# Development and Field Testing of the FootFall Planning System for the ATHLETE Robots

**Vytas SunSpiral**
SGT, Inc./NASA Ames Research Center
Moffett Field, CA 94035
vytas.sunspiral@nasa.gov

**DW Wheeler**
SGT, Inc./NASA Ames Research Center
Moffett Field, CA 94035
dw.wheeler@nasa.gov

**Daniel Chavez-Clemente**
Department of Aeronautics and Astronautics
Stanford University
Stanford, CA 94305
dchavez@stanford.edu

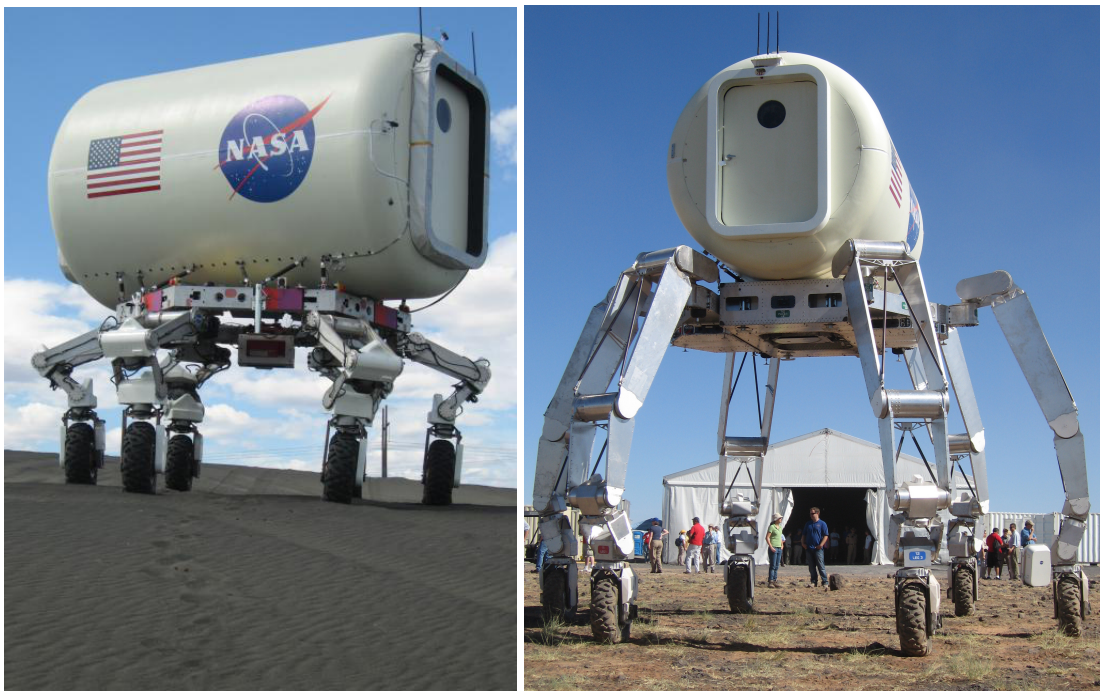
**David Mittman**
Jet Propulsion Laboratory,
California Institute of Technology
Pasadena, CA 91109
David.S.Mittman@jpl.nasa.gov

## Abstract

The FootFall Planning System is a ground-based planning and decision support system designed to facilitate the control of walking activities for the ATHLETE (All-Terrain Hex-Limbed Extra-Terrestrial Explorer) family of robots. ATHLETE was developed at NASA's Jet Propulsion Laboratory (JPL) and is a large six-legged robot designed to serve multiple roles during manned and unmanned missions to the Moon; its roles include transportation, construction and exploration. Over the four years from 2006 through 2010 the FootFall Planning System was developed and adapted to two generations of the ATHLETE robots and tested at two analog field sites (the Human Robotic Systems Project's Integrated Field Test at Moses Lake, Washington, June 2008, and the Desert Research and Technology Studies (D-RATS), held at Black Point Lava Flow in Arizona, September 2010). Having 42 degrees of kinematic freedom, standing to a maximum height of just over 4 meters, and having a payload capacity of 450 kg in Earth gravity, the current version of the ATHLETE robot is a uniquely complex system. A central challenge to this work was the compliance of the high-DOF (Degree Of Freedom) robot, especially the compliance of the wheels, which affected many aspects of statically-stable walking. This paper will review the history of the development of the FootFall system, sharing design decisions, field test experiences, and the lessons learned concerning compliance and self-awareness.

## 1 Introduction

The motivation of the FootFall Planner is to improve the process of commanding the ATHLETE robot while walking by providing situational awareness to operators, ensuring stability requirements, planning collision-free paths, and automatically generating command sequence and monitoring execution. In keeping with mission-realistic operations, it is assumed that ground-control operators will be part of the command loop and that a 2-10 second communications delay may be present between the robot and ground control. Thus, we designed FootFall as a decision support tool to ensure safe and error-free operation of the robot, rather than targeting fully autonomous operations. A central challenge to this work was the compliance of the high-DOF robot, especially the compliance of the wheels, which affected many aspects of statically-stable




(a) SDM-B at Moses Lake

(b) T12 at D-RATS2010. The central cargo pallet holding the habitat is visible between the two Tri-ATHLETE robots.

Figure 1: The Two Generations of ATHLETE Robots. For relative scale, notice that the simulated habitat module is the same on both robots. Not including the habitat, SDM-B can stand up to 2 meters tall and T12 can stand just over 4 meters tall.

walking. Over the four years of the project, we adapted FootFall to two generations of the ATHLETE robot, and tested it at two analog field sites.

ATHLETE is a technology development project managed by NASA's Jet Propulsion Laboratory in California (JPL). ATHLETE is capable of rolling over relatively flat terrain and walking over extremely rough or steep terrain. Two identical prototypes were constructed in 2005 and one of these, named SDM-B, is still operational. The second generation ATHLETE prototype was constructed in 2009 and is implemented as a coordinated system of two Tri-ATHLETEs, fully independent three-limbed robots. When the two Tri-ATHLETEs are operated as a single robot, which is the only configuration used by the FootFall Planning System for walking tasks, the resulting system is called T12. To avoid confusion between the two generations of the ATHLETE robots, this paper will refer to SDM-B or T12 when discussing aspects that are unique to the respective generations of ATHLETE.

The first generation SDM-B, shown in Figure 1(a), consists of six identical limbs each with six-degrees-of-freedom (36-DOF total) that enable the robot to stand approximately two meters tall at its central hex ring. Attached to the end of each limb is a wheel that can be used for mobility when driving over benign terrain. Alternatively, the wheels can be locked rotationally so that the limbs can be used for walking over rough terrain. The rover body is a hexagon, giving six flat faces that can be used to dock to similar ATHLETE vehicles, or to other systems such as refueling stations, rappelling winches, etc. SDM-B is able to travel at speeds of 3 km/h, climb vertical steps of 1.7 m, and carry payloads of up to 300 kg in Earth gravity. Additionally, ATHLETE can fold into a flat ring, allowing for easier transport to the lunar surface. (Wilcox et al., 2007), (Heverly and Matthews, 2008), (Wilcox, 2009), (Wilcox, 2011)

For sensing its environment, SDM-B has a number of stereo camera pairs. Each face of the hexagon has a

pair (NavCams) that provides the long distance view required for tele-operated driving. There are three more pairs (HazCams) mounted on the inner vertices of the hex that give visibility to the area below the robot and between the legs. Finally, some of the legs have cameras mounted above the wheels (ToolCams) that are designed to assist with the deployment and use of tools. For walking, we mainly rely upon the HazCams because the NavCams do not see the ground within reach of the legs. See Figure 7 for an example of viewable terrain. Other sensors on the vehicle include an Inertial Measurement Unit (IMU) for pose measurement, and dual absolute and relative encoders on all the joints, which enable the calculation of torques. (Collins, 2007)

The second generation ATHLETE prototype, T12, is implemented as a coordinated system of two Tri-ATHLETEs, fully independent three-limbed robots. This innovation allows a straightforward cargo handling strategy: two Tri-ATHLETEs dock to opposite sides of a cargo pallet to form a six-limbed symmetrical vehicle, work together to move and emplace the cargo, then undock and depart. This strategy provides all the advantages of the six-limbed design for cargo or habitat transport with the additional benefits of flexibility and modularity. The second-generation prototype is designed to demonstrate cargo handling at one half the anticipated lunar scale. Looking at Figure 1(b) it is possible to see the joint between the cargo pallet and the Tri-ATHLETE on the left (the line running front-to-back on the underbelly of the robot). This joint is a source of additional compliance which was not present in the SDM-B robot. Managing the various sources of extra compliance introduced to T12 will be discussed in Section 5.1.3.

The full T12 robot stands to a maximum height of just over 4 meters, and has a payload capacity of 450 kg in Earth gravity. The legs of the T12 system are twice as long as those of SDM-B and have been given an extra segment and joint to be able to step off a lunar lander. (Fig. 2) This brings the robot to a total of 42 kinematic Degrees of Freedom (DOF). (Heverly et al., 2010)

Because SDM-B and T12 have wheels at the ends of their legs, they can drive over smooth terrain and walk over rough terrain. This duality of movement style is ideal for robots operating in unstructured natural environments. The most appropriate mode of locomotion can be selected depending on the conditions. Driving provides fast energy-efficient motion but is limited to smooth terrain, while walking is slower and less efficient but also more robust to obstacles and challenging terrain.

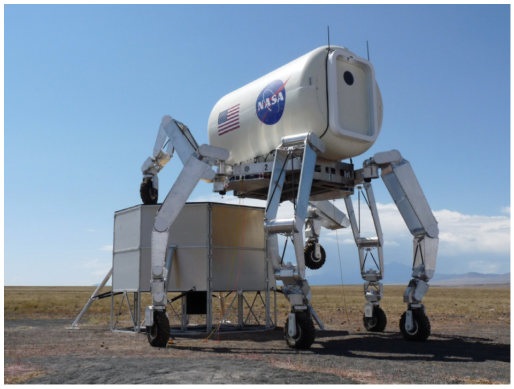



Figure 2: Second generation of ATHLETE, T12, stepping off a simulated lander.

The FootFall Planning System was developed over a period of 4 years, from 2006 through 2010, with the two field tests as the major milestones. The remainder of this paper follows the development process chronologically and is structured as follows: we begin in Section 2 by reviewing related works on walking robots. Section 3 will give an overview of the architecture of the system as developed for the 2008 field test at Moses Lake. Section 4 discusses the results from that field test and key lessons learned. Section 5 covers the development of the FootFall Planner from 2008 through 2010 and includes details on implementation, testing, and evaluation of new capabilities to address compliance, self-imaging, and gaited walking; a detailed discussion of the compliance model run-time performance metrics as measured at the JPL Mars Yard; and the migration of FootFall to the new T12 robot. Experiences from the D-RATS 2010 Field Test are discussed in Section 6, and finally conclusions and lessons learned from the four year project are given in Section 7.

# 2 Review of Walking Robots

The control and planning for legged walking robots is an extremely active field where research can be generally grouped into two categories: dynamically controlled robots and statically stable robots.

## 2.1 Dynamic Walkers

A large fraction of the current research in legged locomotion is focused on dynamically controlled robots (Campbell and Buehler, 2003), (Kimura et al., 2007), where the center of gravity (CG) is allowed to move outside of the polygon of ground contacts. This approach, which uses closed loop controllers to dynamically react to experienced forces and positions, has led to the success of well-known robots such as BigDog (Playter et al., 2006), which is capable of recovering its balance after slipping on ice.

These approaches require dynamic modeling (Poulakakis et al., 2005), and an actuation system with very high power-to-weight ratios. Many of these systems rely on high-pressure hydraulic actuators, which are unlikely to be used in a vacuum due to their large mass and the potential for leaks. ATHLETE uses highly geared, energy-efficient motors, and as a result cannot produce the rapid power outputs required for dynamic control. Instead, ATHLETE is a statically stable robot, where the CG is always maintained within the support polygon and dynamic effects are negligible.

## 2.2 Static Walkers

Some of the earliest examples of walking machines are from the late 1800's, such as the Steam Man of Dederick and Grass (Dederick and Grass, 1868), which was patented and built in 1868. The development of actively controlled walking machines acquired momentum in the 1960's, spurred by the advances in computer technology brought about by the space program. Notable robots from this period include the Iron Mule Train (Morrison, 1968), the GE Walking Truck (Mosher, 1968), the Phoney Pony (McGhee, 1967) and the Big Muskie (Cox, 1970).

The 1970's and 1980's saw further advances in multi-legged robot development. The first European walking robot was developed at the University of Rome in 1972 (Petternella et al., 1974), and others were built in Russia (Okhotsimski et al., 1979; Okhotsimski, 1980). In the United States, the OSU Hexapod was created in 1977 by McGhee (McGhee and Ishwandi, 1979), followed by the much larger Adaptive Suspension Vehicle by Waldron and McGhee in 1985 (Waldron and McGhee, 1986). In Japan, Hirose developed the PV II, a sophisticated quadruped which was the precursor of the impressive Titan family of robots still under development at the Tokyo Institute of Technology (Hirose et al., 1985; Hirose et al., 1991; Hirose and Arikawa, 1999; Kato and Hirose, 2001). Hirose also developed a Roller-Walking robot (Hirose and Takeuchi, 1995) with wheels on the ends of its legs, which gives it some similarity to ATHLETE. Hirose's robot has passive wheels and moves by a rollerblading motion, which is quite different from the deliberate stepping approach taken on ATHLETE.

