A Controller for the LittleDog Quadruped Walking on Rough Terrain

John R. Rebula, Peter D. Neuhaus, Brian V. Bonnlander, Matthew J. Johnson, Jerry E. Pratt

Florida Institute for Human and Machine Cognition, Pensacola, Florida 32502, USA

http://www.ihmc.us/research/projects/LearningLocomotion/ *(jrebula@alum.mit.edu)*

Abstract— We present a controller for a quadrupedal robot statically walking on known rough terrain. The controller has both deliberative and reactive components for task specific control issues, such as impassable terrain and unmodeled foot slippage. The controller architecture supports multiple gaits, and we present both a stable omnidirectional gait and a faster directional gait. The robot successfully negotiates obstacles up to 7.5 cm ($\approx$40% leg length) tall and navigates over rocky terrain.

I. INTRODUCTION

This paper describes the design and implementation details of a controller for the Boston Dynamics LittleDog quadruped robot walking over rough terrain. The LittleDog (Fig. 1) has 12 actuated degrees of freedom with high gain servo motors powering each joint. In order to focus on the controller and not be distracted by the complex problem of accurately sensing the robot and environment, we employ a Vicon motion capture system comprised of six cameras fixed around the experimental area and a computer to process images. The LittleDog has 22 reflective balls attached to the body and legs. The motion capture system detects these markers and determines the robot pose and joint angles from known models. In addition, the LittleDog also has an inertial measurement unit which provides measurements of linear acceleration and angular velocities which are accurate over short time periods. This data is then fused with the motion capture data, which accounts for drift and integration errors. Predefined terrain boards can also be placed into the area with motion capture markers to identify the location. One of the terrain boards is shown in Fig. 2. Each terrain board has a corresponding high resolution model, which when combined
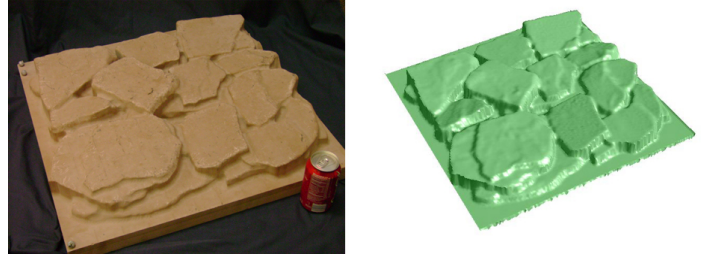

Fig. 2. An example terrain board (left) with a rendering of its corresponding model

with the motion capture data, provides an accurate model of the robot and its environment updated at 100Hz. The task is to successfully navigate across the rough terrain to reach an arbitrarily specified goal.

A. Control Issues

- **Impassable Terrain:** A robot traversing over rough terrain may encounter an obstacle that is kinematically impossible for the robot to traverse. This situation creates a search problem over possible paths over the terrain that requires a deliberative path planner to exploit terrain knowledge.
- **Foot Slippage:** One implication of rough terrain is the magnification of error in foot contact models. Small foot slips, even on locally flat terrain, can lead to a foot falling off of an edge of the terrain, resulting in unexpected stances and a possible loss of stability.
- **Accidental Collision:** Rough terrain can obstruct a leg's path while swinging from one location to another. In addition, even with fixed foot positions, shifting the body center of mass can result in a leg segment contacting the terrain accidentally. Such collisions must be sensed and accounted for in a controller.
- **Modeling Errors:** Stability of a statically walking quadruped is very sensitive to the center of mass location, and therefore normally requires an accurate model of the robot. Due to variation during the LittleDog calibration procedure it was desirable to design a controller robust to model error.
- **Sensor Errors:** The motion capture system occasionally fails to locate some of the markers, resulting in momentary loss of robot position information on the terrain. Although filtering the motion capture output mitigates this problem, a failure at a critical time such as foot placement planning can lead to robot instability if the algorithm depends too heavily on the assumption

Fig. 1. The BDI LittleDog robot. Each leg has three actuated degrees of freedom, two at the hip ($\theta_{hip,x}$, $\theta_{hip,y}$), and the knee joint θ_{knee}. The whole system therefore has 18 degrees of freedom, three per leg plus (x,y,z,roll,pitch,yaw) of the body.

of perfect sensing.

