SUNFEST Technical Report TR-CST01DEC05, Center for Sensor Technologies, Dept of Electrical and Systems Eng, Univ. of Pennsylvania, Philadelphia, PA 2005

University of Pennsylvania

SUNFEST

NSF REU Program
Summer 2005

LEARNING LEGGED LOCOMOTION OVER EXTREME TERRAIN

NSF Summer Undergraduate Fellowship in Sensor Technologies
David Cohen (Mechanical Engineering) – The University of Pennsylvania
Advisor: Dr. Daniel Lee

ABSTRACT

Currently, legged locomotion over obstacles remains a great challenge in robotics. The equations describing so-called “extreme” walks are highly complex, and the contact forces between the robot and its environment are not easily modeled. A major research goal in the field of robotics is to develop capacity for “extreme” walks while keeping computations tractable.

This study was designed to create a method for making a Sony Aibo walk up a 1-inch step consistently. Project activities focused on finding ways to apply simple models for generating walks onto the step. Once the study identified a class of “extreme” walks in lower dimensions, parameters were hand-tuned in an effort to assess whether a prospective class of walks was appropriate or not.

As a result of the work completed in this study, both a 35 mm and a 50 mm step were scaled successfully by the Sony Aibo. Avenues for future work include: 1) the development of an automatic system for determining footfall order and step displacements, 2) a method for automatic primitive identification and switching, 3) refinements to the fields employed, and 4) extensions to extreme dynamic locomotion.

Table of Contents

1. INTRODUCTION

A major research goal in the field of robotics is to develop capacity for “extreme” walks while keeping computations tractable. In order to create a class of agile legged robots capable of assisting troops in the field, DARPA (Defense Advanced Research Projects Agency) has issued solicitation BAA 05-25, which has as its primary objective the development of learning techniques that will permit a quadruped to walk across an obstacle-strewn terrain.

This study was designed to create a method for making a Sony Aibo walk up a 1-inch step consistently, in preparation for the associated DARPA project. The Sony Aibo was selected because, although the DARPA-supplied robot is not currently available, the Aibo has a similar size and structure. Project activities focused on finding ways to apply simple models for generating walks onto the step. The rationale behind developing the simple models was that the models would eventually be machine learned. To address the present objective, hand-tuning alone was sufficient.

As a result of the work completed in this study, both a 35 mm and a 50 mm step were scaled successfully by the Sony Aibo. The method used will be explained fully in the following sections. Areas requiring future work -- such as footfall order determination and primitive switching -- will also be described.

2. OVERVIEW OF STEP METHOD

The method used for scaling the obstacles mixed a potential field formulation for torso placement with pre-set geometries for footpaths during steps. The torso was subjected to rigid-body translation and rotation from fields that were determined by leg extension, leg orientation, and body balance. At each frame, the step function would attempt to find a local minimum in the potential field by iteratively “riding” the field, and the torso would move toward this minimum as quickly as possible. The feet were pre-sequenced to place themselves at certain end positions in turn and, as such, were not affected by the fields.

Interspersed between the footfalls were periods in which the robot would shift its center of balance over the three feet that were to be stationary over the next cycle. This insured that the robot did not immediately fall over when the next foot was raised.

A set of primitives (namely, front-up move-forward and rear-up) were found that allowed the robot to more effectively negotiate large obstacles. By separating the larger task of obstacle-scaling into these primitives, field parameters could be tuned more specifically and different footfall sequences could be determined for different situations.

Part 3 will explain the types of potential fields exerted on the torso. Part 4 will explain the rigid-body dynamics used to evaluate the effect of these fields. Part 5 will briefly explain the implementation of trapezoidal step paths. Part 6 will discuss the various primitives’ parameters and the footfall sequences for these primitives.

3. EXPLANATION OF POTENTIAL FIELDS

Early on, it was recognized that it would be necessary to have the torso displace and pitch in order to keep the robot's feet stable and within their configuration spaces. The original study proposal [1] called for the use of explicit primitives, like "pitching" and "climbing", to accomplish these tasks. This, however, would have required complex tuning and switching. The use of potential fields allowed vast simplification.

There were three main types of fields applied to the torso: radial leg fields, angular leg fields, and a balance field. The leg fields were applied to the hips of each leg, and resulted in both translation and rotation for the torso. The balance field was applied at the center of mass of the robot, and thus allowed for only translation. As will be explained in section 4, these fields determined instantaneous momentum, not force.

The radial leg and angular leg potential fields were modeled to keep the feet within their configuration spaces. The balance field was added to keep the dog from falling over. These fields are explained in greater detail below.

3.1 RADIAL LEG FIELDS

The radial leg fields, depicted in Figure 1 as springs, were nicknamed "shock-absorber fields" since their effect was similar to attaching shock absorbers between the hip and foot of each leg. Stretching a leg near the limit of its configuration space tugged the torso after it; likewise, bringing a leg close in to the torso tended to push the torso away. The purpose of these fields was to avoid the imaginary angles and odometry problems associated with attempts to position the leg beyond the singularities.

