid valve, commanded by off-board computing. The prototype was outfitted with encoders, MEMS gyroscopes, and a 3-axis accelerometer to measure the rotation and motion of the leg wheel. This prototype served to validate the pneumatic leg piston design and also provided the groundwork for proprioceptive sensing and control strategies for the RATS system. Sensing allowed for tracking of the orientation of the leg wheel and a control strategy was devised to fire an individual leg when it rotated to a specific angle with respect to the ground surface [14]. This controller provided an effective technique for achieving rolling locomotion from sequential valve openings and could be adapted to induce a hopping behavior to clear obstacles.

2.6.2 12-Legged Electric Servo Prototype

The second RATS prototype was developed to explore the full spherical geometry to be employed in the 12-legged prototype while removing some of the complexity associated with pneumatic leg actuation and control. This prototype had 12 legs pointing straight

outward from the center of the sphere (Fig. 2.10). A servo motor provided linear actuation of each leg via a five-bar linkage. This robot carried on-board electrical power and had a wireless communication link to the control computer, so it was completely untethered and had full freedom of motion. Due to the relatively slow actuation of the servo-driven legs, this robot could only achieve discrete, quasi-static locomotion by pushing with one leg to tip over the other two legs on the ground. The planning strategies developed for this prototype to generate sequences of tipping actions for walking locomotion along a path were also applicable for tipping control of the latest version of RATS.

Chapter 3

Design of 12-Legged Pneumatic RATS

The latest version of RATS was designed to be the first fully functional prototype, shown in Fig. 3.1. It incorporates features from both of the prior prototype generations and is capable of all of the locomotion modes envisioned for the RATS concept. It can tip, roll, and hop, giving it precise position control, high speed motion, and obstacle avoidance capability. The following sections describe the design of the prototype robot to achieve such mobility.

3.1 Mechanical Design

3.1.1 12-Legged Geometry

The robot geometry is unique and impacts the mobility and control of the prototype. The body and leg arrangement reflects the geometry introduced in the servo-actuated prototype. It has 12 legs pointing straight outward, each centered on a face of the dodecahedron-shaped core. The resulting arrangement places the legs at the vertices of an icosahedron (the dual polyhedron of a dodecahedron) with equal angular spacing of 63.44 degrees between adjacent legs. Platonic solids were chosen as the basis for the core structure to ensure symmetry (Fig. 3.2) [15]. Using 12 legs (dodecahedron) provides better spatial coverage than 6 legs (cube) without the excessive complexity of having to fabricate and control 20 legs (icosahedron).

Each leg cylinder is supported by a network of curved steel beams that span between neighboring legs, creating a rounded icosahedron external structure that is approximately 30 cm in diameter. When the robot rests on a flat surface, only 3 legs contact the ground. Twenty such three-legged stances are possible, corresponding to the twenty sides of the icosahedron.

The symmetry inherent in this geometry has benefits and disadvantages. Since only three legs are required for supporting the robot, RATS will always come to rest on some set of three legs, as long as the landing surface is relatively flat. Additionally, since any three-

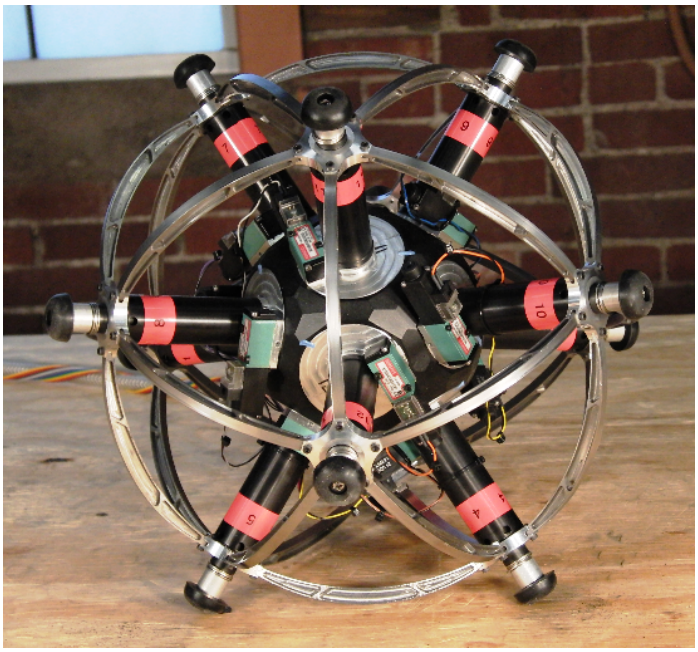


Figure 3.1: 12-legged pneumatic RATS prototype.

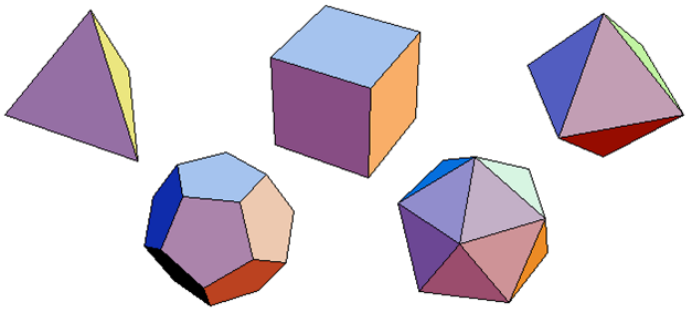


Figure 3.2: Symmetry of platonic solids was crucial to the RATS design. The prototype's dodecahedron-shaped core resulted in an external icosahedron structure.

legged stance is identical to the nineteen other possible stances, no particular orientation of the robot is favored for initiation of motion (there is no right-side-up). The spherical outer shape of the robot is also conducive to rolling, allowing the robot to take advantage of downhill slopes and making for more efficient travel over flat surfaces. This same ease of rolling can also lead to unintended motion, such as tumbling after a hop or difficulty holding a position on a steeply inclined surface. The symmetrical geometry presents the biggest challenge for the control system, which must decide which of the identical legs are appropriate to actuate as the robot tumbles and rolls around. The robot requires a decent estimate of its orientation with respect to the ground in order to know which legs will push against the surface and which will thrust uselessly into the air. This and other controls issues are discussed in Chapter 6.

3.1.2 Pneumatic System

The RATS prototype's ability to move is dependent on its pneumatic actuation system, consisting of the tank, control valves, and leg piston. Each leg cylinder is seated against the central core, which is a high-pressure aluminum tank with an internal volume of approximately 0.84 L. This core tank stores the compressed gas that is the lifeblood of the robot. The core is pressurized from an off-board nitrogen gas (N_2) tank through a detachable hose. The robot has a mass of 3.446 kg when fully charged. A 3-way solenoid control valve is positioned at the base of each leg to draw gas directly from the pressure core and feed it into the leg cylinder upon actuation (inset in Fig. 3.3). When the valve is unpowered, the cylinder is opened to the atmosphere, allowing the pressurized gas to exhaust.

The leg piston is the part of the leg assembly that extends linearly when gas is released into the cylinder. There is no piston seal included in the design so as to minimize friction. Instead, the radial face of the 2.54 cm diameter piston head is polished and rides directly on the inner wall of the cylinder. The leg assembly relies on tight tolerances to prevent excessive gas leakage around the piston. The cylinder endcap limits the outward extension of the piston leg to a maximum of 6.3 cm. A custom-wound spring returns the leg when the valve shuts off the supply pressure, expelling the cylinder gas out the exhaust port of the valve.

The leg piston has a 1.65 cm diameter shaft that extends out through the cylinder endcap. This leg shaft terminates in a medium stiffness, HBR rubber foot pad. The rubber feet serve to absorb some shock from impacts with the ground and to improve traction when pushing off due to deformation of the foot against the ground surface.

3.1.3 Exchangeable Hardware Components

The RATS prototype has a few exchangeable features that greatly affect the behavior of the system, including the pneumatic control valves, leg length, and rubber feet. During most of the system characterizations included in Chapter 4, the prototype was outfitted with

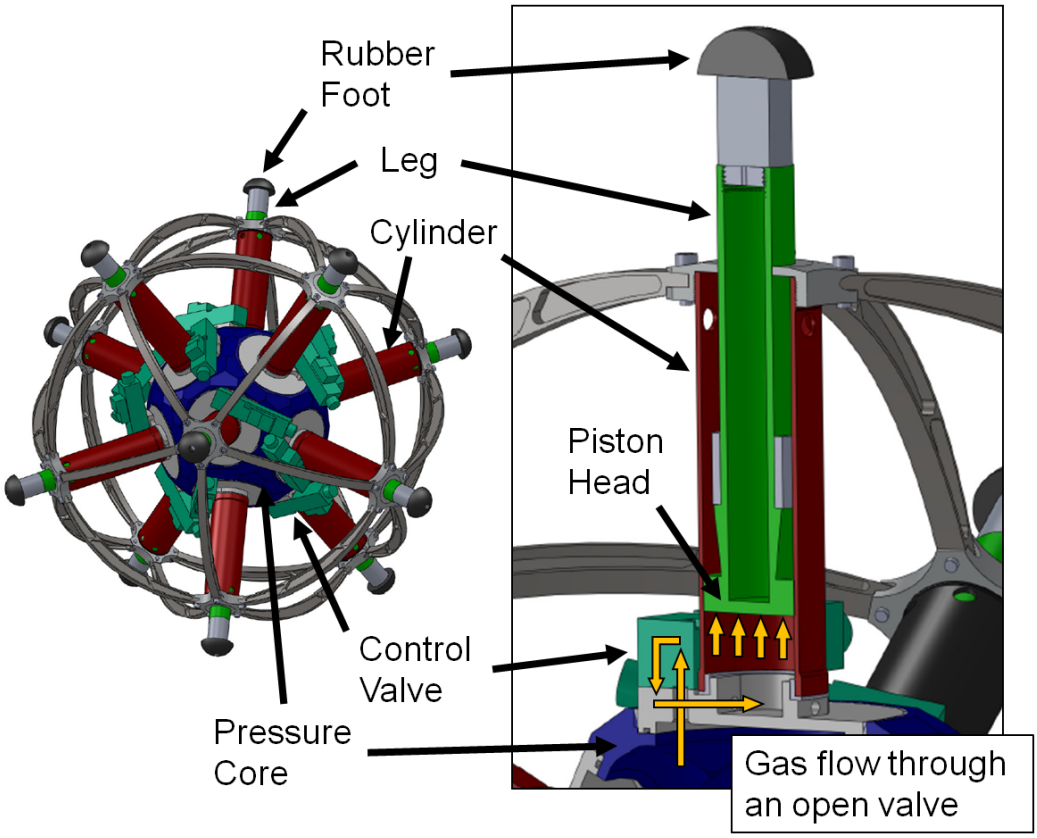
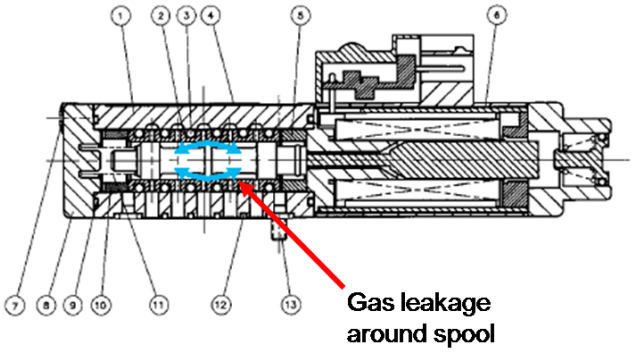


Figure 3.3: CAD drawing of latest 12-legged pneumatic RATS prototype. Cut-away shows leg piston design with the path of gas flow from the pressure core, through the control valve, and into the cylinder.



Parts List

DET. NO.	NO. REQ'D	PART NAME
1	1	Valve Body
2	1	Sleeve Assembly w/seals
3 *	6	Seals
4	1	Nameplate
5 *	1	Spacer
6	1	Solenoid Capsule Assembly
7	2	Screw
8	1	Spring Cover
9 *	2	Seal
10 *	1	Bumper
11 *	1	Spring
12 *	1	Gasket
13	2	Screw

Figure 3.4: Diagram of the internals of the spool-and-sleeve valves that constantly leaked gas.

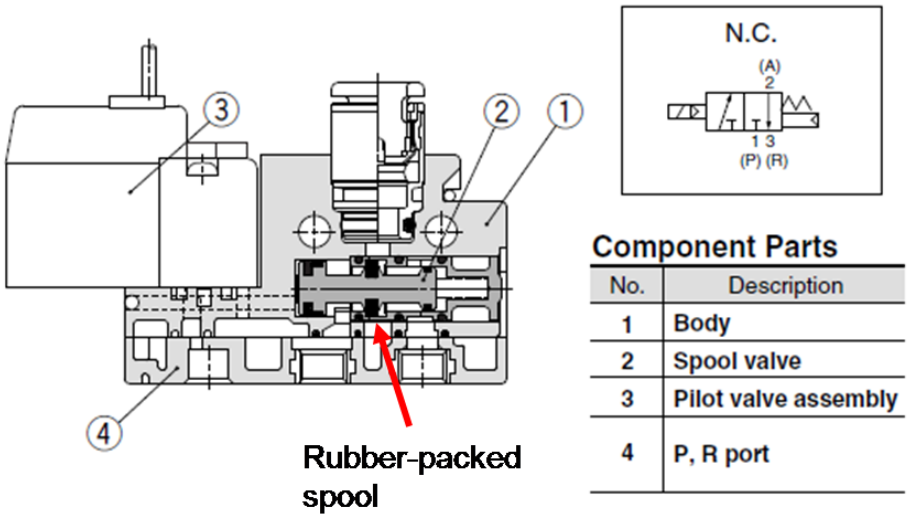


Figure 3.5: Diagram of the internals of the air-pilot rubber-packed valves used to eliminate gas leakage.

solenoid-actuated lapped spool-and-sleeve valves, which had a maximum pressure capacity of 0.758 MPa (110 psi). In these valves, a metal spool is shifted through a metal sleeve, connecting different pairs of valve ports as it moves (Fig. 3.4). This particular model of valve had no rubber seal at the spool/sleeve bearing surface, so a small quantity of air leaked through the valve constantly, even when not actuated. Such leakage critically limited the endurance of the RATS prototype as the finite supply of gas contained in the core was lost to the atmosphere.

In an effort to address this leakage problem, the spool-and-sleeve valves were replaced with air-pilot rubber-packed solenoid valves. These valves have a higher maximum operational pressure of 1.034 MPa (150 psi). The addition of the rubber seal ensures no gas is leaking through the valve when unpowered, but it does result in higher forces needed to shift the valve mechanism. In order to provide enough shifting force without a larger solenoid coil, these new valves use a small solenoid coil to open a preliminary pilot valve, which then uses the high-pressure supply gas to shift the internal poppet of the main valve (Fig. 3.5). The air-pilot mechanism does spend a small quantity of supply gas for each actuation, but this loss is insignificant compared to the gas saved by preventing leakage.

The characteristics of the leg ends are also adaptable. Initially, the prototype had cylindrical rubber feet on the end of short legs that did not even touch the ground when the robot was at rest on a flat surface. Instead, the robot rested on the curved steel beams that connected the 3 bottom-most feet. The legs can be lengthened by adding a spacer between the leg end and the foot pad, allowing the robot to rest on its rubber feet. The cylindrical feet can also be exchanged for different shapes and material to affect the interaction between ground surface and the robot.

3.2 Electronics and Control Architecture

The RATS prototype is controlled by sending actuation signals to the control valves on specific legs. The sequence in which legs are fired, the timing of the actuation, and the duration of actuation signals are what determines the nature of the robot's locomotion. In the initial control configuration of the spherical RATS prototype, all computing and electrical power occurred off-board and was transmitted to the robot through a tether cable. A PC running Scientific Linux sent control commands to a control board, communicating through a NI-DAQ system. The control board had 12 opto-isolated relays that supplied 24 VDC from an off-board power supply to individual valves. The control logic was implemented in C-code on the computer, operating at 1000 Hz to provide 1 ms resolution for the valve commands. Once the controller determined that a leg should be actuated, a countdown timer was initialized and the corresponding valve was sent a signal for a predetermined number of milliseconds. Varying the length of time the valve was powered allowed control over the strength and duration of the push from the pneumatic piston leg. In that configuration, control capabilities were strictly open-loop due to the lack of sensors aboard the robot. Instead, the controller received a scripted sequence of leg firings, detailing the specific time and signal duration for each firing.

Eventually the system was updated to implement a tetherless control system. The robot is now outfitted with two 11.1 V, 800 mAhr Lithium-polymer batteries for on-board electrical power. It also has a Netburner microcontroller (ColdFire chip) for on-board computing and real-time control. This on-board controller provides low-level management of the solenoid valves, including control of valve timing to millisecond precision. The prototype is able to communicate with an off-board computer via an X-bee (Zig-bee protocol) radio link. The off-board computer still executes the high-level control logic, determining when to fire legs. This high-level controller transmits firing commands to the microcontroller consisting of which valve to actuate, how long to apply power, and how much time to wait before the next command is executed. The microcontroller stores these commands on a queue and uses an interrupt timer to execute them with proper timing. This configuration allows the prototype full freedom of motion while retaining use of off-board computing for increased processing power.

All of the on-board electronics and power are mounted on the exterior of the pressure core. These components are separated into 3 groups: the battery package, the microcontroller board, and the radio transmitter board. The robot mobility is affected by the distribution of this added weight, so efforts were taken to arrange the three groups so as to distribute the weight as evenly as possible.

3.3 Sensors

The latest development in the RATS prototype has been the addition of inertial sensing. The robot currently has a 3-axis MEMS accelerometer rigidly mounted on the robot core. The accelerometer has adjustable sensitivity and is set to measure accelerations up to $\pm$ 1.5 g. The sensor output is read and digitized with a 12-bit ADC included with the Netburner microcontroller package. Because the microcontroller is limited in memory and processing speed, all sensor data must be transmitted by radio to the off-board computer for processing.

Chapter 4

Mobility Characterization and Testing

While the concepts behind the RATS design are simple, it is difficult to predict the exact nature of the prototype's locomotive behavior based solely on the design. In order to fully understand the motion capabilities of the robot, a detailed characterization of the system was necessary. Initial testing of the robot was mostly qualitative in nature. The basic principles of the robot's mobility were investigated by studying how different combinations of leg actuation and valve firing duration affected motion. The effects of geometry and some implementation details of the pneumatic system were also studied. A set of quantitative tests was then used to examine what factors influenced the performance of the leg actuators. These characterizations were useful in providing insight for mobility and control strategies. The results also illuminated several system characteristics that must be compensated to enable successful implementation of control on the prototype.

It should be noted that the majority of these tests were executed using the original spool-and-sleeve valve configuration. The effects of using the new rubber-packed valves were assessed at a later point and these results are discussed briefly at the end of the chapter.

4.1 Basic Mobility Characterization

The 12-legged pneumatic RATS prototype exhibits different behavior depending on many factors including the number of legs fired, the duration of the leg actuation, and the physical configuration of the robot. This section addresses some of the general behavior trends observed during initial testing of the prototype.

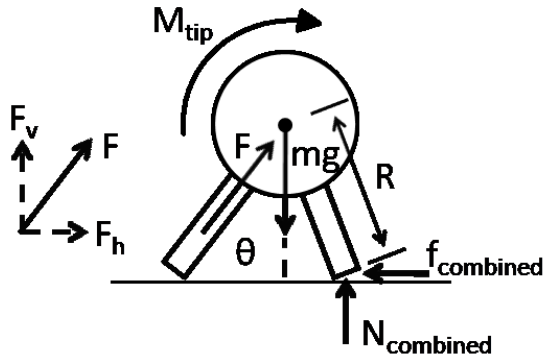


Figure 4.1: Free body diagram of RATS firing a single leg.

4.1.1 Leg Firing Effects

The design of the 12 piston actuators on RATS is such that the only method for dynamically controlling the strength of a leg firing is to alter the duration for which the valve is held open. Longer valve firing times generate longer piston strokes and allow the actuator to apply launch forces for longer duration when pushing off the ground.