B. Previous Work

Due to the difficulties involved with legged robots, quadrupedal locomotion on rough terrain has mostly been studied in simulation. For example, the task of searching through known terrain with impassable zones lends itself to formulation as a search problem, so this problem is often treated only in simulation with simplified models as in [19]. Oomichi and colleagues have worked on determining suitability of terrain for legged locomotion as part of a hierarchical controller [18]; their path and gait planning algorithm validation was again performed in simulation. Another approach avoids the issue of sensing the environment and path planning by writing controllers that adapt in real time to the given terrain. Prajoux and Martins [24] discussed a controller that adapts to foot slippage using force control in simulation. Estremera and de Santos approached quadrupedal rough terrain navigation by combining gaits that maintain constant body movement with gaits that break the movement phases into separate body movement and leg movement phases [7]. The algorithm was validated using a rotationally symmetric robot on level terrain with software defined "forbidden zones" and the hybrid gait was derived assuming pitch and roll remained close to zero. Our robot is asymmetric and overcoming the terrain requires significant pitch and roll so the algorithm was not directly applicable. Their more recent work [8] presents a gait that does function on modestly uneven terrain (6.5 cm step compared to 48 cm extended leg length), but the terrain is composed of simple steps.

C. Proposed Solution

We present a controller that combines aspects of deliberative and reactive controllers to address the issues with rough terrain quadrupedal locomotion outlined in section I-A. The main loop of the controller is a deliberative algorithm that plans the next foot step and determines the motion of the legs and body to achieve the step while maintaining stability. This deliberative layer addresses the problem of impassable terrain but neglects foot slippage, accidental collision, and modeling and sensor errors. Reactive modules were created to address specific problems that arose from these issues.

II. DELIBERATIVE MODULES

Considering the issue of impassable terrain and the fact that we have fairly accurate sensor information, we felt a deliberative layer was appropriate as the basis of our controller.

A. State Machine

The deliberative controller was implemented as a state machine, depicted in Fig. 3. The first phase is Gait Selection, which chooses the next leg to swing. The second phase is an optional body shift with all four feet on the ground, which we refer to as a Quad Shift. The Quad Shift phase moves the body to position the center of gravity (COG) inside
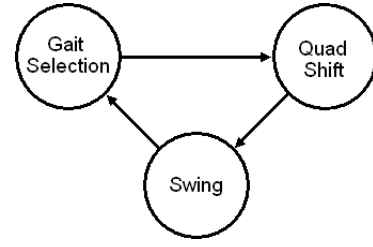

Fig. 3. A partial state machine of a single swing leg cycle. The algorithm starts by deciding which leg to swing in gait generation. The body is then shifted on all four legs during Quad Shift to position the COG inside the support triangle of the non swing feet. The robot then executes the swing while possibly shifting the body.

the upcoming support polygon to ensure stability. The third phase is the Swing, which swings the leg and may involve a simultaneous body shift. This simple three phase model allowed us to achieve several gaits. The only knowledge shared between these states is the swing leg identified in the gait generation phase.

B. Stance List Control

We use the concept of a *stance* to define the configuration of the robot at any point in time. The stance is composed of body position and orientation and the location of all four feet. Any state that results in motion will output a series of waypoint stances that start with the current stance and end with the final desired stance of the state. Stances are then generated between these waypoints, smoothly transitioning from one to the next such that playback at 100Hz generates the desired position trajectories. These stances are then played back by performing inverse kinematics and passing the resulting desired joint angles to a low level servo motor controller each control cycle. The transition to the next state occurs when all stances in the current stance list have been played. The reactive modules discussed in III then modify the stances as they are replayed.