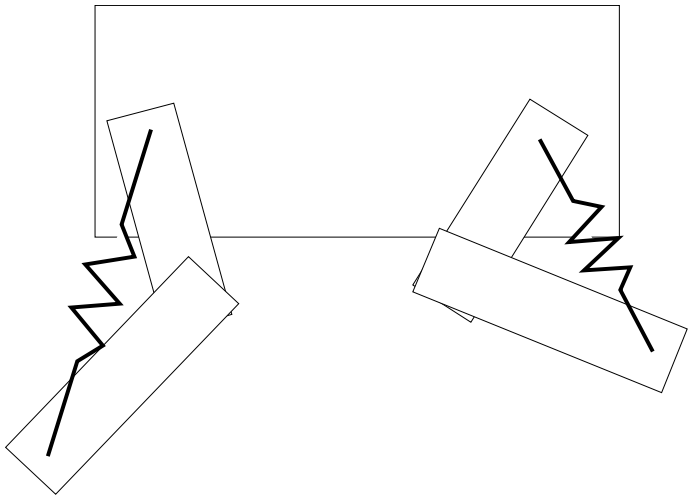


Figure 1 – Radial Leg Fields

The characteristic force-displacement response, though, was quite dissimilar to the classical linear spring. The study showed that it was advantageous to create a large "dead

zone" within which the robot was unaffected by the shock-absorber field. Therefore, the field was modified to be the product of a line with the sum of two exponentials:

$$\text{Force} = A \cdot (x \cdot \text{natural\_length}) \cdot (\exp(B \cdot (x - \text{max\_length})) + \exp(C \cdot (\text{min\_length} - x)))$$

In the function above, 'x' is the extension between the hip and foot. Note that, by modulating the constants B and C, the operator can make the force-displacement profile asymmetric – an advantage not afforded by other functions with "dead zones" (such as hyperbolic sine or high-degree polynomials). An example of the force-displacement profile of this function, with real parameters taken from the code, is given in Figure 1. Note that, since the constants B and C are equal, the asymmetry is not fully employed.

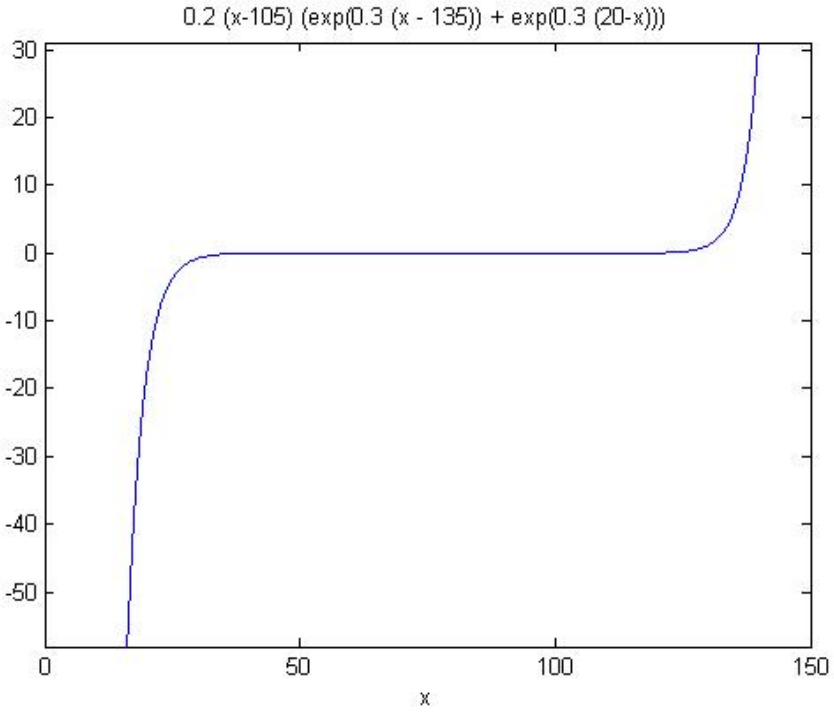


Figure 2 – Force-Displacement profile of Radial Leg Field

3.2 ANGULAR LEG FIELDS

The radial leg fields served to keep the legs within their extension limits. However, using radial fields alone led to situations in which the robot would try to position its legs outside their angular limits. Therefore, it was necessary to add in angular leg fields, so that extreme angles could be avoided.

The angular fields were decomposed into two fields for each leg: a "flap" field, and a "swing" field. Figure 3 shows the axis convention employed for the robot, and Figures 4 and 5 show the "swing" and "flap" angles, respectively. The axis convention is consistent across the three figures.

