Autonomous Evolution of Dynamic Gaits with Two Quadruped Robots

Gregory S. Hornby*, Seichi Takamura, Takashi Yamamoto, and Masahiro Fujita
*corresponding author:
QSS Group Inc., Mail Stop 269-3
NASA Ames Research Center
Moffett Field, CA 94035-1000

***Abstract*— A challenging task that must be accomplished for every legged robot is creating the walking and running behaviors needed for it to move. In this paper we describe our system for autonomously evolving dynamic gaits on two of Sony's quadruped robots. Our evolutionary algorithm runs on board the robot and uses the robot's sensors to compute the quality of a gait without assistance from the experimenter. First we show the evolution of a pace and trot gait on the OPEN-R prototype robot. With the fastest gait, the robot moves at over 10m/min., which is more than forty body-lengths/min. While these first gaits are somewhat sensitive to the robot and environment in which they are evolved, we then show the evolution of robust dynamic gaits, one of which is used on the ERS-110, the first consumer version of AIBO.**

***Index Terms*— genetic algorithm, evolutionary algorithm, dynamic gaits, evolutionary robotics, legged robot, legged locomotion, quadruped robot**

I. Introduction

DEVELOPING locomotion controllers for legged robots is a problem that has been studied for over twenty years at a variety of different research groups (such as: Sony, [6]; Honda, [10]; and the University of Tokyo, [4] and [23]) and on a variety of different platforms (1-legged, 2-legged, 4-legged). In most projects, the gaits are static and programmed by hand (for surveys see [21] and [20]). Here we describe our work in the autonomous acquisition of dynamic gaits using an evolutionary algorithm (EA).

A system for automatically generating dynamic gaits is especially important for the entertainment robot industry for a number of reasons. When an entirely new entertainment robot is constructed it is necessary to produce gaits for it, and these new gaits need to be created quickly so that other members of the software development team can use the gaits in developing higher-level behaviors. Similarly, to create a personality for a robot it is useful to be able to continuously generate a variety of different styles of gaits for it. Also, in the process of going from a prototype to a consumer version of a robot, several different versions of a given robot are created. Each one of these prototypes has slightly different physical characteristics and a method for automating the tuning of pre-existing gaits to new versions of a robot is extremely useful. Finally, we are interested in an autonomous process for developing gaits as part of a gait-learning behavior to match the artificial maturing process that comes with the consumer version of our robots.

(a) (b)

Fig. 1. Sony's entertainment robots: (a) OPEN-R prototype; (b) ERS-110.

For the most part, previous work in the development of gaits for real robots has either required assistance from a source external to the robot or has focused on acquiring a static gait in simulation that was later transferred to the physical robot. On actual robots, EAs have been used to evolve neural network controllers for a six-legged robot [16] and an eight-legged robot [9]. In both cases the experimenter evaluated the performance of a gait by measuring the distance traveled and entering this result into the computer running the EA. One case of a robot evaluating itself is Genghis, a six-legged robot which learned a tripod gait [18]. There the learning algorithm used feedback from a wheel that was attached to the end of the robot to adjust the parameters of a behavior-based controller. More common is the evolution of controllers in simulation that are later transferred to the real robot: such as neural networks for a six-legged artificial cockroach [7], [8] and an eight-legged OCT1 [14], and a binary string of on/off flags for powering the nitinol actuators of a Stiquito II [19]. Similarly, reinforcement learning has been used to train neural networks for a biped robot first in simulation, and later to fine-tune the networks on the actual robot [3]. Finally, EAs have been used to evolve both the morphology and controller of crawling robots [17] and modular, walking robots [13] in simulation, with these robots then constructed and shown to work in reality. Yet evolution/learning in simulation is not always feasible because the level of fidelity may not be attainable (such as for actuators or pliable parts) or may require large amounts of computational power (such as fluid dynamics simulators for underwater robots).

G. S. Hornby is with QSS Group Inc. at NASA Ames Research Center.
S. Takamura and T. Yamamoto are with the Entertainment Robot Company, Sony Corporation.
M. Fujita is with the Intelligent Dynamics Laboratory, Sony Corporation.

In this paper we describe a system for the autonomous acquisition of dynamic gaits for two versions of Sony's entertainment robots. The system for developing gaits uses an EA to optimize a vector of parameters that specify a gait. All processing is handled by the robot's onboard processor and each set of gait parameters is evaluated using the robot's sensors. First we describe the evolution of pace and trot gaits with the OPEN-R prototype (figure 1.a), and then the development of a robust trot gait on its successor, the ERS-110, which is more commonly known as the first AIBO[1], (figure 1.b). Not only does our implementation successfully evolve dynamic gaits for both of our quadruped robots – in both cases evolving better gaits than were developed manually – but one of the evolved gaits is used in the first consumer version of AIBO.

