To appear in *RoboCup-2006: Robot Soccer World Cup X* (RoboCup 2006), 2007.

Autonomous Learning of Stable Quadruped Locomotion

Manish Saggar[1], Thomas D'Silva[2], Nate Kohl[1], and Peter Stone[1]

[1] Department of Computer Sciences
[2] Department of Electrical and Computer Engineering
The University of Texas at Austin

Abstract. A fast gait is an essential component of any successful team in the RoboCup 4-legged league. However, quickly moving quadruped robots, including those with learned gaits, often move in such a way so as to cause unsteady camera motions which degrade the robot's visual capabilities. This paper presents an implementation of the policy gradient machine learning algorithm that searches for a parameterized walk while optimizing for both speed and stability. To the best of our knowledge, previous learned walks have all focused exclusively on speed. Our method is fully implemented and tested on the Sony Aibo ERS-7 robot platform. The resulting gait is reasonably fast and considerably more stable compared to our previous fast gaits. We demonstrate that this stability can significantly improve the robot's visual object recognition.

1 Introduction

In the robot soccer domain, a fast gait is an important component of a successful team. As a result a significant amount of recent research has been devoted to the problem of developing fast legged locomotion for Sony Aibo ERS-7 robots, leading to considerable improvement in gait speeds [1–5].

However, learned gaits optimized solely for speed tend to produce body motions that cause the camera to shake. Such unsteady gaits lead to camera images in which objects are rotated, translated, or blurred compared to camera images from a steady gait. These images make it difficult for the robot to identify objects. For example a pink over yellow beacon is usually identified as a pink blob over a yellow blob, however the pink does not appear above the yellow when the image is rotated. Thus, unstable gaits degrade a robot's object recognition and localization abilities which can cause problems during a game.

This paper proposes optimizing both gait speed *and* stability simultaneously, using a multi-criteria objective function. In addition, experiments are described that explore the idea of using active head movements to compensate for uneven body motion.

The remainder of this paper is organized as follows. Section 2 presents existing machine learning techniques that have been applied to optimize gait parameters for speed. Section 3 describes the parameterized Aibo gait, head motions, and the policy gradient algorithm used to train new gaits. Section 4 describes our training experiments in detail and compares two different methods to offset unstable body movements. In Section 5, applications of stable gaits and future work are outlined, and Section 6 concludes.

2 Related Work

When generating quadrupedal robot gaits, the machine learning (ML) approach offers several advantages over hand-tuning of parameters. Using learning can reduce the amount of time required to find a fast gait and can be easily applied to different surfaces and different robots. ML techniques also do not suffer from the bias a human engineer might have when hand-tuning a gait. For example, there is evidence that when walking the actual joint angles of the Aibo differ considerably from requested joint angles, because of the force exerted by the ground [6]. ML techniques may be less susceptible to this problem than humans who often hand-tune gaits based on the locus of points the foot ideally moves through, as opposed to the actual locus the foot moves through.

Applying ML techniques to directly control an Aibo by manipulating joint angles is a difficult task. Evaluations on physical robots are noisy and take a long time compared to evaluation in simulation. Moreover, some of the intermediate exploratory gaits that ML algorithms generate may cause physical damage to the robot. The Aibo also does not have sensors that can be used during training that can provide closed loop feedback to the controller.

Nonetheless, reinforcement learning (RL) has been used to learn several similar control problems, not limited to Aibo locomotion. RL has been used to control a model helicopter than can hover while inverted in air [7]. Other ML techniques have been applied to directly control simulated bipedal robots: in [8] a central pattern generator was used for rhythm generation in the hips and knees of a simulated bipedal robot, and a dynamics controller was used to control the ankles of robot.

Similarly, previous work has shown that ML algorithms can excel at generating fast gaits for the Aibo by taking advantage of algorithms to optimize parameterized gaits for desirable characteristics. The earliest attempt to use ML algorithms to learn a gait used a genetic algorithm to optimize parameters describing joint velocities and body positions [9].

More recent approaches attempt to learn parameters for gaits that move the Aibo's four feet through a locus of points. In previous work, the policy gradient algorithm has been used, with a half-elliptical locus, to learn an Aibo gait that is optimized for speed [2, 3]. Powell's method of multidimensional minimization has been used to optimize a parameterized gait with a rectangular locus [4]. A genetic algorithm that used interpolation and extrapolation for the crossover step was used to optimize a parameterized gait with a half-elliptical locus [1]. Odometry was used in order to evolve an omni-directional parameterized gait using a genetic algorithm by training the robot to move forward with its target orientation constantly changing [10]. In [5], a genetic algorithm and an acceleration model of the Aibo body was used to optimize a parameterized Aibo gait.

