Autonomous Evolution of Gaits with the Sony Quadruped Robot

G. S. Hornby* **M. Fujita** **S. Takamura** **T. Yamamoto** **O. Hanagata**
Group 1, D-21 Laboratory
Sony Corporation
6-7-35, Kitashinagawa, Shinagawa-ku
Tokyo, 141-0001
JAPAN
hornby, mfujita, takam, yamamoto, hana@pdp.crl.sony.co.jp
ph: 81-3-5448-5901

Abstract

A trend in robotics is towards legged robots. One of the issues with legged robots is the development of gaits. Typically gaits are developed manually. In this paper we report our results of autonomous evolution of dynamic gaits for the Sony Quadruped Robot. Fitness is determined using the robot's digital camera and infrared sensors. Using this system we evolve faster dynamic gaits than previously manually developed.

1 INTRODUCTION

In this paper we present an implementation of an autonomous evolutionary algorithm (EA) for developing locomotion gaits. All processing is handled by the robot's onboard computer and individuals are evaluated using the robot's sensors. Our implementation successfully evolves trot and pace gaits for our robot for which the pace gait significantly outperforms previous hand-developed gaits. In addition to achieving our desired goal of automatically developing gaits these results show that EAs can be used on real robots to evolve non-trivial behaviors.

A method to automatically create locomotion controllers is directly applicable for use with our own robot. Previously, we proposed OPEN-R[1] [Fujita & Kageyama, 1997], a set of standard definitions of interfaces for entertainment robot architecture. The most significant feature of OPEN-R is style flexibility. This flexibility allows users to reconfigure their own robots. The current implementation of OPEN-R components allows us to build both a wheel-based robot and a quadruped robot. For the current quadruped robot we have already developed a crawl gaiting pattern. Now it is easy for us to build a different quadruped robot with different mechanics and sensors based on OPEN-R. We would like a method for adapting existing locomotion controllers for new robots as well as being able to generate new gaits for our robot – a quadruped robot with 16 degrees of freedom (DOF) and various sensors, [Fujita & Kitano, 1998].

We also wish to show that the autonomous evolution of behaviors with complex, physical robots can be practical. Arguments against evolutionary robotics attack both evolution with real robots and evolution with a simulator, [Mataric & Cliff, 1996]. Recently, a methodology for developing simulators for evolution has been proposed and shown to be successful in 4 sets of experiments, [Jakobi, 1998]. But it is acknowledged that there are cases when this methodology will not work – such as when a high degree of accuracy is necessary. When this occurs it is desirable to be able to evolve with a physical robot.

Problems with using real robots are power, maintenance and time, [Mataric & Cliff, 1996]. Power can be supplied by a tether (as has been done with Kheperas as well as in our experiments) or a power floor, [Watson et al., 1998]. Maintenance is not a large problem when an evolutionary run is measured in hours and not days. In our case, interchangeable parts allowed us to easily replace malfunctioning legs with working ones. Time is not a problem when an evaluation can be done quickly. For example, evolving 100 individuals for 100 generations will take 16 hours if an evaluation takes 6s. This is not unreasonable. In contrast, evolution will take approximately 40 days if an evaluation takes 6 minutes. We propose using physical robots when accuracy cannot be readily simulated and evaluation times are short.

*Gregory Hornby is a Visiting Reseacher from Brandeis University, hornby@cs.brandeis.edu.

[1]OPEN-R is a trademark of Sony Corporation.

Previous autonomous evolution with actual robots has not evolved behaviors with comparably complex robots nor evolved controllers for tasks requiring precise control of actuators. Examples of autonomously evolved behaviors are: forward, backward and stopping behaviors with a wheeled robot in [Steels, 1994]; homing navigation with a Khepera in [Floreano & Mondada, 1996]; and pursuer-evader behaviors with Kheperas in [Floreano & Nolfi, 1998]. None of these behaviors would be particularly difficult to implement by hand nor would they be difficult to evolve in simulation (comparable behaviors have been successfully transferred from simulation to physical robot in [Jakobi, 1998]).

Our test problem is that of developing locomotion controllers for dynamic gaits. By their nature, dynamic gaits are sensitive behaviors for which building a simulator would be difficult and for which hand development of parameters has been difficult. The gaits evolved in our experiments outperform hand-developed gaits. These results show that complex, physical robots can be used for evolution and with these robots non-trivial behaviors can be evolved.