Many statically stable robots have been built since (Kar, 2003), as their design and control is better understood than dynamically stable robots. Only a few have been deployed in natural unstructured terrains, like Dante II (Wettergreen, 1996), which descended into the caldera of an active volcano.

The design of these robots has generally been optimized to make the task of walking as simple as possible. Dante II, for example, is a frame walker with two sets of rigidly connected 4-leg frames that translate about each other. In many cases, the physical constraints on the problem have greatly simplified planning and the need for situational awareness. Likewise, in robots designed specifically for walking, lower DOF limbs can be stiffer and have fewer torque related challenges.

### 2.3 ATHLETE as a Static Walker

ATHLETE is unique among multi-legged statically stable walkers due to the complexity of its legs, which have seven degrees of freedom (DOF) each on the current T12 generation. While most walkers have the luxury of optimizing their mechanical design to facilitate walking, ATHLETE is constrained by many factors such as driving, tool use, launch vehicle considerations, load carrying, docking, the ability to step off a tall lunar lander, and maintaining a horizontal load platform. These considerations lead to the general-purpose seven-DOF legs used on ATHLETE, which in-turn increase the complexity of the necessary motion planning. Finally, the wheels which give ATHLETE access to energy efficient driving modalities also introduce a significant source of compliance and pose uncertainty, which must be accounted for during walking tasks.

Different approaches for generating gaits for various terrains have been studied. Hauser et.al. (Hauser et al., 2006) explored planning in the full 42-dimensional configuration space of the robot, an approach that is computationally expensive but is ideal for traversing very complex terrain. Overshadowing the time required to plan, a real world implementation of whole body motion planning also requires a full model of the environment, which requires processing 9 pairs of stereo images on ATHLETE and merging them together. Because a major motivator for the FootFall project was to simplify and speed up walking operations for current ground based development efforts and future flight missions, we decided early on in this project that this approach was too computationally expensive. Instead, FootFall plans for a single leg at a time and alternates body shifts with foot placements. Extending the current work to full terrain modeling and whole body motion planning will become reasonable after spending engineering effort on distributed multi-core processing and runtime optimization.

Beyond the complexity of the motion planning, a number of other complications exist for ATHLETE that differ from most statically stable robots in the literature. The length, mass, and high DOF of the legs make the torque considerations extremely prominent, with torque-limit exceptions common in the initial years of FootFall development. Similarly, the near omni-directionality of the robot and its ability to radically reconfigure itself complicate terrain reconstruction, situational awareness, and avoidance of self collisions. We will explore these issues in depth throughout the rest of the paper.

## 3 FootFall Planning System Development for SDM-B

From 2006 through 2008 the first version of FootFall was developed for SDM-B and then tested at the Moses Lake field test. The following sections will give an overview of the architecture of the FootFall Planning System at the time of the Human Robotic Systems Project's Integrated Field Test at Moses Lake, Washington, June 2008. Much of the basic structure deployed at Moses Lake has stayed the same even as we added enhancements and adapted the system to work with T12 in recent years.

### 3.1 FootFall Architectural Overview as of June 2008

FootFall facilitates walking planning for ATHLETE by suggesting command sequences, displaying the predicted results of the command sequences, and allowing the command sequences to be sent to the robot. The command sequences are displayed in VERVE (Fong et al., 2010), a three-dimensional visualization environment developed at NASA Ames Research Center which displays terrain as well as current and future configurations of the robot (see Section 3.6). Using VERVE, the operator examines the possible robot configurations before sending those commands to the robot. The initial version of FootFall planned commands for single steps; later, we introduced body shift commands and gait sequencing.

The single-step FootFall architecture tested at Moses Lake is illustrated in Figure 3; it consists of telemetry integration and processing modules that feed the display in VERVE. When the operator requests a terrain map, FootFall requests stereo images from ATHLETE and sends the images through the Ames Stereo Pipeline

(Moratto et al., 2010) to build a 3D model of the surrounding environment, and then colors the terrain map to indicate which areas are reachable and safe for the robot to step onto. FootFall displays the terrain map aligned with the robot model in VERVE. Next the operator selects a leg to step. FootFall uses robot mass models, the terrain map, and current robot configuration to evaluate reachable, stable step locations for the selected leg. The stable step locations are colored on the terrain map in VERVE.

When the operator selects a step location from the highlighted stable area, FootFall plans a trajectory for the leg by invoking the motion planning module which produces a sequence of joint angles, accounting for stability as well as for environmental and self collisions. FootFall displays the text of the commands and allows the operator to preview the predicted results of the commands in VERVE. If the plan is acceptable, the user sends the commands through FootFall to the robot. We discuss individual components of the architecture in more detail in the following sections, and an in-depth review of the architecture is available in (SunSpiral et al., 2008).

It is important to note that the entirety of the FootFall Planning System runs on the ground control computers. As such, all the work discussed in this paper is geared towards off-board planning for walking, and ultimately results in the generation of Cartesian and joint space commands to be uploaded to the robot for on-board execution by the flight avionics and controllers. For this work we were constrained to using existing on-board controls. For general information on the on-board controls, see the Motion Control section of (Wilcox et al., 2007), and for a detailed discussion of on-board controls for active terrain compliance during driving see (Townsend et al., 2010).

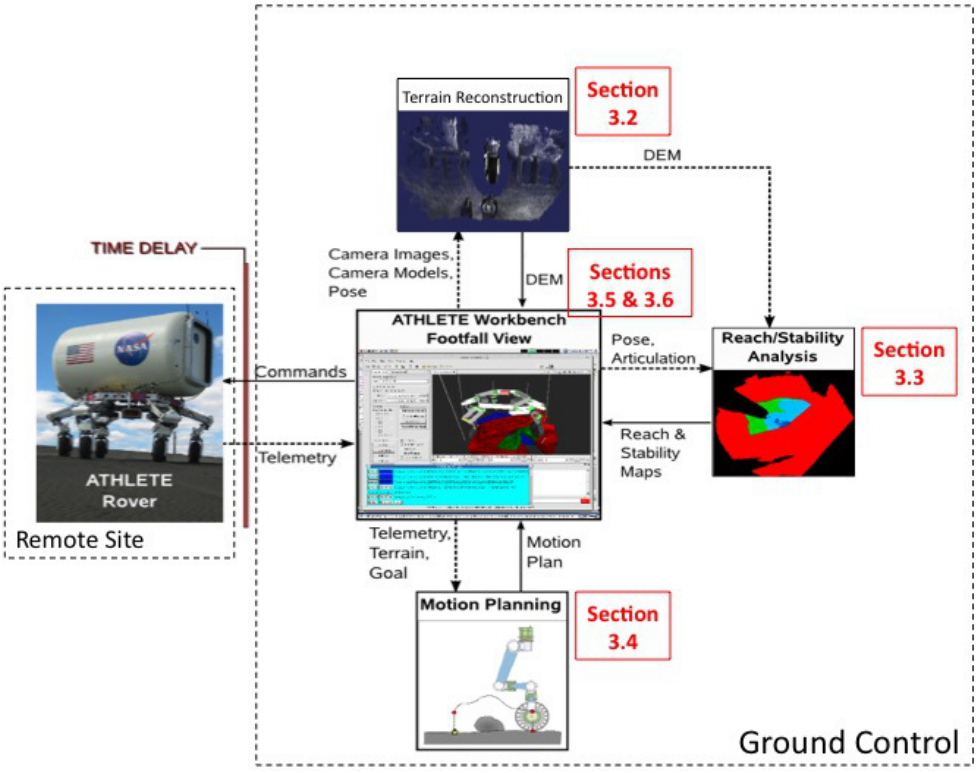



Figure 3: Overview of the FootFall Planner architecture as of Moses Lake Field Test in 2008. Modules are discussed in the sections indicated. See Figure 18 for the final architecture at DRATS 2010. Notice that all FootFall components run on ground control computers.

### 3.2 Terrain Reconstruction

The operator initiates terrain reconstruction by selecting stereo cameras and requesting a new terrain model from those camera images. SDM-B has nine stereo camera pairs, not including ToolCams, and many images show terrain that is not near the leg that will be stepping. Letting the operator select fewer images for terrain reconstruction saves processing time and communications bandwidth without discarding relevant information.

Each returned stereo pair is processed through the Ames Stereo Pipeline, which is part of the NASA Vision Workbench (Vision Workbench, 2010). The Vision Workbench is an open-source, modular, extensible computer vision framework developed at Ames; it supports a variety of space exploration tasks including automated science and engineering analysis and 2D/3D terrain reconstruction. FootFall uses the Stereo Pipeline to produce a Digital Elevation Map (DEM) of the terrain seen by the ATHLETE stereo cameras. Stereo Pipeline includes the functionality to align and blend DEMs created by different camera pairs. FootFall displays the integrated DEM in VERVE and uses the integrated DEM to check for environmental collisions while planning motions for the robot.

### 3.3 Reachability and Stability Analysis

Once the DEM has been generated, the user requests a stability map for the leg they wish to move. The stability mapper checks each point on the terrain map that is within a leg radius of the hip. If the mapper finds an inverse kinematic solution that puts the foot at the selected point, it evaluates the new leg configuration for stability. The stability mapper uses the new leg configuration to calculate the Center of Gravity (CG) and the Conservative Polygon of Support (CPS). The CPS is the intersection of all support polygons drawn with the feet in ground-contact minus one, to simulate the failure of one of the supporting legs. The mapper then assesses vehicle stability using either the Static Stability Margin (SSM) or the Normalized Energy Stability Margin (NESM). The SSM is the distance of the CG from the closest edge of the polygon and is efficient to compute. When walking on sloped terrain, the NESM method of calculating stability can be used. NESM, which was proposed by Hirose (Hirose et al., 2001), computes a robot-mass normalized measure of the minimum increase in potential energy required to rotate the CG over an edge of the CPS.

Stable positions are marked green on the terrain map; unstable positions are marked red. To compensate for robot compliance (discussed in Section 4.1), all step sequences go through waypoints directly above the start and goal positions. These waypoints ensure that the wheel breaks contact with the ground as the leg is unloaded and the robot sags. The compliance waypoints are also checked for reachability.

### 3.4 Motion Planning

#### 3.4.1 Single-Query Bi-Directional (SBL) Motion Planner

The user selects a point on the stability map to tell FootFall to plan a step to that point. FootFall finds the inverse kinematic solution for the foot at that point and explores the configuration space of the leg to find a collision-free path between the current and desired configurations. The algorithm FootFall uses to find the path is a sampling technique known as the Single-query Bi-directional planner with Lazy collision checking (SBL) that was developed by G. Sanchez and JC Latombe. (Sanchez and Latombe, 2003)

As with all other sampling-based motion planning approaches, SBL works on the principle that it is considerably cheaper computationally to check if a single configuration is collision-free than to project the environmental model into the high dimensional Configuration Space. It conducts the search for feasible paths by sampling configurations between the start and goal and then verifying if (a) the configurations are feasible and if (b) the configurations can be connected without collisions.

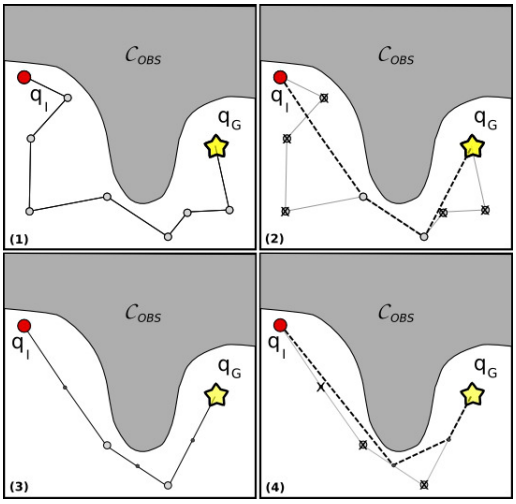



Figure 4: Path smoothing in a hypothetical 2D C-space: the initial and goal configurations ($q_I$ and $q_G$) are separated by a C-obstacle. (1) Motion plan with $N_1$ nodes generated by SBL is usually not smooth; (2) the shortest path is found using Dijkstra's algorithm, resulting in a smoother motion plan with $N_2 \leq N_1$ nodes; (3) the simplified path is bisected, a step that adds $N_2$-1 nodes; (4) Dijkstra's algorithm is re-run.

In the literature we find exact and approximate techniques to check collisions along the edges that connect configurations (Schwarzer et al., 2005). FootFall uses an approximate technique that iteratively bisects a linear edge between the two configurations, until a collision is detected or a maximum pre-specified level of resolution is achieved without collisions. This technique is $\epsilon$-accurate with $\epsilon$ currently set to 0.01 rad for our application. The bisection technique works well in detecting collisions because segments in collision have a high probability of having their middle point being in collision. Thus, performing successive sub-segment bisection will generally detect collision along the path quickly.