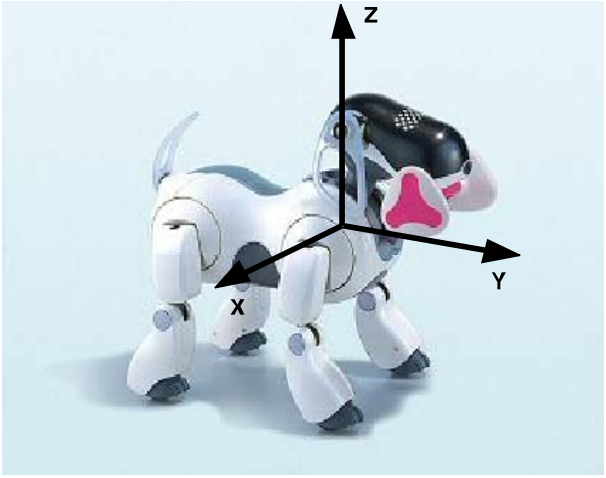


Figure 3 – Sony Aibo Axis Convention

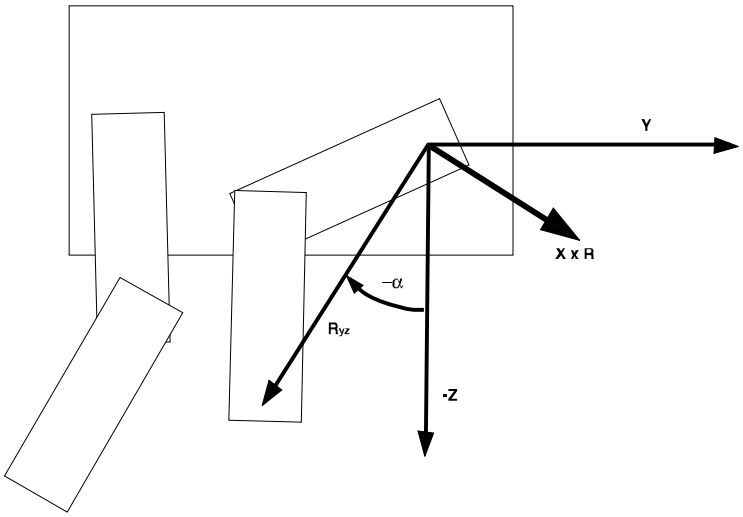


Figure 4 – Y-Z Plane of Robot with "Swing" Angle Indicated (Front Right Leg)

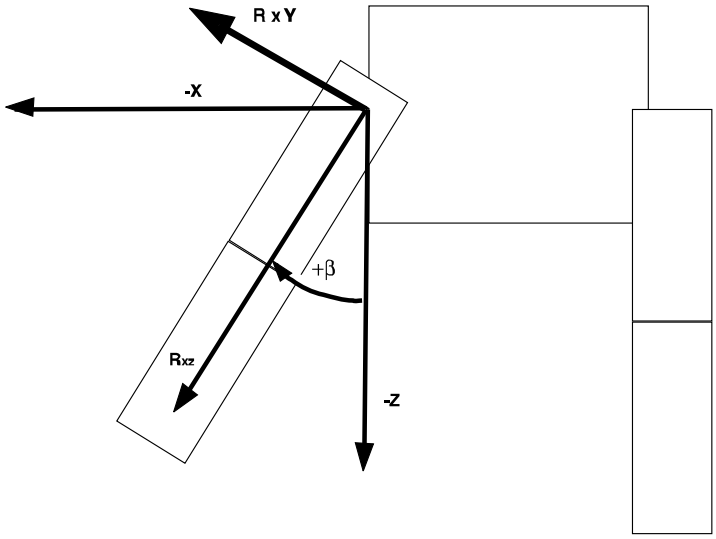


Figure 5 – X-Z Plane of Robot with "Flap" Angle Indicated (Front Right Leg)

The "swing" angle, marked α on Figure 4, is the angle between the projection of **R** (the vector from the hip to the foot) on the Y-Z plane and the **–Z** vector. The sign convention used made the angle depicted negative, so it is marked as such. Likewise, the "flap" angle, marked β on Figure 5, is the angle between the projection of **R** on the X-Z plane and the **–Z** vector (the sign convention makes the angle depicted positive).

For each field a force function similar to those used by the radial fields was implemented, since it was found that a "dead zone" was advantageous with the angular fields as well. Thus, both "flap" and "swing" had a natural angle, a minimum angle, and a maximum angle associated with it.

The direction along which the force was applied to the torso, however, was different between the two fields and distinct, obviously, from the radial fields as well. The direction for the force application due to the "swing" field was along the cross product of the **X** vector and **R**, and the direction for the force application due to the "flap" field was along the cross product of **R** and the **Y** vector. Both directions are denoted on their respective figures with the heavier arrows. The rationale behind using the cross products for the directions of application was that they would provide the most efficient angular change without radial change, given that the foot position remained constant.

3.3 BALANCE FIELD

The final potential field force implemented on the torso was a balance field. In order to keep the robot statically stable (the gaits generated were all static, not dynamic, gaits) it was necessary both to keep at least three legs on the ground at a time and to ensure that the robot's center of mass was within the polygon determined by the feet on the ground. Therefore, a field was implemented that drew the torso towards the centroid of the polygon determined by the planted feet.

Figure 6 shows a schematic of the balance field for the case where the front right foot is off the ground. The planted feet are shown and labeled (no legs are shown, since they are unnecessary for the calculation of the force or direction). The force runs from the centroid of the torso to the centroid of what is, in this case, a triangle whose vertices are the positions of the planted front left foot, rear left foot, and rear right foot.

The magnitude of the force is calculated using a hyperbolic tangent:

$$\text{Force} = A \cdot \tanh(B \cdot x)$$

Variable 'x' is the magnitude of the vector from the torso centroid to the foot centroid.