From a rest position on a flat surface, firing a single leg generates a thrust force F through the robot center at an angle $\theta = 37.38$ degrees from the surface normal (Fig. 4.1). This results in a vertical thrust component F_v tending to make the robot leave the ground.

$$F_v = F \cos\theta \tag{4.1}$$

Since the thrust force passes through the robot center, the horizontal thrust component F_h must couple with the combined friction reaction $f_{combined}$ at the other two stance feet a distance $R\cos\theta$ away to produce a tipping moment M_{tip}.

$$M_{tip} = f_{combined} R \cos\theta \tag{4.2}$$

The friction reaction is equal to the horizontal component of the thrust force up to a maximum value f_{max} related to the combined normal contact force on the ground at the stance feet $N_{combined}$ and the friction coefficient μ.

$$f_{combined} = \min \left\{ \begin{array}{l} F_h = F \sin\theta \\ f_{max} = \mu N_{combined} \end{array} \right\} \tag{4.3}$$

If a short valve time is used, F_v is insufficient to lift the robot and ground friction remains strong enough to produce a tipping moment. In this case, the robot simply tips over the two stance feet. (Fig. 4.2a,b). For longer valve times, F_v grows strong enough to lift the two stance feet off the ground. F_h briefly produces a rolling moment before the stance legs lose contact, and then continues to accelerate the robot forward. The result is a

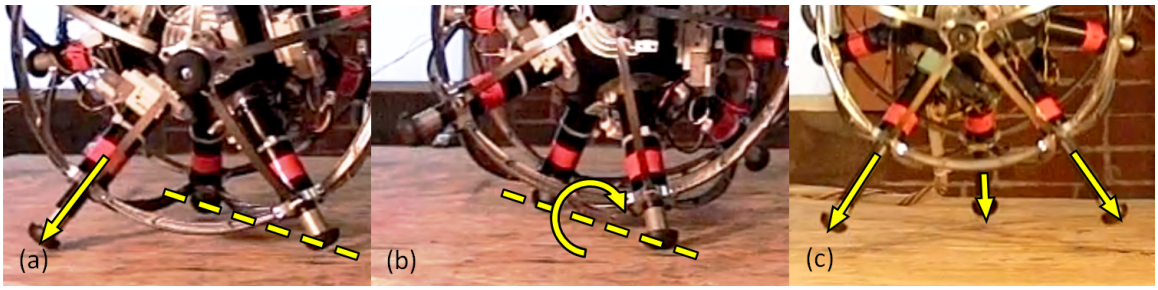


Figure 4.2: Leg actuation combinations to achieve varied locomotion: (a) Short duration firing of a single leg to cause (b) tipping about the other two stance legs. (c) Firing three legs for a vertical hop.

forward hop with induced tumbling. Firing three legs simultaneously from rest produces a vertical hop, as the horizontal thrust components from all three legs cancel each other out (Fig 4.2c). Any imbalance in leg strength during the launch phase results in rotation once the robot is airborne.

Landing from a 3-legged hop depends strongly on which part of the robot contacts the ground first. With shortened legs, the majority of initial contact occurred on the robot's steel support beams. This caused a hard impact that was sometimes strong enough to jar another valve to open and cause an extraneous leg firing. Once the legs were lengthened with spacers, the rubber feet were the only landing contact points, which softened the impact of landing. If the robot touches down on three feet nearly simultaneously, the landing is fairly stable with minimal rebound and tumbling. If the robot rotates in the air so as to cause a landing on fewer than three feet, it frequently tumbles away and comes to rest 10 to 30 cm from its launch position.

This basic behavior underscores the importance of characterizing the pneumatic leg actuators to understand the factors that determine leg strength. Deterministic tipping behavior requires individual leg firings that are strong enough to successfully tip but not so strong as to cause additional stochastic tumbling. Controlling the differences in leg strength when simultaneously firing multiple legs is also useful for controlling the direction, height, and induced rotation of a hop.

4.1.2 Foot Shape Effects

The contact conditions between the rubber feet and the ground have a large effect on tipping and hopping behavior. Since the legs rarely push exactly perpendicular to the ground surface, ground sliding friction is an important consideration. On one hand, the legs must slip against the surface in order to allow leg extension. On the other hand, traction is what provides the horizontal reaction force to cause forward hops and tipping. This friction interaction is largely determined by the interaction between foot material and the particular ground surface compliance and texture of the environment. The foot shape also

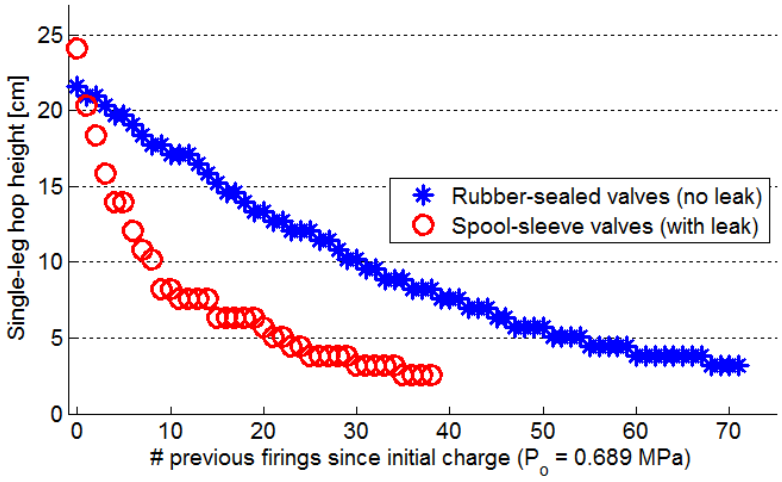


Figure 4.3: Observed drop in hopping height (leg strength) as supply gas is depleted through repeated 40 ms valve actuations. Strength decay is even more dramatic when using leaking valves.

plays a role in determining locomotive behavior. After comparing tipping behavior using various combinations of cylindrical and hemispherical feet, a tradeoff became apparent. Cylindrical feet have a right-angle edge that provides better pushing traction, but also is more resistant to rolling. The hemispherical feet give up some traction, but improve the rolling behavior. In the end, hemispherical feet were selected to give consistent tipping and stable landing performance.

4.1.3 Core Pressure Effects

The pressure of the gas entering the control valve also directly affects leg strength. Higher supply pressure means higher pressures inside the cylinder, and consequently higher piston forces. Since the prototype was not designed with an on-board pressure regulator between the source tank and the control valves, the supply pressure to the valve intake is always the same as the pressure of the core tank. This pressure drops over time as gas is expended by leg firings and the leg strength of each subsequent firing consequently drops as well. This effect was measured using leak-proof valves (which are discussed later) to hop repeatedly using a 40 ms firing time after a single 0.689 MPa charge of the core. The robot could not lift itself off the ground more than a few centimeters after 75 successive firings, as shown in Fig. 4.3.

This leg strength loss from pressure drop is further exacerbated by continuous leakage of gas through the spool-and-sleeve valves. The leakage is severe enough to result in a 50% drop in pressure in 5 minutes from an initial 0.758 MPa (110 psi) without firing any legs. Leg strength decays even more quickly than the pressure. A 50% decrease in leg strength (measured by vertical hop height) was observed after only 2.5 minutes of leaking without

previous leg firings. This severely limits the endurance of the robot and the distance it can travel. Using the leaking spool-and-sleeve valves, a single 0.758 MPa charge of the core tank is only good for 40 single-leg hops in rapid succession (using 40 ms valve opening times) until the robot is unable to leave the ground. This endurance trend is also shown in Fig. 4.3.

4.2 Detailed Mobility Testing

4.2.1 Leg Actuation Response Time

Solenoid control valves do not immediately open for gas flow upon receiving a voltage signal from the controller. All valves have a characteristic latency, or response time, that is most often defined as the time between application of the energizing signal and complete opening of the valve mechanism. Response times vary for every type of valve, but this is only one contributing factor to system latency. Details of the leg fabrication, such as the friction of the piston inside the cylinder, also affect how quickly the leg responds. It is important to identify the delay between actuation signal and first motion of the unloaded leg piston in order to control the timing and strength of leg actuation on RATS.

It was decided that the most straight-forward method to measure the overall effect would be to use a high-speed camera to record the firing of each leg. Conveniently, the valves have a built-in LED that illuminates as soon as the actuation voltage signal is received, which signifies the beginning of the response time measurement. A DSLR camera with a video frame rate of 1200 frames per second (fps) was used to record the event. The number of frames between LED illumination and perceptible leg motion was counted and divided by the video frame rate to calculate the elapsed time. The test was repeated twice for each of the 12 legs to establish an average empirical response time for each leg. The results from these measurements are shown in Fig. 4.4.

For the spool-and-sleeve valves, the average response time from all legs was 13.16 ms. The quickest response was Leg 6 at 11.25 ms and the most delayed response was Leg 3 at 14.58 ms. Since the valve commands are issued with a resolution of 1 ms, these response times are clearly significant and must be considered in the development of mobility control. Another key observation is the variation in response time between individual legs. The variation might be caused by differences in valve construction, piston friction, or any other inconsistencies in prototype fabrication. This lack of uniformity across the actuation of all legs is a recurring theme that will also need to be compensated for in future controller development.

4.2.2 Effects of Valve Time on Leg Strength

As observed earlier, altering the valve opening time is the only available method to control the strength of a leg firing after the pressure core has been charged. In order to observe the

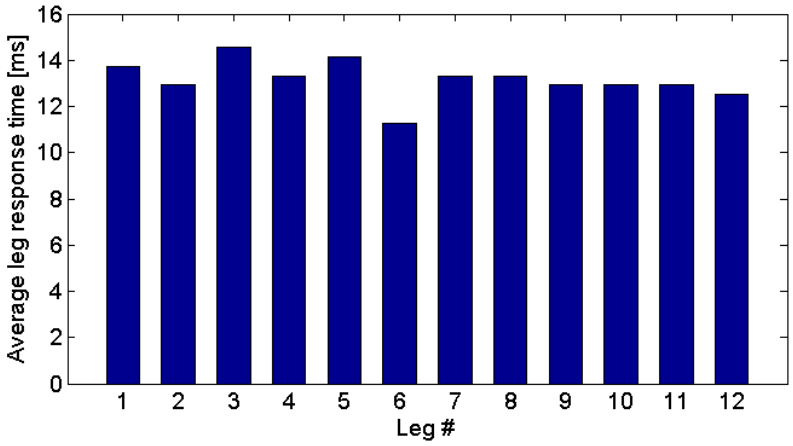


Figure 4.4: Average response times found for each of the 12 valves and legs. Latencies are inconsistent across legs; the average from all legs was 13.16 ms.

empirical effects of varied valve firing times on the pushing strength of the leg, a single-leg vertical hopping experiment was devised. Individual hops were recorded using the high speed camera, filming at a frame rate of 600 fps. A launching platform was fashioned to allow RATS to balance on a single leg without obstructing the vertical hopping motion. A ruler marked in half-inch increments was positioned next to the robot to measure the height of each hop. After the trial was recorded, the video frame containing the robot at the apex of its hop was manually identified and the height of the bottom edge of the curved metal frame was compared to the ruler (Fig. 4.5).

The tests were performed by initially charging the core with N_2 gas to a regulated 0.724 MPa (105 psi) level. The feed hose was disconnected immediately prior to the hopping test. An actuation signal was then sent to the robot with a specified duration. Valve times varying from 10 to 65 ms in 5 ms increments were tested, with each test using the same leg (Leg 9) for hopping to avoid inconsistencies between legs. Each valve signal duration was tested 3 times. The results from this testing are shown in Fig. 4.6. Data are plotted from two trial runs with identical conditions performed on two different days to demonstrate repeatability.

The robot exhibited nearly linear increase in hop height as valve time increased from 15 to 50 ms. For valve firing times below 15 ms, the robot did not perceptibly leave the ground, which is largely explained by the actuation latency effect explored above. Sufficient gas pressure in the cylinder must also build up to generate enough force to overcome piston friction and gravity. At valve times above 50 ms, the hop height plateaued at a maximum of 31.75 cm. It is proposed that this plateau exists because the robots leg hits its maximum-extension dead-stop and its foot loses contact with the ground as the robot continues airborne. With no ground contact, RATS is unable to convert additional valve time into applied force.

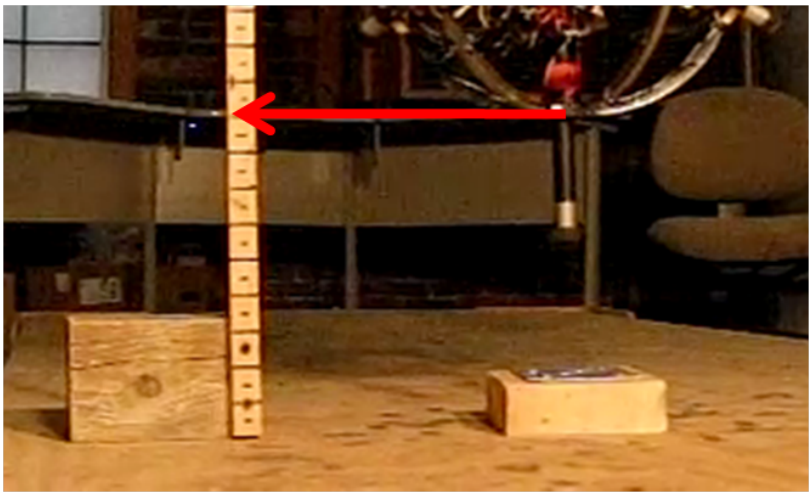


Figure 4.5: Measurement technique for single leg vertical hopping height. A high speed video camera captured the apex of the hop. The bottom edge of the robot was compared against the ruler in the video frame.

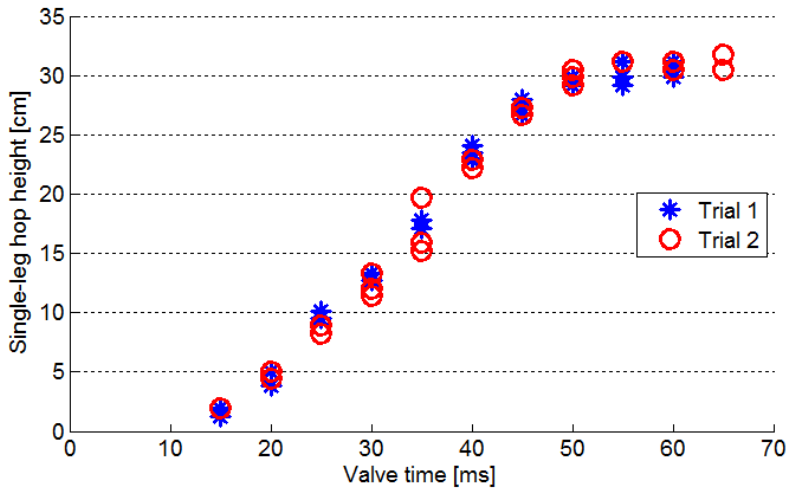


Figure 4.6: Hop height trend with increasing valve opening times at a constant core pressure of 0.724 MPa (105 psi). Valve times below 15 ms did not propel the robot off the ground. Valve times longer than 50 ms did not increase hop heights. Trials from two separate days are plotted and each valve time was tested 3 times in each trial.

4.2.3 Strength Variation Between Legs

Earlier tests demonstrated that each of the legs on the RATS prototype provides a different response upon actuation. In order to quantify this variation, two different indicators of leg strength were used to survey all 12 legs.

Tipping Threshold Variation

The relative strength of an individual leg is easily explored through tipping behavior. For a given short duration of valve actuation signal, a strong leg might execute a complete tip, coming to rest on a set of 3 legs different from its initial set. A weaker leg might fail to completely tip the robot over for the same signal duration.

The external icosahedron geometry dictates each leg is included in 5 tripod stances, and thus each leg may tip over 5 edges (defined by the other two legs of the particular tripod stance). It was noted that even for a given firing leg, the robot's tipping success varied with the edge over which the tip was being attempted. Such variation can be explained by the non-uniform distribution of mass on the robot, especially when considering the placement of batteries and electronics hardware. Some tipping edges might have a higher moment of rotational inertia, or the center of mass of the robot might be positioned so as to require a larger change in potential energy from the start to the midpoint of the tip where the COM is at its highest elevation. Such edges would require more force and launch energy from the firing leg to successfully tip. In order to avoid inclusion of mass distribution into controls considerations for tipping behavior, I opted to identify the threshold valve time for each leg as the valve command duration that resulted in a successful tip over the most tip-resistant of the 5 edges.

A set of experiments was carried out to identify this threshold valve time for each leg. Starting with a valve opening duration of 5 ms, one of the legs supporting the robot was fired to attempt a tip. The valve signal was increased by 1 ms increments until the leg actuation resulted in a full tip on at least 9 out of 10 attempts. The threshold valve time that achieved this tipping consistency for each leg is plotted in Fig. 4.7.

Single-Leg Hop Height Variation

Another set of experiments investigated the hop heights achieved by each leg when fired at a single valve time of 40 ms. Stronger legs would be expected to hop higher for the same valve time and core pressure. Using the same high-speed camera technique for measuring the height of a single-leg hop as described above, each leg was tested 3 times with a 0.379 MPa (55 psi) core pressure. The series of tests was carried out twice more for 0.552 MPa (80 psi) and 0.758 MPa (110 psi) core pressures. The results are plotted in Fig. 4.8.

The data show clear trends in both the effect of pressure on leg strength and the relative strength of each leg. As expected, higher pressures resulted in higher hops for all legs. Legs

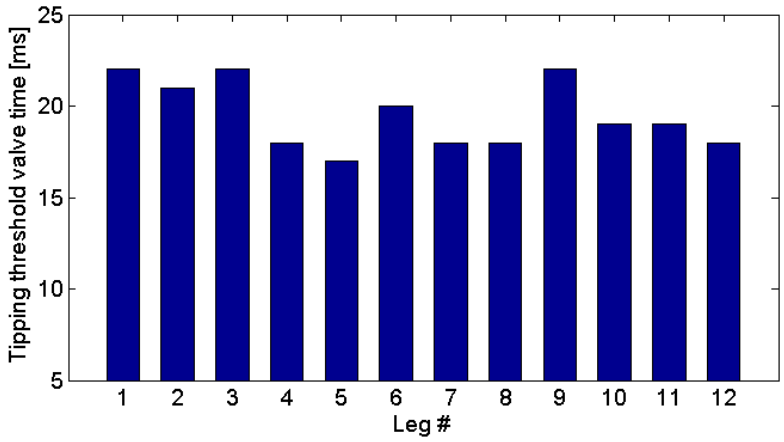


Figure 4.7: Valve opening times required to achieve consistent tipping behavior for each leg. Values represent threshold for $\geq$90% tipping success rate.

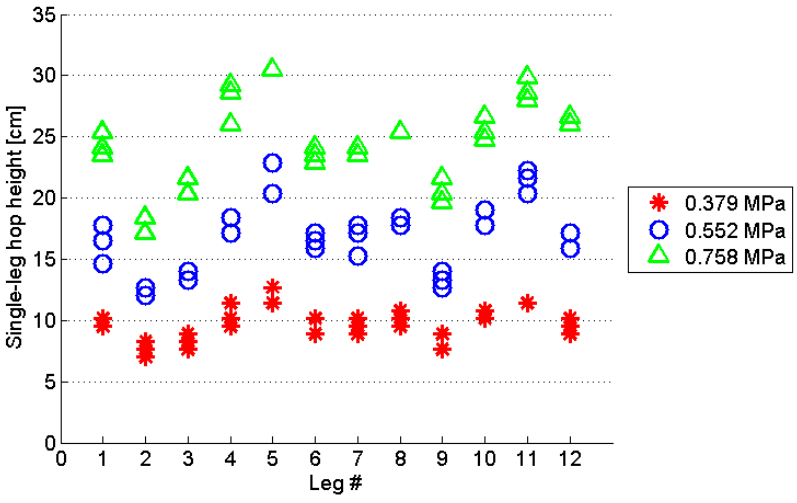


Figure 4.8: Results from single-leg vertical hops for each leg at 3 different core pressures, all using the same 40 ms valve signal duration. Legs 4, 5, and 11 were consistently stronger and Legs 2, 3, and 9 consistently weaker than the others regardless of core pressure.

4, 5, and 11 were stronger relative to the other legs regardless of the supply pressure. Similarly, the Legs 2, 3, and 9 were consistently weaker than the other legs at all pressures. These differences between legs seemed more pronounced at higher supply pressures.