The rest of this paper is organized as follows. In the next section we describe the two quadruped robots, the locomotion module, the evolutionary algorithm and the procedure by which a robot evaluates its performance. In sections III and IV we present and discuss the results of our experiments. The final section is a summary and conclusion of our work.

II. Method

The two robots used in these experiments are the OPEN-R prototype, which is the pre-AIBO entertainment robot, and the ERS-110, the first consumer version of Sony's entertainment robot AIBO. Gaits on both robots are controlled by the locomotion module, which uses a set of real-valued parameters to specify a gait. To autonomously acquire gaits an evolutionary algorithm optimizes gait parameters by sending sets of parameters to the locomotion module and then evaluating the resulting performance using the robot's onboard sensors. In this section we describe first the two robots and the locomotion module followed by a description of the evolutionary algorithm and the method for evaluating a set of gait parameters.

A. Robot Platform

Both robots used in these experiments have over a dozen degrees of freedom and various sensors. Fifteen degrees of freedom come from actuators in the head and the four legs, each of which has three degrees of freedom. The OPEN-R prototype (figure 1.a) has a tail with a single degree of freedom, giving it a total of sixteen, and the ERS-110 (figure 1.b) has a two-degree of freedom tail and two actuated ears, each with a single degree of freedom, giving it a total of nineteen degrees of freedom. Both robots have a micro-camera, stereo microphone, position sensitive device, and touch sensors located on the top of the head and on the bottom of each leg. The robots' body houses the CPU and battery as well as a gyroscope and accelerometers. See [6] for a more detailed description of the robots' hardware and software architectures.

[1]AIBO is a registered trademark of Sony Corporation.

B. Locomotion Module

The locomotion module is the software module that controls the movement of the robot's legs to perform different gaits. A gait is defined by a vector of real-valued parameters that are used by a mathematical function of sines and cosines to obtain a cyclic movement of the bottom of each leg. Using the dimensions of the robot's physical form and three kinds of coordinate systems – for the ground, the robot's body and one for the bottom of each leg – the desired joint angles for the three joints in each of the legs are calculated. The joints in each leg are controlled by a special ASIC chip and its current angle is sensed with a potentiometer. Every 8ms the software sets the target joint position and PID gain value for each joint based on the gait parameters and then the ASIC chip calculates the output to the motor to drive the joint to make its position error zero. This reduces the problem of developing a gait to that of finding a set of parameters for the locomotion module. In total, the locomotion module uses sixty-one real-valued parameters to define a gait.

1) The Locomotion Module on the OPEN-R Prototype: The first version of the evolutionary algorithm ran on the OPEN-R prototype and searched a space of twenty parameters by setting some of the sixty-one parameters for the locomotion module to fixed values (eg. setting body roll orientation to 0°) and using the same value for multiple parameters (eg. setting the swing time for each leg to be the same). These twenty parameters are listed in table I. They 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. With a set of parameters, the locomotion module moves the OPEN-R Prototype in any specified two-dimensional translation and rotation – although for our experiments we test the OPEN-R prototype only on its ability to move forward.

TABLE I
GAIT PARAMETERS FOR THE OPEN-R PROTOTYPE

parameter	unit	initial range	best trot	best pace
body center x	mm.	85 - 95	82.7	89.2
body center z	mm.	-5 - 5	6.2	-2.0
body pitch	degrees	-5 - 5	-11.3	3.2
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	2.4	1.7
switch time	ms.	500 - 900	799	617
ampl. body x	mm.	-2 - 2	0.6	-0.4
ampl. body y	mm.	0 - 20	10.3	5.1
ampl. body z	mm.	-2 - 2	0.7	-1.3
ampl. yaw	degrees	-2 - 2	-2.9	1.6
ampl. pitch	degrees	-3 - 3	-0.4	3.7
ampl. roll	degrees	-3 - 3	2.2	0.4
min. gain	n.a.	25 - 175	103	101
shift	degrees	60 - 120	64	125
length	degrees	90 - 150	117	103

Whereas the first seventeen parameters are used to specify the trajectories of each leg, the last three parameters are used to smoothen the movement of the robot by varying the gain of the leg motors during the movement cycle. The first of these last three parameters, *min. gain*, 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, at 180° it starts to swing backward, and at 360° it is back at the beginning of the cycle. 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.

2) The Locomotion Module on the ERS-110: Based on our experience from running experiments with the first set of parameters on the OPEN-R Prototype, a new set of parameters was used with the ERS-110 (table II). Gain variation was removed because the new locomotion module for the ERS-110 did not support it. Parameters were added to allow different trajectories for the front and rear legs as well as to add control of the body posture. Finally, instead of hard-coding the type of gait (crawl, pace or trot) the ability to evolve the type of gait was added by creating two parameters that control the relative swing times of the legs.