At the lowest level, FootFall checks for two kinds of collisions: self-collisions and environmental collisions. For this purpose, FootFall uses the Proximity Query Package (PQP) library (Gottschalk et al., 1996) (Gottschalk et al., 1999) from the University of North Carolina. PQP detects collisions efficiently by representing triangulated mesh models as hierarchical trees of oriented bounding boxes (OBB-Trees). PQP also provides the option of signaling a collision when two objects come within some distance $\epsilon$ from each other. Initial testing with SDM-B showed that the appropriate value of $\epsilon$ is not uniform between all robot parts. Components that are ordinarily far from each other need a larger $\epsilon$ than close component pairs because their relative positions have a greater uncertainty due to compliance and encoder noise. To accommodate these differences we have manually generated a lookup table indicating values of $\epsilon$ for each pair of components of the robot. In contrast, environmental collisions use one value of $\epsilon$.

### 3.4.2 Path Smoothing

The plans that SBL generates are feasible, but in general they are neither smooth nor acceptable for execution. Large swings of the legs and jerky back-and-forth motions are typical due to the nature of randomized sampling. For better plans, FootFall implements post-process smoothing (Fig. 4). FootFall iteratively discards unnecessary nodes via Dijkstra's algorithm and bisects the simplified plan to provide new nodes used for further path simplification. New path segments are checked for collisions to ensure that the smoothed path is still safe. These steps are repeated until the improvement between successive iterations is no longer significant (measured by the reduction in total Euclidean path length in configuration space). The result of combining this form of path smoothing with the output of the SBL planner are very compelling. One can think about this two-step process as first finding a feasible path (SBL planner) through the high dimensional configuration space, and then using that path as a seed for an optimization process (path smoothing) that

produces a simple and efficient set of motions to arrive at the goal.

### 3.5 Advanced ATHLETE Ground Software

The Planning Software Systems Group at the Jet Propulsion Laboratory has developed a suite of software applications that supports the development of advanced operations technologies for the ATHLETE rover. Each of these applications is built on NASA's Ensemble software application framework, an Eclipse-based platform for graphically rich application software development. The ATHLETE suite of ground software applications is centered around the AthleteWorkBench and is designed to manage the complexity of the ATHLETE rover and to increase the efficiency of the operator in conducting prototypical lunar-style tasks. The FootFall software has been designed to interoperate with this suite of ground tools using the ATHLETE Telemetry Bridge, and it is encapsulated as a perspective available to the AthleteWorkBench.

The ATHLETE Telemetry Bridge (AthleteBridge) software application provides a realistic ground data telemetry processing system in a research environment. The AthleteBridge receives telemetry from and provides commands to the ATHLETE robot. The received telemetry is logged to a persistent storage medium and is made available to interested clients through a broadcast over the network.

Both command and telemetry distributed data streams are provided by the Robot Application Programming Interface Delegate (RAPID) system, which is designed to promote interoperability between robot software modules (Torres et al., 2009). RAPID defines a standard messaging interface and data distribution middleware. RAPID was created to support basic research within the NASA Human-Robotic Systems (HRS) project including laboratory experiments and field tests. RAPID helps: (1) facilitate integration of experimental robot software modules created by a distributed development team; (2) improve the compatibility and reusability of robotic functions; and (3) speed prototype robot development in a wide range of configurations and environments.

The ATHLETE Application Workbench (AthleteWorkbench) provides ground-based command and telemetry monitoring displays for the ATHLETE robot. The AthleteWorkbench combines several different operational tasks into a single integrated application, including commanding, telemetry monitoring, image browsing, mapping, database annotation, and advanced user input mechanisms. To operate the ATHLETE robot, the driver sits at an immersive cockpit that supplies the computer and display hardware necessary to support the advanced operations software, including a stereo image viewer that provides the operator with depth perception without the need to wear special glasses. This immersive system is excellent for many tasks (especially driving), but does not provide robot operators with the full situational awareness required for many walking tasks. Given the extreme reconfigurability of the robot it helps to have a third-person perspective of the full robot situated in the environment.

### 3.6 VERVE (3D Viewing Environment) and the User Interface

To complement the immersive stereo displays with a third-person perspective, we have integrated the Visual Environment for Remote Virtual Exploration (VERVE). VERVE is an interactive 3D user interface for visualizing high-fidelity 3D views of the rover's state on a terrain map in real-time that has been developed at NASA Ames. VERVE has been in use by several robots in the Human Robotic Systems project (Fong et al., 2010). In FootFall, VERVE displays the current configuration of ATHLETE in the context of surrounding terrain. As discussed in Section 3.3, the terrain is false-colored according to the stability analysis to aid the operator in selecting a desired step location. When the step location is selected and the planner suggests a sequence of commands, the operator previews the results of each command in the VERVE window by adding a ghost robot to the display. Once the operator is satisfied with the plan, he sends the commands to the robot through the FootFall GUI. (see Figure 5)

VERVE runs within the NASA Ensemble framework and supports a variety of robot telemetry, including

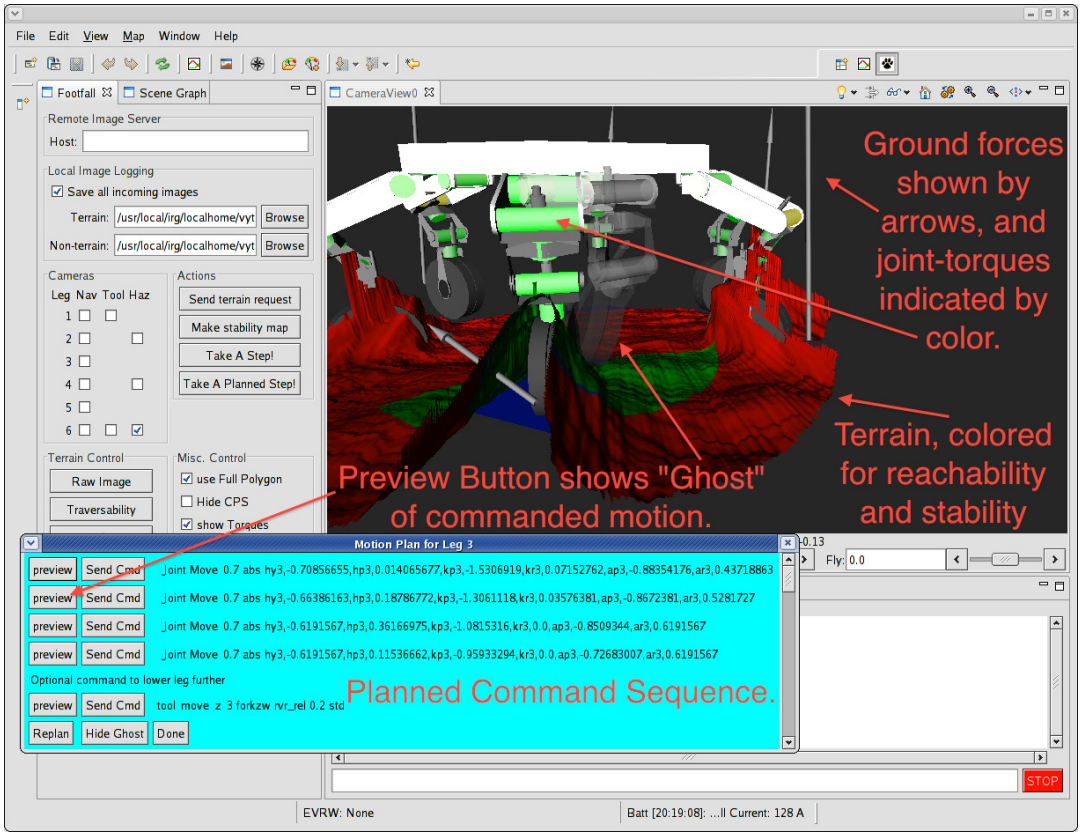


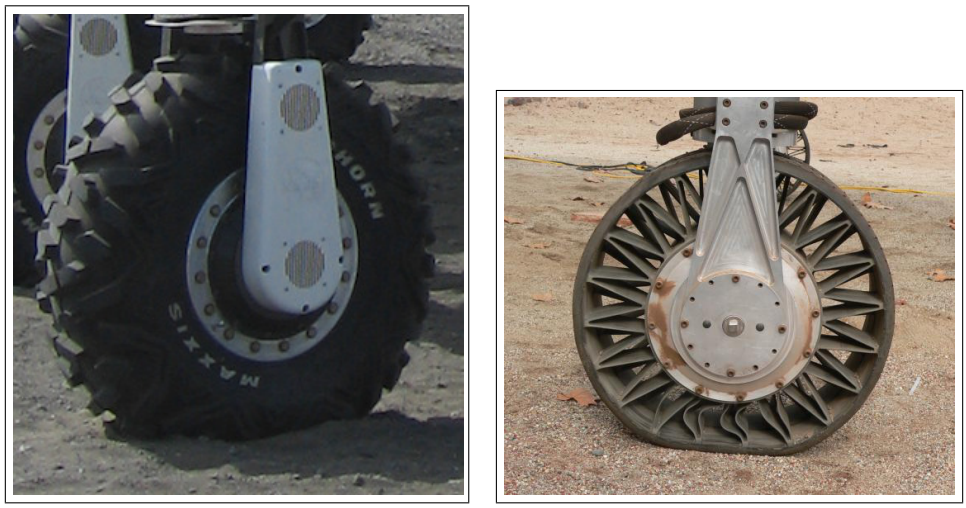

Figure 5: The FootFall UI, with a motion plan. The VERVE window shows the robot and terrain, ground reaction force arrows, joints colored to indicate torque levels, and a ghost of the next step in the motion plan. The terrain is colored to show areas that are unreachable or unstable (Red), and areas that are safe and reachable (Green).

RAPID (see Section 3.5). Since FootFall is designed as a plug-in to the AthleteWorkBench application, it allows access to all the other ground control capabilities normally used by operators. These controls include manual command entry, detailed telemetry monitoring, and emergency stop activation.

VERVE is a RAPID-enabled successor to Viz (Stoker et al., 1999) which was originally developed at NASA Ames for the 1997 Mars Pathfinder mission and successfully used by the Mars Exploration Rover (MER) mission science teams for a variety of geo-morphological measurements and virtual exploration of the area surrounding the rovers. Additionally, the 2009 Mars Phoenix mission science team used Viz as a decision-making aid for planning activities.

# 4 Moses Lake Field Test

In June of 2008, NASA's Human Robotic Systems Project, part of the agency's Exploration Technology Development Program, held an Integrated Field Test at Moses Lake, Washington. During this test, robotic systems from a variety of NASA centers were brought to the sand dunes outside of Moses Lake where they tested possible mission scenarios. SDM-B was involved in a number of these tests, including long distance traverses, docking, use of the new habitat mockup, navigating steep slopes, and testing the FootFall software. The environment in the sand dunes was harsh, and challenged many of the robots with rain, sand storms, mechanical failures, and schedule slips. Even with these challenging conditions we were able to successfully demonstrate that FootFall worked and was capable of generating valid motion plans for SDM-B to step over




(a) Pneumatic tire flattening (b) Mission-relevant tweel flattening

Figure 6: ATHLETE's wheels are compliant and flatten under load, especially as legs are lifted. The compliance is beneficial for driving, but complicates walking tasks.

a rock to an operator specified location. These tests were also successful from the perspective of highlighting which parts of FootFall were fragile and required improvement. We identified the most challenging aspects of operations as: the compliance in the robot, visibility, and self imaging.

### 4.1 Compliance

SDM-B is compliant, which results in a discrepancy between planned motions and actual executed motions. Most of this compliance comes from the joints and from the wheels, which flatten when under load. The flattening of the wheel is a direct result of its ability to absorb shocks and protect the system from damage when driving over rough terrain, and this compliance has been observed in both the pneumatic tires and the mission relevant Tweels (Figure 6). As a result, it is common to command the leg to lift by 20 cm and watch while the wheel does not break contact with the ground, but rather only unloads tension in the leg while the frame flexes and droops. This discrepancy is a problem because the motion planner may generate a plan which avoids collision with the environment, but at runtime the compliance in the robot results in a leg colliding with a rock because the leg never lifted itself high enough.

For the Moses Lake field test, we implemented a number of heuristics that compensated for the compliant behavior and allowed us to generate valid collision-free paths. For instance, we modified the generated DEM to lift the terrain up by 15 cm. This change simulates the e?ect of the robot sagging closer to the ground, and thus forces the planner to find a path that lifts the leg higher above the terrain.

While our heuristics worked and allowed the robot to safely step over the rock, they came at a cost. Each modification effectively reduced the workspace that the robot could plan through, making it more difficult to find a valid motion plan. Even when the operator could see that some locations should be possible to step to, the motion planner still could not nd a valid plan. This lack of workspace was compounded by the fact that the robot carried a new habitat for the field test. The habitat sat above the chassis while the generators were relocated below and severely limited the range of motion of the hip joint. These challenges led us to develop a compliance model for SDM-B after the first field test and to eventually adapt it for T12. We discuss our compliance model in Section 5.1.1, and provide quantitative metrics on its performance in Section 5.1.2.