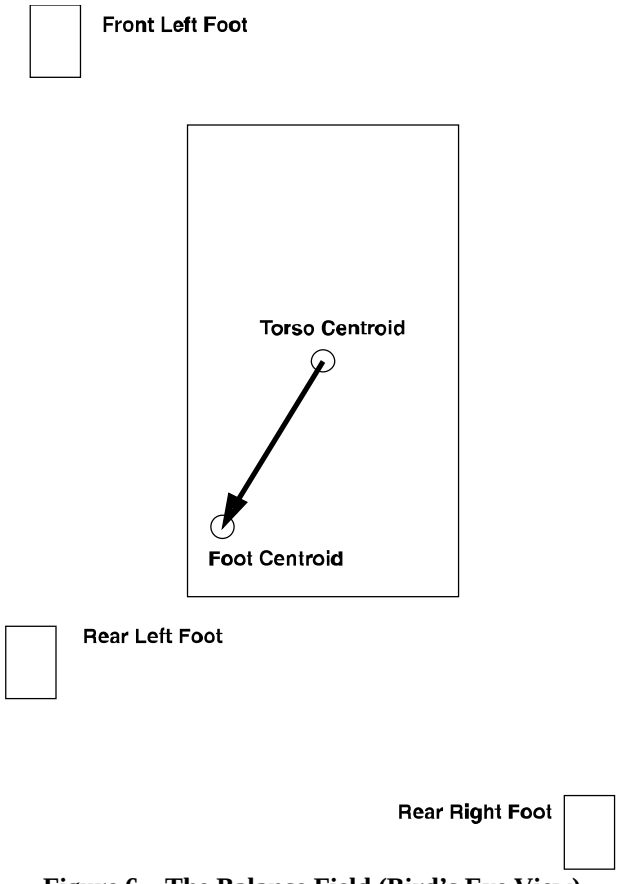


Figure 6 – The Balance Field (Bird's Eye View)

4. EXPLANATION OF THE RIGID-BODY REACTIONS

The torso acted as a 4-DOF rigid body, limited to x-y-z translation and pitching. The "forces" at the hips of the four legs and at the center of mass of the torso were summed by standard techniques to provide a net resultant force and a net moment (it was assumed that the center of geometry of the torso was also the center of mass). The way the force and the moment were applied, however, were different from the standard convention:

$$\text{Force} = \text{mass} \cdot \text{velocity}$$

$$\text{Moment} = \text{moment of inertia} \cdot \text{angular velocity}$$

Thus, the fields implemented are probably most appropriately called "momentum fields". In the interest of simplicity, these "momentum fields" were implemented instead of "force fields". Since the function was aiming for the local minimum, it didn't make a difference whether velocity was integrated from accelerations or not.

The mass and moment of inertia were defined as parameters in the step function. Since the system was 4-DOF, the moments of inertia corresponding to yaw and roll were effectively set to infinity. There were also caps on the maximum angular and translational velocities that could be achieved.

5. EXPLANATION OF THE LEG PATHS

The footpaths used were inspired by the trapezoidal steps simultaneously developed by Newcastle University and the University of New South Wales for the Robocup 2003 competition [2, 3]. The rationale behind using a trapezoidal step in robot soccer was the speed advantage. Disengaging the claw of the Sony Aibo from the field material yielded a speed increase of about 10%. While, for the case at hand, speed was not important, it was crucial that the robot's feet not be hindered by the material. Consequently the trapezoidal method was employed.

A schematic of the trapezoidal step is provided in Figure 7. There was no bottom stroke to the step. Traversal was accomplished by torso repositioning across different foot positions. The parameters ϕ, γ, and 'minimum clearance' determine the shape of the step. Additional parameters determine the speed with which the step is executed in the Rise, Traverse, and Lower Stages.

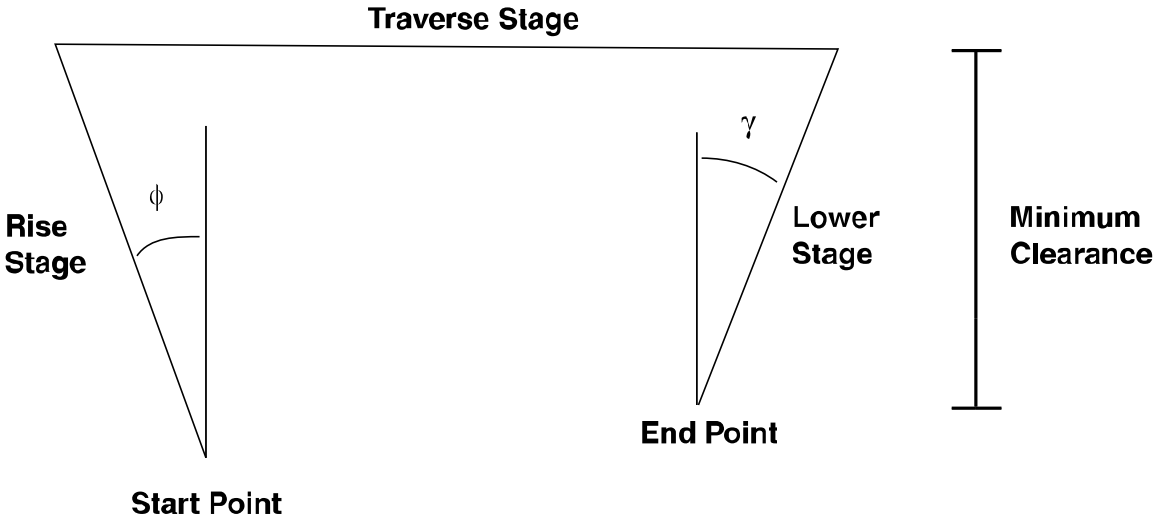


Figure 7 – Trapezoidal Step Path (Side View)