The rest of this document is organized as follows. Section 2 is a review of related work in evolving gaits for legged robots. Section 3 is a description of our robot and the locomotion module. Section 4 consists of a description of the evolutionary algorithm used for our evolution. In section 5 we describe the setup of our experiment and how the robot's sensors are used. Section 6 presents the results of our experiments. We discuss these results in section 7. Finally, section 8 is a conclusion of this work.

2 RELATED WORK

Development of locomotion gaits for legged robots is a problem that has been studied for over 20 years and is a research issue becoming more popular with different research groups (such as: Sony, [Fujita & Kitano, 1998]; Honda, [Hirai et al., 1998]; and University of Tokyo, [Buehler et al., 1998] and [Yamaguchi et al., 1998]). In this section we limit our review of related work to those evolving locomotion controllers for physical robots. First we describe three different gaits for a quadruped robot.

Of the gaits a quadruped robot is capable of three of the most common are the crawl, trot and pace gaits. A crawl gait consists of moving each leg in turn while the other three legs are used for support. This is a static gait and the robot's center of gravity can be kept inside the triangle of the three support legs. In contrast trot and pace are dynamic gaits where there is no such safe area for the robot's center of gravity. A trot gait consists of the matching two diagonal legs moving together whereas with a pace gait the two legs on the same side of the robot move together. Figure 1 shows the leg position for crawl, trot and pace over a complete swing cycle. Up indicates the specified leg is off the ground, down indicates the leg is on the ground.

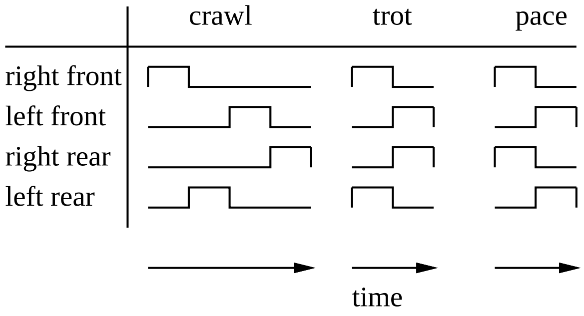

Figure 1: Quadruped Gaits

Most research in evolving gaits for physical, legged robots uses a simulator for the evolution. In [Gallagher & Beer, 1992] neural controllers were evolved for a simulated cockroach. Later, [Gallagher et al., 1996], these networks were transferred to a physical hexapod robot where they produced a smooth walking behavior. Again, in [Parker et al., 1997], a simulator was built for a 6-legged Stiquito II robot and a genetic algorithm was used to evolve locomotion controllers for it. Unlike other examples this robot used nitinol actuators to control the legs for which the controllers consisted of binary strings where each bit indicated whether or not to apply a voltage to a nitinol wire. An OCT1, an 8-legged robot (2-DOF/leg), was used in [Jakobi, 1998]. Neural controllers were evolved in simulation for obstacle avoidance as well as the locomotion gait. Performance on the physical robot was similar to that in the simulator and walking gaits were clearly observable. Finally, another example of using both simulator and physical robot is that of [Porta et al., 1998]. A quadruped robot and simulator were developed for evolving walking and other behaviors. As with the other examples controllers evolved in the simulator produced working walking-behaviors for the actual robot.

More similar toour work is that of using a real robot for the evolution. In [Lewis et al., 1992] ANNs were evolved as controllers to get a tripod gait for a 6-legged robot (2-DOF/leg). They used staged evolution to evolve first oscillators then to evolve a walking behavior. Fitness was determined by the experimenter. Cellular encoding was used in [Gruau & Quatramaran, 1996] to interactively evolve a controller for an OCT1. Again, the experimenter input the fitness of the individual and assisted the evolutionary process through staged evolution.

Before taking an evolutionary approach to the development of gaits our lab created gaits by hand. We developed a crawl gait of 5m/min and a fast-crawl gait (halfway between a crawl and a trot) of 6m/min. A pace gait was also developed but this was not very good and would at times move the robot backwards. A trot gait was not developed.

The significant differences between this work and those described above are that our evolution is completely autonomous and we evolve dynamic gaits. Autonomous evaluation involves coordinating different sensors – for determining location and measuring distance – which is a challenging problem on its own. Also, we compare the performance of our evolved controllers with those developed by hand and find that the evolved controllers are better. Finally, even though joint information is available, our locomotion module does not use sensor feedback. Evolved controllers must be able to recover from stumbles without being able to detect their occurrence.