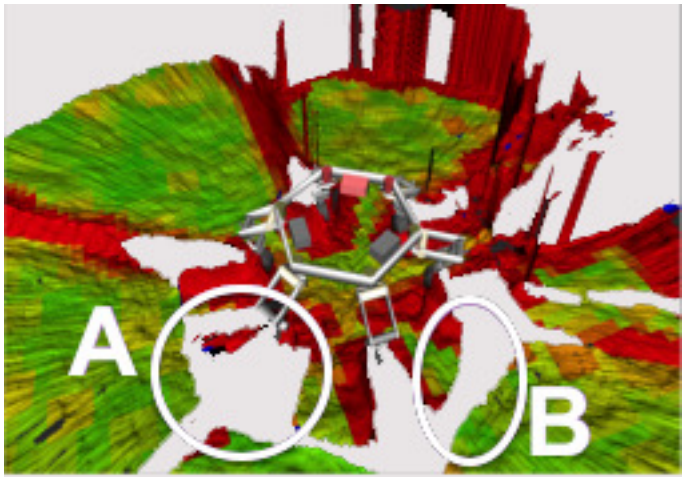



Figure 7: Limitations to Terrain Visibility on SDM-B. Circled regions show (A) Area not visible because of occlusion by the leg - which is also the very area that ATHLETE must step into for forward motion, and (B) Visibility gap between HazCams and NavCams, which make terrain integration difficult.

### 4.2 Visibility

While SDM-B has many stereo camera pairs, their placement is optimized for driving, not walking. The six NavCams in the outer ring are pointed out towards the horizon and on flat terrain they only see ground that is beyond the maximum reach of the legs. Thus, the three inward facing HazCams must be used. Figure 7 shows a terrain map using all of the NavCams and HazCams, and one can see the empty gap between the two where no data is gathered. Careful inspection will reveal that while the above-mentioned gap exists, the bulk of the missing data is caused by the occlusion of the terrain by the legs themselves. Because we must use the HazCams, which look through the body of the robot, the very leg we want to move blocks the view of much of the workspace of that leg. Operationally we are able to take steps that are tangential to the circle of the robot. During the Moses Lake field test we were able to accommodate this limitation, though it also further reduced our available workspace, limiting the range of possible rock-avoiding steps we could take.

While this approach was sufficient for testing a single step, a better solution was needed to make forward progress when taking multiple steps. The simplest solution is to lift a leg, image and reconstruct the terrain behind that leg, and then plan and execute the step. While this is a functional approach it adds an extra command cycle to walking operations which slows down execution, especially in the face of communication induced time delays. The ideal solution is to integrate images over time so that distant terrain seen by the NavCams during prior steps can be used for planning as the robot moves forward. The planned approach was to use bundle adjustment to integrate the various terrain images from the different cameras to create a single uniform terrain model incorporating data from multiple points in time. The Bundle Adjustment algorithm does best when there is a reasonable overlap between images. Given the poor overlap between camera views this was not feasible on SDM-B. Thus, we used these insights to influence the design and placement of cameras on T12 to ensure better overlap of image view planes to support future work on terrain integration. We discuss this view analysis work in Section 5.2.1. Since T12's camera systems, calibration, and telemetry systems did not become fully functional until immediately before the second field test, terrain visibility was handled at the D-RATS field test by lifting the leg and imaging the terrain.

### 4.3 Self Imaging

The two issues of compliance and visibility combine to form another challenge: the ability to account for portions of the robot structure that appear in the images it acquires. Since we are using the HazCams, which

look through the robot, they image the legs of the robot and reconstruct them as part of the terrain (Figure 8). Inclusion of the legs has many undesirable side effects, such as causing our collision detector to think the robot is in collision with the terrain when it is not. The first effort to deal with this was to use software from JPL that uses the forward kinematic solution to mask out chunks of the image based on where the legs are supposed to be. Unfortunately, with the compliance in the robot, the results were not usable, as the masks often missed large parts of the actual legs.

The approach used at the time of the Moses Lake field test was to flatten part of the image mesh used by the collision checker around the location of the foot. Flattening allows us to plan without being in a false-positive collision, but it has two limitations. First, there are often stereo-reconstruction artifacts left in the image beyond the flattened area which do not exist in reality, such as edge distortions which extend away from the camera along perspective lines. Furthermore, the large flattened area means that a rock immediately next to a wheel may be flattened away, and then ignored during the planning process. A better approach based on image segmentation, which was used at D-RATS 2010, is described in Section 5.2.

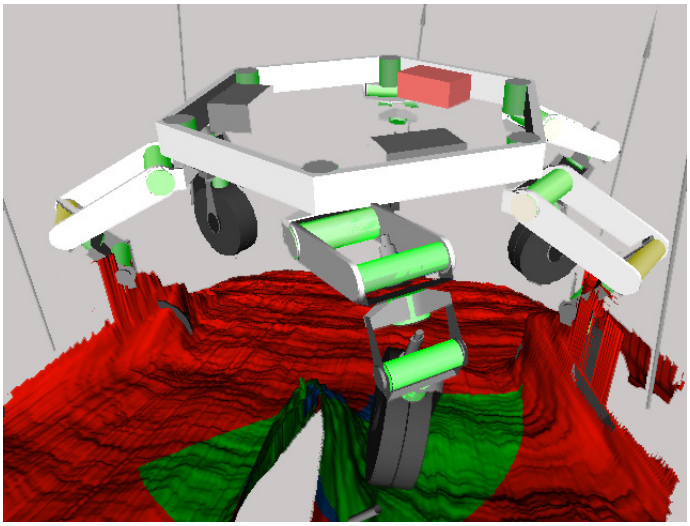


Figure 8: Self-imaging: Because the legs are visible in the camera image, stereo reconstruction will include the legs as part of the terrain. In this image one can see large "lumps" in the terrain which are caused by the legs. The closest leg has been moved to highlight the impact it had on the terrain model.

# 5 FootFall Development from 2008 to 2010

After the tests at Moses Lake we spent 2009 implementing and evaluating new features to address the lessons learned at the test. These primarily focused on: modeling the compliance and adding sag compensation to the motion planner, evaluating the effectiveness of the new methods, and developing an image segmentation approach to handle challenges of self-imaging the leg as part of the terrain. Furthermore, to move from single stepping to full cycle gaited walking we worked on gait production and optimization to avoid joint torque limits. These features were developed on SDM-B while plans were developed for the next generation T12 robot. A number of design choices for T12 were influenced by the experiences from Moses Lake. Finally, most of 2010 was spent in transition to T12 as the physical platform slowly became fully functional. This culminated with the D-RATS 2010 field test where FootFall was used to successfully command T12 for a cycle of steps around a number of large rocks.

### 5.1 Implementing Lessons Learned

#### 5.1.1 Sag Compensation

To ensure that executed motions match our planned motions, we extended our off-board motion planner to model the compliance in the robot. For SDM-B, the majority of sag was observed to come from the deformation of the wheels, and this led to the adoption of a model that assumes that all the compliance happens at the contact points. Although strictly speaking there are several sources of compliance distributed throughout the robot, this model has been found adequate for our purposes. The model would also be applicable for any other legged robot whose legs and body are very rigid compared to the contact points, e.g. wheel-on-leg robots or rigid robots walking on soft terrain.

The robot's reaction forces are commonly calculated by solving the system of six equations representing the sum of forces and moments without the presence of sag. This system is under-constrained, since there are only six equations for $3N$ unknown force components, when $N$ feet are in contact with the ground. Therefore, the solution is obtained using the pseudoinverse of the resulting linear system, which has been proven to yield the *zero interaction solution*. This means that if a line is drawn connecting any two feet in contact, the difference of the projections of their reaction forces along that line will be zero; i.e., the legs are not "fighting each other".

Two limitations exist with this technique. The first limitation is that the balance of forces and moments is done assuming perfectly rigid contacts, which constitutes an approximation when compliance exists, but the solution is often good enough. The second and more important limitation, for the purpose of this work, is that the pseudoinverse technique is unable to capture the transition that occurs while lifting or placing a foot. For a robot with flexible feet, such as tires, a foot's loading gradually decreases as it is lifted. This becomes important when we try to optimize the pose for the purpose of sag compensation.

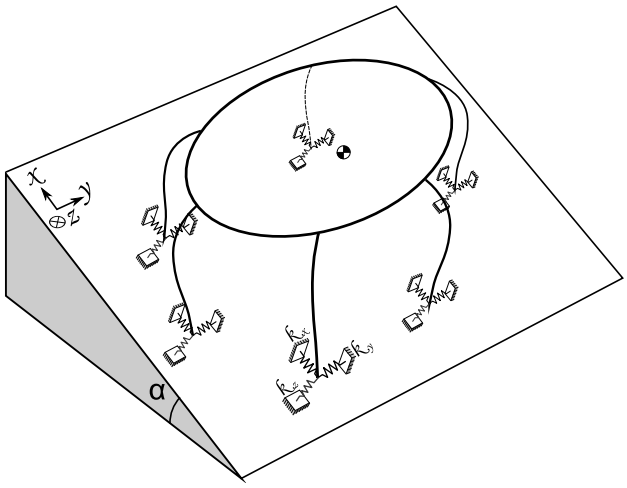



Figure 9: Spring-mass model for reaction force and sag calculations.

To address these two drawbacks, we model the contact points as an array of three springs oriented with the tangential-normal frame of reference as shown in Figure 9. We can express a system of equations as a function of the reaction forces. The resulting system, as will be seen, is nonlinear and therefore we will solve for the reaction forces numerically.

We begin by defining the initial (before sag) and final (after sag) locations of the feet and CG.

$$\textit{Initial:} \qquad \vec{r}_{0,i}, \text{ and } \vec{r}_{0,CG} \tag{1}$$

$$\textit{Final:} \quad \vec{r}_{f,i} = \vec{r}_{0,i} + \Delta\vec{r}_i, \text{ and } \vec{r}_{f,CG} = \vec{r}_{0,CG} + \Delta\vec{r}_{CG} \tag{2}$$

Where for a linear spring:

$$\Delta\vec{r}_i = \begin{bmatrix} \frac{1}{k_{xx}} & 0 & 0 \\ 0 & \frac{1}{k_{yy}} & 0 \\ 0 & 0 & \frac{1}{k_{zz}} \end{bmatrix} \cdot \vec{f}_i \tag{3}$$

where $k_{xx}$, $k_{yy}$, and $k_{zz}$ are the spring constants of the tires in the $x$, $y$, and $z$ directions respectively.

We start by writing the sum of forces and moments about the fixed frame of reference (initially collocated with the rover frame, but not moving). This gives us the first 6 equations:

$$\sum_{Contact} \left(\vec{f}_i\right) + m\vec{g} = \mathbf{0} \tag{4}$$

$$\sum_{Contact} \left(\vec{r}_{f,i} \times \vec{f}_i\right) + \vec{r}_{f,CG} \times m\vec{g} = \mathbf{0} \tag{5}$$

Given the assumption that the robot is rigid other than at the wheels, we next write equations that constrain the geometry before and after spring deformation. With $N$ feet in contact, we first have $N$ equations for the distance between successive feet, taking into account the feet in contact:

$$\forall_{i\in[1,6]} \left[\|\vec{r}_{0,i} - \vec{r}_{0,i+1}\| - \|\vec{r}_{f,i} - \vec{r}_{f,i+1}\| = 0\right] \tag{6}$$

The next set of equations requires that alternate loaded feet remain fixed with respect to each other.

$$\forall_{i\in[1,6]} \left[\|\vec{r}_{0,i} - \vec{r}_{0,i+2}\| - \|\vec{r}_{f,i} - \vec{r}_{f,i+2}\| = 0\right] \tag{7}$$

Up to this point, we have $3N$ equations with $3N$ unknown force components for the feet in contact. However, if we inspect equation (5) carefully, we can see that it depends also on the final position of the CG. Therefore we'll need to solve for this as well, and so strictly speaking need three more equations which define the CG location before and after spring deformation. In practice, it was found that over-constraining the system by adding more CG equations results in better numerical convergence. For this reason we write $N$ more equations for the feet in contact:

$$\forall_{i\in[1,6]} \left[\|\vec{r}_{0,CG} - \vec{r}_{0,i}\| - \|\vec{r}_{f,CG} - \vec{r}_{f,i}\| = 0\right] \tag{8}$$

Equations (4) through (8) are solved numerically using the Levenberg-Marquardt algorithm, with the optimization variables being the reaction force components and the final location of the CG. Note that the model outlined above can be used for any combination of feet in the air and on the ground.

Despite the generality of the above model, it still fails to capture the force redistribution that occurs when lifting or setting down a foot. In other words, it assumes that a foot is either fully loaded or bears no load. This assumption is an appropriate calculation for many situations, however, we are interested in pose optimization to redistribute the loads and mitigate the effects of sag. To calculate pose optimization we must extend our model to account for transitions between a leg being unloaded and fully loaded.

For this purpose, we define contact points for each foot and denote them by $\vec{r}_{C,i}$. More precisely, $\vec{r}_{C,i}$ represents the location of *fork i*, expressed in the inertial reference frame used thus far, at which the bottom of tire $i$ touches the ground and starts bearing load. On any arbitrary terrain, these contact points are $(x_i, y_i, z_{gnd@(x_i,y_i)} + R_{tire})$, and change for a given foot only when its $(x,y)$ coordinates change.