In the case that there is an x-displacement in the step, the path must skew in the X-Y plane. In this case, both the Rise and Lower Stages have no component in the x-direction. This is to ensure that there is still a clean disengagement from the field material. Figure 8 gives a top down view of what a skewed step might look like.

6. PRIMITIVE PARAMETERS AND FOOTFALL SEQUENCES

Once all the fields were in place, the final challenges were to find a suitable sequence of footfalls to climb the step and to tune the field parameters.

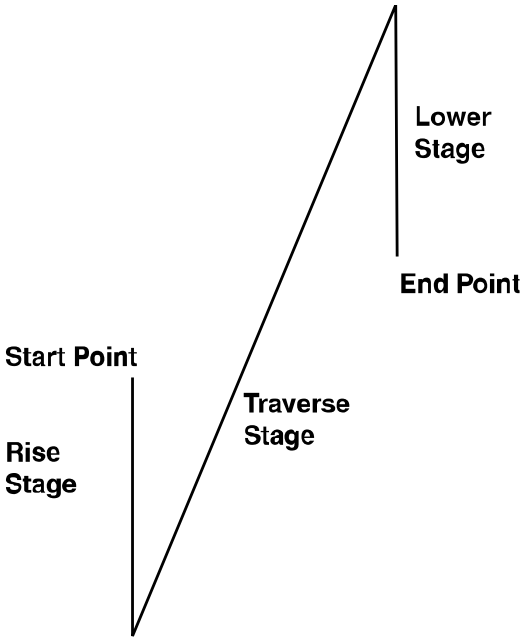


Figure 8 – Skewed Trapezoidal Step Path (Bird's Eye View)

The footfall sequence was a list of 4 element vectors, which denoted the x, y, and z displacements of the step as well as an index of the leg that was stepping. The step function assumed that the foot was just resting on a surface at the end of each footfall. The function was completely open-loop and this "foot resting" condition was necessary so that the robot's orientation could be accurately calculated. Unfortunately, it was the general case that for the scaling footfalls some experimentation was required to find the exact z-displacement that would allow the foot to just set down. This was because the Sony Aibo has large plastic casings around its forepaws, making it very difficult to get exact measurements for the point of contact.

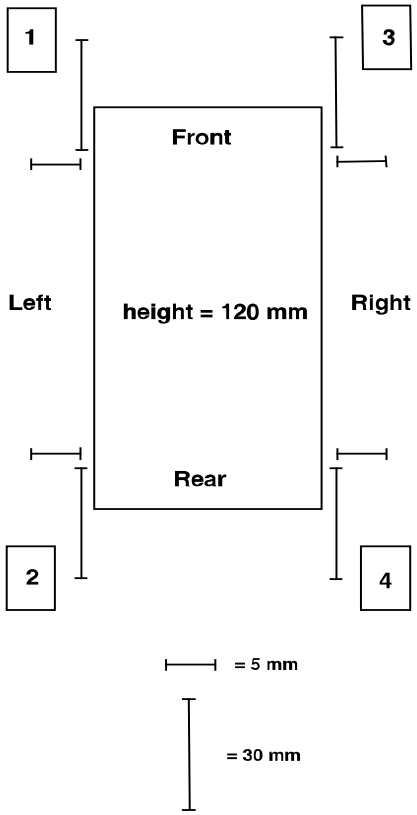
The method for parameter tuning, likewise, was trial and error: any set of parameters that led the robot to try to effect positions outside its configuration space was deemed unsuitable, as was any set that resulted in the robot's losing its balance. Tuning the parameters thus proved exceptionally difficult, as the robot seemed to inevitably violate at least one of the two criteria. This was remedied, however, by the development of a shifting foot-cycle: while traditional non-extreme static gaits use 4 step foot-cycles, it was found that an irregular cycle led to a large qualitative improvement in the robot's performance. Once the irregular cycle was implemented, it was not long before the parameters were tuned appropriately, and the robot was able to successfully scale a 35 mm step.

The next step after scaling a 35 mm, or 0.25 L (L = leg length) step was to scale a 50 mm (0.35 L) step. Unfortunately, the advances that resulted in the scaling of the smaller step were not quite sufficient to permit scaling the larger one. In general, parameters that allowed good performance for one portion of the step led to significant failure in other portions.

The solution was to simply use different parameters for different parts of the step. The three primitives that resulted (front-up, move-forward, and rear-up) each had its own footfall sequence and parameters – thus the irregular cycle used to scale the 35 mm step became a concatenation of three distinct cycles.