One of the fastest known forward Aibo gaits, which has a speed of 451 mm/s, was learned using a genetic algorithm and an overhead camera to quickly determine walk speeds [11].

To the best of our knowledge, all of these approaches have optimized exclusively for walk speed. This paper is based on the observation that the resulting gaits are often unstable, thus degrading the robot's visual capabilities. We demonstrate that this problem can be solved by optimizing the gait for both speed *and stability* by incorporating stability information into the objective function. This paper applies two different approaches to learning a stable walk. In the first approach, the objective function incorporates stability information. In the second approach, compensatory head movements are performed to counter the unstable body motions of a fast gait.

3 Background

The Sony Aibo ERS-7 robot is a quadruped with three degrees of freedom in each leg [12]. A controller must specify the set of twelve joint angles at each instant in order to specify a gait. Learning a controller for a fast gait by directly manipulating joint angles is a difficult non-linear control problem. One solution to this problem is parameterizing a gait by specifying the loci of points that the Aibo's feet moves through. Doing so can constrain the search space both to make it easier to search and to avoid gaits that can damage the robot. This paper uses a modified version of a half-elliptical parameterized gait modeled after that presented by Stone et al. [13]. Four additional parameters were added to this parameterization that govern compensatory head movements designed to improve head stability.

3.1 Parameterized Motion

The half-elliptical locus used by the fast gait is shown in Figure 1. Each foot moves through a half-elliptical locus with each pair of diagonally opposite legs in phase with each other and out of phase with the other two legs (a trot gait).

The four parameters that define the half ellipse are:

1. The length of the ellipse
2. The height of the ellipse
3. The position of the ellipse on the x axis
4. The position of the ellipse on the y axis

The symmetry of the Aibo is used to reduce the number of parameters that have to be optimized. The length of the ellipse is the same for all four legs to ensure a straight gait. The left and right sides of the body use the same parameters to describe the locus of the gait. The height, x position and y position of the elliptical loci of the front and back two legs use different parameters.

In addition to the leg movements, the head was allowed to make elliptical compensatory movements in order to cancel the effect of body motions that

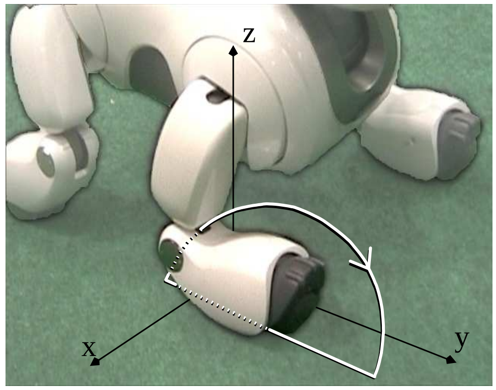


Fig. 1. The half-elliptical locus of each of the Aibo's feet is defined by length, height and position in the x-y plane.

cause the camera to shake. Figure 2 depicts the two types of head movement that were used, which have the overall effect of moving the head in an ellipse. Two parameters were used to specify the head tilt angle limit and head tilt increment at each timestep. Similarly, two parameters describe the head pan motions. Initial values for these parameters were determined by testing just a few sets of values. We leave it to future work to determine how big of an effect these initial values have.

The 15 parameters that completely define the Aibo's movements are:

- The front locus: height, x position and y position (3 parameters)
- The rear locus: height, x position and y position (3 parameters)
- Locus length (same for all loci)
- Front body height
- Rear body height
- Time taken for each foot to move through locus
- The fraction of time each foot spends on the ground
- Head tilt limit and increment (2 parameters, with a limit from -10^o to 10^o)
- Head pan limit and increment (2 parameters, with a limit from -10^o to 10^o)

3.2 Policy Gradient Algorithm

This paper uses a policy gradient algorithm modeled after that presented by Kohl and Stone [2] to optimize the Aibo gait in the continuous 15-dimensional parameter space. The objective function F to be optimized is a function of the gait speed, acceleration and stability, and is described in detail in Section 4.