Now the force exerted by any given foot is redefined in terms of the contact points. Let the distance that fork $i$ has been lifted off the ground (in the unsagged situation) be:

$$\Delta h_i = \vec{r}_{C,i}(z) - \vec{r}_{0,i}(z) \tag{9}$$

Note that $\Delta h_i > 0$ if the leg has been lifted (+z is down). Assuming for now a flat plane, we see that the contact springs are affected differently – X and Y are able to apply their full forces as long as the tire is in contact. Z, however, has constantly diminishing action as the leg is lifted. This must be adequately portrayed in the corresponding equations. Thus for the linear spring model the forces will be given as follows:

$$f_i(x) = \begin{cases} -k_{xx} \cdot \Delta\vec{r}_i(x), & \text{if in contact} \\ 0, & otherwise \end{cases} \tag{10}$$

$$f_i(y) = \begin{cases} -k_{yy} \cdot \Delta\vec{r}_i(y), & \text{if in contact} \\ 0, & otherwise \end{cases} \tag{11}$$

$$f_i(z) = \begin{cases} -k_{zz} \cdot (\Delta\vec{r}_i(z) - \Delta h_i), & \text{if in contact} \\ 0, & otherwise \end{cases} \tag{12}$$

The geometry constraints guarantee that the fork displacements satisfy the rigidity assumption. As opposed to the previous case, the number of equations remains constant at 3(N+1) because contact or lack thereof is now detected automatically.

After solving for the reaction forces numerically, it is possible to backtrack the corresponding spring deformations, by means of the force-deformation relation for the spring (in this case a linear equation). Finally, from the spring deformations we find the change in position and orientation of the robot, which is our prediction of sag. Further details on our sag-compensation approach can be found in (Wheeler et al., 2010).

#### 5.1.2 Sag Compensation Performance Metrics

To test and calibrate our compliance model we measured the heights of various points of the robot with the robot in several configurations. These configurations included ones with all six leg on the ground, with one leg lifted, and with two legs lifted. We chose to measure the heights of the wheel axles (to verify the spring constant of the tires), and the corners of the hex ring, where the legs are attached, so we would know the total amount of sag in the robot. We used a hand-held laser distance meter placed at corresponding spots on the chassis and wheel axles for our measurements (Figure 10). Comparing the measurements with our predictions, we found our model fit the data very well (see Fig. 11). With slight adjustments to the spring constant provided by the tire manufacturer, we were able to get within five centimeters of the actual height of all the points, with an average error of 1.2 cm. The likely causes for the observed error are unmodeled compliance of the legs, and slight variations in inflation pressure and mechanical properties of the different

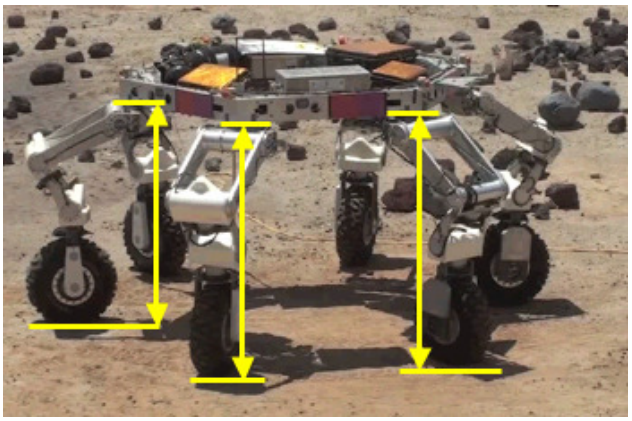
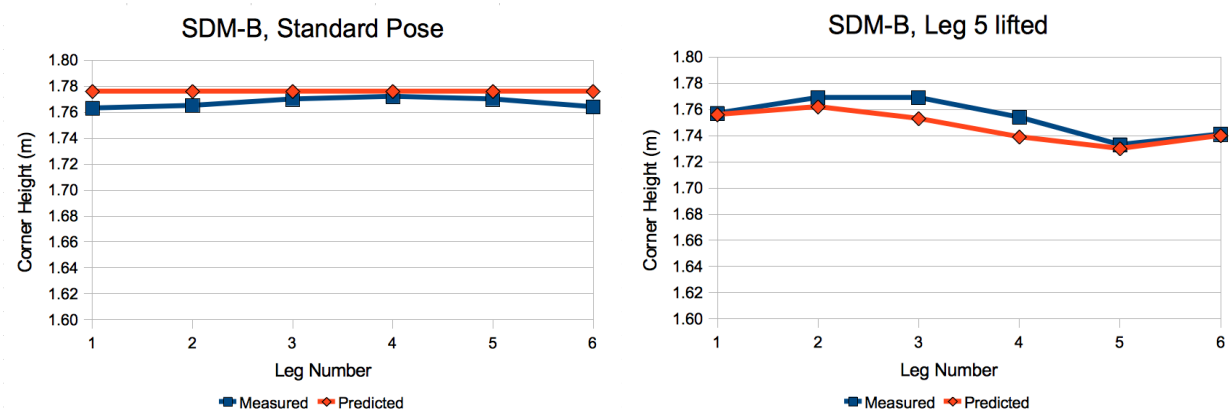


Figure 10: Measurement points on the chassis of SDM-B. Measurements were also taken from the wheel axles, and with the robot in several configurations.




(a) SDM-B Standard Pose

(b) SDM-B Leg 5 Lifted

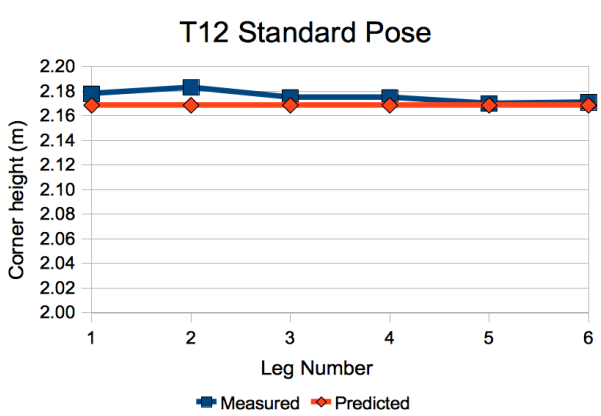
Figure 11: Predicted and Measured heights of the corners of the chassis of SDM-B when (A) all six feet are load bearing, and (B) When leg 5 is lifted. These graphs show the sag of the chassis when a leg is lifted, and the success of our model at predicting sag.

tires. The latter effect can be accommodated by the model if different spring constants are used for each tire in Equations (10)-(12).

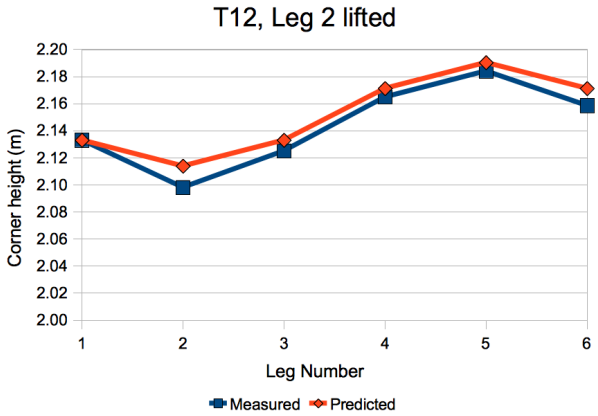
### 5.1.3 Sag Compensation for T12

Unlike the compliance of SDM-B, the compliance of T12 does not appear to be concentrated in the wheels. The deck of T12 is composed of three parts (two Tri-ATHLETE decks and a central pallet) that allow considerable flexing at the attachment points. In addition, the pallet is somewhat weak in torsion, resulting in a twisting compliance when a leg at the pallet-corner is lifted. Finally, the legs of T12, with seven joints rather than six, are more compliant than the legs of SDM-B.

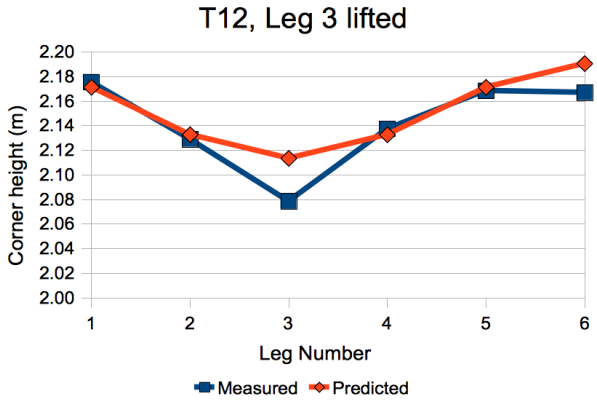
Despite the structural and kinematic differences between SDM-B and T12, our simplified compliance model is able to capture the resulting sag successfully on T12 as well as on the model for which it was originally designed. The only change required is that we treat as springs the legs themselves, rather than just the tires. Because the calculations deal only with the contact points of the feet and the location of the center of gravity, this change of perspective requires no changes to the equations in the model. The only change is




(a) T12 Standard Pose - all legs on ground




(b) T12 Leg 2 Lifted




(c) T12 Leg 3 Lifted

Figure 12: Predicted and Measured chassis corner heights of T12 under different stances. T12 sags more than SDM-B, but the compliance pattern is similar when (B) the lifted leg is one farther from the pallet. However, when T12 lifts a leg that is near a corner of the pallet (C), that corner sags dramatically.

that the height of the wheel axles is not used while determining the spring constant. In this case, we used only the heights of the corners of the chassis to determine the spring constant, which models the cumulative compliance in the entire kinematic chain. (Fig. 12(a)) The ideal spring constant was determined by evaluating model error of all spring constants weaker than the tire spring constant; the value which yielded the lowest average error for all measured chassis points was selected.

Using this method, we were able to find a spring constant that yielded an average error of 0.9 cm for the heights of the chassis corners. This spring constant was significantly lower than the spring constant of the tires alone, because it took into account the compliance of the 7 DOF leg as well. The two legs that were not near a corner of the flexible pallet produced a compliance pattern that fit our model well (Fig. 12(b)). Legs at the corner of the pallet caused greater sag, but the prediction was still within centimeters of the measured value (Fig. 12(c)).

### 5.1.4 Compliance Model Discussion

SDM-B and T12 differ in size, kinematic complexity, and sources of compliance. Despite those important differences, the FootFall compliance model was able to predict the sag of both robots to nearly the same degree of accuracy. In the pose studied, SDM-B has a chassis height of 1.75 m. The average error of 1.2 cm

is less than 0.7 percent of the length of the leg. T12, in its studied pose, has an average height of 2.15 m. The average error of 0.9 cm is 0.4 percent of the length of the leg. Motion accuracy of a few centimeters is sufficient for FootFall gait planning. FootFall depends on stereo vision for a map of surrounding terrain, and the stereo vision system produces maps with centimeter-scale error. Tests with the FootFall planning system have demonstrated that this compliance model is sufficient to plan steps, for both robots, that can be executed open-loop with ample ground clearance. Developing a compliance model that is orders of magnitude more precise than the map used for planning would be unnecessary.

The experiments with SDM-B and T12 also demonstrate the effect of uneven spring constants in prediction accuracy. In the first case, tire pressures were measured but not equalized and were found to vary by $\pm 1$ psi, equivalent to about 12.5 percent given the nominal operating pressure of 8 psi. The unequal pressures caused the predictions for SDM-B to be slightly less accurate than predictions for T12, even though the first robot is structurally more rigid. Because additional accuracy was not required for our application, equalizing the tire pressures was unnecessary.

These results are important for similar robots because they demonstrate that a simple model is capable of predicting sag in a legged robot without detailed modeling of the robot. To calibrate this model for a new robot, or for significant changes to an existing robot, all that is necessary is a few sets of measurements of the heights of the leg attachment points with the robot in characteristic walking poses (for instance, with one foot raised). Then the best leg spring constant can be solved for by exhaustive search as previously explained. This simple model is ideal not only because the on-line computation of specific sag numbers is reasonably quick, but also because there is no need to build a new model of the robot every time a component changes. Having an easily adaptable model allows for frequent upgrades of robot hardware without requiring the repetition of lengthy calculations. Another potential use of this compliance model is to account for soft deformable terrain and the pose changes resulting from foot sinkage while walking. This use would be appropriate even for a very rigid robot, with the core challenge being to appropriately classify the terrain such that the correct spring constants are used.

### 5.2 Image Segmentation

The Moses Lake Field Test exposed the need to separate the robot's legs from the surrounding environment in the visual scene to enable the motion planner to work without a false-positive collision. (Section 4.3) The ideal solution would be to use forward kinematics, the new sag model, and an image plane projection of a model of the leg to mask it from the image. However, given our experience with SDM-B we observed that even under the best of circumstances the forward projected image mask would never perfectly match the actual leg in the image, leaving a few clusters of pixels in the scene. These clusters would show up as obstacles in the 3D scene and complicate the motion plans. Thus we intended to combine the forward projection with an image segmentation technique which would use the mask as a seed and segment out all the pixels related to the robot's leg. Unfortunately the feature to project the leg model onto the image plane was not migrated to the new robot before the field test. Thus we ended up relying entirely on the image segmentation approach, which worked well when we applied it to the depth map instead of the pixel intensity map.