C. Gait Selection

For terrain that allows a fairly straight path, we use the standard gait identified by Muybridge [17]. This gait is sufficient to provide reasonably fast movement on flat terrain and adapts well to variation in mildly rough terrain. However, as the terrain becomes increasingly complex, it is likely that a straight path may not be possible. For this terrain we use an omnidirectional gait selection algorithm that chooses the next leg to swing based on three simple heuristics.

```
start with all legs
IF direction has not changed
    THEN remove the previous swing leg
IF lifting a leg leaves an insufficient
   support polygon
    THEN remove it
CONSIDER the remaining legs
    CHOOSE the leg which can move
        furthest in the desired direction
```

D. Quad Shift Phase

The Quad Shift phase ensures stability of the initial support triangle of the upcoming swing phase by shifting the body center. In practice, it is useful to minimize the time spent in the Quad Shift phase since no leg is swinging. To do this the body shifts the COG along the shortest path across the *trot line*, defined as as the line segment connecting the two support legs not diagonal to the swing leg, allowing for a stability margin as shown in Fig. 4. Using a fixed gait this algorithm results in a shift only on hind leg swings, as the COG is already inside the support polygon after the hind swing. As the terrain becomes increasingly difficult, speed becomes less of an issue than stability and kinematic reachability. For this type of terrain the desired shift position is the centroid of the upcoming support polygon as shown in Fig. 5; this sacrifices speed to maximize stability for omnidirectional walking.

E. Swing Phase

The swing phase determines a trajectory for the swing leg to take and possibly a shift as well. Since the swing phase must generate the swing foot position, it must classify the terrain as passable or impassable. Therefore, the terrain evaluator is presented in this section along with the submodules that generate the shift and swing trajectory.

1) Terrain Evaluation: A terrain scorer classifies potential foot placement at a given point on the terrain as acceptable or unacceptable. The scorer checks small grids around the point and compares the heights to the height at the point. It then rejects the point if it is on too large of a slope, too

close to the top edge or the base of a cliff, or inside of a hole. The grid sizes for each check and the height difference thresholds are parameterized and have been tuned to provide acceptable classifications for the LittleDog robot. For run time efficiency, the scorer precomputes these points on a fine (0.5mm) grid at startup and stores them in a table. An image of the scored terrain is shown in Fig. 6.

2) Shift Generation: Part of our task is to maximize speed while traversing the rough terrain. Our initial terrain selections were a single step and some fairly uniform height rocks. While these required some care in foot placement, they permitted a continuous standard gait with the body staying in constant motion while traversing the terrain. Since the terrain allowed reasonably straight traversal without large direction changes, we maximized our speed by making use of the robot's full body range to extend its reach during the swing phase. The swing shift algorithm is

```
IF hind leg:
    THEN shift half way to the
        midpoint of the front
        stability boundary
IF front leg:
    THEN shift to the midpoint of
        the front stability boundary
```

The resulting behavior stretches the body forward during the swing phase to obtain a further reach as shown in Fig. 7. Notice that a front leg swing shift will shift directly to the
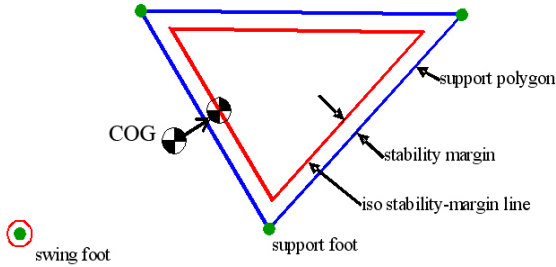

Fig. 4. The Fast Quad Shift algorithm. This ensures the smallest Quad Shift movement and minimizes the time spent in the quad shift phase.
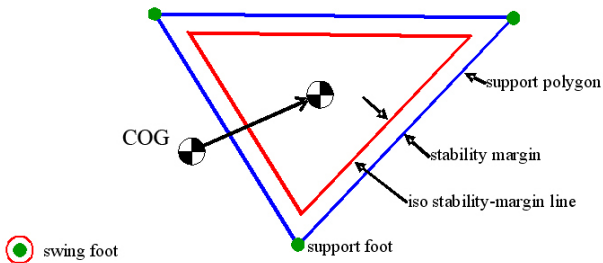