The policy gradient algorithm uses an initial parameter vector $\pi = \{\theta_1, ..., \theta_N\}$ and estimates the partial derivative of the objective function F with respect to each parameter. This is done by evaluating t randomly generated policies

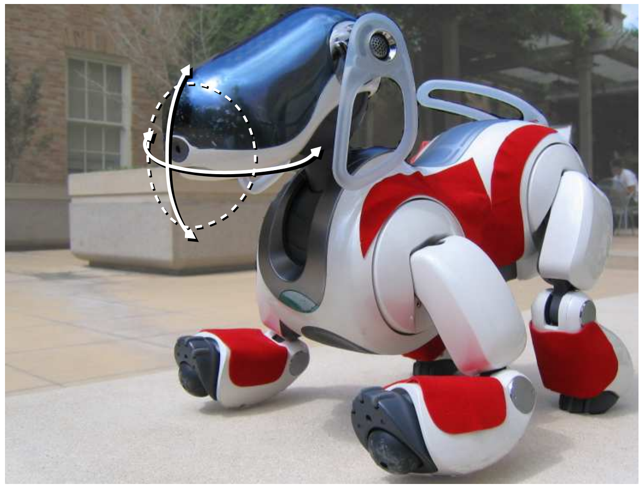


Fig. 2. The Aibo can combine pan and tilt head movements (shown as solid lines) to move the head through an elliptical locus (shown as a dotted line). The center of ellipse is determine by the landmark the Aibo is looking at. The locus is defined by four variables: tilt limit, tilt increment, pan limit, and pan increment.

$\{R_1, ...R_t\}$ near π, such that each $R_i = \{\theta_1 + \delta_1, ..., \theta_N + \delta_N\}$ and δ_j is randomly chosen to be either $+\epsilon_j, 0, or - \epsilon_j$, where ϵ_j is a small fixed value relative to θ_j.

After evaluating each neighboring policy R_i on the objective function F, each dimension of every R_i is grouped into one of three categories to estimate an average gradient for each dimension:

- $Avg_{-\epsilon,n}$ if the nth parameter of R_i is $\theta_{n-\epsilon_n}$
- $Avg_{+0,n}$ if the nth parameter of R_i is θ_{n+0}
- $Avg_{+\epsilon,n}$ if the nth parameter of R_i is $\theta_{n+\epsilon_n}$

These three averages enable the estimation of the benefit of altering the nth parameter by $+\epsilon_n, 0$, and $-\epsilon_n$. An adjustment vector A of size n is calculated where $A_n \in$

- 0 if $Avg_{+0,n} > Avg_{+\epsilon,n}$and $Avg_{+0,n} > Avg_{-\epsilon,n}$
- $Avg_{+\epsilon,n} - Avg_{-\epsilon,n}$ otherwise

A is normalized and then multiplied by a scalar step size $\eta = 2$ to offset small ϵ_j. Finally A is added to π, and the process is repeated for the next iteration. Figure 3 describes the pseudocode for the policy gradient algorithm.

4 Empirical Results

The policy gradient algorithm described above was implemented and run on the Aibo as seen in Figure 4. In order to evaluate a particular gait parameterization,

$\pi \leftarrow InitialPolicy$
while !done **do**
  $\{R_1, R_2, \ldots, R_t\} = t$ random perturbations of π
  evaluate($\{R_1, R_2, \ldots, R_t\}$)
  for $n = 1$ to N **do**
    $Avg_{+\epsilon,n} \leftarrow$ average score for all R_i that have
      a positive perturbation in dimension n
    $Avg_{+0,n} \leftarrow$ average score for all R_i that have a zero
      perturbation in dimension n
    $Avg_{-\epsilon,n} \leftarrow$ average score for all R_i that have a
      negative perturbation in dimension n
    if $Avg_{+0,n} > Avg_{+\epsilon,n}$ and $Avg_{+0,n} > Avg_{-\epsilon,n}$ **then**
      $A_n \leftarrow 0$
    else
      $A_n \leftarrow Avg_{+\epsilon,n} - Avg_{-\epsilon,n}$
    end if
  end for
  $A \leftarrow \frac{A}{|A|} * \eta$
  $\pi \leftarrow \pi + A$
end while

Fig. 3. During each iteration t policies are sampled around π to estimate the gradient, then π is moved by η in the direction that increases the objective function the greatest.

the Aibo was instructed to record various data while repeatedly walking back and forth between two landmarks.