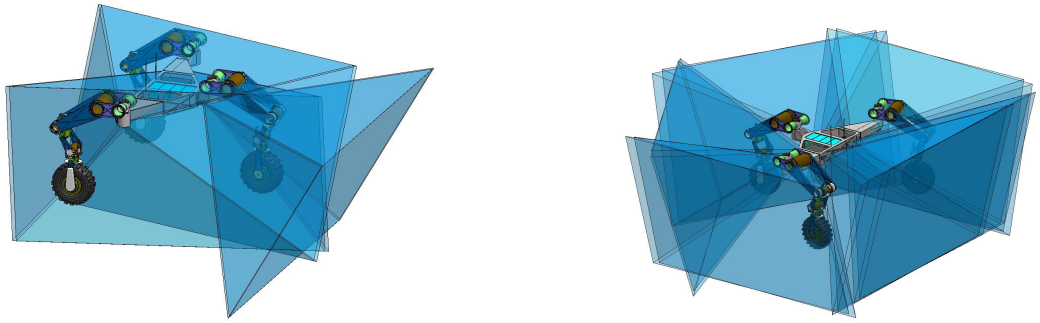
We used a simplified form of the watershed segmentation algorithm (Gonzalez and Woods, 2002), which takes the Laplace transform of a picture so that edges in the picture have the highest pixel values. In the transformed image, each local minima starts a segment; each segment adds progressively higher-valued neighbors until another segment is encountered.

Because we capture the stereo images mid-step, with the leg raised, we know that the leg is separate from the ground in the depth map. We also know that once the chassis is removed from the image, the leg is the closest object to the camera besides the ground. Using these two facts, we use the watershed algorithm on the depth map to find the segments closest to the camera. We assume that the segment that is the right size and not connected to the ground must be the leg.

Our approach worked well, failing only when the sharp contrast between sunshine and shadow prevented the stereo correlator from generating an accurate disparity map. We expect that combining our method with forward-projected image masking and image-based segmentation will give a more robust solution, and the inclusion of High Dynamic Range imaging techniques will enable the handling of a wider range of lighting conditions.

#### 5.2.1 View Analysis and Improved Camera Placement

As the second generation ATHLETE robot was designed, our experiences at the Moses Lake field test were used to improve the locations of the cameras. The goal of the planning process was to enable camera placements which would support both driving and walking tasks, which have different field of view requirements. Furthermore, the constraint was added that the fields of view of the cameras should overlap sufficiently to assist in terrain integration when using algorithms like Bundle Adjustment. Beyond these needs, there were a number of other strict placement requirements, such as not interfering with folding and packing the robot for launch. Figure 13 shows example models from the visibility analysis performed by JPL engineers.



(a) Haz cam (b) Nav cam

Figure 13: Visibility Analysis for Tri-ATHLETE to ensure that the fields of view from the different cameras provided situational awareness for both driving and walking, and that the images would overlap enough to support Bundle Adjustment algorithms for terrain integration.

### 5.3 Gait Production, Optimization, Weight Distribution, and Slopes

To go from a single step to full cycle walking we developed new capabilities for FootFall around gaited walking. This required Gait Planning to: choose the order of steps, locate optimal footfalls to maximize the forward progress, and ensure that all legs stayed within their maximum reach throughout the gait cycle. Further work was done on gait optimization to minimize torque saturation limits, which also highlighted the importance of weight distribution. The basic gait was also tested on slopes to see how well ATHLETE could climb hills while walking. A summary of these efforts is provided below, with full details available in (Chavez-Clemente, 2010). A full architecture for multi-resolution planning, including long distance route planning, gait planning, and configuration space step planning can be found in (Smith et al., 2008).

#### 5.3.1 Gait Planning

Many walking robots are bilaterally-symmetric and have a dominant direction of travel. It is often sufficient to design gaits off-line and then use the most appropriate one for the terrain being traversed. Since ATHLETE is essentially omnidirectional and we would like FootFall to be able to plan steps in any direction of motion

from any arbitrary initial configuration of the robot, we implemented a gait planner which can provide the initial and final locations for a sequence of steps.

For ATHLETE a **discontinuous** gait has been selected due to its ease of implementation. This gait is a common choice for walking robots because it requires less sophisticated coordination during stepping. In the discontinuous gait, the steps are executed with the body stationary, and then the body is shifted with all feet on the ground. A reverse wave sequence has been chosen where the feet are stepped in sequence from front to rear, starting with the left side (relative to the direction of travel chosen). The reverse wave sequence was selected because it maximizes the distance between consecutive feet being moved, thus minimizing the possibility for self-collisions. However, the optimization techniques developed for ATHLETE are not dependent on the use of this specific gait.

It is convenient to decompose the gait design process into two parts: The Skeleton Gait, and Step Planning. The Skeleton Gait consists of the sequence of robot configurations without the details on how each individual step is executed. Step Planning takes the initial and goal configurations from the Skeleton Gait and uses a motion planner to determine the sequence of waypoints that individual legs must follow to complete a step.

The above breakdown is advantageous because it is often desirable to optimize the gait to minimize some metric of interest (e.g. power, proximity to saturation). This optimization can be applied to the skeleton gait directly, preventing unnecessary computation. The total displacement of the body after a full gait cycle is a function of the step length, which is numerically calculated by successive inverse kinematics tests for a given start configuration of the robot and desired direction of motion. After the second gait cycle on flat, unobstructed ground, the motion converges to a regular, periodic pattern. At this point all the body shifts are of equal length, and so are the steps.

### 5.3.2 Gait Optimization

The load distributions during walking, coupled with design constraints, were observed to cause the motors on SDM-B to operate near their maximum torque capabilities and even reach saturation. Consequently, gait optimization algorithms that guarantee successful walking with maximum margin to saturation were developed.

A zero-interaction technique called the 'Sway-Gait' was formulated to increase the margin to saturation through optimal displacements of the robot's body in 3D space. The optimization produces a swaying motion of the body (Figure 14) while preserving the original footfall locations. Improvements of over 20% in the margin to saturation throughout the gait were achieved with this approach. The zero-interaction technique is the safest in the absence of precise knowledge of the contact mechanical properties and friction coefficients.

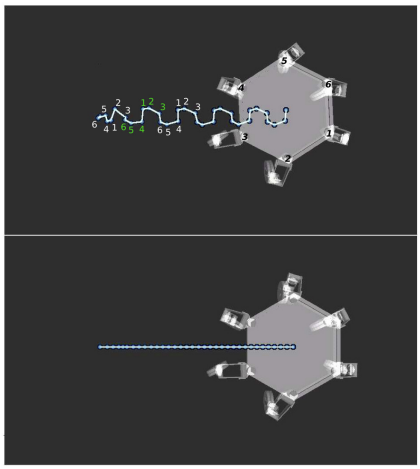


Figure 14: Top figure shows the robot-center path when executing the sway-gait compared to the reference gait shown below.

The zero-interaction technique was implemented and validated on SDM-B, with experiments conducted at the JPL Mars Yard in Pasadena, California (Figure 15). A second approach that uses the null space of contact forces was also developed and tested in simulation. The details of these gait optimization techniques are provided in (Chavez-Clemente, 2010).

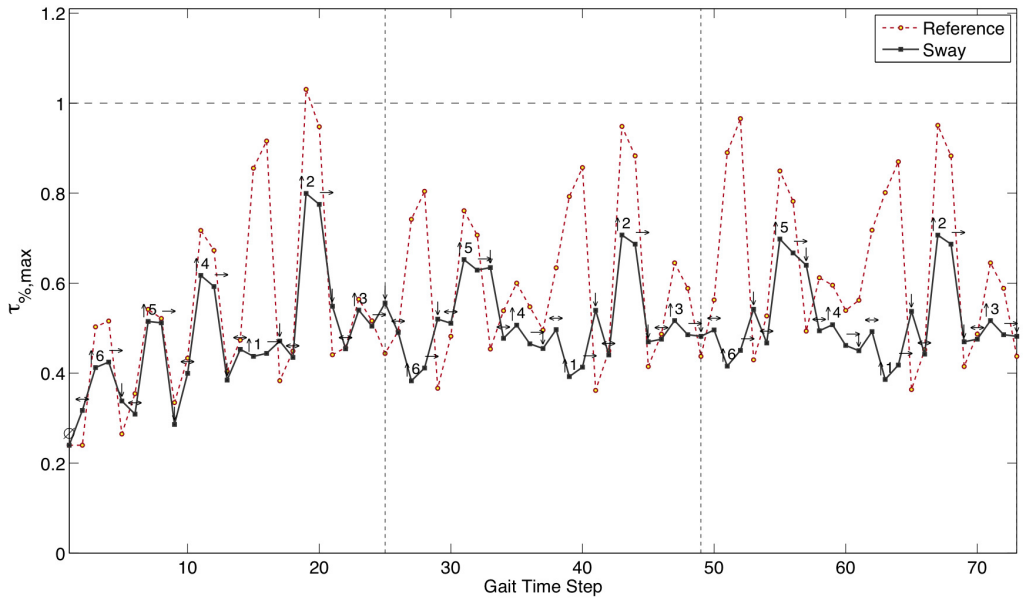
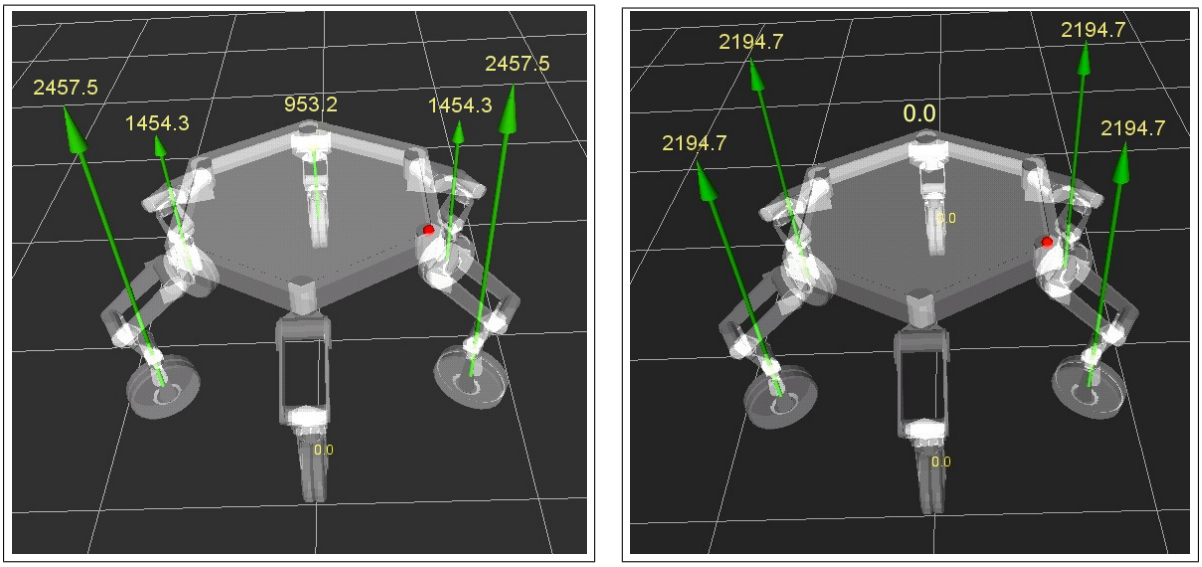



Figure 15: This graph shows peak torques as a percentage of maximum allowable torque for the reference and sway gaits. A value of 1 represents joint saturation. Lower peak torques are desirable so that the gait has torque margin to handle uneven terrain features. The sway gait provides more torque margin by generating lower peak torques compared to the reference gait.




(a) Forces with one leg lifted (front leg). Maximum torque is 50.28% of limit.

(b) Forces with two legs lifted (front and back leg). Maximum torque is 43.43% of limit.

Figure 16: Reaction Forces in Newtons when SDM-B lifts one and two legs. Note that while the average force is higher, the maximum force is lower when two legs are lifted. Minimizing the maximum torque helps the robot avoid torque saturation limits.

### 5.3.3 Weight Distribution For Walking

A surprising and interesting result of the torque analysis performed when optimizing the gaits was that often peak joint torques could be lower by lifting two diametrically opposed legs at once. The naive assumption

is that lifting a single leg would result in lower joint torques than lifting two legs, but this assumption does not take into account the effects of weight distribution. As Figure 16 shows, when only one leg is lifted the weight is unevenly distributed with higher peak loads on the feet neighboring the lifted leg. On the other hand, with two opposing legs lifted (Figure 16(b)), the weight is distributed evenly around the standing legs. Thus, while the average load per leg may be higher, the peak torques are lower. While this effect was studied mathematically (Chavez-Clemente, 2010), the two-legs-lifted hypothesis was not fully integrated into the FootFall Planner. Given the ongoing challenges of operating the robot at or near joint saturation limits, this technique would be worth pursuing further.

### 5.3.4 Walking on Slopes

To further explore the effects of weight distribution and torque limits, SDM-B was tested walking on slopes of 7° and 14°, as shown in Figure 17. The operational requirement to keep the chassis as level as possible became a limiting factor during these tests. Even at these relatively shallow inclinations, SDM-B was observed to be kinematically constrained with the workspace of the uphill legs severely limited while the legs on the downhill side reached maximum extension.