3 ROBOT PLATFORM AND LOCOMOTION MODULE

the robot used for our experiments is the Sony Quadruped Robot. The head and each of the four legs has 3 degrees of freedom. There is also 1 DOF for the tail giving a total of 16 DOF. The body length (not including the head or tail) is approximately 18cm and the length of the leg (from shoulder to foot) is just under 12cm. Onboard the head is a micro-camera, stereo microphone, infrared sonar, and a touch sensor. There is also a touch sensor at the bottom of each leg. In addition to housing the CPU the body also houses a gyroscope and accelerometers. See [Fujita & Kitano, 1998] for a more detailed description of the robot.

Movement of the legs is controlled by a locomotion module. This module controls the robot's gait with a user-specified set of real-valued parameters. Every 20ms it updates the joint angles of the legs to the next position in their swing phase. This module also detects, and recovers from, the robot falling over.

To smoothen the movement of the robot we developed a variable gain control algorithm that varies the gain according to a schedule specified by three real-valued

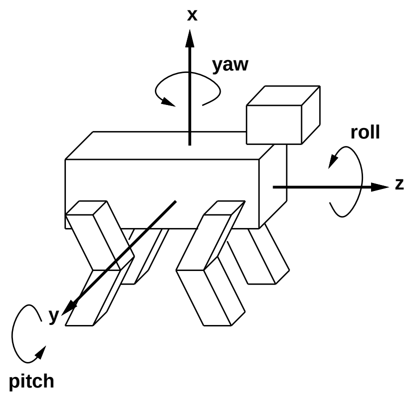

Figure 2: The Robot's (right-handed) Coordinate System

parameters. The first parameter specifies the minimum gain to use – the maximum value is fixed to the maximum possible. The second parameter, shift, specifies when in the swing cycle to start reducing the gain. The third parameter, length, is the duration over which the gain is reduced from the maximum to the specified minimum and then back to maximum following a sin wave:

$$gain = min + (max - min) * (1 - \sin(leg\ phase - shift)) \quad (1)$$

The leg phase starts at 0° swinging forward and up, starts swing back at 180° and ends at 360°. For example, with the pace gait, legs on the same side of the body have the same leg phase and legs on the opposite side of the body are 180° out of phase.

In total, there are twenty real-valued parameters are used to define a gait for the locomotion module. Table 1 lists these parameters, which are are also the genes for individuals evolved by the evolutionary algorithm. These parameters specify the position and orientation of the body, the swing path and rate of swinging of the legs, the amplitude of oscillation of the body's location and orientation, and how the gain varies during the course of a swing cycle for each leg.

4 EVOLUTIONARY ALGORITHM

The evolutionary algorithm is a steady-state EA running onboard the robot. This consists of an initialization phase followed by the evolution of the population. Parameters to the different phase of the EA were chosen based on values used by other researchers and our own experience. In this section we describe the initialization, selection and reproduction phases of the EA. Evaluation is described in the following section.

Table 1: Parameter List For A Gait

parameter	unit	initial range	best trot	best pace
body center x	mm.	85 - 95	82.7	89.2
body center z	mm.	-5 - 5	6.18	-2.05
body pitch	degrees	-5 - 5	-11.3	3.17
all legs y	mm.	5 - 25	10.6	10.0
front legs z	mm.	24 - 40	24.7	25.0
rear legs z	mm.	15 - 29	24.3	25.3
step length	n.a.	80 - 220	152	182
swing height	mm.	15 - 29	19.6	29.5
swing time	ms.	200 - 400	421	222
swing mult.	n.a.	1.5 - 2.5	2.42	1.69
switch time	ms.	500 - 900	799	617
ampl body x	mm.	-2 - 2	0.55	-0.39
ampl body y	mm.	0 - 20	10.3	5.09
ampl body z	mm.	-2 - 2	0.74	-1.27
ampl yaw	degrees	-2 - 2	-2.93	1.57
ampl pitch	degrees	-3 - 3	-0.44	3.68
ampl roll	degrees	-3 - 3	2.15	0.44
min. gain	n.a.	25 - 175	103	101
shift	degrees	60 - 120	64	125
length	degrees	90 - 150	117	103

The initial population is created with a uniform distribution over a given search range. Table 1 lists the twenty real-valued parameters used as genes and their initial search range. This initial range was determined from experience in hand developing gaits. Once individuals are created they are evaluated. With a dynamic gait many parameter configurations result in the robot falling over. To generate an initial population of non-falling individuals, sets of parameters in the initial population that cause the robot to fall are replaced with new, randomly generated individuals. When all individuals in the initial population are non-falling evolution begins.