The lack of uniformity in leg strength observed in the previous two characterizations may be attributed to several factors. Given the same duration of valve actuation signal, the variation in valve response time previously studied would mean different legs were experiencing different effective durations of the valve actually being open. This would cause some variation in imparted energy, and thus hop height. However, upon comparing the response time profiles with the leg strength trends, no strong correlation is evident. Leg 6 had the smallest observed latency, but Leg 6 did not exhibit a low tipping threshold or high hop heights to indicate the shorter response time gave it increased strength. Another explanation is non-uniformity in the fabrication of the individual leg pistons. Differences in friction between the piston and cylinder, gas leakage around the piston, or return spring stiffness might cause the variation in leg strength.

4.3 Discussion

4.3.1 Locomotion Feasibility

The 12-legged RATS robot was designed for multiple modes of locomotion in rough terrain that take advantage of its unique geometry. The characterizations described above indicate the feasibility of achieving such locomotion with this prototype. Tipping behavior has already been demonstrated through the ability to adjust leg firing strength sufficiently to differentiate between tipping strength and forward hopping strength. The spool-and-sleeve valve configuration of the robot has been shown capable of hopping to heights of 31.75 cm - higher than its own diameter. Such hopping ability indicates that RATS should be able to scale tall obstacles nearly its own height using the same actuation mechanisms as are used for forward locomotion on flat terrain.

4.3.2 Robustness

The prototype mechanism is also durable. It has proven to be robust to the mechanical shock of landing impact on hard surfaces like plywood and cement. By this time, the prototype has been subjected to over 20,000 actuation cycles, individual legs experiencing as many as 3,000 cycles, with no failure or significant wear of the piston mechanisms. After some design enhancements to increase resistance to dust contamination, the robot will soon be ready for operation in the soil and grass of outdoor environments.

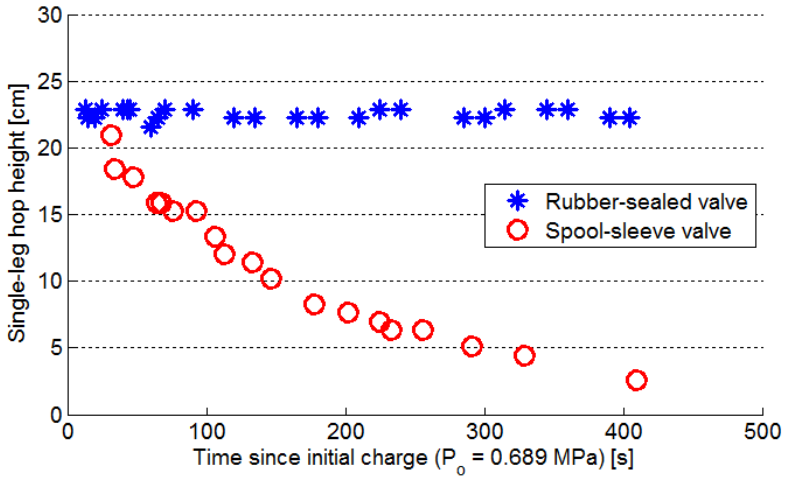


Figure 4.9: Loss of leg strength due to gas leakage over time with no prior firings. The significant strength loss with leaking spool-sleeve valves is eliminated using rubber-sealed valves with no leakage.

4.3.3 Effects of New Rubber-Sealed Valve Configuration

The switch to new air-pilot, rubber-sealed control valves has positively affected the mobility performance of RATS, most notably in the robot's endurance. The valve leakage was characterized by charging the core then waiting a variable length of time before attempting a vertical hop with a single leg. The trend in hop height as a function of leakage time prior to firing is shown in Fig. 4.9 for both valve configurations. As expected, the internal rubber seal prevented any significant leakage over the test period so there was no decrease in hop height. This improvement in gas retention also increased the distance RATS can travel on a single tank of gas. RATS can reliably travel 17 meters or more (160 firings) in tipping mode on a single charge of 0.689 MPa (100 psi) gas with the new valves. At best, the prototype with the old leaking valves could only travel 9.5 meters (92 tips) with the same initial conditions.

The robot's response to valve command duration was tested again using the same single-leg vertical hop procedure from before. For the same core pressure, the new valves achieved a lower maximum hop height (26 cm) than the old valves, as seen in Fig. 4.10. This indicates a slightly slower gas flow rate through the new valves. The rubber-sealed valves do have a higher maximum pressure rating, however, so the robot can operate at core pressures as high as 1.034 MPa (150 psi). As expected, higher supply pressure leads to a faster rate of leg extension. This causes the robot to lose ground contact (reach the hop height plateau) at smaller valve times. It also enables the newly configured robot to achieve maximum hop heights of 33 cm, which is an improvement over the old valves at their maximum operational pressure of 0.758 MPa (110 psi).

Another observed change in the system's performance was in response time. The actu-

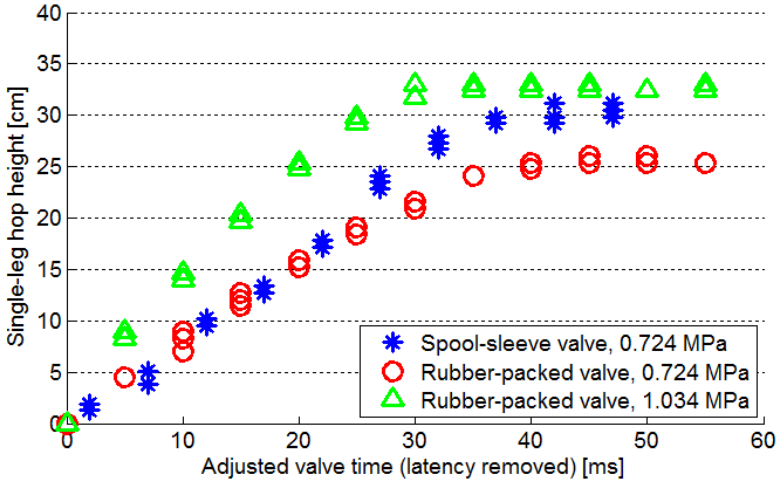


Figure 4.10: Comparison of leg strength trends with increasing valve duration for different valves and pressure conditions. Plotted valve times have been adjusted to remove response latency. Spool-sleeve valves enable higher hops at 0.724 MPa (105 psi), but rubber-packed valves can handle higher gas pressures up to 1.034 MPa (150 psi) to achieve even higher hopping.

ation delay of the new valves is much lower than the old valves at 5 ms, and is relatively consistent for every leg. Such reduction in latency is always good for improving controllability. This characteristic, combined with the slight reduction in leg strength, results in a new set of tipping valve time thresholds, which are given in Fig. 4.11.

4.3.4 Impact on Controls

The findings from the characterization tests discussed earlier have important ramifications for the development of future locomotion controllers. Actuation response time must be accounted for in control design, introducing inefficiencies and error due to the need to act early. This latency most dramatically affects the rolling controller, where precise timing of leg actuations is needed. The intent of the robot's symmetrical geometry was to permit control strategies that are indifferent to which particular legs happen to be on the ground. Instead, the pneumatic piston actuators exhibit non-uniform behavior. Legs have varying response times; pistons have differing amounts of internal friction; other unknown asymmetries are causing variation in leg strength. Accordingly, the controller needs to fire legs with different actuation signal durations in an effort to balance leg strengths. This is especially crucial for tipping and hopping maneuvers. Non-uniform mass distribution has an effect on all modes of locomotion and needs to either be accounted for in controllers or treated as noise to which the controller must be robust.

It should be noted that these effects were studied on ideal flat terrain, where behavior is largely deterministic. Many of the compensation strategies previously mentioned to achieve

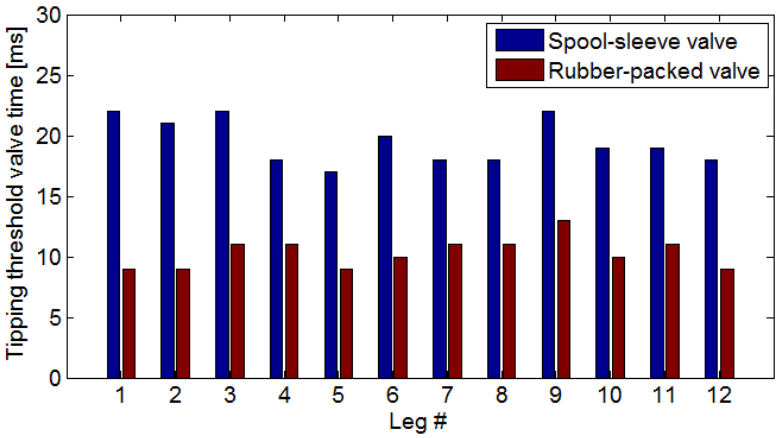


Figure 4.11: Comparison of threshold valve times for tipping using different valves. Rubber-packed valves require shorter actuation times due to reduced latency.

predictable motion become irrelevant as soon as the robot encounters natural terrain with surface irregularities, slopes, and debris. Locomotion controller development for RATS must therefore emphasize robustness and be capable of reacting to stochastic motion resulting from both unpredictable terrain and non-uniform actuator performance.

Chapter 5

RATS Simulation

Dynamic simulation of the pneumatic RATS prototype was an important step in the development of control capabilities. Experimental techniques could be attempted on the simulated robot without the risk of damaging the prototype. The simulation also provided unrestricted access to the robot state, making it possible to decouple controls development from the sensing problem. Artificial restrictions on state access can be implemented, but simulation work may provide valuable insight about what types of sensors and feedback are required for effective control.

RATS simulation was carried out using *Open Dynamics Engine* (ODE). ODE is an open-source rigid body physics engine that uses first order fixed-step integration [16]. Its accuracy may not be sufficient for quantitative engineering, but its speed and built-in framework for creating rigid body assemblies with pre-defined joint constraint types made it a very useful tool for RATS simulation. The simulation was implemented in C/C++ and output was visualized using the built-in "drawstuff" library, which is based on OpenGL.

ODE can only handle rigid bodies with uniform density and shapes from a small set of primitives, including spheres, cylinders, and boxes. Rigid bodies can be linked together using common joint types including hinge, slider, and universal. A collision detection system is included in ODE, along with a basic contact model. External interaction is possible at every timestep of the simulation, allowing for the application of torques and forces to the bodies to augment the forces from gravity, joint constraints, and contact that ODE calculates and applies automatically.

5.1 Previous Simulation Work

5.1.1 5-Legged Pneumatic RATS Simulation

The use of ODE for simulations in the RATS project started with work on the 5-legged prototype. Allan Lüders constructed a simulation of the 5-legged robot and the rotating boom arm on which it pivoted in order to develop control strategies for the real prototype.

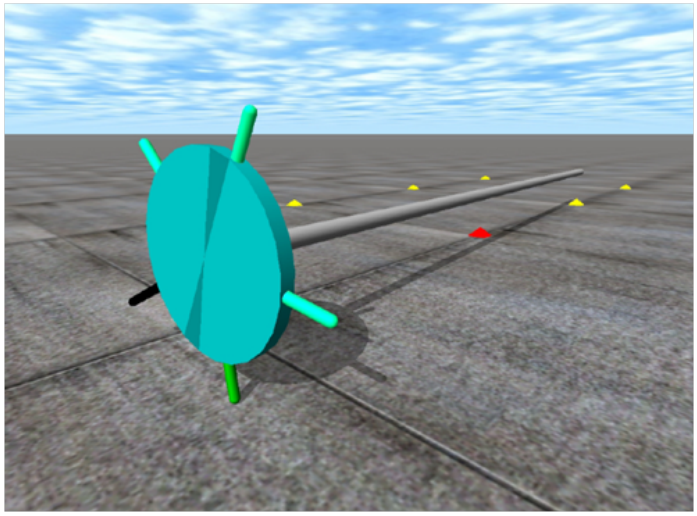


Figure 5.1: Rendering of 5-legged RATS model in ODE.

The simulation consisted of the boom arm, a rotating disk representing the robot body, and the 5 piston legs constrained with slider joints to the disk (Fig. 5.1). The leg geometry was represented as a solid cylinder with spherical caps.

Pneumatic actuation of the legs was simulated using a thermodynamic model based on the work of Richer and Hurmuzlu [17]. It took into account the dynamics of the spool valve, the friction of the piston with the cylinder wall, flow restrictions through the valve, and the compressibility of the gas in the piston chamber. Unknown parameters for the pneumatic model included effective valve orifice size, friction coefficients, dead volume at end of stroke, and the valve discharge coefficient. These were determined by optimizing parameter values to minimize error between the model predictions and experimental results from a benchtop single-leg test setup. Using the equations from the gas model, net forces on the leg pistons were computed at each simulation timestep and then applied to the ODE model of the leg.

The latency resulting from solenoid valve response time was simulated using a delay timer that counted down before the virtual valve was allowed to open. The ground contact model was the default ODE spring-damper model with friction cone approximation by pyramid. Lüders' simulator provided results that accurately modeled the real robot's behavior. Jump heights from single leg firings of the robot matched well with the predicted heights from the simulator. Dynamic rolling behavior also matched very closely.

5.1.2 12-Legged Servo RATS Simulation

ODE was used again to model the servo-powered, 12-legged RATS prototype (Fig. 5.2). This time, the body of the robot was modeled as a uniform sphere with 12 capped-cylinder legs, constrained by slider joints with minimum and maximum extension limits. Leg forces were calculated from a modeled 4-bar linkage with applied torque from an assumed servo. The same default ground contact model from the previous simulation was used. This

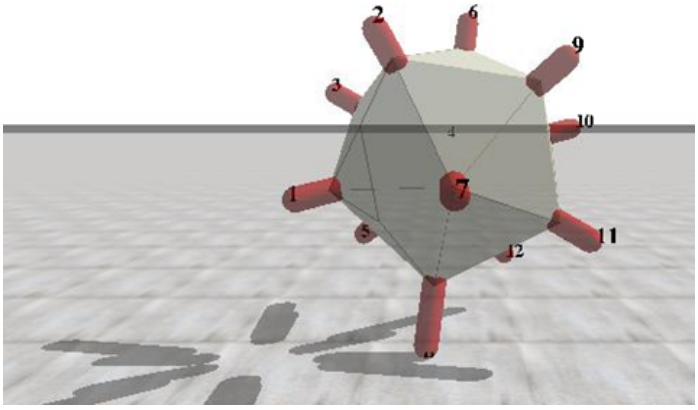


Figure 5.2: Rendering of 12-legged RATS simulation model in ODE. The body assembly and geometry was developed for simulation of servo-actuated RATS and used again for 12-legged pneumatic RATS.

simulation mimicked the tipping behavior of the servo-actuated RATS and served as a testbed for the development of the robot tipping controller.

5.2 12-Legged Pneumatic RATS Simulation

Elements from both of the previous simulations were combined in the latest simulation of the 12-legged pneumatic RATS prototype. Implemented in ODE, the geometry and body assembly were borrowed from the Servo RATS simulation. The gas model from the 5-legged simulation was then used to compute piston leg actuation forces. The unknown parameters for the thermodynamic piston model were assumed to be the same as the fitted values from the 5-legged prototype simulation, despite small changes to the leg design that might have changed friction conditions or dead volume. Physical dimensions and masses (but not mass distributions) used in the simulation were measured from the real robot. The ground contact model was again the ODE default. The contact stiffness, damping, and friction between the feet and ground were parameters that were adjusted by hand to get simulation behavior that resembled observations from the real robot. The simulated robot was controlled with the same control frequency (1000 Hz) as the real robot. The simulation timesteps (integration) occurred faster, at 4000 Hz.

5.2.1 Rigid Body Assembly

The assembly of rigid bodies used to model the 12-legged pneumatic RATS robot was identical to that employed in the servo RATS simulation. A solid, uniform density sphere was used to represent the main robot body. The sphere's diameter matched the outer diameter created by the struts between legs and its mass was set equal to the mass of the entire robot less the mass of the 12 leg pistons and feet. The 12 legs were modeled as uniform

Table 5.1: Dimensions and masses used in ODE model.

Rigid Body	**Mass Distrib.**	**Mass [kg]**	**Contact Geometry**	**Dimensions [m]**	
Robot body	Uniform density sphere	2.858	Sphere	Sphere radius	0.15
Leg + foot (each)	Uniform density cylinder	0.049	Cylinder + hemi-sphere endcaps	Cylinder length Cylinder radius Leg stroke limit (slider joint)	0.114 0.012 0.063

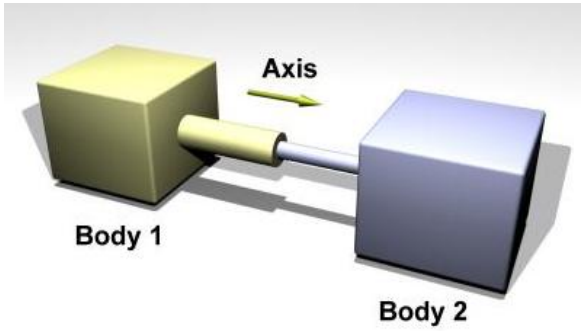


Figure 5.3: A slider joint constraint was defined between each leg and the robot body sphere in ODE [16].

density cylinders with hemi-spherical caps at the ends. The cylinder (and cap) radius was set to match the radius of the hemi-spherical rubber feet used on the robot, and the mass was equal to the total mass of the robot leg, including piston, leg shaft, leg extender, rubber foot, and securing screw. The dimensions and masses used in the simulation are listed in Table 5.1.

The legs of the simulation were arranged to match the true geometry of the robot, aligned to be perpendicular to the faces of an imaginary dodecahedron at the robot's core (the actual pressure core and leg cylinders were not modeled individually but were instead grouped together and represented by the body sphere in simulation). A slider joint was defined between each leg and the body sphere, allowing the leg to translate along its length-wise axis (Fig. 5.3). The slider joint was also given minimum and maximum extension limits, permitting a 6.3 cm leg travel equal to the travel of the robot's legs. By ODE conventions, the slider joints were frictionless, so friction between piston and cylinder, along with other leg forces like return spring and gas pressure, had to be applied manually elsewhere in the simulator.

5.2.2 Ground Contact

Ground contact between the robot's rubber feet and the ground greatly affects the simulated robot's behavior. The default contact model included in ODE was that of a spring-damper system with friction. Whenever the collision checker detected overlapping geometry between the robot's legs and the ground plane, temporary joints were assigned to the collision points and were used to calculate and apply contact forces. At each contact "joint", the simulator applied a stiffness resistance force proportional to the depth of geometry penetration and a damping resistance force proportional to the rate at which that penetration was changing. Predefined contact stiffness and damping parameters were used to calculate these contact forces.

$$\vec{F}_{stiffness} = -k_{contact}\vec{x}_{penetration} \tag{5.1}$$

$$\vec{F}_{damping} = -b_{contact}\left(\frac{d}{dt}\vec{x}_{penetration}\right) \tag{5.2}$$

ODE calculates contact friction with a predefined friction coefficient and a pyramid approximation to the coulomb friction cone where slip occurs when the contact force falls outside the cone defined by

$$|F_{tangential}| \leq \mu\,|F_{normal}| \tag{5.3}$$

The parameters of the ground contact model (stiffness, damping, and friction coefficient) could be hand-tuned to qualitatively match the behavior of the real robot. The challenge was to identify a single set of contact parameters that accurately portrayed all RATS behavior. I was not able to find such a set through hand-tuning. In order for the simulated robot to tip over two feet without sliding along the ground, the friction coefficient needed to be rather high. Unfortunately this high friction negatively affects 3-legged vertical hopping. With the friction set sufficiently high for tipping, the robot is completely unable to leave the ground when firing 3 legs simultaneously with suitable valve times. The friction keeps the feet from slipping across the ground, which prevents the necessary extension of the hopping legs to provide the upward push. A parameter set was finally chosen for development of rolling control that sacrificed accuracy in low-dynamics situations (like tipping) for better performance in highly dynamic situations like tumbling and rolling. These final parameters are listed in Table 5.2.

5.2.3 Thermodynamic Piston Actuation Model

The nonlinear thermodynamics model used to derive piston forces for the latest RATS simulation was mostly borrowed from the 5-legged pneumatic simulation. The model takes into consideration the compressibility of the supply gas, nonlinear gas flow through the control valve, internal friction between the piston and the cylinder, and the resistance from

Table 5.2: Ground contact parameters used in ODE.

Parameter	**Values for Accurate Tipping**	**Values for Accurate Rolling**
Friction coefficient μ	3	1.2
Contact stiffness $k_{contact}$	7500	25000
Contact damping $b_{contact}$	200	400

the piston return spring.