During the 14° slope experiment SDM-B was observed to slide downhill a short distance ($< 5$cm) on occasion. For the safety of the robot we did not attempt to climb on steeper slopes and concluded that this was near the maximum inclination that SDM-B could walk up for soils of this type (hard-packed base with a loose sandy top layer). While this may be a limit for walking up a continuous slope, it should be noted that SDM-B can drive on steeper slopes because more legs are in contact with the ground providing more friction and avoiding torque limits. Likewise, SDM-B can climb over more significant obstacles, like rock ledges, where there is less chance of sliding downhill. SDM-B could also ascend steeper slopes by keeping its body parallel to the local ground, but it may not always be possible depending on the payload. Given the significant torque and workspace improvements of T12, the current generation robot should be capable of ascending steeper slopes than SDM-B. Repeating these experiments on the new platform would be worthwhile.

(a) 7° slope (b) 14° slope

Figure 17: SDM-B walking on slopes of 7° and 14°. The latter starts to place severe kinematic constraints on the motion of the robot.

## 6 DRATS-2010 Black Point Lava Flow Field Test

The Desert Research and Technology Studies (D-RATS) 2010 field test was held at Black Point Lava Flow in Arizona. D-RATS is an annual field test led by NASA in collaboration with non-NASA research partners. The goal of D-RATS is to do earth-based experiments of future mission concepts and technologies. In 2010, the primary activity was sending the Space Exploration Vehicle (SEV) rovers (with a two person astronaut team on board each) and the ATHLETE rover on a 14-day traverse over the lava fields. After the robots returned to base camp a number of other tests and demonstrations were performed, including the final

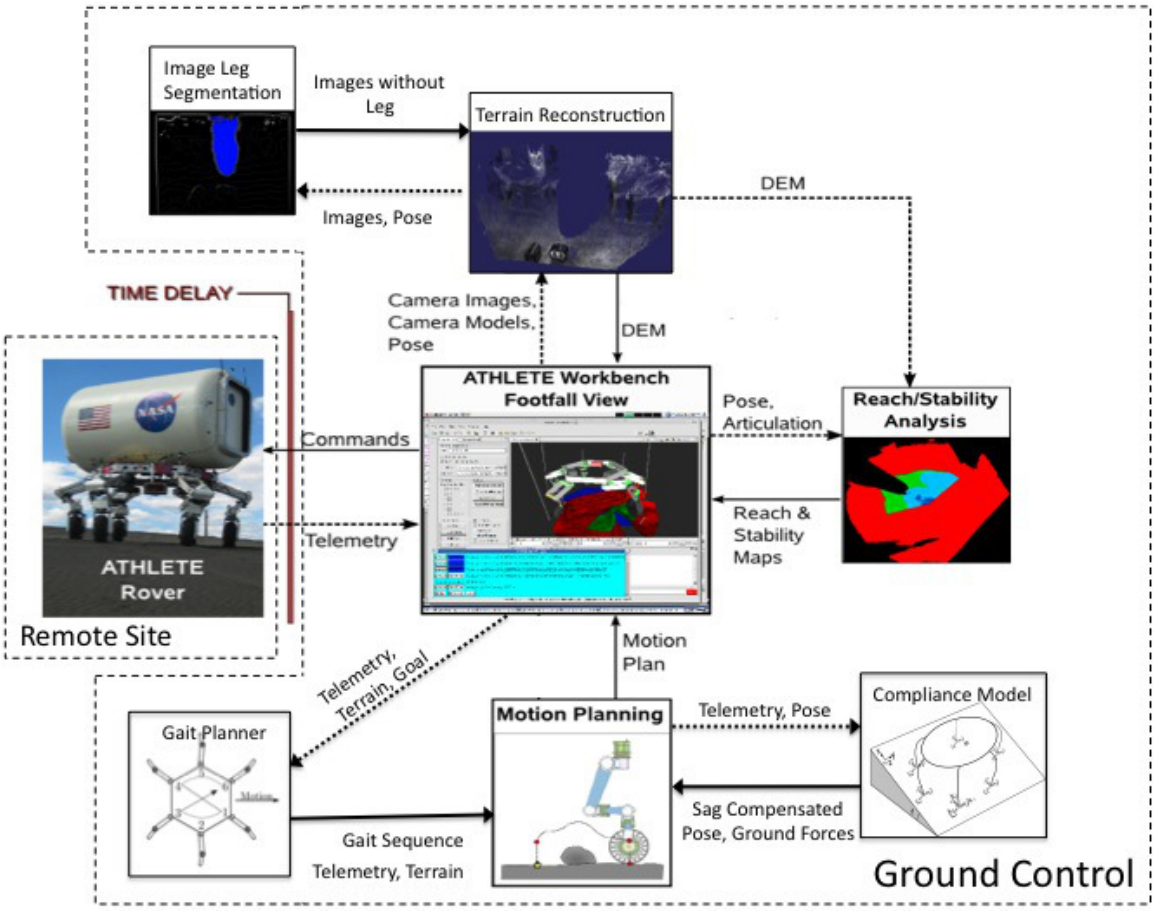



Figure 18: FootFall Planning System software architecture at D-RATS 2010 Field Test. Compare with Figure 3 which shows the architecture from the 2008 Moses Lake Field Test.

experiments for the FootFall Planner.

The objective of the FootFall tests at D-RATS was to perform a full walking gait cycle in the presence of rocks, which must be avoided. This objective required that the sag compensation models work accurately along with image segmentation to remove the robot's legs from the environment model. Figure 18 shows the software architecture for FootFall at the time of the field test, with all the components developed over the previous years integrated. This can be compared to Figure 3 which showed the architecture at the time of the first Field Test at Moses Lake in 2008.

As can be seen from Figure 19, the test was successful and FootFall was able to command ATHLETE through a gait cycle. During the gait cycle, the chassis of the robot progressed forward approximately 1/3 of a body length. This test validated the performance and characterization of the compliance model (discussed in Section 5.1.1) which had been developed and tested on SDM-B in the JPL Mars Yard. D-RATS was the first end-to-end test of FootFall on the new T12 robot and it is noteworthy that the compliance model was easy to re-tune as hardware changes occurred. It is also significant that the torque values reported on T12 appear to be more reliable than on SDM-B and did not overtly loose calibration during our tests. This opens the possibility of developing on-board closed-loop torque controls for walking tasks.

The primary challenges encountered during the field test were related to image processing, and were somewhat expected. The image segmentation, as described in Section 5.2, worked well in most cases, though it ran into

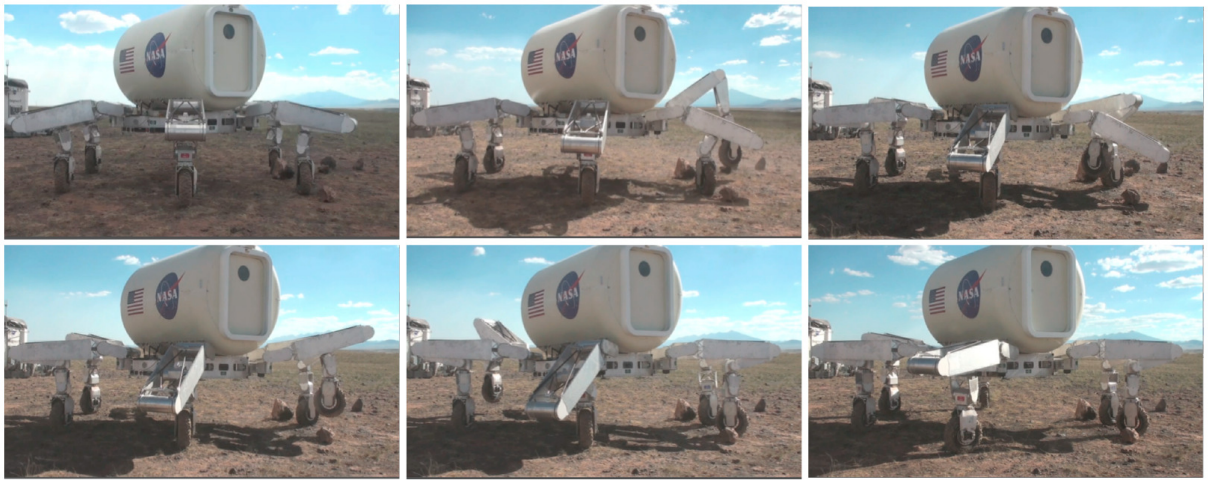


Figure 19: Stepping around rocks at D-RATS 2010

some trouble with strong shadowing and saturation conditions. It is anticipated that further development of the image processing pipeline, such as inclusion of High Dynamic Range imaging techniques and integration of forward kinematic masking, will make this process more robust. As with all robotic technologies, the FootFall Planner can be further improved in a number of ways, and has provided many useful observations and lessons learned during the effort of developing FootFall to its current state.

# 7 Conclusions

Stepping back from the many implementation details, this section will look at the larger lessons learned from the four-year effort, and make recommendations for future research. The most important lesson learned is that for robots of the scale and complexity of ATHLETE, one must pay careful attention to forces and joint torques, even though the robot is controlled in a statically stable manner. This focus has implications ranging from the physical design of the robot, to motion planning, and the types of on-board commands that are available. The secondary lessons of this work focus on visibility and situational awareness.

## 7.1 Forces and Torques

Traditionally, most space robots have focused on position control as the primary means of ensuring safe operation. Walking entails complex contact dynamics and load-bearing forces with a magnitude of the mass of the vehicle itself. This centrality of forces is being actively explored by dynamically controlled walking robots, but is not as common in work on statically stable robots. Our initial approach to statically stable walking follows work by Wettergreen (Wettergreen, 1996) and others that focused on calculating the systems center of gravity and polygon of support to ensure that tip-over events were avoided. Once we were successfully planning collision-free paths through the robot's kinematic space, we found that compliance in the robot caused execution of those paths to run into environmental collisions. As described in section 5.1.1 above, the structural compliance was handled by modeling compliant contact points and forces and the resulting change in the pose of the central hex ring. This analysis enabled the executed motions to match the planned ones. This model may also be appropriate for rigid robots walking on soft compliant soils. With compliance modeled we continued to encounter force related challenges as the robot regularly encountered joint-torque limits during execution.

Problems of joint torque saturation were particularly common on SDM-B, which was over-weight for its motors. Among various issues, the motors would ratchet (i.e. skip teeth) and the joint-torque measurement

system would lose calibration. As a result, the reported joint torques could not be relied upon to be accurate and closed-loop torque-based control was not realistic without regular recalibration. Thus, FootFall focused on position-based planning and controls. While this approach worked well on fairly benign terrain, it quickly became apparent that even mild unevenness in the terrain could cause significant buildup of torques in the joints, causing saturation and an interruption of execution.

Joint torque saturation led to research into torque-based gait optimization to plan for shifting the robot's body to minimize the peak joint torque, as described in section 5.3.2. Looking at weight distribution and peak torques led to some surprising insights, such as the realization that lifting two opposing legs at once would lead to a lower peak torque than lifting a single leg, and thus more robust operation. This level of torque analysis should be taken one step further and incorporated into the single step motion-planning algorithm directly. At issue is that the kinematically reachable workspace of the robot is much larger than the load-bearing workspace of ATHLETE. Thus, footfalls would be identified which were reachable and safely kept the CG within the polygon of support, but which violated joint-torque limits. Further, it would be worth exploring torque-based optimizations for the path-smoothing step of the motion planner. Currently, once a feasible collision-free path is found to the desired footfall, the path is smoothed to minimize the Euclidian distance of the path. While the resulting motions are efficient, it may be preferable to optimize for a minimal peak joint-torque.

Once a torque-based motion plan is specified, it must be executed. The existing approach of using position-based commands is sub-optimal because small discrepancies between the reconstructed and actual terrain can cause very large torques to propagate though the system. The sensitivity to terrain errors was analyzed in (Chavez-Clemente 2010). The use of pure position based motion commands was motivated by the unreliable nature of torque telemetry on the original SDM-B robot due to various factors, including joint-ratcheting. The redesign of ATHLETE into the new T12 robot improved the joint motor designs and enabled operations that rely on the proper calibration of the joint-torque telemetry. As a result it is now reasonable to implement torque-based controls on the robot, such as impedance control or motion commands which terminate upon achieving a certain contact force. Torque-based commands will allow torque-space motion planning to be tied to execution such that equal weight distribution of the robot can be maintained and peak torques can be avoided, even in the face of uncertainty in terrain reconstruction.

As mentioned above, the load-bearing workspace of the robot is much smaller than its reachable workspace. This observation leads to a few thoughts on the physical design of the robot. Having motors directly at the joints makes sense from an implementation perspective, but leads to significant walking challenges. To make the legs light enough to be lifted by the shoulder/waist motors, a common approach (as done on ATHLETE) is to place smaller motors at the distal joints (i.e. the ankle motor is the smallest and lightest). This approach is inherited from manipulator designs which consider the arm in isolation and for which the longest moment arm (and thus the largest torques) are experienced at the shoulder. But in standing and walking tasks, the pose of the entire robot must be considered and the ankle joints can experience the greatest torques. This occurs in certain poses where the ankle experiences long moment arms originating at the ground contact point of other feet while also supporting much of the robot's mass. As a result, the ankle motors can be easily overloaded. This has motivated operational constraints placed on ATHLETE such that the last leg segment is kept near vertical when load bearing. This constraint effectively removes a degree of kinematic freedom from the leg for walking tasks.