A tournament selection is used to select individuals for parents and the individuals to be replaced. First the algorithm decides whether to perform recombination or mutation. Then a number of individuals is randomly selected to be in the tournament. For recombination, 3 individuals are randomly selected and for mutation 2 individuals are randomly selected. The parent(s) is the individual(s) with higher fitness, and the individual with the lowest fitness is replaced by the offspring of the parent(s).

Both mutation and recombination are used as variation operators, with an equal probability of selecting either mutation or recombination. Recombination takes two individuals as parents ($p1$ and $p2$) and creates one child individual (c). Each gene of the child is given a value according to the equation, $c_i = p1_i + \alpha_i(p1_i - p2_i)$. Here, c_i is the ith gene of the child individual; $p1_i$ and $p2_i$ are the ith gene of parents p1 and p2; and α_i is a random number in the range of -1 to 1. Mutation takes one parent individual and perturbs a few values in it by a small amount to generate a child individual. A random number with a Gaussian-like distribution is used to determine the number of genes (1 to 8) to mutate. The genes to be mutated are selected randomly (it is possible to select a gene more than once) and the mutated value is, $c_i = p_i + \delta i_{mutate}$; where δ is a uniform random value in the range of -1 to 1. Values for $v_{i,mutate}$ are set to 5% of a parameter's initial search range.

A problem experienced in initial experiments was that sometimes an individual would receive a significantly higher fitness than it deserved–such as through an inaccurate measure of distance from the infrared distance sensor. This resulted in pulling the search towards poor parameters. To reduce this problem an individual's age, the number of times it has been used as a parent, is stored. Age is incremented each time the individual is used as a parent for either recombination or mutation. When an individual reaches the age of 4 it is re-evaluated and its age is reset to 0.

5 EXPERIMENTAL METHOD

The desired result of our experiment is a set of parameters that moves the robot both quickly and in a straight line. In this section we describe our experiment and how individuals are evaluated.

Evolution takes place inside a pen (see figures 3 and 4). At each end of the pen there is a strip of colored cloth to mark the center of that end. Using its camera the robot turns until it is centered on one colored strip of cloth. Once centered, the robot measures its distance with its infrared sensor and proceeds to locomote for a fixed amount of time (7s for these experiments). The robot stops either at the end of this time or if it encounters a wall. Then the robot pans its head to find the color strip and measures its stopping distance. Using these two distances the robot scores the tested locomotion parameters by calculating its average speed during the trial. An individual's fitness is the average of three locomotion trials.

Centering the robot on the color strip is done by use of the Micro-Camera-Unit (MCU). For the MCU we have a dedicated LSI chip with 8 color detection tables (CDTs) for detecting colors within a given range at each pixel position. A CDT is used for each color

Figure 3: Picture Of The Experimental Environment

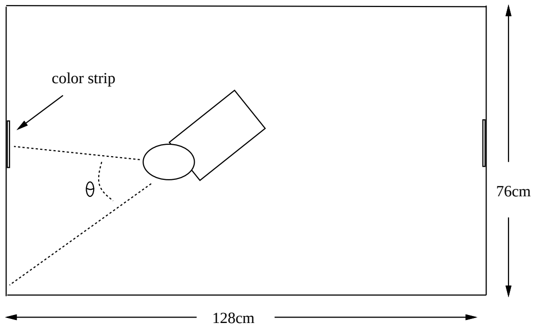

Figure 4: The Robot Pen

strip. For centering, the robot uses a turning gait to turn in place. The robot turns in a fixed direction until the desired color is detected. Once detected the robot finds the average horizontal location of all the pixels containing the desired color and converts this to an angle (the camera covers an angle of 52°). When the average location falls within ±6° of center for a period of 1.6 seconds the robot is centered.

Head panning uses the CDT in a way similar to the centering behavior. In this case, the robot's body remains fixed and the head turns. When the color strip is detected the robot calculates the offset from the current head angle and rotates its head to face the color strip.

Distances are measured using an infrared sensor located in the robot's head. The infrared sensor returns a value that must be converted to a distance. To create a function for this conversion the robot was placed in the pen at fixed distances from a color strip and the average of 200 readings was taken. Before a test of a set of parameters begins the robot tests its distance to the color strip to determine if it is within its reliable range. When the robot is further than its maximum reliable range (80cm) it uses a hand-built crawl gait to move closer. Closer than 10cm the infrared sensor readings become unreliable. When the robot comes within 20cm of a wall it is stopped, giving it 10cm in which to slow down and stop. This also prevents the robot from running into walls.