At every timestep of the simulation, the force on the piston $F_{leg,actuation}$ depended on the position of the leg x, the translational velocity $\dot{x}$, and the absolute gas pressure P_{cyl} inside the cylinder chamber.

$$F_{leg,actuation} = F_{pressure} + f_{coulomb} + f_{viscous} + F_{return} \tag{5.4}$$

$$= (P_{cyl} - P_{atm})\, A_{piston} - F_f - \beta\dot{x} - k_{return}x \tag{5.5}$$

where P_{atm} is the atmospheric pressure,A_{piston} is the cross-sectional area of the piston, F_f is the coulomb sliding friction, β is the coefficient of viscous friction, and k_{return} is the stiffness of the custom-wound return spring.

For simplicity, the gas model was only developed and implemented for the first half of the pneumatic process. The following equations apply only to the actuation phase where the valve opens to fill the cylinder with gas and push the leg outward. The exhaust phase, where the gas is pushed out of the cylinder through the deactivated valve to the atmosphere, was ignored since it did not affect the thrusting behavior of the legs. The force applied in the simulator to return the leg to its rest position $F_{leg,return}$ had no gas pressure component, leaving only the force from the return spring countered by friction.

$$F_{leg,return} = -F_f - \beta\dot{x} - k_{return}x \tag{5.6}$$

The design of the leg actuation mechanism was such that the only restriction in the gas flow path was the valve. The mass flow of gas through this restriction $\dot{m}_{valve}$ determined how the pressure of the cylinder changed over time. For ratios of cylinder pressure P_{cyl} to supply pressure P_s below some critical level P_{cr}, flow through the valve can be considered linear and proportional to the supply pressure. When the ratio is above the critical value, the flow is nonlinearly dependent on the pressure ratio. The mass flow through the spool-and-sleeve valve under these two conditions is described as

$$\dot{m}_{valve,spool-sleeve} = \begin{cases} C_f A_v C_1 \frac{P_s}{\sqrt{T}} & \text{if } \frac{P_{cyl}}{P_s} \leq P_{cr} = 0.528 \\ C_f A_v C_2 \left(\frac{P_s}{\sqrt{T}}\right) \left(\frac{P_{cyl}}{P_s}\right)^{1/\kappa_{gas}} \sqrt{1 - \left(\frac{P_{cyl}}{P_s}\right)\left(\frac{\kappa_{gas}-1}{\kappa_{gas}}\right)} & \\ & \text{if } \frac{P_{cyl}}{P_s} \leq P_{cr} = 0.528 \end{cases} \tag{5.7}$$

where C_f is a non-dimensional flow coefficient, A_v is the area of the valve opening, C_1 and C_2 are model constants, T is the gas temperature, and κ_{gas} is the specific heat ratio of the propellant gas [17].

The second valve configuration with rubber-sealed valves required a different model to describe gas flow. Based on the manufacturer's specification for the valve's flow coefficient C_v, the volumetric flowrate q (in standard cubic feet per hour) through the valve is described by

$$q_{SCFH} = Y C_v \sqrt{\frac{(P_s - P_{cyl})\, P_s}{(SG_{gas})\, T}} \quad \text{where} \quad Y = \begin{cases} 907 \text{ if } \frac{P_{cyl}}{P_s} \leq P_{cr} = 0.528 \\ 1360 \text{ if } \frac{P_{cyl}}{P_s} > P_{cr} = 0.528 \end{cases} \tag{5.8}$$

where P_s and P_{cyl} are the core and cylinder pressures in psia, respectively, T is the gas temperature in Rankine, and SG_{gas} is the specific gravity of the gas [18]. This volume flowrate in SCFH must be converted back into m^3/s using

$$q\ \frac{m^3}{s} = \left(q_{SCFH}\ \frac{cu\ ft}{hr}\right)\left(\frac{m^3}{35.315\ cu\ ft}\right)\left(\frac{hr}{3600\ s}\right) \tag{5.9}$$

then the mass flowrate $\dot{m}$ is calculated using

$$\dot{m}_{valve,rubber-sealed} = \frac{P_s q}{R_{gas} T} = \frac{P_s\,(q_{SCFH}/127134)}{R_{gas} T} \tag{5.10}$$

where R_{gas} is the special ideal gas constant for the specific gas being used, T is the temperature in Kelvin, and P_s is the core pressure in Pascals.

The change in cylinder gas pressure $\dot{P}_{cyl}$ was also dependent on the position and velocity of the leg. The equation for change in cylinder pressure is derived assuming adiabatic gas expansion in the cylinder.

$$\dot{P}_{cyl} = \left(\frac{\kappa_{gas}}{V_0 + A_{piston} x}\right)\left(R_{gas} T \dot{m}_{valve} - P_{cyl} A_{piston} \dot{x}\right) \tag{5.11}$$

where V_0 is the inactive volume of the cylinder stroke.

The equation for cylinder pressure depends on the motion of the piston, while the piston motion is dictated by the forces applied to it, which including the gas pressure force. Since the two equations are coupled, they must be solved simultaneously in the simulator in

Table 5.3: Thermodynamic piston model parameters. (* indicates value from Lüders optimization)

Parameter	**Symbol**	**Value**	**Units**
Viscous piston friction	β	* 5.2843	kg/s
Coulomb piston friction	F_f	* 9.7003	N
Piston area	A_{piston}	$5.0165 \cdot 10^{-4}$	m^2
Return spring stiffness	k_{return}	100	N/m
Spool-sleeve discharge coeff.	C_f	* 0.5378	-
Spool-sleeve valve opening area	A_v	* $3.2781 \cdot 10^{-6}$	m^2
Spool-sleeve flow model constant	C_1	* 0.040418	-
Spool-sleeve flow model constant	C_2	* 0.156174	-
Rubber-sealed valve flow coeff	C_v	0.3	-
End of stroke inactive volume	V_0	* $5.4 \cdot 10^{-6}$	m^3
Critical pressure ratio	P_{cr}	0.528	-
Atmospheric pressure	P_{atm}	101325	Pa
Temperature	T	298.15	K
Specific heat ratio of nitrogen	κ_{N_2}	1.4	-
Specific gas constant of nitrogen	R_{N_2}	296.8	J/kgK
Specific gravity of nitrogen	SG_{N_2}	0.9669	-

numerical fashion. At every timestep, the current state of the piston (position and velocity) was used to calculate the pressure change, which was then integrated. The actuation force to be applied to the leg in the simulation was then calculated using the new cylinder pressure.

Many of the parameters used in this thermodynamic model are very difficult to measure directly. During development of the simulation for the 5-legged prototype, Lüders performed model fitting by optimizing the unknown parameters to minimize the error between the model predictions and actual position and pressure data obtained from a testbed setup. The fitted parameters he determined for the spool-and-sleeve valve configuration are given in Table 5.3, along with all of the other constants and parameters used in the model. Although the design of the leg piston changed slightly in some dimensions, the same spool-and-sleeve valves from the 5-legged prototype were employed in the initial configuration of the 12-legged pneumatic version. Because of the similarity in actuator design and for lack of resources to obtain position and pressure data for characterization of the new legs, the same parameters from Lüders' model were employed in the latest simulation of RATS.

5.2.4 Gas Depletion Model

The limiting factor in the single-charge endurance of the robot is the drop in supply pressure due to gas depletion. Under the assumption of no gas leakage from the system when the valves are deactivated (which was found to be approximately true for the newer air-pilot, rubber-sealed valves), the pressure drop in the supply tank can be easily simulated using the mass flow equations from above. In this case, we assume the gas tank depletion process

to be isothermal, resulting in the following equation for the change in supply pressure $\dot{P}_s$.

$$\dot{P}_s = \dot{m}_{valve} \frac{R_{gas} T}{V_{tank}} \tag{5.12}$$

where V_{tank} is the volume of the supply tank.

Gas depletion was incorporated in the simulation by integrating this equation at every timestep after calculating the mass flowrate and cylinder pressure change.

5.3 Discussion of Simulator Accuracy

The simulation behavior resembled actual robot behavior in a general sense, but did not accurately represent a few details. The first was that the ground contact model implemented in ODE was not perfect. The ground contact parameters could be tuned to allow proper tipping behavior or proper hopping behavior, but not both. The quasi-static nature of tipping with the real prototype makes measurement reasonably easy to obtain. More dynamic behaviors such as tumbling, hopping, and rolling present much more of a challenge in the area of sensing and state measurements. This highlights the value of using a simulation, which grants full access to the robot's state without the need for implementation and filtering of noisy sensors. The decision was made to focus the simulation on maintaining accuracy in the more dynamic behaviors of RATS, enabling the development of more complex feedback control techniques in simulation. Such controls work would otherwise require significant advancements in sensing and pose filtering before it could be tested on the real prototype.

Another simulation inaccuracy was the pneumatic model of the legs. As mentioned before, the model in use was tuned to match the legs of the 5-legged pneumatic prototype. Without re-optimizing the model parameters to fit experimental data from the new leg pistons, the pneumatic behavior in the simulation only provided a rough estimate of the true forces being applied to the legs. Beyond this, each of the 12 legs on the robot has a slightly different firing response for the same actuation command, as observed in Chapter 4. This may be caused by differing amounts of leg friction or other manufacturing details, and each of the 12 valves has slightly different response behavior upon initial actuation. This inconsistency in leg strength could be approximated by altering the friction term of the leg force equation for some legs more than others, or by implementing a longer delay timer for some valves. But unless the source of the true inconsistency is identified, these approaches would merely be hacks and the simulation's behavior would still not be truly accurate.

The third large inaccuracy in the simulation was the issue of mass distribution. ODE only allowed for uniform densities in its rigid bodies. In reality, the robot is by no means perfectly symmetric or uniform in mass distribution. The microcontroller board is attached to one side of the robot; the batteries are attached on another; the radio transmitter is in still another location. The robot is largely hollow with dense concentrations of mass

located at the legs, valves, electronics components, and leg support struts. All of these effects could be more accurately modeled by introducing many smaller rigid bodies, fixed with joints at precise locations to represent all of the intricate components of the prototype. But introducing such an assembly into ODE would have resulted in an extremely slow simulation (ODE would not handle so many interlocking constraints in an efficient manner) and would allow for many small numerical errors to compound as ODE attempted to find solutions to a system with so many constraints.

The bottom line is that the ODE simulator is a good approximation to the dynamic motions of the RATS prototype. Currently we lack the ability to obtain ground-truth state measurements from the real robot to compare with the simulator, which might allow us to optimize some of the model parameters. The simulation, despite its flaws, still served well as a platform for the development of control strategies for robot mobility. Knowledge of the inaccuracy simply provides motivation to develop robust control methods that will compensate for real-world differences. Additionally, all controllers must be designed with tunable parameters that can be adjusted to optimize performance when implemented in the real prototype.

5.4 Future Simulation Work

As mentioned in the above discussion on simulation inaccuracies, there are many areas for improvement in the current simulation of 12-legged pneumatic RATS. ODE allows for implementation of customized contact models, which could be more expressive of true contact behavior. More research into the intricacies of ground contact between the foot rubber material and the ground surface material would be required to develop a better model. The thermodynamic piston model could also be improved by applying model fitting techniques for matching with the new leg piston characteristics. More accurate modeling of the mass distribution would improve the simulation as well.

Beyond improving the existing simulation, there are some aspects of the robot that have not been modeled at all. Development of pose estimation techniques (which are necessary for implementation of most control techniques on the real robot) would greatly benefit from accurate modeling of sensors on the robot. Instead of directly using the true state information from the simulation, this data could be combined, rotated, modified, or corrupted to more accurately represent what would be expected from real sensors. Another area for enrichment is in the terrain model. Currently the ground is a smooth, flat, solid plane that may be inclined. The addition of discontinuities, smoothly changing slopes, surface compliance, and obstacles would provide a better testing environment for controller robustness in rough terrain locomotion. These and other developments are left as future work in the RATS project.

Chapter 6

Control Strategies for Mobility

The spherical RATS prototype was designed with the intent of achieving three modes of locomotion. The robot should be able to use tipping for precision motion, hopping for obstacle avoidance, and rolling for high-speed travel. The characterization work presented in Chapter 4 demonstrated the feasibility of these diverse mobility goals. Furthermore, that work established an understanding of how the mechanism design and actuator performance affects controls. Given this knowledge, I set forth in this chapter to present some control techniques to enable RATS locomotion. Here I seek to exploit the robot's symmetries and geometric leg arrangement to simplify the control policies and arrive at strategies that are robust and capable of being implemented on the prototype.

6.1 Tipping Control

I first investigate tipping locomotion as a means for following a predefined path on a plane. The quasi-static nature of this motion is the most predictable and repeatable of the locomotion modes, which makes it the simplest to implement in an open-loop manner on the prototype. Added feedback in the form of gravity detection via 3-axis accelerometer enables a tipping system that is much more robust. In this section, I present the techniques used to develop tipping sequences to follow a path, and also describe the adaptations that were needed to establish such tipping behavior with the prototype.

6.1.1 Path-Following by Tipping

Each of the RATS locomotion modes has certain situations where it is most effective or useful. There are some situations where RATS may need to navigate along a specific path. Examples of this could be the need to maneuver through a narrow gap in obstacles, move around the edge of a crater without falling in, or reach a precise goal point with minimal error. In such cases, RATS should employ its tipping mode because the quasi-static behavior is minimally stochastic and provides the most controlled, predictable motion. Here

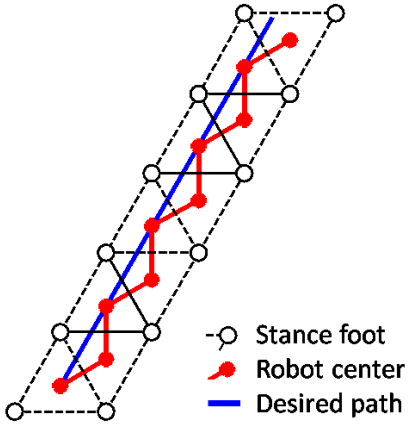


Figure 6.1: Closest approximation to straight line motion by RATS tipping. The robot can only move in one of three directions from each tripod stance.

I describe a tipping sequence planner for RATS that allows it to follow a pre-determined path using tipping.

Tipping, by nature, is a discrete form of locomotion that imposes unique constraints on how the robot moves. Since RATS can only tip over one of three edges from its current stance, it is constrained to motion in three discrete directions, 120 degrees apart. Additionally, the robot can only achieve positions in discrete locations of the plane (a grid of evenly spaced points with triangular arrangement, spreading out across the plane with orientation based on the robot's initial position and orientation). The three-legged passive dynamic walking robot STriDER at Virginia Tech moves very much like tipping RATS in its directional motion restrictions [19]. It swings one leg of its tripod stance through the other two and lands in another tripod stance without the pivot legs moving on the surface. As a result, it too is only capable of movement in one of three directions.

This discrete behavior presents problems when trying to follow a path, because the curve will most likely not coincide with the discrete locations RATS can reach, as shown in Fig. 6.1. It is also not likely that the steering direction needed to follow the curve at any given time will exactly correspond with the three discrete directions in which RATS can move. Thus the navigation problem becomes one of finding the best approximation to the desired path by points that the robot is capable of achieving.

The form of the proposed controller for path-following behavior in RATS is shown in the block diagram below (Fig. 6.2). Initially, a path was specified as a series of waypoints along which the robot was expected to follow. Given the desired path and the robot's initial position and orientation, the controller then generated a heading vector $\vec{H}$ that indicated the ideal direction of motion that RATS should try to follow. Once the heading was established, the controller proceeds to select the leg of the current support triangle that would tip the robot in a direction that most closely followed the heading vector. The correct leg was then fired and the robot tipped. RATS does not have feedback to know its location or complete

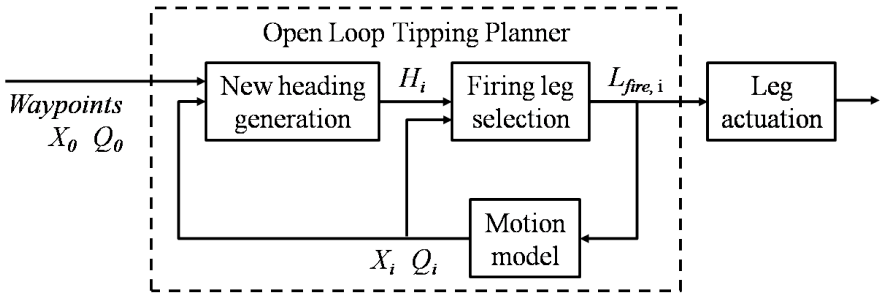


Figure 6.2: Block diagram of open loop tipping planner for path-following based on state estimate $\{X, Q\}$.

orientation, so a simple motion model was used to internally track the robot's progress along the path and maintain an estimate of which feet are on the ground. The process then started again with the selection of a new heading vector.

The best method found for selection of an appropriate heading was to choose the vector directed from the current estimate of the robot position toward the next waypoint W ahead along the path. The planner aimed for a certain waypoint W_{old} until it had gotten as close as possible to that waypoint. The "close as possible" condition was established once the target point lay within the triangle defined by the current estimated locations of the stance feet. This check was implemented by computing the barycentric coordinates (λ_1, λ_2, λ_3) of the target waypoint W relative to the triangle vertices (x_1, y_1), etc., a technique often used in computer graphics.

$$M = \begin{bmatrix} x_1 - x_3 & x_2 - x_3 \\ y_1 - y_3 & y_2 - y_3 \end{bmatrix} \tag{6.1}$$

$$\lambda_1 = \frac{(y_2 - y_3)(x_W - x_3) - (x_2 - x_3)(y_W - y_3)}{\det M} \tag{6.2}$$

$$\lambda_2 = \frac{-(y_1 - y_3)(x_W - x_3) - (x_1 - x_3)(y_W - y_3)}{\det M} \tag{6.3}$$

$$\lambda_3 = 1 - \lambda_1 - \lambda_2 \tag{6.4}$$

$$\text{Point in triangle iff} \quad 0 \leq \lambda_i \leq 1 \; \forall \; i \in \{1, 2, 3\}$$

Once this condition was achieved, a new waypoint W_{new} was selected by looking along the path for the next discrete point from the list that was at least a certain distance away. The minimum look-ahead radius R (shown in Fig. 6.3) can be varied to make the robot more far-sighted or near-sighted in its selection of the next waypoint. R was varied as a multiple of the support triangle side length L according to $R = \gamma L$ so that γ became a

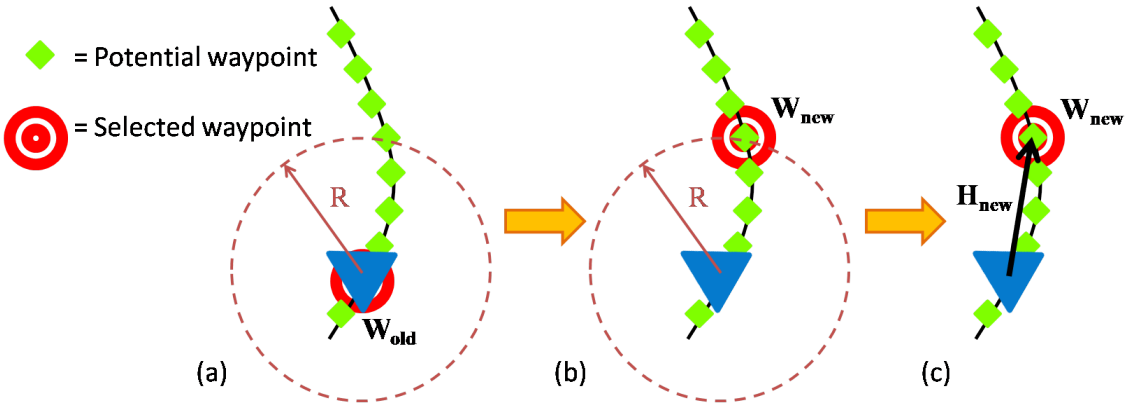


Figure 6.3: Selection method for a new heading in tipping path-following. Upon reaching the target waypoint (a), the planner selects a new waypoint beyond a minimum search radius (b), then sets the new heading to point toward the new target (c).

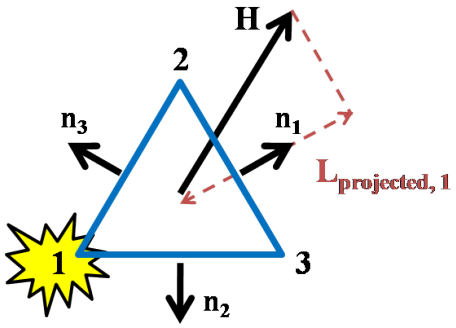


Figure 6.4: Firing leg selection for tipping path-following using projection of the heading vector onto edge normals.

dimensionless tuning parameter.