Fig. 5. The Stable Quad Shift algorithm. This requires a longer quad shift time but results in a more stable stance and puts the robot in a good position to move in any direction.
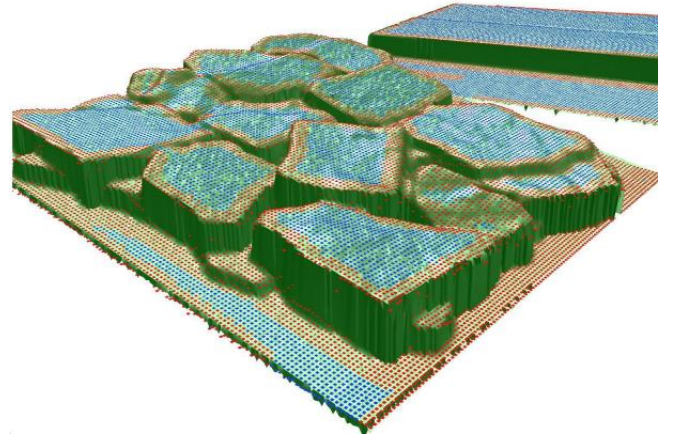

Fig. 6. An example of the terrain scorer run on the tilted rough terrain board. The painting resolution is [5mm]. The blue points represent acceptable foot locations; the red points represent unacceptable locations.
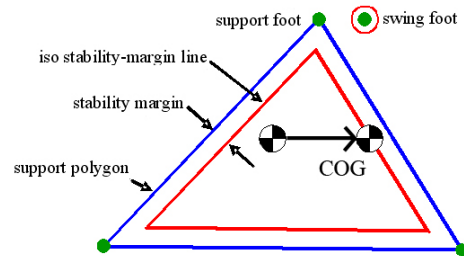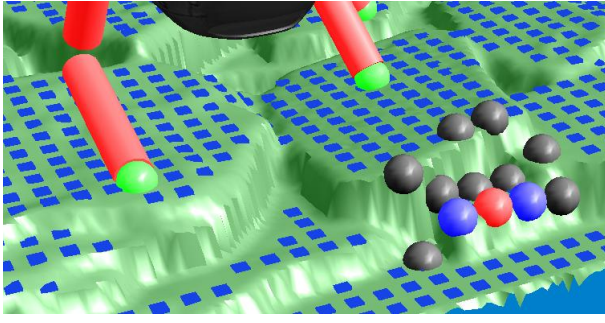

Fig. 7. The swing shift for a front leg swing phase.

Fig. 8. An example set of points considered for the current foot target. The red ball is the initial candidate, the blue and black balls are subsequent candidate points. The green balls represent current foot locations. The blue squares on the terrain represent acceptable foot locations.
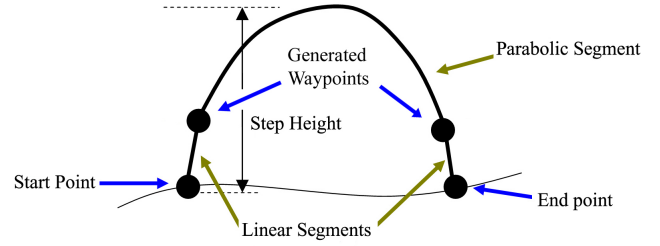

Fig. 9. The swing leg foot trajectory. The trajectory consists of three segments. The first is a straight line upwards. The second is a parabola with an apex at the desired step height. The final segment is a straight line down to the end point.

front stability boundary of the support triangle – a marginally stable position. This allows the robot to fall onto the front leg, a characteristic of quadrupeds that the controller takes advantage of to maximize forward movement.

The shift generator also determines the desired pitch and height of the stance so that the swing leg can clear the terrain and reach the target location. The initial target pitch of the body is zero degrees. The target body height is an average of the lowest body height such that no knee joint angle is less than a minimum threshold, and the highest body height such that no leg is over extended. When adjusting the pitch and height from these targets, the terrain in front of the swing leg is scanned to determine the highest point. A check is then performed to determine if there is enough clearance based on the target pitch and height for the swing leg to step over and onto the highest point. If there is insufficient clearance, the pitch of the robot is adjusted up to a maximum allowed pitch. If there is still insufficient clearance, the height of the robot is adjusted up to the maximum allowed without over stretching a leg.