In order to generate a gait that was both stable and fast, the learning algorithm had to be given an appropriate objective function. In previous work, the objective function was focused primarily on generating a fast gait. In this paper, since stability is desired, the objective function was modified. Figure 5 depicts the images a robot would see with a perfectly stable gait and with an unsteady gait. The image taken with the unstable gait is rotated and translated compared to the image taken with a stable gait.[3]

In order to find a stable gait, the original objective function (which was designed to optimize only for speed) was modified to include stability information. This modified objective function consists of four components:

1. M_t - The normalized time taken by the robot to walk between the two landmarks.
2. M_a - The normalized standard deviation (averaged over multiple trials) of the Aibo's three accelerometers
3. M_d - The normalized distance of the centroid of landmark from the center of an image.
4. M_θ - The normalized difference between the slope of landmark and the ideal slope (90^o)

[3] Videos of a fast gait and a stable gait from the perspective of the robot can be found at `http://www.cs.utexas.edu/~AustinVilla/?p=research/learned_walk`

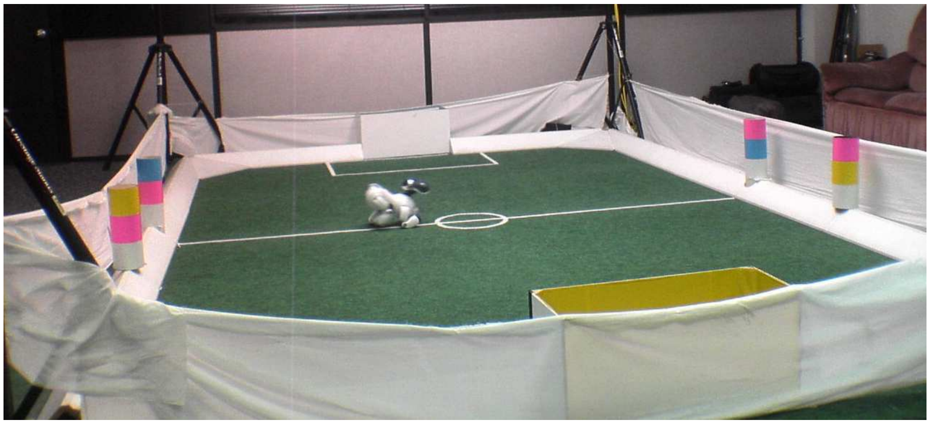


Fig. 4. The training environment during the gait parameter optimization experiment. The Aibo records how long it takes to move between two beacons. It also records the average accelerometer values, the average difference in the position of the centroid of the beacon and the center of the image, and the average slope of the beacon in the image.

These four components are combined to create a single objective function F:

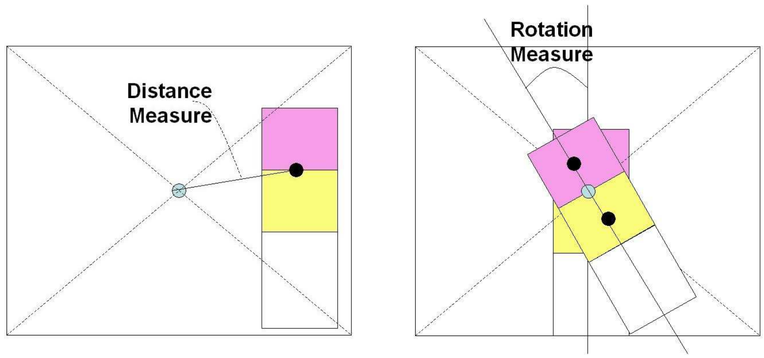
$$F = 1 - (W_t M_t + W_a M_a + W_d M_d + W_\theta M_\theta) \tag{1}$$

The different components of the objective function are weighted by W_t, W_a, W_d, and W_θ, respectively, to optimize for desirable attributes. These weights are constrained such that their sum is equal to one. For example, if stability is more important than speed, the time taken to walk between landmarks W_t can be assigned a smaller value than the other three weights. The next section describes experiments that compared different weightings of this objective function.

4.1 Learning a stable gait

The first experiment we performed was designed to determine how best to train for stability while learning a gait. To do this, we used two different parameterizations for weighting the subcomponents of the objective function. The first parameterization used $W_t = 0.4$, $W_a = 0.1$, $W_d = 0.4$ and $W_\theta = 0.1$, which weighted speed slightly more than stability. The second parameterization used $W_t = 0.3$, $W_a = 0.3$, $W_d = 0.2$ and $W_\theta = 0.2$, which more evenly weighted all four components.