With the heading vector $\vec{H}$ determined, the planner selected the tipping edge so as to move RATS in a direction as close to $\vec{H}$ as possible. This edge selection was carried out using a standard inner product of the $\vec{H}$ vector with each of the three unit vectors normal to the edges of the support triangle $\hat{n}_i$ (where i denotes the triangle edge opposite from leg i). The value of the inner product was the projected length of $\vec{H}$ onto each of the edge normals, as shown in Fig. 6.4.

$$L_{projected,i} = \left\langle \vec{H}, \hat{n}_i \right\rangle = \vec{H} \cdot \hat{n}_i \tag{6.5}$$

The edge with the largest positive inner product value $L_{projected}$ was then selected to be the tipping edge, and the leg opposite from this edge was chosen to be the firing leg for the next step.

The planner tracked an estimate of the robot's pose using a simple motion model. This model first reflected the fired leg and the robot's center estimate over the tipping edge to

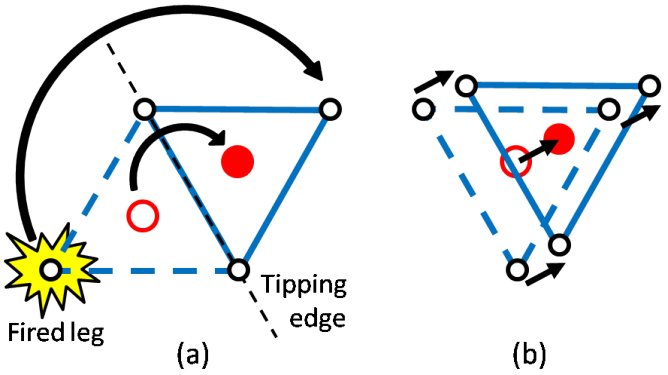


Figure 6.5: Motion model of tipping used in path-following planner. Model (a) reflects the fired leg and robot center over the tipping edge, then (b) translates the coordinates of feet and center by a small constant distance.

model the tipping motion (Fig. 6.5a). The coordinates of all four points (3 feet and center) were then translated by a small constant distance to model the slipping or short forward hop that the robot makes when it tips (Fig. 6.5b). This translation distance could be tuned to match the actual performance of the robot depending on the food/ground contact conditions. The new set of stance feet was determined by replacing the label of the fired foot with the new foot label of the expected stance, found in a lookup table representation of the finite-state transition graph shown in Fig. 6.6. The labels of the two tipping edge feet were not changed.

The path-following planner was tested in simulation on circular and figure-8 shaped paths. Fig. 6.7 shows the planner's solution to following a figure-8 path. The two key metrics of performance were number of firings required and the L_2 distance from robot center to the nearest point on the path, summed over all steps then divided by the number of steps. Clearly the planner should seek to minimize the average path-following error, but it should also spend as little gas energy as possible by minimizing the number of legs fired. These metrics appear to be contradictory, since cutting corners on a path would save firings but increase the tracking error. Different situations dictate the relative importance of these two objectives. If the path is leading the robot through a narrow gap, the planner should place higher importance on minimizing path error. If the mission hinges only on the robot's final position and does not place weight on the robot's path, minimizing the number of tips is desirable.

While experimenting with different look-ahead radii for new waypoint selection, I found the radius multiplier γ to be a suitable parameter for managing the error/energy tradeoff. Small search radius ($\gamma < 1$) caused the planner to step the robot back toward the path line whenever possible, resulting in a very jagged robot trail that oscillates around the pathline and tracks with minimal error (Fig. 6.8a). Using a large radius ($\gamma > 4$) caused the planner to focus on making forward progress at each step rather than trying to minimize error.

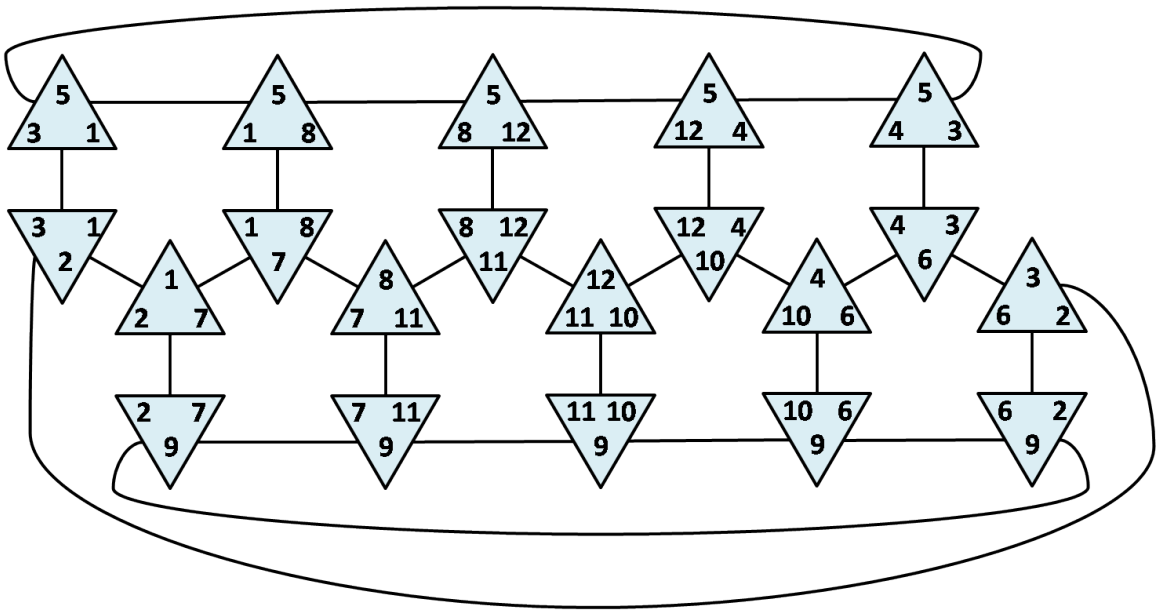


Figure 6.6: Finite-state transition graph used to predict the next stance resulting from a leg actuation. Firing one leg results in a stance transition along the graph edge departing from the opposite side of the stance triangle (the tipping edge).

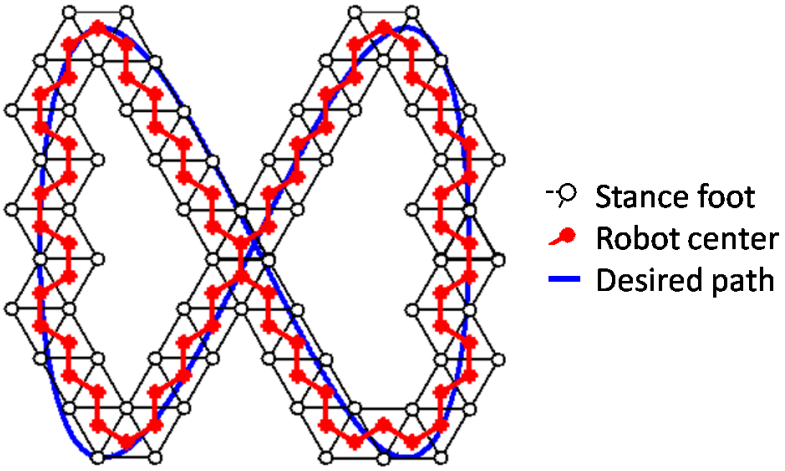


Figure 6.7: Tipping planner approximation of a figure-8 path.

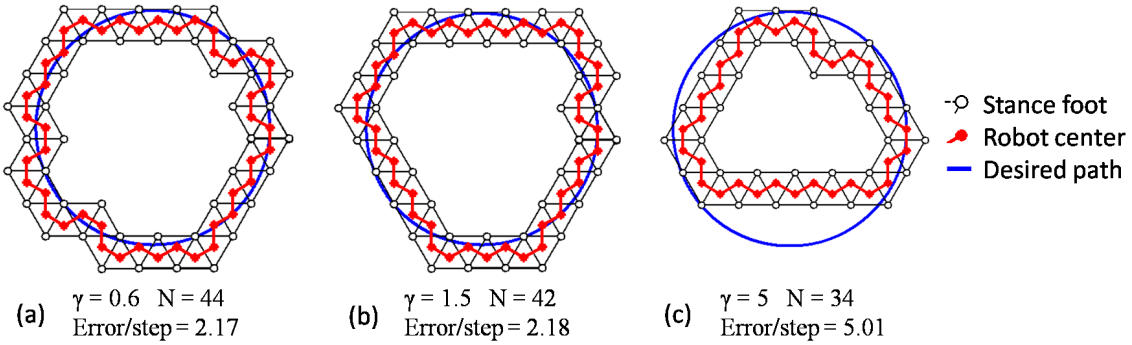


Figure 6.8: Tipping planner solutions for following a circle using different search radii to balance the tradeoff between path error and number of firings. $\gamma = 1.5$ was suitable for generic curved paths.

Sharp bends or small meanderings of the path may be missed using a long search radius and the robot was more likely to cut corners (Fig. 6.8c). This greedy strategy exhibited a lower quantity of legs fired but likely had higher path error for curved paths. For generic paths, a moderate search radius of $\gamma = 1.5$ was found to strike a satisfactory balance (Fig 6.8b).

6.1.2 Open Loop Tipping

Executing tipping as a mode of locomotion is a matter of selecting the proper leg to fire and determining the valve opening duration that has just sufficient leg strength for a successful tip. Firing events were initiated according to a timer (a leg fired every N milliseconds). In the initial open loop implementation of tipping with RATS, the firing leg was selected at each step by the tipping planner described above and all firing events were assumed to result in tipping success.

Selection of the appropriate valve time proved more difficult. As noted in Chapter 4, each of the 12 legs had a different threshold valve time for successful tipping due to differences in leg strength. Since I already knew these threshold values, the controller could simply look up the tipping valve time corresponding to the leg to be fired. This strategy only worked for the first 30-50 tips of a sequence, however. With each tipping actuation, the gas pressure in the supply tank was dropping, making the legs weaker after each tip. The valve opening times needed to be increased to compensate for gas depletion in order to achieve prolonged tipping performance from a single tank charge.

Gas Depletion Compensation

Compensating for gas depletion was a two-part problem. First, the controller needed to know the pressure of the gas in the core. The lack of pressure transducer in the core meant the pressure must be estimated using the initial pressure and a model of gas flow from the

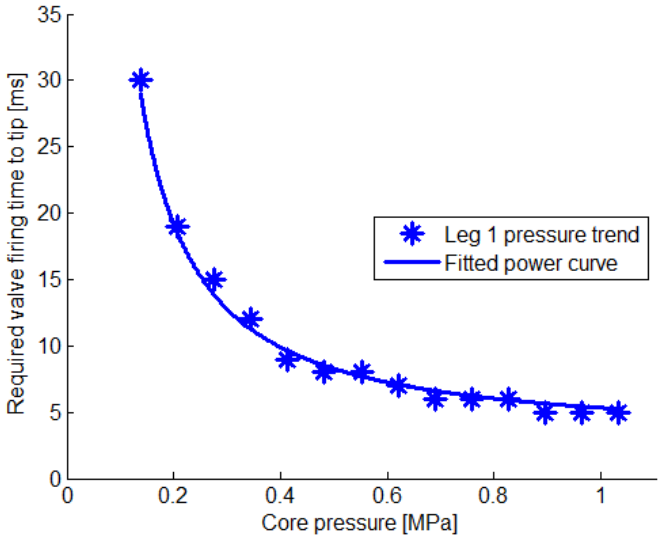


Figure 6.9: One of 12 power curves fitted to data from individual legs allowing for interpolation of required valve time for a successful tip at different core pressures.

core when the valves were opened. Conveniently, a thermodynamic model of the pneumatic system had already been derived for the simulation described in Chapter 5. The same equations (using the mass flowrate equation for the appropriate valve type) were employed to simulate how the gas flowed from the core tank. At each firing event, the gas flow model was integrated for the same amount of time as the actual valve was opened (the valve command duration minus the valve latency), resulting in a new core pressure to be used at the next firing event.

The second part of the compensation technique required the controller to determine the proper valve opening duration to achieve a successful tip given the current core pressure estimate and the firing leg. The simulation in its current state was not accurate enough to predict the necessary valve time for the real robot, and the pressure force inside the piston is very nonlinear over time. To get the best prediction of leg strength given pressure, I performed another characterization of the prototype legs and fit a regression curve to the data I recorded. For each leg, the most difficult tipping edge was identified, then the threshold valve time was found to achieve tipping over that edge at different core pressure levels. For three legs, I performed this test at pressures from 30 to 150 psi at intervals of 10 psi. Each of the legs exhibited a trend shaped like a power curve, shown in Fig. 6.9. I then performed the test for the other 9 legs at five pressure levels of 30, 50, 70, 100, and 140. All of these characterizations were performed on a sealed concrete floor.

In order to capture the fact that all legs were experiencing the same pneumatic process but that some of the legs had faster response times or were experiencing greater resistance to extension, I used a regression curve of the form

Table 6.1: Tipping leg strength model parameters used for gas depletion compensation.

Curve Shape Parameters	a	b
(for all legs)	2.2584	-1.1795
Curve Shift Parameters	$c_{resistance,i}$	$c_{latency,i}$
Leg 1	0.01175	3.0517
Leg 2	0.05455	2.1951
Leg 3	0.05705	2.8776
Leg 4	0.06298	2.7901
Leg 5	0.01396	2.0872
Leg 6	0.02718	4.7386
Leg 7	0.05915	3.5766
Leg 8	0.04534	2.5904
Leg 9	0.07336	2.8272
Leg 10	0.04756	2.2999
Leg 11	0.01646	4.1676
Leg 12	0.04789	2.0860

$$t_{fire}\left(P_s, L_i\right) = a\left(P_s - c_{resistance,i}\right)^b + c_{latency,i} \tag{6.6}$$

where P_s is the estimated core pressure in MPa and a and b are universal parameters describing the shape of the strength model and are used for every leg. Parameters $c_{resistance,i}$ and $c_{latency,i}$ are unique for each leg i and shift the model curve to fit the specific behavior of each leg. The parameters were optimized to fit the model to the observed valve times for all legs. The parameter values are given below in Table 6.1.

Given the firing leg, the proper leg curve shift parameters $\phi_i = \{c_{resistance,i}, c_{latency,i}\}$ were retrieved from a lookup table. These parameters, along with curve shape parameters a and b, could then be used to interpolate the desired valve time as a function of the core pressure. Interpolated valve times were rounded up to the next millisecond since valves are only controlled with millisecond resolution and conservative valve times are desirable (want to make sure the robot actually tips).

The entire compensation process, including the leg strength model and the thermodynamic gas depletion model is illustrated in Fig. 6.10.

Tipping performance is strongly dependent on the interaction between the foot and the ground surface. Some surface materials provide better friction, which has complex effects on tipping. Ideally, the tipping strength model should be recalibrated for each new surface, although this exercise was not undertaken during this course of research. To facilitate this calibration in the future, a new parameter $c_{surface}$ could be introduced to adjust the model for different surfaces as

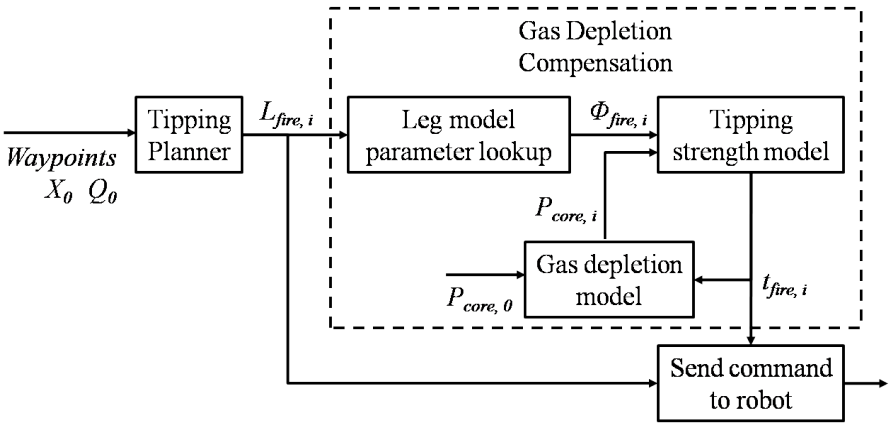


Figure 6.10: Block diagram of gas depletion compensation technique for prolonged tipping.

$$t_{fire}\left(P_s, L_i, surface_n\right) = a\left(P_s - c_{resistance,i} - c_{surface,n}\right)^b + c_{latency,i} \tag{6.7}$$

Open Loop Tipping Results

The framework for tipping control discussed to this point was sufficient to establish prolonged tipping performance with the RATS prototype. Without compensating for gas depletion, the robot was able to travel as far as 6.34 m in a straight line before it failed to complete a tip. The robot could follow a pre-specified path with some accuracy, but tracking error was introduced along the way. Fig. 6.11 shows the final robot location at its first failed tip from 10 attempts to follow a straight line path without gas depletion compensation. The deviation of the robot from the path occurred because of a slight rotation of the robot about the yaw angle (vertical axis) during each tip. This yaw motion is the result of unbalanced friction at the two stance legs while the firing leg applies forward thrust and is highly unpredictable. Such yaw motion was not captured in the motion model, and with no orientation or position feedback, the robot eventually strayed from the desired path. The robot also slips forward or backward somewhat randomly in the process of each tip, and while the motion model can express this motion, it is not consistent enough to predict without feedback.

A few trials of the straight line tipping were conducted on a ground surface that had been marked out with a grid to measure the locations of each stance foot after each tip was made (setup shown in Fig. 6.12). The robot's actual position from one trial is reproduced in Fig. 6.13, plotted over the internal motion model's estimate of the position. The effects of the slip and yaw while tipping are easily seen by the skewing and overlapping of the red stance positions.

The gas depletion compensation method described above had a large impact on the longevity of the robot's tipping. The robot could travel as far as 17.3 meters from a single initial tank charge of 0.689 MPa (100 psi) with an average distance of 17 meters.

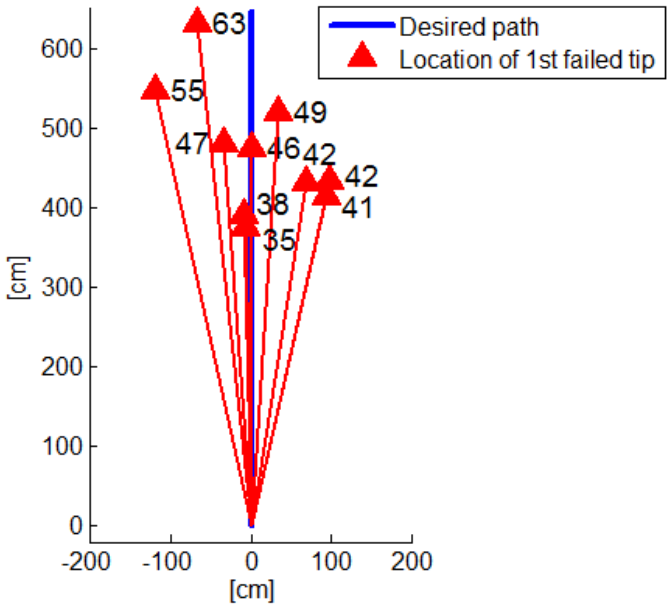


Figure 6.11: Location of first failed tip from 10 attempts to follow a straight line with an initial pressure of 0.689 MPa (100 psi). Different sets of initial stance legs were used for each attempt. Numbers at the location markers indicate the number of successful tips prior to failure.