3) Foot Swing Generation: The swing generator determines the desired trajectory of the swing foot. We attempt to place the foot at the furthest possible position in the direction of travel for the current height. The initial estimate assumes level terrain and uses trigonometry to determine the maximum reach. We enumerate possible foot locations in a grid running backwards from the maximum reach estimate, choosing the first acceptable foot location found. An example swing enumeration is shown in Fig. 8. A foot location is considered acceptable if it is kinematically reachable, on acceptable terrain, and does not result in the swing leg contacting the terrain. In the case that no point is considered acceptable, the maximum reach point is chosen. Once the target point is found the swing generator creates a set of waypoints for the swing leg to pass through. A typical path is depicted in Fig. 9. These waypoints are set based on terrain, and for rough terrain the step height is set to the maximum kinematically reachable value. On rough terrain, the robot's stability is very sensitive to errors in properly locating the swing foot on the terrain. Errors caused by issues such as those described in section I-A integrate over the

swing leading up to the final foot placement. To address this we replan the final foot placement when we begin to lower the foot. This new trajectory is based on the current observed stance and yields a swing trajectory that can more accurately place the foot.

III. REACTIVE MODULES

Due to the control issues outlined in section I-A, the deliberative algorithm alone fails under certain conditions. These failure modes were identified and addressed by specific modules with control components combining aspects of reactive and deliberative schemes appropriate to the problem they address.

A. Pitch and Roll Controller

During stance list playback, foot slippage can cause discrepancies between the current stance of the robot and the current desired stance in the list. These foot location errors translate into pitch and roll errors, and the pitch and roll controller reactively cancels these errors by adjusting each support leg's extension.

B. Hind Leg Tip Detector

The shift generator is designed to maintain the center of mass inside the support polygon of the upcoming swing, but due to modeling errors the Fast Quad Shift algorithm does not always succeed. In the case of front leg swings, this is not disastrous because the robot falls forwards onto the swing leg while attempting to shift forwards, effectively catching itself. In the case of a rear swing leg, particularly when the body is pitched up to overcome an obstacle, the robot can tip back onto the rising swing leg and roll over. We solve this problem reactively by monitoring the pitch and roll dynamics during the lifting phase of hind leg swings to determine if the robot is tipping back. If this condition is detected, the stances that have been executed since the start of the swing are replayed backwards, thus putting the leg back on the ground. Once the leg is back in its starting position, the body is repositioned by moving the center of mass closer to the centroid of the stance feet polygon and the swing is recalculated and executed.
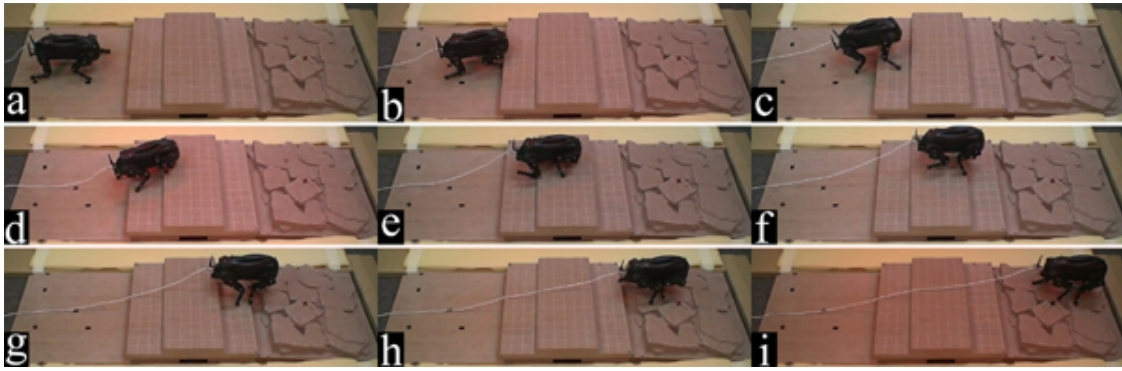
Fig. 10. A time lapse of the robot traversing rough terrain. The robot was given a target destination on the far rock terrain board and began facing the desired direction. Images are evenly spaced in time, total run time is 47 sec.
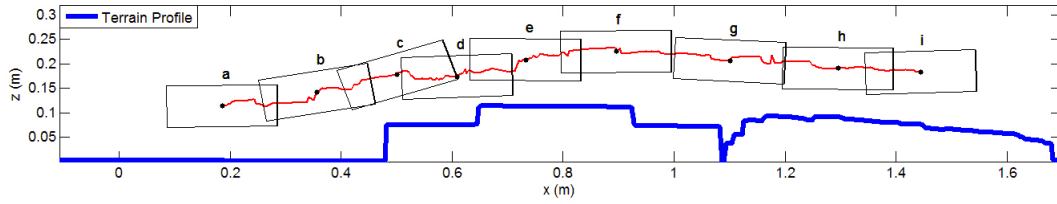