The infrared sensor is used as follows. Before taking readings the robot's body is moved to a normalized position. The start distance is determined by averaging seven consecutive infrared sensor readings. If the robot is more than 80cm from the target color strip it moves forward (using a hand-designed, crawl gait) until it is within 80cm. Then the robot moves for a specified amount of time (7 seconds) using a set of locomotion parameters and stops. If the robot has fallen (detected by the onboard accelerometers) the current individual is given a score of 0, then the robot gets up by itself (using a hand-coded behavior) and the next individual is tried. Otherwise, if the robot does not fall, the trial ends successfully and the robot pans its head until it finds the color strip. The stop distance is determined by averaging seven consecutive infrared sensor readings.

To simplify optimizing both velocity and straightness the score of a trial is the product of its velocity and straightness scores (averaged over three trials). Velocity, $v()$, is the average velocity of the robot during the trial. Straightness is a function of the angle between the robot's forward direction and the direction to the target color strip, θ, and the distance to the target strip, (see figure 4). Before calculating the straightness function, θ is converted to a 0-1 measure of offset by the function $f(\theta)$. The straightness function, $s()$, normalizes this value to account for the robot's distance from the color strip – with the robot at a fixed orientation θ will be larger when the robot is closer to the color strip. These functions are defined as:

$$score = v(d_{start}, d_{stop}, time) \times s(\theta, d_{stop}) \tag{2}$$

$$v(d_{start}, d_{stop}, time) = \frac{d_{start} - d_{stop}}{time} \tag{3}$$

$$s(\theta, d_{stop}) = \frac{d_{stop}(f(\theta) - 1) + 80 - 10f(\theta)}{70} \tag{4}$$

$$f(\theta) = 1 - \frac{|\theta|}{90°} \tag{5}$$

For the function $s()$, 80 and 10 are used as the constants because they are the maximum and minimum measurable distances. Table 2 lists values of $s(\theta, d_{stop})$ for different values of θ and d_{stop}. If the robot cannot find the color strip it is assumed that the robot's gait caused it to turn so sharply that it cannot pan its head far enough to face the color strip. In this case the individual receives a score of 0 for the trial, the same score it would receive if θ is $0°$. An individual's fitness, used in the selection phase, is the average score over three trials.

Table 2: Sample Values For $s(\theta, d_{stop})$.

θ	d_{stop}	$s(\theta, d_{stop})$
$\pm 90°$	80	0
$\pm 90°$	45	0.5
$\pm 90°$	10	1.0
$\pm 45°$	80	0.5
$\pm 45°$	45	0.75
$\pm 45°$	10	1
$\pm 0°$	80	1
$\pm 0°$	45	1
$\pm 0°$	10	1

Using the method described above we run two experiments. First we evolve a trot gait then we evolve a pace gait. The difference between the two experiments is in the time we set the legs to swing. For the trot gait we set the front-left and rear-right leg swing together at a 180° shift from the other two legs. With the pace gait we set legs on the same side of the robot to move together, with the left and right sides at a 180° shift of each other.

6 EXPERIMENTAL RESULTS

In evolving a trot gait we used a population size of 30 and ran for 21 generations. 30 random individuals were created to make the initial population of non-falling individuals. In the initial population most individuals did not move very well. Some moved backwards and the best, while moving as far as 26cm, were awkward. Individuals also had a tendency to walk in a curve. By the end of the evolution the individuals were propelling the robot more smoothly and almost straight. The best individual had a fitness score of 6.3 and moved 6.5m in a one minute trial. Figure 5 is a graph plotting the results of evolving a trot gait.

Figure 6 contains a graph showing the results for the evolution of a pace gait. Initial individuals for this gait are much less stable than those of the trot gait. It took 84 randomly generated individuals to create the initial population of 30, non-falling parameters. Like with the trot gait, the initial population had a couple of good individuals that moved quickly but did so awkwardly. Most individuals had a fitness less than 1.5. With this gait there was a large variance in performance between different trials of the same locomotion parameters. Averaging scores over three trials alleviated this problem, as did maintaining an individual's age and re-evaluating after 4 reproductions. After 11 generations of evolution the best individuals could move 10.2m/minute. Figure 7 contains a sequence of images of the best evolved pace gait with a $\frac{1}{15}$s interval between frames.