The requirements-conflict for the distal joint to be both light and strong is leading some manipulation and walking robots to draw inspiration from biological systems by placing actuators further up the limb and using cables to drive the distal joints (Lovchik and Diftler, 1999), (Nichol et al., 2004), (Tsusaka and Ota, 2006), (Bowling, 2007). This change moves the center of gravity of the arm closer to its base joint (waist or shoulder) and allows more powerful motors to be used for the distal joints. While there are known engineering challenges with such a cable-driven design, they can be overcome and such a route should be seriously considered for high DOF walking robots given the fundamental challenge of joint-torque saturation of distal joints during load-bearing walking tasks.

### 7.2 Situational Awareness

Remote operations of a robot are very sensitive to the quality of situational awareness that is provided. Acquiring usable information about the terrain around ATHLETE was a central challenge throughout the project. While some of the difficulties were with implementation details, such as camera calibrations and camera models, which were mitigated through ongoing engineering effort, there were also structural lessons learned. The key question is: "Can the robot build a terrain map of the areas where it wishes to step next?" There are two approaches to dealing with this: camera placement for direct visibility, and camera placement to facilitate terrain integration over time.

Despite improvements during the redesign, camera placement on ATHLETE is well suited for driving tasks, but was never fully optimized for walking tasks due to competing design requirements. This is largely due to the design of the robot which has its cameras on the main hex ring, with the legs extending out beyond the ring. Fundamentally this arrangement causes the legs to block the view of the terrain where the robot is going to step next. Looking at quadruped mammals the design choice is to place the 3D sensors (eyes and ears) out in front of the body such that the terrain to be walked on is directly visible. This arrangement was not an option for ATHLETE due to launch volume considerations. Instead, ATHLETE had to rely upon a slower method of lifting a leg, imaging the terrain beyond the leg, and then planning and executing the leg motion to the desired location.

Given the limitations to direct visibility, the alternative is to build an integrated terrain over time. An integrated map would allow the robot to plan the next step given terrain that was visible during the previous step and/or from multiple camera perspectives. This approach is intuitively compelling, though it does compound a number of sources of error in the final terrain model. To support such terrain integration it is important to look at the overlap of the fields of view of various camera systems on the robot. Techniques such as bundle adjustment work best with significant overlap between images to properly calibrate them to each other for integration. This overlap requirement applies to images over time, and also images of the terrain from various cameras on the robot. As the terrain patch of interest may be visible to different cameras at different points in time, the cameras on the robot should all maintain a reasonable overlap with each other to facilitate terrain integration.

Finally, the challenges we faced dealing with self-imaging were particularly philosophically interesting. Biological systems clearly have an ability to segment themselves from the surrounding environment in every sensory modality used (vision, sound, force, etc). It is fascinating that we are now dealing with the same issues as we push the boundaries of robotic operation in natural environments. Ultimately, we suspect that this issue will be solved by ever richer multi-modal models of the environment which are integrated over time, and which enable robots to more clearly define "self" versus "not-self." This possibility lets the imagination run wild with ideas of intelligence and self-awareness deriving from the basic computational requirements of planning collision free motions in natural unstructured environments.

## Acknowledgments

This work was made possible by the support and commitment of Brian Wilcox and the entire ATHLETE team. In particular, the authors would like to thank Jay Torres, Curtis Collins, Julie Townsend, Jaret Mathews, Chet Joswig, Michael McHenry, Matt Heverly and everyone else from JPL who pitched in and made FootFall a success. Likewise, many members of the Intelligent Robotics Group and the Planning And Scheduling Group at NASA Ames contributed to this effort and we would like to thank Terry Fong, Leslie Keely, Michael Broxton, Mark Allan, Tristan Smith, Javier Barreiro, David Smith, and our two interns, Patrick Mihelich and James Snow who helped develop many critical components of the FootFall system. Finally, from Stanford we would like to thank Steve Rock and Kris Hauser.

This research was carried out as part of the Human-Robotic Systems (HRS) project, which is funded

by NASA's Exploration Technology Development Program under NASA Ames Prime Contract No. NNA08CG83C. Part of this work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Further support was provided by the Aerospace Robotics Lab at Stanford and the NASA Ames Education Associates Program.

## References

Bowling, A. P. (2007). Mass distribution effects on dynamic performance of a cable-driven hexapod. *Journal of Mechanical Design*, 129(8):887–890.

Campbell, D. and Buehler, M. (2003). Preliminary bounding experiments in a dynamic hexapod. *Experimental Robotics VIII*, pages 612–621.

Chavez-Clemente, D. (2010). *Gait Optimization for Multi-Legged Walking Robots, with Application to a Lunar Hexapod*. PhD dissertation, Stanford University, Aerospace Robotics Lab, Stanford, CA.

Collins, C. L. (2007). Stiffness modeling and force distribution for the All-Terrain Hex-Limbed Extra-Terrestrial explorer (ATHLETE). ASME.

Cox, W. (1970). Big muskie. *The Ohio State Engineer*, pages 25–52.

Dederick, Z. and Grass, I. (1868). Improvement in Steam-Carriage, united states patent 75874.

Fong, T., Bualat, M., Deans, M. C., Adams, B., Allan, M., Altobelli, M., Bouyssounouse, X., Cohen, T., Flckiger, L., Garber, J., Palmer, E., Heggy, E., Helper, M., Hodges, K. V., Lee, P., Lee, S. Y., Lees, D., Lum, J., Lundy, M., Smith, T., To, V., Utz, H., Wheeler, D., and Young, K. (2010). Robotic follow-up for human exploration. In *Proceedings of AIAA SPACE 2010 Conference and Exposition*.

Gonzalez, R. C. and Woods, R. E. (2002). *Digital Image Processing, Second Edition*.

Gottschalk, S., Lin, M., Manocha, D., and Larsen, E. (1999). *PQP: the proximity query package*.

Gottschalk, S., Lin, M. C., and Manocha, D. (1996). OBBTree: a hierarchical structure for rapid interference detection. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 171–180. ACM.

Hauser, K., Bretl, T., Latombe, J. C., and Wilcox, B. (2006). Motion planning for a six-legged lunar robot. New York, NY.

Heverly, M. and Matthews, J. (2008). A wheel-on-limb rover for lunar operations. In *Proceedings of the Ninth International Symposium on Artificial Intelligence and Robotics, and Automation in Space (iSAIRAS)*, Hollywood, CA. iSAIRAS.

Heverly, M., Matthews, J., Frost, M., and McQuin, C. (2010). Development of the Tri-ATHLETE lunar vehicle prototype. In *Proceedings of the 40th Aerospace Mechanisms Symposium*.

Hirose, S. and Arikawa, K. (1999). Development of quadruped walking robot TITAN-VIII for commercially available research platform. *Journal of the Robotics Society of Japan*, 17(8):1191–1197.

Hirose, S., Masui, T., Kikuchi, H., Fukuda, Y., and Umetani, Y. (1985). Titan III: a quadruped walking vehicle. In *Robotics Research-The Second International Symposium*, pages 325–331.

Hirose, S. and Takeuchi, H. (1995). Roller-walker: A proposal of new leg-wheel hybrid mobile robot. In *International Conference on Advanced Robotics*, pages 917–922.

Hirose, S., Tsukagoshi, H., and Yoneda, K. (2001). Normalized energy stability margin and its contour of walking vehicles on rough terrain. In *IEEE International Conference on Robotics and Automation*.

Hirose, S., Yoneda, K., Arai, K., and Ibe, T. (1991). Design of prismatic quadruped walking vehicle TITAN VI. In *Fifth International Conference on Advanced Robotics, 1991.'Robots in Unstructured Environments', 91 ICAR.*, pages 723–728.

Kar, D. C. (2003). Design of statically stable walking robot: A review. *Journal of Robotic Systems*, 20(11):671–686.

Kato, K. and Hirose, S. (2001). Development of the quadruped walking robot, TITAN-IX mechanical design concept and application for the humanitarian de-mining robot. *Advanced Robotics*, 15(2):191–204.

Kimura, H., Fukuoka, Y., and Cohen, A. H. (2007). Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts. *The International Journal of Robotics Research*, 26(5):475.

Lovchik, C. and Diftler, M. (1999). The robonaut hand: a dexterous robot hand for space. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 907 –912 vol.2.

McGhee, R. B. (1967). Finite state control of quadruped locomotion. *Simulation*, 9(3):135.

McGhee, R. B. and Ishwandi, G. I. (1979). Adaptive locomotion of a multilegged robot over rough terrain. *IEEE Transactions on Systems, Man, and Cybernetics*, 9:176–182.

Moratto, Z. M., Broxton, M. J., Beyer, R. A., Lundy, M., and Husmann, K. (March, 2010). Ames Stereo Pipeline, NASA's open source automated stereogrammetry software. In *41st Lunar and Planetary Science Conference*, The Woodlands, Texas.

Morrison, R. A. (1968). Iron mule train. In *Mobility testing: proceedings of the off-road Mobility Research Symposium*, page 381. Cornell Aeronautical Laboratories.

Mosher, R. S. (1968). Test and evaluation of a versatile walking truck. In *Mobility testing: proceedings of the off-road Mobility Research Symposium*, page 359. Cornell Aeronautical Laboratories.

Nichol, J. G., Singh, S. P., Waldron, K. J., Palmer, L. R., and Orin, D. E. (2004). System design of a quadrupedal galloping machine. *The International Journal of Robotics Research*, 23(10-11):1013 –1027.

Okhotsimski, D. E. (1980). Motion control system development for a mobile robot. *Automatic control in space*, pages 251–256.

Okhotsimski, D. E., Gurfinkel, V. S., Devyanin, E. A., and Platonov, A. K. (1979). Integrated walking robot. *Machine intelligence*, page 313.

Petternella, M., Salinari, S., delle Ricerche, N., and Eudossiana, V. (1974). Feasibility study on six-legged walking robots. In *Proceedings*, page 33. IIT Research Institute.

Playter, R., Buehler, M., and Raibert, M. (2006). BigDog. In *Proceedings of SPIE*, volume 6230, page 62302O.

Poulakakis, I., Smith, J. A., and Buehler, M. (2005). Modeling and experiments of untethered quadrupedal running with a bounding gait: The scout II robot. *The International Journal of Robotics Research*, 24(4):239.

Sanchez, G. and Latombe, J. C. (2003). A single-query bi-directional probabilistic roadmap planner with lazy collision checking. *Robotics Research*, pages 403–417.

Schwarzer, F., Saha, M., and Latombe, J. C. (2005). Adaptive dynamic collision checking for single and multiple articulated robots in complex environments. *Robotics, IEEE Transactions on*, 21(3):338–353.

Smith, T., Barreiro, J., Smith, D., SunSpiral, V., and Chavez, D. (Sept. 2008). Athlete's feet: Multi-resolution planning for a hexapod robot. In *International Conference on Automated Planning and Scheduling (ICAPS)*, Sydney, Australia.

Stoker, C. R., Zbinden, E., Blackmon, T. T., Kanefsky, B., Hagen, J., Neveu, C., Rasmussen, D., Schwehr, K., and Sims, M. (1999). Analyzing pathfinder data using virtual reality and superresolved imaging. *Journal of Geophysical Research*, 104(E4):8889–8906.

SunSpiral, V., Chavez-Clemente, D., Broxton, M., Keely, L., Mihelich, P., Mittman, D., and Collins, C. (2008). FootFall: a ground based operations toolset enabling walking for the ATHLETE rover. In *Proceedings of AIAA Space Conference*, San Diego, CA. AIAA.

Torres, R. J., Allan, M., Hirsh, R., and Wallick, M. N. (2009). RAPID: collaboration results from three NASA centers in commanding/monitoring lunar assets. In *Proceedings of IEEE Aerospace Conference*.

Townsend, J., Biesiadecki, J., and Collins, C. (2010). Athlete mobility performance with active terrain compliance. In *Aerospace Conference, 2010 IEEE*, pages 1 –7.

Tsusaka, Y. and Ota, Y. (2006). Wire-driven bipedal robot. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3958 –3963.

Vision Workbench (2010). NASA vision workbench. http://ti.arc.nasa.gov/tech/asr/intelligent-robotics/nasa-vision-workbench/.

Waldron, K. and McGhee, R. (1986). The adaptive suspension vehicle. *IEEE Control Systems Magazine*, 6(6):7–12.

Wettergreen, D. S. (1996). *Robotic walking in natural terrain: gait planning and behavior-based control for statically-stable walking robots*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA.

Wheeler, D., Daniel Chavez-Clemente, and Vytas SunSpiral (2010). FootSpring: A compliance model for the ATHLETE family of robots. In *Proceedings of 10th International Symposium on Artificial Intelligence, Robotics, and Automation in Space*.

Wilcox, B. H. (2009). ATHLETE: a cargo and habitat transporter for the moon. In *Proceedings of IEEE Aerospace Conference*.

Wilcox, B. H. (2011). Athlete: A cargo-handling vehicle for solar system exploration. In *Aerospace Conference, 2011 IEEE*, pages 1 –8.

Wilcox, B. H., Litwin, T., Biesiadecki, J., Matthews, J., Heverly, M., Morrison, J., Townsend, J., Ahmad, N., Sirota, A., and Cooper, B. (2007). ATHLETE: A cargo handling and manipulation robot for the moon. *Journal of Field Robotics*, 24(5):421–434.