Fig. 11. A graph of the terrain height and the center of mass trajectory. The rectangles represent the position and attitude of the robot body as it traverses the terrain from left to right, corresponding to the images in Fig. 10
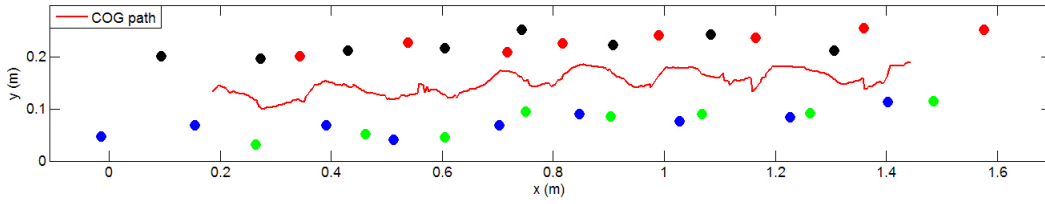

Fig. 12. An overhead graph of the body center of gravity and footstep positions. The red, green, blue, and black points represent the front left, front right, hind right, and hind left steps, respectively.
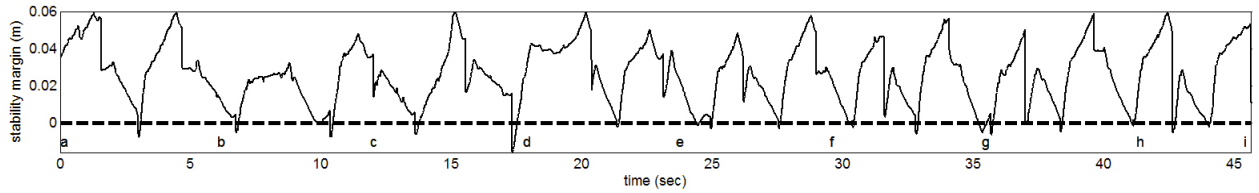

Fig. 13. A measure of static stability S_{SM} throughout the run. The stability is measured as the shortest distance of the horizontally projected COG from the edge of a support polygon. Notice the stability margin periodically drops below zero at the end of each front leg swing.
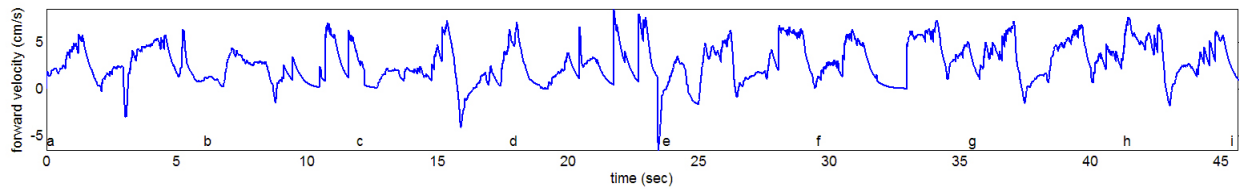

Fig. 14. A graph of the filtered forward velocity $\dot{x}$ of the robot body center throughout the run.

C. Soft Force Control

With a high gain position control scheme, small errors can cause the robot to punch into the ground and destabilize itself or stop the foot before touching down on the terrain. We use a compliant foot placement algorithm to guarantee both that the foot is on the ground before the transition to the next phase and that the placement is smooth enough to avoid accidentally shifting the body. We switch from position control of the leg to a pseudo-force control when the foot is close to touching the ground, and generate a desired trajectory from the current foot location to the target point on the ground. When in pseudo-force control mode, a desired current (torque) is given to the knee joint such that the foot moves down with a constant desired force. The desired path of the foot is parameterized by the knee joint so as the foot moves down and the knee joint changes, the desired position of the foot can be determined. From the desired position of the foot, the desired values for the two hip joints can be calculated. The swing state is over when the velocity of the knee joint is close to zero or the leg is fully extended.