Once the primitives were set in place, it was a simple matter of re-tuning the parameters and footfalls to scale the new step. The footfall sequence remained unchanged from the 35 mm step to the 50 mm step. Indeed, the ease with which the robot was retuned suggests the robustness of the field method, although it should be noted that the motion returned from the front-up primitive had to be smoothed and sped up. This was, however, a problem for finding the minimum of the force field and not a problem inherent in the fields themselves. Figures 9 and 10 give information on the primitive parameters and footfall sequences employed for the 50 mm step.

	Front-Up	Move-Forward	Rear-Up
k_radial	1	1	1
coeff_radial	0.3	0.3	0.3
m	1	1	1
l	50	50	50
max_v_ride	1	2	2
max_v_trap	1	1.5	1.5
max_o	0.1	0.01	0.01
b	0.5	0.5	0.5
b_coeff	0.05	0.05	0.05
phi	0.15	0.15	0.15
gamma	-0.15	-0.15	-0.15
front_h	25	20	10
rear_h	25	25	25
rise_speed	2	4	4
traverse_speed	2	4	4
lower_speed	2	4	4
natural_flap	0.2	0.2	0.2
max_flap	1	1	1
min_flap	-0.05	-0.05	-0.05
natural_swing	0	0	0
max_swing	1.4	1.4	1.4
min_swing	-1.4	-1.4	-1.4
k_flap	1	1	1
k_swing	5	5	5
coeff_flap	5	5	5
coeff_swing	5	5	5



	1	2	3	4
natural_length	105	110	105	110
max_length	135	140	135	140
min_length	60	65	60	65

Figure 9 – 50 mm step Parameter Summary

The picture in the upper right of Figure 9 identifies the four legs by the numerical convention employed in the footfall sequence. The lower table in Figure 9 uses this convention. Figure 9 also identifies the initial stance of the robot.

The natural_length, max_length, and min_length given in the lower table refer to the associated radial field values (all in mm). The radial parameter k_radial serves as A and

coeff_radial serves as both B and C in the following expression (given earlier in section 3.1):

$$\text{Force} = A \cdot (x - \text{natural\_length}) \cdot (\exp(B \cdot (x - \text{max\_length})) + \exp(C \cdot (\text{min\_length} - x)))$$

Likewise, k_flap and k_swing are both A in their respective functions, and coeff_flap and coeff_swing are each both B and C. The angles natural_flap, max_flap, etc. are all given in radians.

The parameter b serves as A and b_coeff serves as B in the following expression (given earlier in section 3.3):

$$\text{Force} = A \cdot \tanh(B \cdot x)$$

The parameter max_v_ride refers, in mm/frame, to the maximum speed the torso is allowed to move during while "riding" the potential field (i.e. with all four feet on the ground); likewise, max_v_trap is the maximum speed the torso is allowed to move while stepping (three feet on the ground). Similarly, max_o is the maximum angular speed the robot is allowed to pitch (in rad/frame).

Phi and gamma are as noted in Figure 7. Rise_speed, traverse_speed and lower_speed are the maximum mm/frame a stepping foot is allowed to move in each of the three stages (noted in Figures 7 and 8).

	Front-Up			
x displacement	0	0	0	0
y displacement	65	70	70	60
z displacement	0	50	50	0
leg index	2	1	3	4

	Move-Forward			
x displacement	0	0	0	0
y displacement	70	70	70	60
z displacement	0	0	0	0
leg index	1	3	2	4

	Rear-Up			
x displacement	0	0	0	0
y displacement	85	75	85	60
z displacement	0	30	-10	30
leg index	1	2	3	4

Figure 10 – 50 mm step Footfall Summary (displacements in mm)

Certainly, a quick examination of Figure 10 reveals that the three primitives are quite distinct, not only in footfall order but in the characteristics of their steps. Note that the Rear-Up z-displacements for the rear left and rear right legs are much less than 50 mm. This factor is attributable to the previously mentioned forepaw casings. Also, across all three primitives the y displacements of leg 4 (rear right) seem to lag behind the others. It

is possible that this asymmetry makes it necessary to bring foot 3 down 10 mm in the Rear-Up primitive.

7. DISCUSSION AND CONCLUSIONS

Extreme Legged Locomotion is an important and very difficult problem. The huge number of dimensions faced by a researcher attempting to solve the problem makes most solution techniques worthless. In the interest of limiting the problem so that part of it could be solved using traditional (gradient based) learning techniques, a field/pre-set geometry system was set up to create an appropriate class of extreme steps. The system was tested on two obstacles (a 35 mm step and a 50 mm step), and was successful at negotiating both.

Problems remain, however, with determining the proper footfall sequence as well as foot displacements during these sequences. Currently, these sequences and displacements are tailored to specific primitives and are associated with specific obstacles. There is no algorithm in place to find a footfall sequence or set of displacements for an arbitrary obstacle field. In addition, the primitive switching employed here will have to be codified if it is to be learned automatically. Also, refinements to the field system used, including allowances for balance improvements, anticipatory fields, 6-DOF motion, and computational optimization might be necessary. Finally, the matter of extreme dynamic stepping must be addressed. Future studies will be required to solve these crucial problems (see section 8).