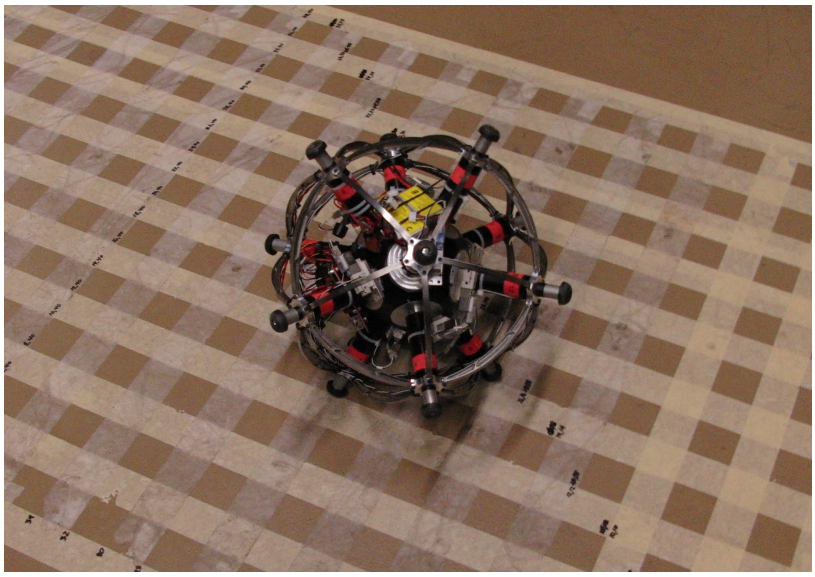


Figure 6.12: Experimental setup used to record foot locations after each tipping action.

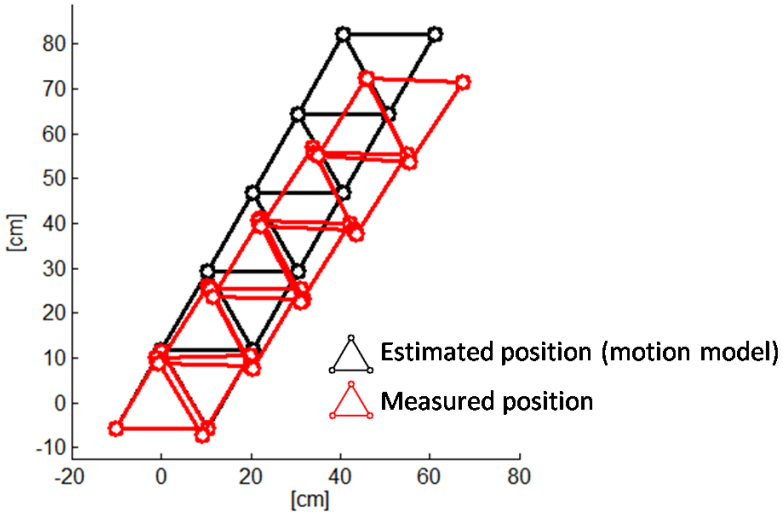


Figure 6.13: Comparison of motion model estimate with measured position of robot feet as RATS attempts to follow a straight line by tipping. Slippage and yaw motion accumulate to cause position tracking error.

This nearly tripled the traveling distance of RATS compared to uncompensated tipping (previous maximum distance of 6.34 m).

Beyond path tracking error, the tipping controller lacked robustness in a few other areas. Whenever the robot failed a tip, the controller had to be restarted because the path-following planner did not receive feedback, blindly assumed the tip was successful, and proceeded to fire the next leg in the sequence without correcting for the failure. These failures to tip were often the result of changes in ground condition, such as encountering dust on the floor or moving the robot onto a new surface to which it had not been calibrated. The robot was able to execute a rapid-fire tipping sequence (waiting 0.3 seconds between each leg actuation) to emulate high-speed rolling, but such behavior involved more highly dynamic and stochastic behavior that often resulted in unintended tips and rolling motion. The controller was incapable of adjusting for this unpredicted behavior, making it incapable of sustained high-speed rolling beyond a few meters traveled.

6.1.3 Closed Loop Tipping

The 3-axis MEMS accelerometers on board the RATS prototype enabled some feedback of the robot's orientation to improve the robustness of the tipping controller. This sensor package was used to detect the direction of the gravity vector, which is the only acceleration the robot experiences at rest. When the robot is tipping or rolling, the measured accelerations also include components from centripetal acceleration and impact forces. This reduces the accuracy of the orientation estimation, so without gyroscopes to augment the accelerometers, the feedback is only useful when the robot comes to rest after each tipping

action. Due to the limited bandwidth of the wireless radio link, the accelerometer output was sampled at a 16 Hz rate and digitized with a 12-bit ADC before being transmitted to the off-board control computer. The signals were passed through a digital first order Butterworth filter with cutoff frequency of 0.5 Hz.

The accelerometer was rigidly attached to the central core of the robot body. The relative orientation of the accelerometer to the body was established by recording gravity measurements while the robot rested on different tripod stances. The best-fit transformation matrix was then found to match the measured gravity vectors against the expected directions based on robot geometry. A lookup table was constructed to contain the expected gravity direction in IMU-frame for each of the 20 stance tripods while resting on a flat level surface.

The closed loop tipping process is illustrated in Fig. 6.14. Just before the tipping controller selected a leg to fire, the robot's current stance triangle was identified by projecting the filtered accelerometer vector onto each of the 20 expected directions from the lookup table and selecting the stance corresponding to the largest positive projection. If this detected stance matched the stance predicted by the tipping planner's stance propagation motion model, the planner proceeded with the next firing action in the sequence. If the stance matched the stance observed before the previous tip, the controller identified a failed tip and repeated the previous firing command. The controller can be modified to increase the valve time for a repeated firing command after a failed tip.

If the detected stance does not match the predicted stance or the previously observed stance, unexpected motion is assumed and the "Reverse location propagation" method is initiated. First, the controller launches a breadth-first search through the state transition graph in Fig. 6.6, starting at the newly detected stance and looking for the path with fewest number of transitions back to the previous detected stance. This technique relies on the assumption that the disturbance that caused the unexpected motion can be modeled as a sequence of additional, uncommanded tips. Once the shortest sequence of transitions is identified, the standard motion model is run forward from the previous location estimate according to that sequence to arrive at a new set of coordinates for the robot's foot locations.

Closed Loop Tipping Results

Closing the loop in the tipping controller further improved the prototype's robustness. The open loop system used the gas depletion compensation technique to prolong the number of tips RATS could execute before the first tipping failure, but when that failure inevitably occurred, the open loop planner could not recover. Gas depletion compensation was used again with the closed loop system, but by detecting the robot's orientation with respect to gravity, the planner could readjust its firing sequence to correct for failed or extra tips. Fig. 6.15a shows how the closed loop planner reacted to an unexpected stance after the robot was disturbed in the act of tipping (disturbance represented by the green line). The actual measured positions of the robot during the same event are plotted in Fig. 6.15b. The

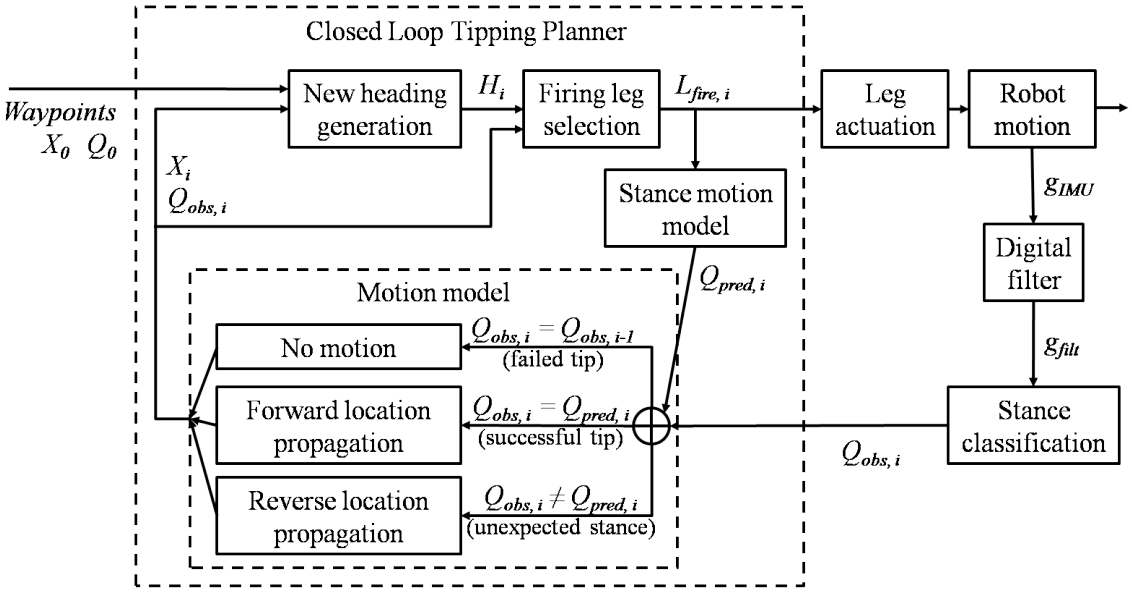


Figure 6.14: Block diagram of the closed loop tipping planner that used gravity detection for orientation tracking and tip error correction. Q represents the robot stance.

actual disturbance caused additional yaw and translation that the gravity-based orientation feedback was unable to detect. As a result, the position estimate from the planner's reverse location model contained error, but the robot was still able to correct for the disturbance and continue along the desired path toward its goal.

The ability to detect failed tips also made the robot more robust to changes in ground surface properties or deficiencies of the gas depletion compensation system. If the controller was set to increase all firing times every time the robot fails to complete a tip, it can quickly adapt to increase leg strength as necessary for more difficult surface conditions or low gas pressure. Failed tip detection could also be used to learn better values for $c_{surface}$, the surface correction parameter for the tipping strength prediction model.

6.2 Rolling Control

While it is possible to achieve high-speed rolling locomotion as an extension of tipping by accelerating the sequence of leg firings in time, truly dynamic and robust rolling requires a different form of control. The rolling and tumbling behavior of RATS is very stochastic due to the discrete contacts it makes with the ground surface as it moves. This makes any high speed rolling behavior difficult to sustain as an open-loop system.

In unstructured and unpredictable terrain, posture control and foot placement planning for the RATS robot would pose undesired complexity and computational requirements on the controller. In order to avoid such complexity and make the system as adaptable to terrain as possible, RATS does not attempt to control its orientation or maintain any specific

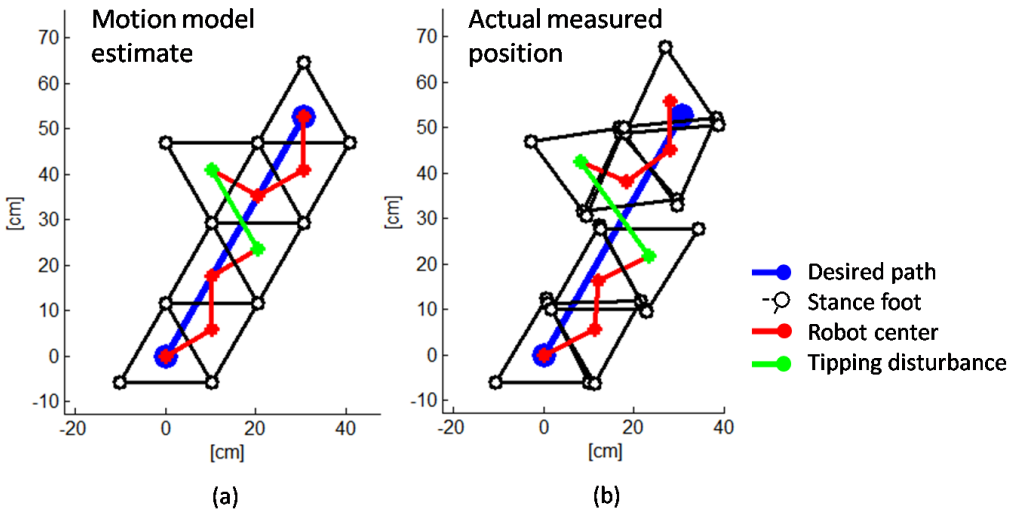


Figure 6.15: Successful performance of the closed loop tipping planner to adjust for a detected motion disturbance (indicated in green). The motion model's estimate of the robot position (a) failed to capture the yaw and translation of the actual robot motion (b).

body pose as it moves. The closed-loop rolling controller is a reactive system that makes actuation decisions in response to the robot state resulting from stochastic motion. Of course this makes the success of the controller heavily dependent on the accuracy of the orientation estimate. In order to separate the challenge of rolling control from orientation estimation, all development of a feedback rolling controller occurred in simulation, where state is readily accessible. Here I present the rolling controller that successfully achieved directed high-speed motion in simulation. I also discuss extensions of the controller to achieve speed control and turning behavior. The lack of adequate feedback and pose estimation at the writing of this thesis prevented implementation on the actual prototype, so I will outline the further developments necessary to achieve prototype rolling.

6.2.1 Straight Line Rolling

The RATS prototype has no control of the extension position or the orientation of its legs. Only the duration of valve openings, which affects the pneumatic forces applied to the leg, and the sequencing of leg actuations are controlled. This limitation combined with the limited number of legs arranged around the spherical body results in available control actions that are very discrete and bang-bang in nature. The robot's symmetry allowed for simplification of the control scheme. Instead of applying "fire or no-fire" logic to each of the 12 legs, the controller needed only to identify which of the 12 legs would help propel it forward and fire them at the appropriate time.

The original work in rolling control for the 5-legged planar prototype [14] provided the foundation for development of a rolling controller for the 12-legged robot. The basic concept

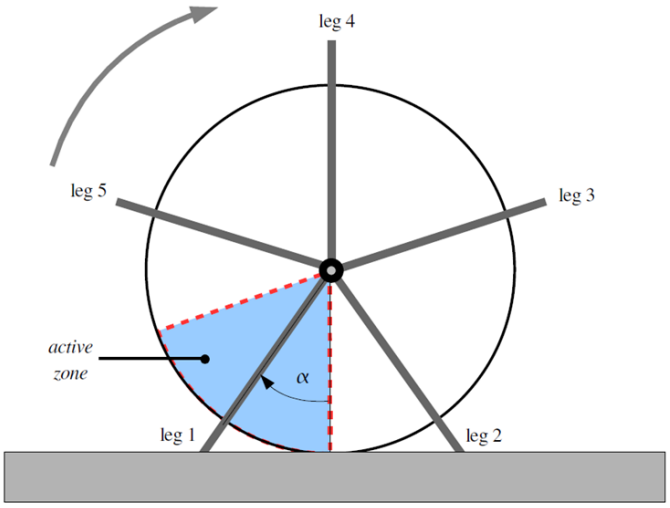


Figure 6.16: Illustration of active firing zone and firing angle parameter used in the 5-legged rolling controller [14].

behind the planar controller was to fire a leg only when the leg was within a certain angular zone of the leg disc. The zone was defined as starting at a certain angle (the firing angle) away from the surface normal vector, and once the angle between leg and surface normal entered the active zone (was greater than the firing angle), the leg was fired. This concept is illustrated in Fig. 6.16. The firing angle and zone were shifted around the wheel as a function of the rolling angular velocity.

The rolling control policy presented here is an extension of that developed for the 5-legged planar prototype. The 3D nature of the new RATS prototype required a new definition of the firing angle. Since legs are rarely ever directly trailing the robot in-plane (180 degrees from the desired heading), legs pointing to the left and right needed to be fired in alternating fashion. This way, the left leg counteracted the lateral force component of the leg-push forces from the right leg against the ground, and vice versa. The legs also needed to be fired when they were sufficiently trailing the robot center so that they provided a component of forward pushing force in the desired direction.

The solution was to project the leg positions onto a plane and reduce the dimensionality of the control problem. I defined the sagittal plane as a vertical plane passing through the center of the robot, containing the desired heading unit vector $\hat{h}$ and the gravity unit vector $\hat{g}$. Both of these vectors were tracked in a body-fixed frame by rotating them by the current robot quaternion. The unit normal vector of the plane $\hat{n}$ (also hereafter called the lateral direction vector) was then given by the cross-product of the heading with gravity. The unit vectors pointing out along each leg $\hat{R}$, which were stationary in the body-fixed frame, were projected onto this plane by

$$R_{sag} = \hat{R} - \left(\hat{R} \cdot \hat{n}\right) \hat{n} \tag{6.8}$$

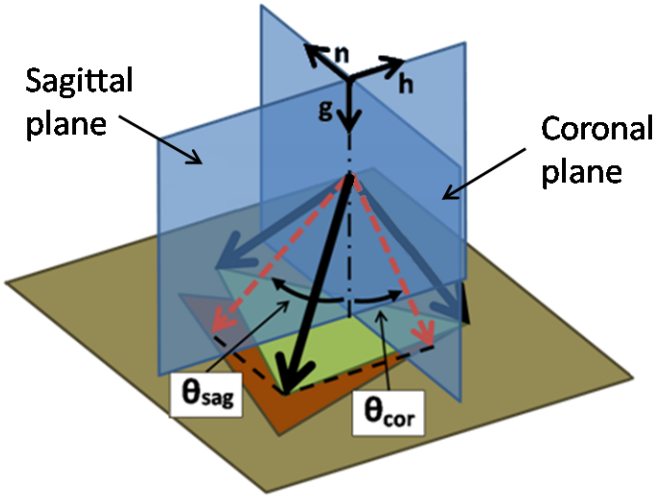


Figure 6.17: Projections of a leg onto the sagittal and coronal planes with corresponding projection angles.

then the sagittal projection angle θ_{sag} between the projected leg vector and gravity was calculated according to

$$\theta_{sag} = \text{sign}\left(\hat{g} \times R_{sag} \cdot \hat{n}\right) \arcsin \frac{|\hat{g} \times R_{sag}|}{|R_{sag}|} \tag{6.9}$$

Similarly, the coronal plane was defined as the vertical plane passing through the center of the robot containing the gravity vector and the lateral direction vector, and was perpendicular to the sagittal plane. The projection of the leg unit vector onto this plane was given by

$$R_{sag} = \hat{R} - \left(\hat{R} \cdot \hat{h}\right) \hat{h} \tag{6.10}$$

and the coronal projection angle θ_cor between the projection and the gravity vector was calculated by

$$\theta_{cor} = -\text{sign}\left(\hat{g} \times R_{cor} \cdot \hat{h}\right) \arcsin \frac{|\hat{g} \times R_{cor}|}{|R_{cor}|} \tag{6.11}$$

The two projection angles are shown in Fig. 6.17.

Two criteria were used to reduce the number of legs under consideration at any given time and eliminate the possibility of wasting gas by firing legs that would not hit the ground. The first was to enact a firing dead-zone based on elevation. When the center of the robot was above a certain height ceiling z_{max}, no legs were allowed to fire, regardless of their orientation. The second criterion was that legs would only be considered for firing if their sagittal and coronal projection angles were within an angular range of straight down.

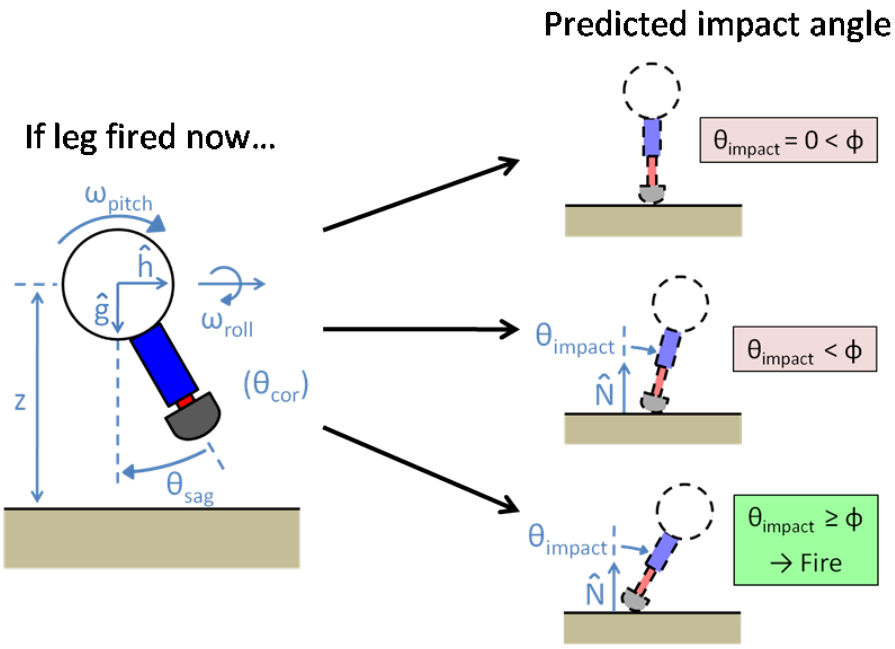


Figure 6.18: Control logic employed by the Impact Angle Rolling Controller (projection of a leg onto the sagittal plane is shown). The angle at which a leg would impact the ground if fired now is calculated using the current robot state. The leg is fired only if its predicted angle of impact with the ground is greater than the optimal angle ϕ.

$$\text{Legs eligible to fire only if} \quad \begin{cases} z \leq z_{max} \\ -\alpha \leq \theta_{sag} \leq \alpha \\ -\beta \leq \theta_{cor} \leq \beta \end{cases} \tag{6.12}$$

The planar rolling controller developed for the 5-legged prototype was based around the leg angle at firing time but had to adjust the angle as the robot's angular velocity increased. Instead, the latest controller focused on the leg angle at the time of impact, which affected the leg thrusting behavior much more directly. The key concept of the rolling controller was to repeatedly predict when each eligible leg would hit the ground if fired at that instant in time. A certain desirable impact angle $\theta_{impact} = \phi$ was defined as the angle formed between the leg's sagittal projection and the surface normal of the ground. The leg was fired at the time when the leg was predicted to hit the ground in the future with the desired impact angle. Accordingly, this control method was called the Impact Angle Controller. Fig. 6.18 illustrates the decision process executed by the controller calculations.