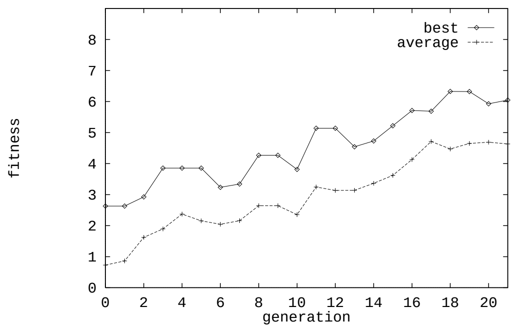

Figure 5: Trot Gait Results

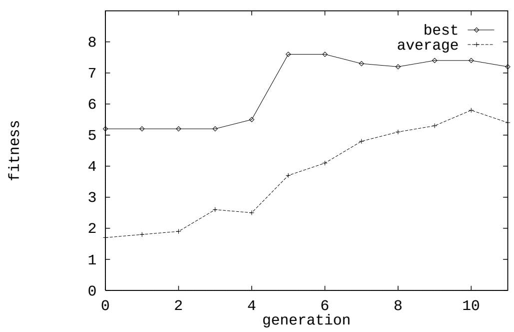

Figure 6: Pace Gait Results

In these experiments each generation took approximately one hour. The evolved parameters of the best individuals for the trot and pace gaits are listed in table 1.

7 DISCUSSION

Our first implementation of an evolutionary algorithm used the experimenter to enter the fitness scores in the same way as [Lewis et al., 1992] and [Gruau & Quatramaran, 1996]. In addition to using the distance moved by the robot an objective measure of aesthetics was used to adjust an individual's fitness. One finding was that parameters for a dynamic gait are sometimes erratic in performance, notably in the early stages of evolution. The same set of parameters may perform well in one evaluation and poorly in an-

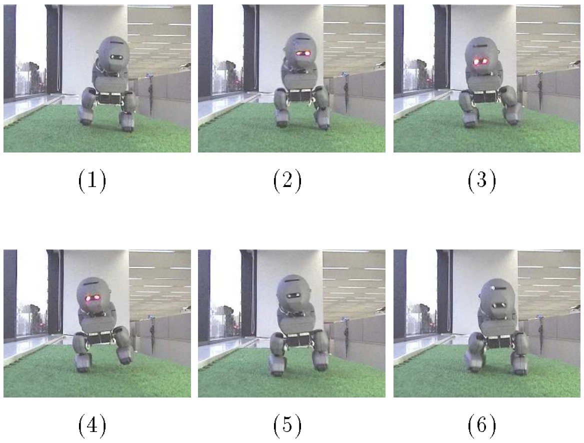

Figure 7: Example Of A Pace Gait

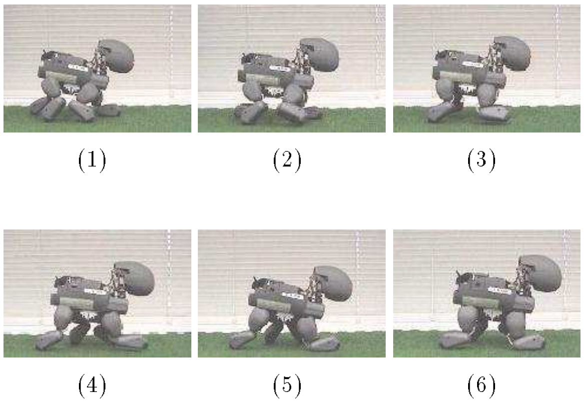

Figure 8: Example Of A Trot Gait

other. Averaging fitness over multiple trials resulted in the evolution of dynamic gaits with consistent performance. This finding was the reason for averaging performance over three trials with autonomous evolution. By hand a trot gait of 5m/min and a pace gait of 8.1m/min were evolved.

With autonomous evaluation of fitness our first evaluation metric used the same metric as our hand-evaluated algorithm: distance over a fixed amount of time. This evaluation metric became inaccurate as individuals improved. Often an individual would start its trial near the middle of the pen. This left it with approximately 50cm to the far wall. Good individuals could do this easily in less than allotted time. Consequently they were re-evaluated. If this happened twice in a row they were given a score of 0 for that trial. Thus very good individuals were frequently given low scores resulting in a low upper limit to what could be evolved (about 5m/min). Changing to variable time trials and scoring for average speed raised this upper limit.