8. RECOMMENDATIONS

The problems listed in the closing of section 7 will now be expanded upon, and possible solutions will be suggested.

8.1 FOOTFALL ORDER AND STEP DISPLACEMENTS

The largest unaddressed problem in the extreme step approach outlined here is that the footfall order and step displacements were pre-set by the operator. Any system that endeavors to provide locomotion over an arbitrary obstacle field cannot use this technique.

One possible way to solve this problem is to use the fields themselves to plan appropriate sequences of extension and recovery. By integrating the fields to create a scalar potential field (or by creating a new, simplified scalar field) important information can be gleaned about the “comfort” of certain poses. It may be possible to use a state machine to decide when to extend into “uncomfortable” positions and when to relax into “comfortable” ones (a naïve approach of only executing “comfortable” positions would greatly hinder the efficacy of the walk). Future footfalls could be planned in advance, and appropriate “set-up” steps could be sequenced to allow low potential and good balance.

This footfall code was not put in place due to its sheer computational complexity. Given that the step sequences themselves took upwards of 2 minutes to calculate, cycling through hundreds of potential future moves was not feasible. In addition, since all steps used in the study were handmade, the limited amount of terrain available for testing was likely to make any experimental system unreasonably condition-dependent. Finally, and perhaps most importantly, there was not enough good feedback to test out different footfall patterns. The DARPA set-up will probably solve the last two problems. Significant re-tweaking might solve the first (see section 8.3).

8.2 PRIMITIVE SWITCHING

The next significant problem with the previously outlined extreme step approach is the arbitrary creation of primitives, with their own field parameters and footfall sequences. The footfall sequence problem might be solved by the method described in section 8.1, but the field parameter switching problem remains.

Enumerating primitives is a possible solution. Such primitives could be identified by the pitch angle and roll angle (see section 8.3) as well as the planned trajectory of the stepping foot. A large, but certainly non-infinite, number of primitives would result, and each could be tuned in turn. One difficulty with this method would be learning the border conditions between primitives.

A promising alternative would be to make the field parameters a function of pitch, roll, and intended step trajectory. A simple linear function might suffice. In this way, the problem of primitive creation and switching might be sidestepped. Also, the parameters that define the functions could themselves be learned with traditional gradient-based techniques. The number of dimensions to be learned, however, would at least double, and most likely triple, under this system.

8.3 FIELD REFINEMENTS

The first clear area for field refinement is the balance field. The radial and angular foot fields enjoyed a “dead zone”, which tended to enhance their performance. The balance field, on the other hand, had no such property. Indeed, the field, with its sole dependence on the distance between the torso centroid and planted-foot-triangle centroid, neglects what often are extreme aspect ratios in the triangle. Given additional time for implementation, the simple fix shown in Figure 11 might be applied. The lines are force directions, leading directly in from each face of the triangle. Within the triangle, there is a much smaller pull towards the center.

The inclusion of anticipatory fields is a second proposed improvement for future studies. Currently, all anticipatory action is hard-coded in: in between footfalls, the robot shifts its weight to the feet that will be planted during the next step. A set of anticipatory fields could accomplish this task automatically, making the entire process more coherent.

The third area for improvement is more obvious: the 4-DOF system should be extended to 6-DOF. This was not implemented in the current project, since it was decided that the symmetric nature of the step made the ability to roll and yaw unnecessary. For arbitrary obstacle fields, however, roll and yaw will require substantial refinement. To this end, a quaternion-based representation of the torso's orientation could be implemented, as described by Mirtich [4].

Finally, the fourth area for improvement is the method for determining field equilibrium. In the present state, the field is evaluated up to 100 times per frame to determine in which direction the torso must move. With proper tuning, it might be possible to eliminate this inefficiency all together, or at least limit the cycles to a more reasonable number. This would allow the robot to execute its moves in real time, reacting to outside information. Computational overhead limited the current approach to being open-loop.