The policy to achieve a specific leg impact angle was implemented in much the same way as the 5-legged rolling controller. At each loop of the controller, a unique firing angle $\theta_{fire,i}$ was determined for each eligible leg of the prototype. Given the current robot state, the firing angle was calculated as the appropriate orientation of the leg that, if fired, should

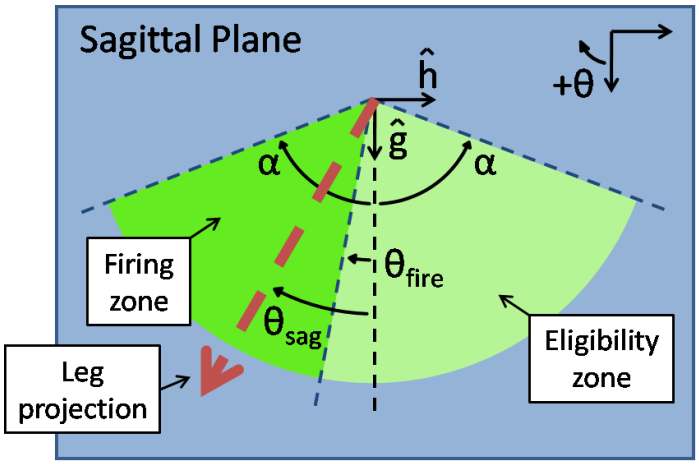


Figure 6.19: Projection of a leg onto the sagittal plane used to initiate actuation when the leg enters the firing zone. Legs are not even considered for firing when their projections lie outside the eligibility zone of $\pm\alpha$.

result in a successful impact at the optimal impact angle. This firing angle then defined the leading edge of a firing zone (shown in Fig. 6.19) and the leg was fired as soon as its sagittal projection entered that zone.

$$\theta_{sag,i} > \theta_{fire,i} \Rightarrow \text{fire} \tag{6.13}$$

The main function of the Impact Angle Controller was to calculate this firing angle according to the current state of the robot. The feedback involved in this calculation included the current elevation z, the pitch and roll angular velocitiesω_{pitch} and ω_{roll}, and the projection angles of the leg vector onto the sagittal and coronal planes, θ_{sag} and θ_{cor} .

First an imaginary leg vector was set to have the same coronal projection angle θ_{cor} as the real leg, but a sagittal projection angle equal to the desired impact angleϕ. The distance r_{total} between the center of the robot and the ground along this imaginary leg vector was then determined by

$$r_{total} = z\sqrt{\sec^2\theta_{impact} + \tan^2\theta_{cor}} \tag{6.14}$$

The distance Δr along this imaginary leg vector that the leg must extend (out from its retracted position) was then found by subtracting l, the length from the center to end of the retracted leg

$$\Delta r = r_{total} - l \tag{6.15}$$

If the necessary leg extension distance was longer than 6.3 cm, the leg would reach the end of its travel and never hit the ground. Due to the approximate nature of the calculations involved, another criterion was added to the leg eligibility conditions with a tunable limit

Δr_{max}

$$\text{Leg eligible to fire only if} \quad \Delta r < \Delta r_{max} \tag{6.16}$$

The leg had to be fired early enough to compensate for the valve response latency and the time necessary for the leg to extend. The predicted time to impact t_{total} was found using the average leg extension velocity $\bar{v}_{extend}$ empirically measured from the simulator and adding the valve latency t_{delay}

$$t_{total} = t_{extend} + t_{delay} = \frac{\Delta r}{\bar{v}_{extend}} + t_{delay} \tag{6.17}$$

The appropriate firing angle θ_{fire} for the given leg under its current orientation and conditions was then given by

$$\theta_{fire} = f\left(z, \theta_{cor}, \omega_{pitch}\right) = \phi - \omega_{pitch} t_{total} \tag{6.18}$$

If the current sagittal projection angle was equal to or greater than this calculated firing angle, the leg was predicted to hit the ground with the desired impact angle, so the controller fired the leg.

If the robot was rotating about the heading axis (roll), it would have a tendency to move in a direction away from the desired heading. To stop this behavior, additional negative feedback was added to adjust the coronal projection angle to compensate for side-to-side roll velocity. If the robot had roll velocity toward the side with the leg being considered, the coronal projection angle was increased by adding a small multiple gain ($\gamma \approx 0.05$) times the roll angular velocity.

$$\theta_{cor,adjusted} = \theta_{cor} + \gamma \omega_{roll} \tag{6.19}$$

This caused the leg to be fired slightly earlier than it otherwise would have, causing it to hit the ground earlier with a smaller impact angle, which stamps out the sideways roll behavior. This added feedback improved the direction tracking performance of the rolling controller.

There were seven tunable parameters in the Impact Angle Controller. The first four were related to the leg eligibility restrictions: the elevation dead-zone threshold h_{max}, the maximum sagittal projection angle α, the maximum coronal projection angle β, and the maximum leg extension distance Δr_{max}. Two others were the desired impact angle ϕ and the coronal angle feedback gain γ. The final parameter was the valve opening duration t_{fire} for each actuation command issued by the controller. These parameters were optimized from hand-tuned values using the simplex-based Nelder-Mead method and the final values are given in Table 6.2.

The optimized controller was found to be capable of generating stable rolling behavior in simulation. Because the firing condition was based on the sagittal projection of each leg, the

Table 6.2: Optimized parameters that govern the behavior of the Impact Angle Rolling Controller.

Controller Parameters	**Symbol**	**Optimized Values (Nelder-Mead)**	**Units**
Elevation dead-zone limit	h_{max}	0.289	m
Max sagittal proj. angle (* not optimized)	α	* $\pi/2$	rad
Max coronal proj. angle	β	0.797	rad
Max leg extension dist.	Δr_{max}	0.077	m
Desired impact angle	ϕ	0.093	rad
Coronal feedback gain	γ	0.032	-
Valve opening duration	t_{fire}	33	ms

Impact Angle Controller naturally produced the alternating left and right leg firing behavior necessary to induce forward motion. It could initiate rolling from rest and would constantly accelerate in the direction of the desired heading, regardless of its initial orientation. Fig. 6.20 shows an overhead view of the paths taken by the robot (blue) over multiple 8 second simulation tests as it tracked the commanded headings shown in red. The robot was started at rest from the same initial orientation in each of these tests.

6.2.2 Extensions to Straight Line Rolling

The Impact Angle Controller was found to successfully produce rolling behavior that could follow a single heading. The controller naturally compensated for the robot's rolling speed by actuating legs earlier as higher rolling rates were detected. This resulted in continuous acceleration of the robot's rolling speed. The controller, as presented so far, was incapable of regulating its rolling to a desired speed and was not robust to large changes in the commanded heading direction. Effective rolling locomotion would require both speed control and the ability to turn, so these capabilities are examined here as extensions and modifications of the existing rolling controller.

Speed Regulation

Since the rolling controller does not enforce specific gait trajectories, it is difficult to produce rolling behavior to achieve specific desired speeds. The methods for speed control considered here involve allowing the controller to accelerate the robot up to the desired speed, then attempting to limit the natural acceleration of the controller to maintain stable rolling at that speed. Such a modification to the controller would essentially establish a "brake" to hold the speed in check. Various methods were considered to counter this acceleration including an adjustment to the desired impact angle ϕ, stopping all leg firings above the desired speed, or reducing the valve opening duration of leg firings. Of these techniques, the one that proved most successful while maintaining rolling motion in the desired direction

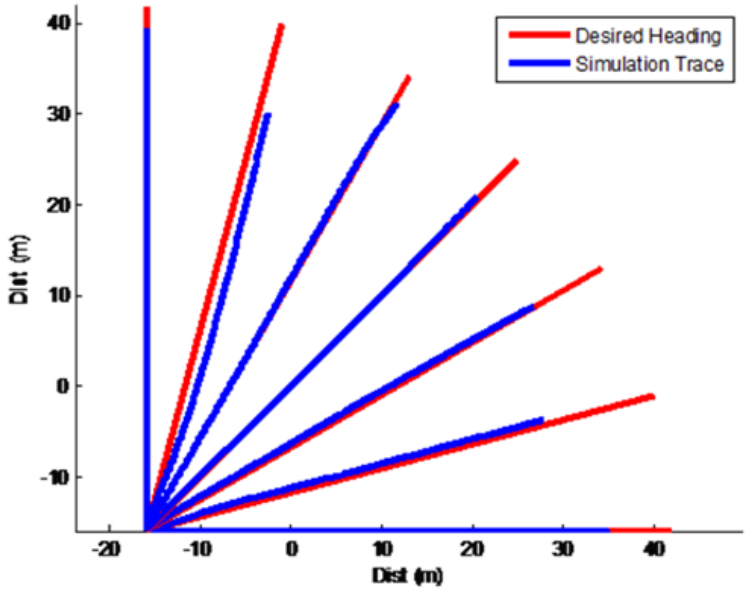


Figure 6.20: Overhead view of straight-line rolling paths of the simulated robot as it tracked different headings for 8 seconds.

was a reduction in valve times.

A simple uniform reduction in valve times for all legs when the robot rolled too quickly was ineffective. The robot would start to tumble off course, and although the rolling speed dropped quickly, the reduced duration firing events did not provide enough leg strength to continue smooth motion in the desired direction. A more intelligent approach was to vary the valve time reduction according to the coronal projection angle θ_{cor} of the legs. Legs with near-zero coronal angles, those more aligned with the sagittal plane and the desired direction of motion, provided the main thrust to propel the robot forward while rolling. Legs with larger coronal angles, those pointing more to the left and right, provided the alternating lateral thrusts to stabilize the robot and maintain directional control while rolling. With this understanding, it became clear that reducing the leg strength of the forward propulsion legs would slow the robot, while maintaining the strength of the lateral thrusting legs would help to retain directional stability.

First, a control signal ξ_{brake} was established as a measure of how strongly to apply the brakes. Initially this signal was set to be proportional to the velocity error $\Delta v = v - v_d$. For robot velocities below or equal to the desired speed ($\Delta v \leq 0$), no valve time reduction would be introduced and the braking signal would be zero. If the robot's horizontal velocity error was greater than some upper limit ($\Delta v \geq \Delta v_{max}$), the braking signal would be set to one. Accordingly, the proportional error signal was scaled and capped to fall in the range [0,1] as

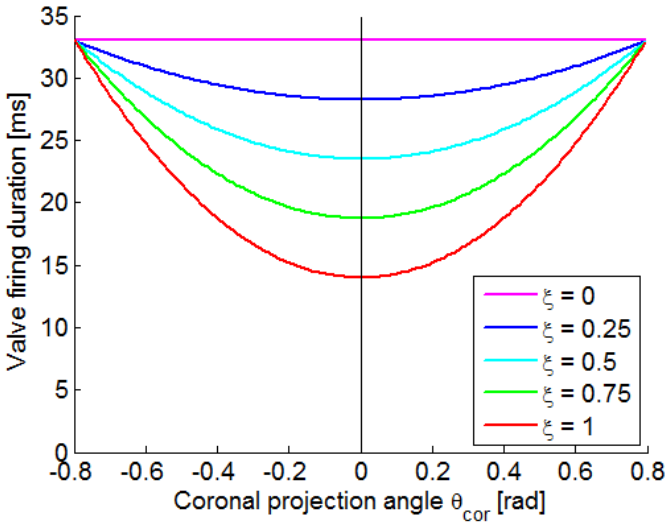


Figure 6.21: Family of quadratic functions used to control rolling speed by adjusting valve firing durations according to a leg's coronal projection angle.

$$\xi_{brake} = \max\left[\min\left[\frac{1}{\Delta v_{max}}\left(v - v_d\right), 1\right], 0\right] \tag{6.20}$$

The proposed method for speed control was then to develop a function that would assign valve firing times to specific legs according to the coronal projection angle of the leg and the braking value as calculated above. Since the function should be symmetric with respect to coronal angle (left-pointing legs treated the same as right-pointing legs), a family of convex quadratic functions was selected (shown in Fig. 6.21). The functions in the family would return the standard optimized valve time for rolling ($t_{fire} = t_{std} = 33$ ms) for the outermost values of the coronal angle in the eligible zone ($\theta_{cor} = \pm\beta$). The vertex of the parabolic curves, corresponding to the valve time of a leg lying in the sagittal plane with $\theta_{cor} = 0$, would be adjustable according to the braking signal. For robot velocities below or equal to the desired speed ($\xi_{brake} \leq 0$), no valve time reduction would be introduced and the parabola's vertex would be set to the same standard firing time value as the endpoints - the function would be a flat horizontal line at $t_{fire} = t_{std}$ and the controller would function as normal. When the braking signal was at the maximum of 1, the function's vertex would be set to the valve latency t_{delay}, resulting in no actuation of a leg if it is aligned with the desired heading. The function vertex would then vary linearly between t_{delay} and t_{std} according to the magnitude of the braking signal between 0 and 1.

$$t_{fire}\left(\theta_{cor}, \xi_{brake}\right) = \left(\xi_{brake}\frac{\Delta v_{max}}{\beta^2}\right)\theta_{cor}^2 + \left(t_{std} - \xi_{brake}\Delta v_{max}\right) \tag{6.21}$$

Better speed control was found to result by adding integral and feed-forward terms to the braking signal. The feed-forward term c_{ff} provided the necessary braking to maintain

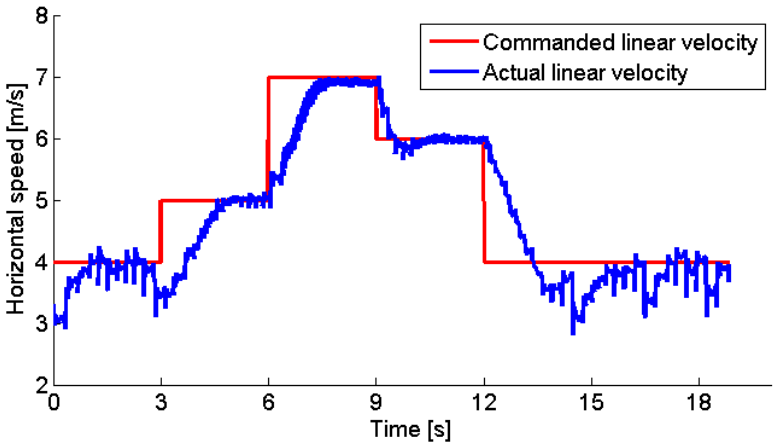


Figure 6.22: Step response of rolling controller to changing linear velocity commands. Robot behavior at low commanded velocity was unstable, but small steady-state error was observed for higher velocities.

steady-state rolling at the commanded speed. The proportional and integral terms then served to handle the transient speed reduction.

$$\xi_{brake} = \max\left[\min\left[\frac{1}{\Delta v_{max}}(v - v_d) + K_I \sum_t (v_t - v_d) + c_{ff}, 1\right], 0\right] \tag{6.22}$$

Hand-tuned values of Δv_{max}, K_I, and c_{ff} were selected and the parabolic braking technique was implemented in the rolling controller. Fig. 6.22 shows how the simulated robot responded to abrupt changes in desired speed. The robot rolling behavior was very unstable at the lowest velocity command, but the controller was able to maintain stable rolling with very small steady-state speed error for higher commanded velocities. Optimization of the braking parameters could improve on this performance. Using different feed-forward values depending on the commanded speed might also result in better speed control and eliminate the need for the integral term.

The speed control outlined here represented the most effective set of tweaks to the existing straight line rolling controller to compensate for the controller's natural acceleration. More active methods for effecting rapid changes to the robot's speed might be achieved by firing legs outside the framework of the Impact Angle Rolling Controller. A trailing leg fired at just the right time might add significant amounts of energy to the system to quickly increase the robot's forward speed. Similarly, a forward-pointing leg fired at just the right time might serve to absorb a large quantity of the robot's energy, significantly reducing speed or even bringing the robot quickly to a stop. Investigations into such techniques are left as future work.

Turning

The basic robustness of the existing rolling controller allowed the robot to follow a heading, even when that heading was dynamically shifting by small increments. For small angular differences between the desired heading and the robot's current rolling direction, the rolling controller was able to adjust and match the heading. Angular heading errors greater than 0.1 radians caused problems with the controller as projections of the legs onto the sagittal plane containing the desired heading no longer generated leg firings that maintained steady rolling. Such large heading error required a different approach for steering.

The angular heading error can be determined by comparing the desired heading $\hat{h}$ with the robot's true direction of rolling motion $\hat{h}_{roll}$. This rolling direction is found as the cross-product of the gravity vector with the projection of the current angular velocity vector onto the horizontal plane ω_{proj}.

$$\vec{\omega}_{proj} = \vec{\omega} - (\vec{\omega} \cdot \hat{g})\, \hat{g} \tag{6.23}$$

$$\hat{h}_{roll} = \frac{\hat{g} \times \vec{\omega}_{proj}}{|\vec{\omega}_{proj}|} \tag{6.24}$$

The heading error θ_{error} is then calculated by

$$\theta_{error} = \text{sign}\left(\hat{h}_{roll} \cdot \hat{n}\right) \arccos\left(\hat{h}_{roll} \cdot \hat{h}\right) \tag{6.25}$$

For small angular error magnitude, no turning actions are needed to augment the natural robustness of the straight-line controller. Outside of this range, the turning effort should increase with the heading error up to a maximum effort level. A turning effort signal ξ_{turn} can be used to capture this logic, similar to the braking effort signal. Fig. 6.23 shows one way to define ξ_{turn} according to the angular error such that it falls in [0,1]. The signal increases linearly from 0 at the empirical limit of rolling controller robustness ($\theta_{error} = 0.1$ radians) to 1 at the upper bound of maximum turning effort (chosen to be $\theta_{error} = 45$ deg $= 0.785$ radians).

For large heading errors ($\xi_{turn} > 0$), the sagittal and coronal planes used in the rolling controller logic were redefined according to the basis vectors $\{\hat{h}_{roll}, \hat{\omega}_{proj}, \hat{g}\}$. This maintained continuity in rolling behavior as the heading error increased to require turning effort.

Changing the rolling direction of RATS is a difficult task. The angular momentum of the rolling robot makes turning very difficult at high speeds. Small radius turning is only possible when the robot is rolling slowly, which means any turning effort should be accompanied by a reduction of the robot's speed. Fortunately, the same mechanisms that were under consideration to control the robot's speed can be used to achieve adjustments to the rolling direction. Adjusting the normal rolling control actuations by modifying valve opening durations and desired leg impact angles can both be used to induce turning behavior.

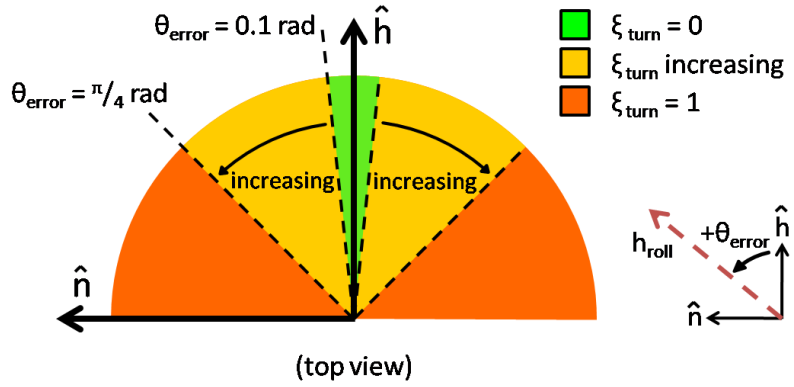


Figure 6.23: Concept for setting the turning effort signal according to the error between the detected rolling direction and the desired heading.