IV. LOW LEVEL CONTROL

The main controller runs on an offboard computer at 100Hz with a wireless connection to the robot. Stances are generated by the deliberative modules for playback on the robot, and are then modified by the reactive control modules to account for accumulated error. Every control tick the controller performs inverse kinematics on the desired stance to produce desired joint positions. These positions are then transmitted to the onboard computer on the robot, which performs low level PD control on the individual joints of the robot at 500Hz. Gains of $K_p = 30 \frac{Nm}{rad}$ and $K_d = 0.6 \frac{Nms}{rad}$ were used on in the low level controller.

V. RESULTS

The robot was run using the Fast Quad Shift algorithm with a standard gait generator. A time lapse of the robot navigating rough terrain is shown in Fig. 10. A corresponding profile view of the center of mass position in relation to the terrain is depicted in Fig. 12. The static stability margin S_{SM}[16] for this run, which is defined as the distance of the center of mass of the robot projected onto the support triangle to the nearest edge of the support triangle is shown in Fig. 13. Notice the stability margin periodically drops below zero at the end of each front leg swing.

VI. CONCLUSIONS AND FUTURE WORK

A. Conclusions

This paper describes a controller for the LittleDog robot walking on rough terrain. We identified several control issues and presented a controller that combines deliberative and reactive modules to address these issues. The basis of the controller is a deliberative algorithm that chooses foot locations and generates a body shift and swing motion. Reactive modules were added to address specific problems arising from foot slippage and other unmodeled errors. These monitor critical events such as hind foot liftoff and foot
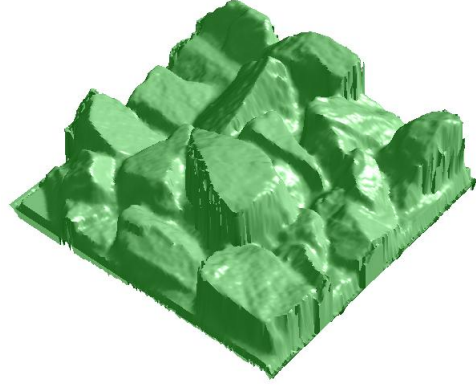

Fig. 15. The model of a terrain board that will be used to test the mobility of a statically walking LittleDog robot over extreme terrain. We hope to explore the limits of what is reachable for a statically stable controller. The tallest rock is approximately 15 cm high.

placement. We also presented an omnidirectional gait for slower, more stable walking.

B. Future Work

We are extending the controller to deal with more challenging terrains such as the one shown in Fig. 15. In particular we wish to explore terrain that significantly limits the possible paths for the LittleDog robot. In addition we are investigating the use of experience based machine learning to further develop the controller to respond to unmodeled disturbances, such as foot slippage. The current controller fails on this terrain due to software constraints on the pitch and roll of the robot; we are currently working on methods for relaxing these constraints while maintaining stability. We also plan to investigate dynamic maneuvers and their applicability to rough terrain.

VII. ACKNOWLEDGMENTS

REFERENCES

[1] H. Adachi, T. Arai, N. Koyachi, and K. Homma. Six degrees of freedom position and posture control for a quadruped robot. *Intelligent Autonomous Vehicles 1995. Postprint Volume from the 2nd IFAC Conference*, pages 133 – 8, 1995.

[2] H. Adachi, M. Koyachi, T. Arai, and Y. Shinohara. Human-operated walking control of a quadruped by event-driven method. *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97*.

[3] J. E. Bares and D. S. Wettergreen. Dante II: Technical description, results, and lessons learned. *International Journal of Robotics Research*, 18(7):621–649, July 1999.

[4] P. Gonzalez de Santos and M.A. Jimenez. Path tracking with quadruped walking machines using discontinuous gaits. *Computers & Electrical Engineering*, 21(6):383 – 396, 1995.

[5] P.G. de Santos and M.A. Jimenez. Generation of discontinuous gaits for quadruped walking vehicles. *Journal of Robotic Systems*, 12(9):599 – 611, Sept. 1995.

[6] C. Eldershaw and M. Yim. Motion planning of legged vehicles in an unstructured environment. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2001.

[7] Joaquin Estremera and Pablo Gonzalez De Santos. Free gaits for quadruped robots over irregular terrain. *International Journal of Robotics Research*, 21(2):115 – 130, 2002.