We used a relatively slow hand-tuned gait as a starting point for the policy gradient algorithm, since previous work suggested that starting from a faster gait could hinder learning [2]. This starting point was determined empirically after trying several different starting gaits. Learning performance was somewhat sensitive to the initial parameter settings, but we did not extensively optimize the initial values.



(a) (b)

Fig. 5. Two visual clues that indicate an uneven gait. (a) shows the average displacement (M_d) of the centroid of the landmark with respect to the center of the image. (b) shows the average rotation (M_θ) of the landmark. If the camera is steady the average position difference should be zero and the average rotation should be 0^o.

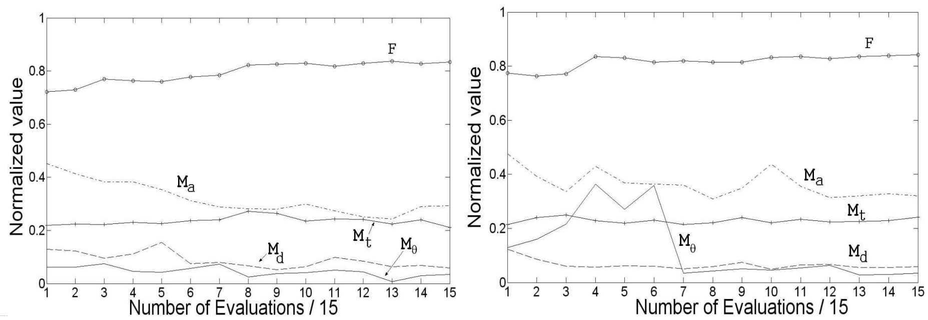
Figure 6 shows the progress of the policy gradient algorithm during training without head movements for the two different objective function parameterizations. The policy gradient algorithm generates 15 exploratory policies per iteration of the algorithm. In both parameterizations, the slope, distance and average acceleration measure of the objective function decrease considerably, while the time measure has a modest increase. This lead us to conclude that the weight parameters are not sensitive to smaller variations. Detailed results are shown in Table 1.

Table 1. Percentage reduction in the four objective function components for two different parameterizations without using compensatory head movements. In both cases the gait becomes more stable while only becoming slightly slower.

	Parameterization 1	Parameterization 2
M_t	-4.76	-4.5
M_a	34.7	32.6
M_d	60	57.14
M_θ	76.9	51.2

4.2 Adding compensatory head movements

The previous results successfully demonstrate the ability of our robots to learn a stable gait while minimizing speed reduction. However, in that case, all of the learning was focused on the leg motion. Since the stability objective measures the robot's head motion, we hypothesized that allowing the robot to make com-



(a) (b)

Fig. 6. (a) The overall fitness and fitness subcomponents (normalized to $[0, 1]$) for a single run with $W_t = 0.3$, $W_a = 0.3$, $W_d = 0.2$, and $W_\theta = 0.2$ without using head movements. The starting gait has an overall fitness of 0.72 and the final gait has an overall fitness of 0.83. (b) A similar plot, but with the parameters $W_t = 0.4$, $W_a = 0.1$, $W_d = 0.4$, and $W_\theta = 0.1$. The starting gait has an overall fitness of 0.78 and the final gait has an overall fitness of 0.83. Both speed and stability increase during learning.

pensatory head movements could effectively improve stability. To test this hypothesis, similar experiments to those described above were performed, but four additional parameters were added that governed compensatory head movements. For these experiments, the position of the landmark in the camera image was used to calculate the center of the ellipse that the Aibo's head moved through, and the tilt and pan angle limits and increments (set by the policy gradient algorithm) were used to calculate the length and height of the ellipse.

Figure 7 shows the progress of the policy gradient algorithm during training with head movements for two different objective function parameterizations. The policy gradient algorithm generated 19 exploratory policies per iteration of the algorithm. As in the experiment that learned a stable gait without head movements, the gait became more stable after learning. However, the results from this experiment were not as good as those from the previous experiment. This suggests that the addition of compensatory head movements does not significantly improve stability or speed.