Added rotation about the heading axis (roll) is one way to add lateral motion and directional change to the robot's rolling. To do this, an imbalance must be introduced between leg actuations on the left and right side of the robot. If the robot needs to turn to the left, for example, it would be desirable to reduce the strength of legs on the inside of the turn (legs left of the sagittal plane) and increase the strength of legs on the outside of the turn (legs right of the sagittal plane). One effective technique to achieve this imbalance is to create a turning policy that does not fire legs on the inside of the turn, thus eliminating any thrust that would push the robot away from the desired turning direction. The imbalance could be further accentuated by increasing the firing durations of legs with larger coronal projection angles on the outside of the turn. This policy closely resembles the valve time variation implemented for speed control except for the asymmetry caused by the cease-fire for all legs on the inside of the turn. Also like the braking implementation, the magnitude of the valve time variations (and therefore the size of the lateral thrust imbalance) can be adjusted according to the strength of the required turning effort (the magnitude of ξ_{turn}).

Yaw rotation about the gravity axis is also desirable for turning behavior. This can be achieved by firing legs earlier on the outside of the turn to impact the ground at smaller or even negative impact angles. Reducing the impact angle of outside legs removes the forward component of the thrust from leg impacts. Instead, more of the thrust would be aimed in the lateral direction to generate turning in that direction. The problem with enforcing small (or negative) impact angles is that the resulting leg impacts also generate increased thrust forces in the vertical direction, adding a vertical velocity component to the existing forward rolling. This vertical motion is disruptive and counterproductive because the robot then loses contact with the ground, which is the only way it can modify its direction of travel. Legs at the outer limits of coronal angle eligibility introduce less of this vertical excitation than legs with very small coronal angle. Accordingly, a compromise could be enacted whereby legs with large coronal angle would have significantly reduced impact angles to

generate the desired yaw behavior, but the impact angles for legs with small coronal angles would not be modified so as to avoid undesired hopping. The magnitude of the yaw effect could be scaled with the turning effort ξ_{turn} by changing the magnitude of the impact angle adjustments.

These turning concepts were implemented in an exploratory manner on the simulator. Initial results indicate that valve firing time and impact angle adjustments are successful at achieving significant, albeit gradual, changes in rolling direction. Further, the method described above for adjusting the strength of the turning effort according the magnitude of the angular heading error maintains a smooth, uninterrupted rolling gait. There is still much work required to produce directional changes with small turning radius, and the results highlighted the importance of reducing the robot's forward speed to enable sharper turns. The added directional control was found to augment the existing rolling controller's ability to withstand significant external disturbances to the robot's direction of motion.

6.2.3 Future Work Needed for Implementation of Rolling Control

All of the rolling control techniques presented above had one thing in common that prevented them from being implemented on the real RATS prototype. The relative reduction in control complexity achieved through reliance on symmetry and fixed-angle leg arrangement comes at the cost of strong dependence on an estimate of the robot's pose. Every actuation decision made by the closed loop rolling controller assumed knowledge of the robot's orientation, rotational velocity, and elevation. At the writing of this thesis, access to the necessary robot state was only available when running in simulation. Much work is needed in the area of sensing and filtering for pose estimation before the prototype can replicate the simulated rolling behavior.

Clearly the current 3-axis accelerometer feedback available on the RATS prototype is far from sufficient for such dynamic pose estimation. The addition of gyroscopes to detect rotational motion would theoretically span all 6 degrees of free robot motion, but integrating the raw accelerations and angular velocities into useful pose estimates is still a difficult challenge. Unfortunately, the bouncing, tumbling, discrete contact nature of RATS motion is extremely difficult to model for the purposes of traditional Kalman filtering. Even multiple-model filtering techniques, where different physics models are used to predict robot motion depending on how many legs are touching the ground, would be challenging to implement because transitions between models are difficult to detect and predict [20]. Other sensing modalities are discussed in Chapter 8 that might provide better feedback to enable more effective pose estimation.

Beyond improving pose estimation techniques, reducing the rolling controller's dependence on the robot state would also improve implementation feasibility. Sensitivity studies should be performed to investigate which parts of the robot state are most important to achieve stable and effective rolling behavior. It may be possible to introduce assumptions

of steady-state values for state variables, or discover intrinsic relationships between multiple state components. The problem of feature selection is always a challenge in control policy development and some machine learning techniques may also be useful to eliminate dependencies on some portions of the robot state.

Chapter 7

Conclusions

The work presented in this thesis has shown the initial success of RATS as a mobility system with varied modes of locomotion for diverse terrain. The system characterizations discussed in Chapter 4 established techniques for actuator control and provided insight for the implementation of mobility strategies on the prototype. The ability to modify leg firing strength by changing valve opening duration has proven crucial to differentiate between locomotive behaviors, all using a single mechanism. Low energy actuation results in tipping, the proper sequencing of moderate energy actuations produces rolling/running, and high energy firing causes hopping. Harnessing 12 of these actuators in a spherically symmetric arrangement permits continuous locomotion that does not require maintenance of any specific body posture.

The specific hardware implementation of the current prototype, such as valve flow and latency, leg and foot geometry, and gas storage, was found to significantly impact actuation and mobility performance. While several adjustments were made to the design over the course of this research, many more key design changes were identified to improve the performance of future prototypes. It must be noted that the current design is less than optimal when considering each mode of locomotion separately. The tipping mode is less effective because the line of action for all legs passes through the robot's center, requiring ground contact friction at the stance legs to establish the necessary moment couple. Vertical hopping is also less efficient because the horizontal components of thrust from each leg are used to cancel the others. Both of these deficiencies would benefit from legs that are oriented more normal to the ground, but such a design would be effective only for certain orientations of the robot. High-speed rolling, however, benefits from spherical symmetry and radially aligned legs. When considering energy conversion, the nature of pneumatic solenoid valve actuation is better suited for hopping, which favors brief high-force impulses, than for tipping, which benefits from longer duration low-force thrust to maintain ground contact. The design compromises in the current version of RATS represent a solution that facilitates three modes of locomotion and omni-orientational mobility using a single form of actuation.

The development of control techniques for RATS mobility was successful on a number of fronts. The tipping of RATS is sufficiently deterministic to make high-precision positioning and path-following feasible. Open loop control does work on flat terrain after compensating for actuator and system characteristics such as non-uniform leg strength. Additionally, compensating for the drop in supply pressure caused by gas depletion increased the distance traveled while tipping by nearly 300%. Pose estimation, even stance determination by simple gravity detection, was found to make behavior like tipping more robust to stochastic ground interactions and unexpected changes in terrain.

The rolling controller developed in simulation is capable of stable rolling behavior in any commanded direction. The controller even exhibited control over rolling speed and the ability to change directions while rolling. The reactive nature of this rolling controller should make it especially robust to the unexpected effects of rough terrain. The heavy dependence of the closed loop rolling controller on the robot's state, including orientation, angular velocities, and elevation, requires significant advances in pose tracking and estimation before this technique can be implemented on the prototype robot. As the RATS project moves forward, significant thought will need to be applied to understanding how locomotion can be achieved using minimal feedback and sensing.

Chapter 8

Future RATS Development

This thesis has shown the feasibility of RATS to achieve three modes of locomotion, making it a viable solution for locomotion in diverse terrain. Experiments on the existing RATS design have investigated the basic principles of pneumatic leg control and actuation. I provided an initial look at controlling the prototype, demonstrating tipping as a low-dynamics mobility mode that can be executed largely as an open loop system. The presented rolling controller is a highly reactive system (thus requiring plentiful feedback) that can achieve high speed motion.

But this work has only scratched the surface of possibilities for RATS. As the program moves forward, we must first seek to understand what it is that RATS should be able to do. A top-down look at the system might reveal that one direction for development is to make a robot that can get from point A to point B as quickly as possible. Another approach might seek to minimize the energy expended in transit. Maybe energetics and stochasticity should be minimized to allow precision control of the robot's motion at all times. There are many objectives that RATS might attempt to achieve and the current embodiment of RATS might be the branching point for all of them, or it might already be way off base. The point is that any attempts to "improve" the RATS concept should be driven by the objective we hope to reach.

8.1 Future Design Considerations

Since I have been exposed to the best and worst aspects of the current RATS implementation, I have given much thought to redesign. Here I offer some insights and suggestions for design changes that may shift RATS through its broad design space toward more optimal capabilities.

8.1.1 Actuator Design

The current RATS actuators, its pneumatic piston legs, strike a fairly even balance between controlled, methodical motion and explosive, dynamic moving ability. If more hopping power is desired, clearly the first step would be to select valves with high pressure capacity to increase gas forces and maximum flowrate to facilitate rapid piston expansion. A larger piston bore will also provide more thrust force (at the expense of leg extension speed) and longer piston travel enables more hopping energy to be drawn from the gas pressure. Minimizing frictional losses in the piston and reducing the strength of the return spring will also improve leg strength.

But perhaps RATS should have finer control of the strength and actuation of its legs. In this case, a valve with low flowrate might be satisfactory, but improving the valve bandwidth could be helpful. The return spring stiffness contributes to the balance between leg strength and actuation cycle time/bandwidth. Adding degrees of control freedom further enables precision leg behavior at the expense of complexity. Position control of leg extension could be achieved using a double-acting piston and adjustable leg angles should make RATS more nimble, perhaps resulting in fewer legs needed for locomotion.

If efficiency is the goal, adding compliant pogo-stick structures to the legs would allow the system to retain more of its energy during impacts with the ground. Other compliance of the leg structure might improve natural stability as well.

8.1.2 Propellant Considerations

Propellant sustainability is an important consideration for many RATS applications. Addressing the loss of gas pressure over time as legs are fired would go a long way for improving the system's endurance. If an external atmosphere is available, an on-board compressor would allow the supply tank to constantly be refilled. A change in gas storage architecture would also help. Rather than pulling gas directly from a single storage tank, I envision a dual tank configuration with a high pressure tank feeding an accumulator tank through a pressure regulator. This would allow the core tank to be charged at much higher pressures while the leg valves draw gas from the accumulator at a lower, more manageable pressure level, further increasing the number of leg firings available from a single tank charge.

Other propellant sources should be considered as well. Liquids have a higher energy density, so gas in a highly compressed liquid form would greatly increase endurance. The initial RATS concept from Boeing involved the catalytic reaction of hydrogen peroxide to produce high pressure water vapor for propulsion. Combustion of a fuel to achieve high pressure gas is another option that has already been employed by Sandia National Labs in their hopper [6]. Each of these sources have advantages over storage of simple compressed gas, but additional technologies need to be developed and added to the RATS system before they become feasible.

8.1.3 Exterior Shape

The shape of the RATS body exterior is a feature that should be very customizable for the specific application. The protruding legs of the current design are useful for deterministic motion, but the biggest reason for adding spacers to the legs was to keep the robot from attempting to roll on the curved frame members. For unimpeded rolling behavior, perhaps the best exterior change would be to add a solid spherical shell outside the support frame, with legs that can retract within the shell body. This would allow the robot to start itself rolling via its legs, then continue rolling without much added energy. Another form could involve surrounding the robot with a single elastic skin that deforms as legs push outward. This would increase the surface area of contact during leg actuation and improve locomotion on compliant surfaces like sand, mud, or even water. Certainly the existing design needs to be made more rugged before outdoor exploration would be feasible. This could be achieved by adding an external skin and wiper seals on the piston shafts to prevent dust and debris from entering the pneumatic system.

8.1.4 Scalabilty and Payload

The size of the robot is also a matter of application. A small version of RATS would be more portable and could gain access to tighter spaces, but would also stand a greater chance of becoming entrapped under or between obstacles. As with all systems there is the usual tradeoff between size and mass, but RATS must also deal with the fact that the quantity of its gas energy supply is directly related to the available storage volume. A larger, heavier RATS version might not be able to jump as high as a smaller robot, but it would certainly be able to store gas for many more leg actuations. Practical matters such as the size/scaling of available control valves, on-board computation, and electrical power also affect the robot design.

The robot's size may also be driven by its mission payload. If RATS is carrying instruments that are to be dropped at various locations, or if it is equipped with additional cameras or radios, the necessary payload volume must be considered. Any volume committed to extra equipment reduces the available volume for gas storage. The mass distribution of these additional components is another key aspect that must be considered, as the dynamic motion of RATS is affected by changes in its center of mass.

8.2 Advancements in Controls

Besides changes to the mechanical design, advances in sensing and controls will have a lot to say about improvements to RATS mobility. It is important to consider how the objective drives the type of control and amount of sensing that is necessary. Given the difficulties in acquiring state feedback from the system, it may be useful to further explore the extent of what can be done in an open loop manner (or with minimal/low frequency feedback).

8.2.1 Investigation of Gait Control

Different gaits, or leg actuation patterns, were observed to arise naturally during the course of rolling controller development. Just as animals use different gait patterns for locomotion at different speeds, the rolling controller I developed in Chapter 6 tended toward certain leg patterns at different speeds. It may be possible to exploit this phenomenon to achieve spontaneous behavior by controlling the robot's gait as a trajectory of actuations. While naturally regulating the robot's speed, certain gaits might improve stability and robustness to terrain. This may be the key to successful open loop control for highly dynamic mobility.

8.2.2 Additional Feedback

If feedback is necessary, it becomes important to identify sensing modes that provide diverse feedback about as many things as possible. Besides the traditional accelerometer and gyroscope state feedback, other sensing modalities might prove more informative for pose estimation and terrain detection. In fact, successful detection of the ground surface should enable more accurate estimates of the robot's relative elevation and orientation. Such surface detection might be implemented in the form of simple mechanical whiskers that extend outward from the robot body and detect impending surface contact well before the robot's legs are within range of touching the ground. Infrared proximity sensors or laser rangefinders mounted at each leg or distributed around the robot's surface could perform a similar function to measure the distance to the ground surface and determine which face of the robot is oriented toward the ground. Distributed multi-scopic imaging from cameras around the robot might enable more detailed estimation of the robot's surroundings, including ground surface conditions and geometry, in addition to information on the robot's pose.

Knowledge of the state of the RATS actuators could also enhance controllability. It would be helpful to know when and which feet are in contact with the ground. This could be detected by contact sensors or force transducers on the robot's feet. Further, knowledge of the extension position of each leg would be possible with a linear position encoder. A pressure transducer tapped into the core gas storage tank would be very useful for adjusting valve durations according to supply pressure. Transducers to read the pressure in each leg cylinder could also be useful to measure and control the forces being applied to each leg.

While the usual information pathway is for environmental sensor data to be used in the control logic, reversing this flow of information may be useful as well. I have already demonstrated how the recognition of repeated failed tipping attempts can be used to adapt to ground conditions. This principle could be extended to draw contextual information about the environment from patterns in the robot's control actions. It seems likely that pattern matching on leg actuations could be used to infer the presence of an incline or an obstacle in the robot's path. This might lead to an obstacle negotiation or motion planning strategy that is less dependent on other sensing modes for knowledge of the terrain.

8.2.3 Optimization and Learning of Controls

Optimization is always an important part of good control systems. The successful straight-line rolling behavior achieved in Chapter 6 is the result of optimizing many of the control parameters in simulation. Optimizing control on the real robot will require learning techniques that permit the robot to adapt its control policies in real time and in a finite number of trials. Lüders demonstrated improved rolling behavior using a Q-learning approach on the 5-legged RATS prototype [14]. The objective of learning and adaptive behavior in the 12-legged RATS system will be to develop mobility strategies and controllers that are robust.

8.3 Applications of RATS

As already discussed, different skills and mobility traits are useful for different objectives. Here I will briefly review possible applications of the RATS technology described in this thesis. One of the main areas for use is enabling planetary exploration. On other planets, RATS might be used to access craters, boulder fields, caves, or other difficult locations that scientists find interesting, but which are not conducive to traditional locomotion systems. RATS could also be a high-mobility, high-speed, supplementary exploration robot to assist a wheeled rover in mapping and surveying broad areas of terrain. RATS might also be used on Earth to explore disaster zones, navigating quickly over and through rubble or dangerous environments.

Another application might see RATS as a deployable mobile sensor. RATS could be tossed through a window and allowed to maneuver through indoor environments to reposition a sensor payload. Outdoors, it might be used to drop beacons or transmitters at specific locations in the field or many RATS robots could carry radio equipment to serve as nodes in a dynamic, mobile, and reconfigurable communication network. The key to RATS is that it should be able to deliver accessibility and speed in rough terrains that are not attainable with other robots of equal size - wheeled, hopping, or otherwise.

Bibliography

[1] M. Buehler, R. Playter, and M. Raibert, "Robots step outside," in *Int. Symp. Adaptive Motion of Animals and Machines*, 2005.

[2] A. Bicchi, A. Balluchi, D. Prattichizzo, and A. Gorelli, "Introducing the sphericle: An experimental testbed for research and teaching in nonholonomy," in *IEEE Int. Conf. on Robotics and Automation*, pp 805-810, 1997.

[3] S. Bhattacharya and S.K. Agrawal, "Spherical rolling robot: a design and motion planning study," in *IEEE Trans. on Robotics and Automation*, vol. 16, no. 6, pp 835-839, 2000.

[4] K. Matsuoka, "A mechanical model of repetitive hopping movements," *Biomechanisms*, 5:251-258, 1980.

[5] M.H. Raibert, *Legged Robots That Balance*, MIT Press, Cambridge, MA, 1986.

[6] Peter Weiss, "Hop...hop...hopbots!", *Science News*, 159(6):88, Feb. 10 2001.

[7] P. Fiorini, S. Hayati, M. Heverly, and J. Gensler, "A hopping robot for planetary exploration," in *IEEE Aerospace Conf.*, vol 2, pp 153-158, 1999.

[8] B. Brown and G. Zeglin, "The bow leg hopping robot," in *IEEE Int. Conf. on Robotics and Automation*, pp 781-786, 1998.

[9] S. Stoeter and N. Papanikolopoulos, "Autonomous stair-climbing with miniature jumping robots," in *IEEE Trans. on Systems, Man, and Cybernetics*, vol 35, pp 313-325, 2005.

[10] M. Confente, C. Cosma, P. Fiorini, and J. Burdick, "Planetary exploration using hopping robots," in *7th ESA Workshop on Advanced Space Technologies for Robotics and Automation*, 2002.

[11] R. Armour, K. Paskins, A. Bowyer, J. Vincent, and W. Megill, "Jumping robots: a biomimetic solution to locomotion across rough terrain," in *Bioinspiration and Biomimetics*, pp 65-82, 2007.

[12] J.B. Jeans and D. Hong, "IMPASS: intelligent mobility platform with active spoke system," in *IEEE Int. Conf. on Robotics and Automation*, pp 1605-1606, 2009.

[13] B. Wilcox, T. Litwin, J. Biesiadecki, J. Matthews, M. Heverly, J. Morrison, J. Townsend, N. Ahmad, A. Sirota, and B. Cooper, "ATHLETE: a cargo handling and manipulation robot for the moon," in *Journal of Field Robotics*, vol. 24, no. 5, pp 421-434, 2007.

[14] R.A. Lüders, D. Apostolopoulos, and D. Wettergreen, "Control strategies for a multi-legged hopping robot," in *IEEE Int. Conf. on Intelligent Robots and Systems*, pp 1519-1524, 2008.

[15] G. Martin, *Transformation Geometry: An Introduction to Symmetry*, Springer, New York, NY, 1982.

[16] R. Smith, "Open Dynamics Engine v0.5 User Guide," Web, http://www.ode.org/ode-latest-userguide.html, 2006.

[17] E. Richer and Y. Hurmuzlu, "A high performance pneumatic force actuator system, part 1 - nonlinear mathematical model," in *ASME Journal of Dynamic Systems Measurement and Control*, vol. 122, no. 3, pp 416-425, 2000.

[18] "Flow Coefficient, Cv," Web, from *Smith Valve*, http://www.smithvalve.com/technical/flow-coefficient.aspx.

[19] J. Heaston, D. Hong, I. Morazzani, P. Ren, and G. Goldman, "STriDER: Self-excited tripedal dynamic experimental robot," in *IEEE Int. Conf. on Robotics and Automation*, pp 2776-2777, 2007.

[20] S. Skaff, "Context-based state estimation for hybrid systems with intermittent dynamics," doctoral thesis, Carnegie Mellon University, 2007.