Another factor limiting the maximum evolvable speed is variation in the starting angle of the robot. Initially we had planned to run the pace experiment for 20 generations but after 11 generations it appeared to cease improving. Differences in fitness seemed to be the result of the starting angle of the robot. After centering on the colored strip the robot spent 1s switching gait parameters to the current individual's. In doing so it would often turn slightly. Then the robot would be penalized through $s(\theta, d_{stop})$, sometimes large sometimes small, even though it ran straight. As a result the difference in fitness between the top individuals was mostly a matter of luck. With little selective pressure on these individuals the population ceased to improve.

Our experiments produced pace gaits that were faster than the trot gait. This was true both for manual evaluation and for automatic evaluation. A reason for this may be because of the hardware design. With the pace gait the robot was able use the shifting of its body to assist in lifting the forward-swinging legs off the ground. The trot gait has a forward-moving leg on each side of its body. Consequently clearance of the ground can be done only by moving the legs and is more difficult on our robot. A robot with a twisting torso may be able to twist its body to help lift the forward-moving legs off the ground and achieve as good performance with trot as with pace.

In fact, our evolved trot gaits are not truly dynamic. The gaits that did evolve would drag one of the forward moving legs along the ground. In one run of our experiments the robot angled its body forward and and semi-crawled forward, resting on its front legs. In another run it was the rear legs that dragged along the ground. Sometimes an individual would crawl with both front and rear legs sliding along the ground.

A dynamic trot gait is not stable on two legs and will eventually fall such that it uses a third leg for stability. By moving quickly, the legs on the ground keep switching and the robot does not have time to fall onto a third leg. In evolving a trot gait, early individuals do not move very fast and are usually resting on a third leg. If an individual always leans to one side it will tend to turn while moving and receive a large penalty through $s(\theta, d_{stop})$. Consistently resting on legs on both sides of the body removes this lopsided drag and allows the robot to move in a straight line.

From graph 6 it appears that the best pace parameters from the initial random population are almost as good as the best hand tailored controllers. This is not the case. Running the best individual from the initial population shows that over a period of 1 minute it does cover almost 5m (a little less than its fitness would indicate) it does not move smoothly or straight. It stutters, bounces in-place, and frequently turns. The average fitness of the population is a better indicator of performance.

Evolved parameters are both robust in some ways, yet not robust in others. By the end of evolution, an individual's offspring tended to be very similar to its parent(s). Successful individuals who were robust to mutation had successful offspring. These parameters flourished. Where a parameter was sensitive to a mutation its offspring were not successful and died out. Yet individuals were somewhat sensitive to their environment. With a different carpet, different leg calibration or different voltage an individual would often perform differently. Typically this consisted of more frequent stuttering in place and moving in a curve. It is possible that adding sensor feedback would reduce these affects. Regardless, this finding suggests that individuals are somewhat specialized to the environment they evolved in and to generate more general controllers individuals should be evaluated in different environments.

Evolving dynamic gaits was hard on the robots. In the four months of developing and testing the evolutionary algorithm frequent maintenance was necessary. The neck needed to be repaired three times, the CPU board was exchanged twice, the wires on the legs had to be replaced several times and more than a dozen rubber feet were used. When a simulator can be easily constructed and the problem does not require a fine degree of control the simulator is likely the better option.

8 CONCLUSION

In this paper we presented our work in the autonomous evolution of dynamic gaits. We evolved vectors of 20 real-value parameters for our locomotion module. The evolutionary algorithm for this was run onboard the robot. Using the robot's sensors the fitness for each individual evaluated without assistance by the experimenter. This algorithm successfully evolved both pace and trot gaits for our robot. Table 3 shows the fitness values for the best individual from the initial population and the best individual found as well as the actual speed of the best individual. The best pace gait moved the robot at approximately 10.2m/min, significantly better than the best hand developed gaits – 6m/min for a fast-crawl and unsuccessful development of a pace gait.

Table 3: Summary Of Results

	initial pop best fitness	overall best fitness	overall best speed
pace	5.2	7.6	10.2 m/min
trot	2.6	6.3	6.5 m/min

Previous work in evolution with real robots used simple robots and evolved controllers for non-brittle tasks. In addition, minimal simulations of these robots has been built and similar controllers have been successfully evolved with them. We used a more complex robot (16 DOF with various sensors) and evolved a sensitive behavior (dynamic locomotion on a quadruped without sensor feedback) achieving better results than hand-tailored controllers. These results show the feasibility of evolving low-level behaviors with real robots. In future work we plan to evolve high-level behaviors in simulation using models of low-level behaviors evolved with real robots.