Table 2 shows that gait parameters for the initial hand tuned gait, the final learned gait using head movements, the final learned gait without using head movements and the previously learned fast gait for comparison. The policy gradient algorithm was able to find a stable gait without much improvement in speed. These results demonstrate there is a tradeoff between gait speed and stability.

4.3 How useful is stability?

The main premise of this paper is that walk stability is an important feature for robot gaits. In particular, we hypothesized that stable gaits would improve the robot's visual capabilities. The vision algorithm used for this work converts

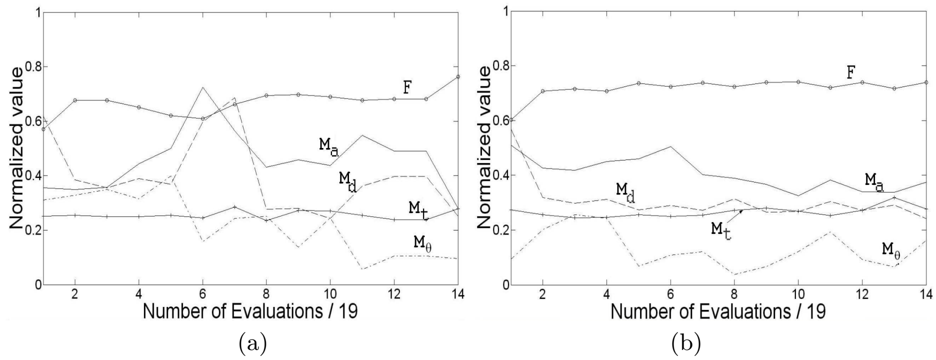


Fig. 7. (a) Scaled overall fitness and fitness subcomponents for a single run with $W_t = 0.3$, $W_a = 0.3$, $W_d = 0.2$, and $W_\theta = 0.2$ when compensatory head movements are enabled. The starting gait has an overall fitness of 0.57 and the final gait has an overall fitness of 0.76. (b) A similar graph, but with the parameters $W_t = 0.4$, $W_a = 0.1$, $W_d = 0.4$, and $W_\theta = 0.1$. The starting gait has an overall fitness of 0.60 and the final gait has an overall fitness of 0.74. In both cases, stability and speed increase, but the overall effect of compensatory head movements is negative.

each image received from the camera into a pixel-by-pixel color-labeled image, then groups regions of similarly-colored pixels into bounding boxes. A variety of heuristics such as size, tilt, and pixel density are used to convert these bounding boxes into high-level objects. If the robot is using an uneven gait, the camera will receive many images from an unexpected perspective, which can wreak havoc on the vision heuristics. The heuristics can always be improved, but this may take valuable processing time away from other components of the robot. Many vision algorithms employ such heuristics, making this a general problem for robotic vision [14, 15].

In order to test whether the stable gaits learned above actually help vision, we conducted two experiments where the Aibo traversed the field while recording the objects that it saw. The number of objects that were correctly classified (averaged over four runs) is shown in Table 3. Using the learned stable walk, the Aibo displayed 39% more true positives and 54% fewer false positives. These results with statistically significant with $p < 0.05$.

5 Discussion and Future Work

The experiments detailed in this paper demonstrate that there is a tradeoff between gait speed and stability. Our version of the fast gait learned according to Stone et al. [13] achieves a speed of 340mm/s. When the objective function is changed to include stability information, the fastest walk that is learned has a speed of 259mm/s. Allowing the robot to make compensatory head motions to counterbalance for the body movements, reduced the speed marginally.

Table 2. The parameterized starting gait, learned gaits with and without head movements and the learned fast gait. The policy gradient algorithm is able to find gaits that are considerably more stable than the learned fast gait while only sacrificing a small amount of speed. The final gait shows a small improvement in gait speed compared to the starting gait.

Parameter	Hand-tuned Gait	Stable gait	Stable gait with head movements	Fast gait
Front locus height	1.1	1.7	1.7	0.97
Front x position	-0.05	0.08	1.17	-0.04
Front y position	0.7	0.76	-0.08	0.3
Rear locus height	1.6	-0.45	1.54	1.61
Rear x position	0	1.54	1.7	-0.11
Rear y position	-0.4	0	0.66	-0.51
Locus length	0.4	0.5	0.68	0.57
Front body height	0.9	0.95	0.96	0.76
Rear body height	0.8	0.75	0.64	0.65
Time on ground	0.5	0.62	0.7	0.27
Time to move through locus	45	45.5	43.4	56
Tilt limit	n/a	n/a	4.93	n/a
Tilt increment	n/a	n/a	0.88	n/a
Pan limit	n/a	n/a	4.81	n/a
Pan increment	n/a	n/a	1.07	n/a
Gait speed	198 mm/s	259 mm/s	237 mm/s	340 mm/s

Table 3. The ratio of objects correctly and incorrectly classified by a vision algorithm using a learned fast gait and a learned stable gait. The stable gait leads to significantly ($p < 0.05$) better visual classification accuracy.