[8] Joaquin Estremera and Pablo Gonzalez de Santos. Generating continuous free crab gaits for quadruped robots on irregular terrain. *IEEE Transactions on Robotics*, 21(6):1067 – 1076, 2005.

[9] B. Hengst, D. Ibbotson, Son Bao Pham, and C. Sammut. Omnidirectional locomotion for quadruped robots. *RoboCup 2001: Robot Soccer. World Cup V (Lecture Notes in Artificial Intelligence Vol.2377)*, pages 368 – 73, 2002.

[10] I. Hiroshi and K. Masayoshi. Local obstacle recognition for a quadruped robot by distance sensors. *Proceedings. 2003 IEEE International Conference on Robotics, Intelligent Systems and Signal Processing*.

[11] Min-Hsiung Hung, Fan-Tien Cheng, Hao-Lun Lee, and D.E. Orin. Increasing the stability margin of multilegged vehicles through body sway. *Journal of the Chinese Institute of Engineers*, 28(1):39 – 54, 2005.

[12] M.A. Jimnez and P.G. de Santos. Terrain-adaptive gait for walking machines. *International Journal of Robotics Research*, 16(3):320 – 39, 1997.

[13] S. Kelly and R. Murray. Geometric phases and robotic locomotion, 1995.

[14] H. Kimura, Y. Fukuoka, K. Konaga, Y. Hada, and K. Takase. Towards 3d adaptive dynamic walking of a quadruped robot on irregular terrain by using neural system model. *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

[15] S. Ma, T. Tomiyama, and H. Wada. Omnidirectional static walking of a quadruped robot. *IEEE Transactions on Robotics*, 21(2):152 – 61, 2005.

[16] R.B. McGhee and G.I. Iswandhi. Adaptive locomotion of a multi-legged robot over rough terrain. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-9(4):176 – 82, 1979/04/.

[17] E. Muybridge. *Animals in Motion*. Dover, 1957.

[18] Takeo Oomichi, Yasutaka Fuke, and Tatsunori Hayashi. Navigation of a quadruped robot in uneven terrain with multiple foot sensors. *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 191 – 198, 1994.

[19] Prabir K. Pal and K. Jayarajan. Generation of free gait–a graph search approach. *IEEE Transactions on Robotics and Automation*, 7(3):299 – 305, 1991.

[20] Sung-Ho Park and Gwang-Jo Chung. Quasi-static obstacle crossing of an animal type four-legged walking machine. *Robotica*, 18(5):519 – 533, 2000.

[21] Ma Peisun and Ma Lie. A study of turning gait control for quadruped walking vehicle. *Journal of Shanghai Jiaotong University*, 29(5):87 – 92, Sept. 1995.

[22] S. Peng, C.P. Lam, and G.R. Cole. A biologically inspired four legged walking robot. *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol.2:2024 – 30, 2003.

[23] L. Pettersson, K. Jansson, H. Rehbinder, and J. Wikander. Behavior-based control of a four legged walking robot. *MECHATRONICS '98. Proceedings of the 6th UK Mechatronics Forum International Conference*, pages 361 – 6, 1998.

[24] R. Prajoux and L. de S.F. Martins. A walk supervisor architecture for autonomous four-legged robots embedding real-time decision-making. *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS 96.*

[25] M. H. Raibert. *Legged Robots That Balance*. MIT Press, 1986.

[26] Reid Simmons, Eric Krotkov, Lonnie Chrisman, Fabio Cozman, Richard Goodwin, Martial Hebert, Lalitesh Katragadda, Sven Koenig, G. Krishnaswamy, Y. Shinoda, William Red L. Whittaker, and P. Klarer. Experience with rover navigation for lunar-like terrains. In *Proceedings of the 1995 Conference on Intelligent Robots and Systems (IROS '95)*, pages 441 – 446, 1995.

[27] W. Wettergreen. *Robotic walking in natural terrain*. PhD thesis, Carnegie Mellon University, 1995.

[28] Jung-Ming Yang and Jong-Hwan Kim. A fault tolerant gait for a hexapod robot over uneven terrain. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 30(1):172 – 80, 2000.