Acknowledgements

The authors would like to thank Ronald Arkin, Dario Floreano, Jun Tani, and the members of Sony D-21, Group 1.

References

[Buehler et al., 1998] Buehler, M., Battaglia, R., Cocosco, A., Hawker, G., Sarkis, J., & Yamazaki, K¿ (1998). Scout: A simple quadruped that walks, climbs and runs. In *Proc. of International Conference on Robotics and Automation*, pp. 1707–1712.

[Floreano & Mondada, 1996] Floreano, Dario & Mondada, Francesco (1996). Evolution of homing navigation in a real mobile robot. *IEEE Transaction on Systems, Man and Cybernetics–Part B: Cybernetics*, 26(3):396–407.

[Floreano & Nolfi, 1998] Floreano, Dario & Nolfi, Stefano (1998). Competitive co-evolutionary robotics: From theory to practice. In Pfeifer, R. (Ed.), *From Animals to Animats V*. MIT Press.

[Fujita & Kageyama, 1997] Fujita, M. & Kageyama, K. (1997). An open architecture for robot entertainment. In del Rey, Marina (Ed.), *Proc. of First Intl. Conf. on Autonomous Agents*, pp. 234–239.

[Fujita & Kitano, 1998] Fujita, Masahiro & Kitano, Hiroaki (1998). Development of an autonomous quadruped robot for robot entertainment. *Autonomous Robotics*, 5:1–14.

[Gallagher & Beer, 1992] Gallagher, John C. & Beer, Randall D. (1992). A qualitative dynamical analysis of evolved locomotion controllers. In Meyer, J.-A., Roitblat, Herbert L., & Wilson, S. W. (Eds.), *From Animals to Animats 2*, pp. 71–80.

[Gallagher et al., 1996] Gallagher, J. C., Beer, R. D., Espenschied, K. S., & Quinn, R. D. (1996). Application of evolved locomotion controllers to a hexapod robot. *Robotics and Autonomous Systems*, 19(1):95–103.

[Gruau & Quatramaran, 1996] Gruau, Frédéric & Quatramaran, Kameel (1996). Cellular encoding for interactive evolutionary robotics. Technical Report 425, University of Sussex.

[Hirai et al., 1998] Hirai, K., Hirose, M., Haikawa, Y., & Takenaka, T. (1998). The development of honda humanoid robot. In *Proc. of Intl. Conf. on Robotics and Automation*, pp. 1321–1326.

[Jakobi, 1998] Jakobi, Nick (1998). *Minimal Simulations for Evolutionary Robotics*. PhD thesis, School of Cognitive and Computing Sciences, University of Sussex.

[Lewis et al., 1992] Lewis, M. Anthony, Fagg, Andrew H., & Solidum, Alan (1992). Genetic programming approach to the construction of a neural network for control of a walking robot. In *Proc. of Intrntl. Conf. on Robotics and Automation*, pp. 2618–2623.

[Mataric & Cliff, 1996] Mataric, Maja J. & Cliff, Dave (1996). Challenges in evolving controllers for physical robots. *Evolutionary Robotics*, 19(1):67–83.

[Parker et al., 1997] Parker, Gary B., Braun, David W., & Cyliax, Ingo (1997). Learning gaits for the stiquito. In Husbands, P. & Meyer, J-A (Eds.), *Proc. of 8th Intl. Conf. on Advanced Robotics*, pp. 285–290.

[Porta et al., 1998] Porta, J., Nicoud, J-D., Floreano, D., & Michel, O. (1998). Katzi, a four-legged evolutionary robot. Internal report, Laboratory of Micro-Processors and Interfaces, Swiss Federal Institute of Technology.

[Steels, 1994] Steels, Luc (1994). Emergent functionality in robotic agents through on-line evolution. In Brooks, R. & Maes, P. (Eds.), *Proc. of Fourth Workshop on Artificial Life*, pp. 8–14.

[Watson et al., 1998] Watson, R. A., Ficici, S. G., & Pollack, J. B. (1998). Embodied evolution. `http://www.cs.brandeis.edu/~richardw`.

[Yamaguchi et al., 1998] Yamaguchi, J., Inoue, S., Nishio, D., & Takanishi, A. (1998). Development of a bipedal humanoid robot having antagonistic driven joint and three dof trunk. In *1998 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pp. 96–101.