	True Positives	False Positives
Fast Gait	0.33	0.052
Stable Gait	0.46	0.028

Even though the stable gait is not as fast as gaits optimized for speed, it could be used in situations where it is important not to lose sight of objects, for example if the robot has the ball and is near the opponent's goal, the stable gait can be used to ensure that the robot does not lose the ball from its vision and thus has a better chance at scoring. We leave deciding which gait to use when to future work.

Another interesting avenue for future work is to examine how different parameterizations for the gait and the head motion affect learning. Although the elliptical head motion described in this paper did not significantly increase head stability, other types of head motions might do better.

6 Conclusion

This paper presented results on using the policy gradient algorithm to learn a stable, fast gait. Experiments were performed using an objective function that optimizes for stability in addition to using head compensatory movements. In both cases, the policy gradient algorithm found a stable gait while sacrificing only a small amount of speed. Videos of a comparison between gaits optimized for speed and gaits optimized for stability are available at:
`http://www.cs.utexas.edu/~AustinVilla/?p=research/learned_walk`.

A Acknowledgments

The authors thank the Austin Villa team for their support and guidance. This research was supported in part by NSF CAREER award IIS-0237699, ONR YIP award N00014-04-1-0545, and DARPA grant HR0011-04-1-0035.

References

1. Rofer, T.: Evolutionary gait-optimization using a fitness function based on proprioception. In: RoboCup 2004: Robot World Cup VIII. Lecture Notes in Artificial Intelligence, Springer (2005) 310–322
2. Kohl, N., Stone, P.: Machine learning for fast quadrupedal locomotion. In: The Nineteenth National Conference on Artificial Intelligence. (2004) 611–616
3. Kohl, N., Stone, P.: Policy gradient reinforcement learning for fast quadrupedal locomotion. In: Proceedings of the IEEE ICRA. (2004)
4. Kim, M., Uther, W.: Automatic gait optimisation for quadruped robots. In: Australasian Conference on Robotics and Automation. (2003)
5. Chernova, S., Veloso, M.: An evolutionary approach to gait learning for four-legged robots. In: In Proceedings of IROS'04. (2004)
6. Stronger, D., Stone, P.: A model-based approach to robot joint control. In Nardi, D., Riedmiller, M., Sammut, C., eds.: RoboCup-2004: Robot Soccer World Cup VIII. Springer Verlag, Berlin (2005) 297–309
7. Ng, A.Y., Coates, A., Diel, M., Ganapathi, V., Schulte1, J., Tse, B., Berger, E., Liang, E.: Autonomous helicopter flight via reinforcement learning. In: Advances in Neural Information Processing Systems 17, MIT Press (2004)
8. In, T.W., Vadakkepat, P.: Hybrid controller for biped gait generation. In: 2nd International Conference on Autonomous Robots and Agents. (2004)
9. Hornby, G.S., Fujita, M., Takamura, S., Yamamoto, T., O., H.: Autonomous evolution of gaits with the sony quadruped robot. In: Genetic and Evolutionary Computation Conference. Volume 2., Morgan Kaufmann (1999)
10. Duffert, U., Hoffmann, J.: Reliable and precise gait modeling for a quadruped robot. In: 9th International Workshop on RoboCup 2005 (to appear). Lecture Notes in Artificial Intelligence, Springer (2006)
11. Rofer, T.: Germanteam robocup 2005. Technical report (2005)
12. Sony: Aibo robot (2005)
13. Stone, P., et al.: The UT Austin Villa 2004 RoboCup four-legged team: Coming of age. Technical Report UT-AI-TR-04-313, The University of Texas at Austin, Department of Computer Sciences, AI Laboratory (2004)
14. Sumengen, B., Manjunath, B.S., Kenney, C.: Image segmentation using multiregion stability and edge strength. In: The IEEE International Conference on Image Processing (ICIP). (2003)
15. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(5) (2002) 603–619