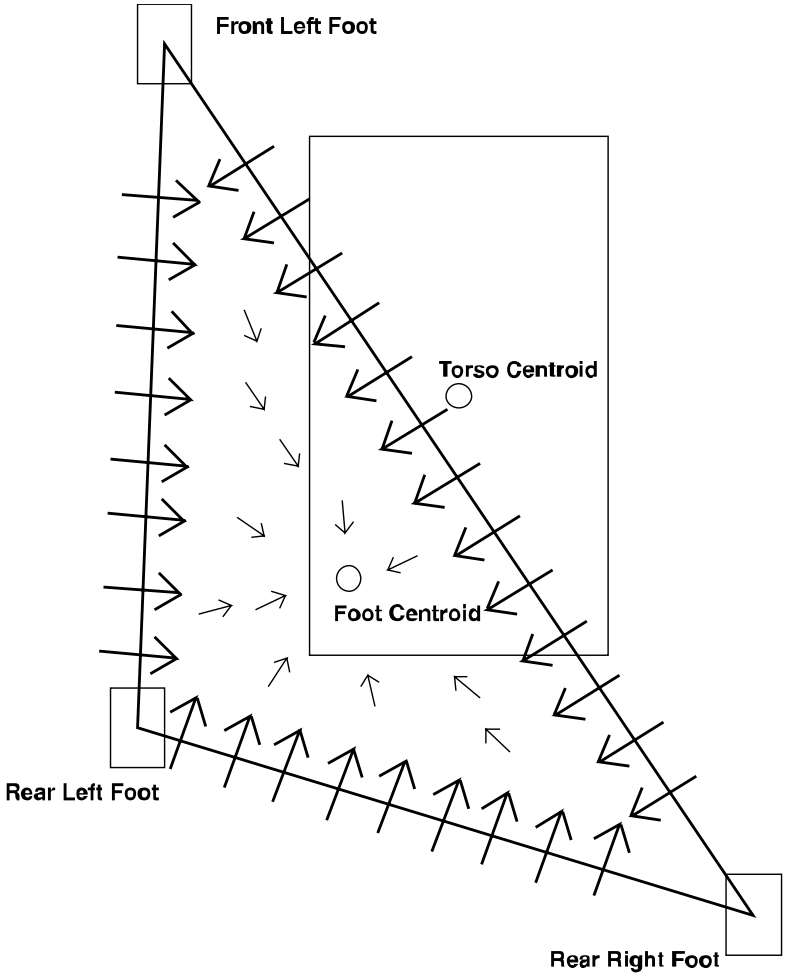


Figure 11 – Refined Balance Field

8.4 EXTREME DYNAMIC LOCOMOTION

The final problem to be addressed is most certainly the most interesting: the difficulty of developing dynamic steps for extreme conditions. Dynamic steps would not only provide

speedier traversal of obstacles, they are also likely to offer improved performance, allowing the robot to cross obstacles which static steps alone cannot surmount.

The same field principles used in the execution of static steps could be used. But they would require significant and complex modifications. In particular, the balance fields and heretofore non-existent anticipatory fields would have to be extremely well tuned to deal with the unpredictable conditions encountered. The balance field would probably require some strategic imbalance as well. More likely, a completely different system will have to be developed to deal with so much additional complexity.

9. ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Daniel Lee, as well as Gilad Buchman and Paul Vernaza at the GRASP lab. I would also like to thank the National Science Foundation for the NSF-REU grant and the Microsoft Corporation for their financial support. It is a pleasure doing significant research as an undergraduate, and I would like to thank all of the people whose tireless efforts make the SUNFEST program possible.

10. REFERENCES

1. D. D. Lee, *Technical Proposal: Learning Low-Dimensional Controllers for High-Speed Quadruped Locomotion*, The University of Pennsylvania, Philadelphia, PA, 2005.

2. J. Bunting et al, *Return of the NUBots*, Newcastle Robotics Laboratory, The University of Newcastle, Australia, Oct. 2003.

3. J. Chen et al, *Rise of the Aibos III – AIBO Revolutions*, The University of New South Wales, Australia, Nov. 2003.

4. B. V. Mirtich, *Impulse-based Dynamic Simulation of Rigid Body Systems*, Dissertation for Ph.D. in Computer Science, The University of California at Berkeley, California, 1996.

APPENDIX: INVERSE KINEMATICS

Inverse kinematics refer to the calculations required to convert an (x, y, z) triple to a set of angles (θ_1, θ_2, θ_3). In the case of the Sony Aibo (and for the DARPA robot, since they share the same limb geometry) there are typically multiple solutions, except at the singularity. Beyond the singularity, any angles returned by the inverse kinematics function will have imaginary components. Presented below is a MATLAB function that takes **XYZ**, an (x, y, z) triple relative to the right front shoulder, and returns **angles**, the angle vector (shoulder swing, shoulder flap, and knee swing) required by the front right leg to effect the position.

The variable 'lone' is the length of the thigh on the robot, and 'ltwo' is the length of the shin. The variable 'upperoffset' is the y-offset from the hip to the knee when the leg is pointing straight down, and 'loweroffset' is the y-offset from the knee to the foot pad under the same conditions. All lengths are in millimeters.

```
function angles = getanglesfrontright(XYZ)

lone = 69.5;
upperoffset = 9;
ltwo = 71.5;
loweroffset = -9;

P = 2*(upperoffset*loweroffset + lone*ltwo);
Q = 2*(ltwo*upperoffset - lone*loweroffset);
R = loweroffset^2 + ltwo^2 + upperoffset^2 + lone^2 - (sum(XYZ.^2));

angles(3) = atan2(Q/sqrt(P^2 + Q^2),P/sqrt(P^2 + Q^2)) - acos(-
R/sqrt(P^2 + Q^2));

if (angles(3) < 0)      %So that we don't snap the dog's knees
    angles(3) = atan2(Q/sqrt(P^2 + Q^2),P/sqrt(P^2 + Q^2)) + acos(-
R/sqrt(P^2 + Q^2));
end

P = loweroffset*cos(angles(3)) + ltwo*sin(angles(3)) + upperoffset;
Q = lone + ltwo*cos(angles(3)) - loweroffset*sin(angles(3));

angles(2) = asin(XYZ(1)/Q);

Q = cos(angles(2))*Q;

components = inv([Q P;P -Q])*[XYZ(2);XYZ(3)];

angles(1) = atan2(components(1), components(2));
```