Relative phases for each leg are specified by a single value in the range of zero to one. This value specifies the point in the gait cycle when the start of a leg's movement cycle occurs. For example, a value of 0.0 indicates that the start of a leg's swing will occur at the start of the overall gait cycle and a value of 0.25 indicates that the start of that leg's swing cycle occurs one quarter of the way through the overall gait cycle. Unlike with the OPEN-R prototype, in which the time in the overall locomotion cycle when each leg would swing was fixed at predetermined values, with the ERS-110 two offset parameters are used to specify the relative phases of each leg. These parameters specify the offset between left and right legs, *L-R*, and fore and hind legs, *F-H*. The right foreleg is fixed to always start swinging at 0.0; the left foreleg starts swinging at L-R; the right hind-leg leg starts swinging at F-H; and the left hind-leg starts swinging at L-R + F-H (this value is adjusted to the range zero to one by subtracting 1.0 if the sum is greater than, or equal to, 1.0). Advantages to representing relative phases with these two parameters is that it allows for symmetries and is a smaller search space than using a separate start time for each leg. Table III displays the starting time for each leg for different types of gaits and F-R and R-L values for these gaits are shown in table IV.

TABLE II
GAIT PARAMETERS FOR THE ERS-110

parameter	unit	initial range	best gait
body center x	mm.	105 - 125	110.1
body center z	mm.	-10 - 10	-4.8
body pitch	degrees	-10 - 10	6.5
posture center x	mm.	0 - 20	-1.8
all legs y	mm.	-5 - 15	7.4
front legs z	mm.	10 - 30	21.5
rear legs z	mm.	-5 - 15	18.5
step length	n.a.	60 - 100	124.4
swing height front	mm.	25 - 45	36.5
swing height rear	mm.	25 - 45	44.1
swing time	ms.	460 - 540	503
swing mult.	n.a.	3 - 5	3.0
switch time	ms.	500 - 900	974
ampl. body x	mm.	-10 - 10	-1.8
ampl. body y	mm.	-25 - -5	-10.0
ampl. body z	mm.	-20 - 0	-13.6
ampl. yaw	degrees	-10 - 10	0.9
ampl. pitch	degrees	-10 - 10	-0.5
ampl. roll	degrees	-5 - 15	4.5
L-R	n.a.	0.25 - 0.5	0.48
F-H	n.a.	0.5 - 0.75	0.67

TABLE III
RELATIVE PHASES FOR DIFFERENT GAITS

Leg	Crawl	Trot	Pace	Skip
right foreleg	0.0	0.0	0.0	0.0
left foreleg	0.5	0.5	0.5	0.0
right hind-leg	0.75	0.5	0.0	0.5
left hind-leg	0.25	0.0	0.5	0.5

TABLE IV
PARAMETER VALUES FOR DIFFERENT GAITS

Parameter	Crawl	Trot	Pace	Skip
L-R	0.5	0.5	0.5	0.0
F-H	0.75	0.5	0.0	0.5

C. Evaluating a Set of Gait Parameters

Evolution takes place inside a walled area (figure 2) with a strip of colored cloth to mark the center of each end. Cables attached to the robot supply power and allow the robot to communicate data back to a host computer. Evaluating a set of gait parameters consists of locomoting with them and then measuring the straightness and distance traveled. The procedure by which a robot evaluates its own performance consists of three parts and was influenced by our experiences from manually evaluating sets of gait parameters.

The first part of an evaluation trial consists of the robot centering on the color strip. Objects are detected from the

(a) (b)

Fig. 2. The experimental environment: (a) OPEN-R Prototype; (b) ERS-110.

image returned by the onboard Micro-Camera-Unit (MCU) by using an LSI chip with eight color detection tables (CDTs). These CDTs detect colors within a user-specified range for each pixel location in the image. Using a CDT for each color strip, the robot centers by turning its body in a fixed direction in search of the desired color. Once detected the robot continues turning, reversing directions if necessary, until the average horizontal location of the color strip falls within $\pm 6°$ of the center of the image for a period of two seconds.

In the second part of an evaluation trial the robot determines how far it is from the color strip and then runs toward it. The distance to the color strip is measured using the robot's position sensitive device (PSD) sensor, which is located on the front part of its head. The start distance is determined by averaging seven consecutive PSD sensor readings. Since the reliable range over which the PSD sensor works is 10 to 80 centimeters, when the robot is further than its maximum reliable range it uses a hand-built crawl gait to move closer. To convert the value returned by the PSD sensor to a distance, a lookup table of distances was created by placing the robot at fixed distances from a color strip and taking the average of two-hundred readings. A sensor reading is converted to a distance by linearly interpolating the average sensor reading between the two nearest values in the distance look-up table. After the robot has determined how far it is from the color-strip it uses the set of locomotion parameters to move for seven seconds and then stops. The trial also ends if the robot detects that it has come within 20cm of a wall.

The third part of an evaluation trial begins after the robot has stopped moving and consists of the robot using its sensors to determine the straightness of its movement and the distance it traveled. If the robot has fallen (detected by the onboard accelerometers) the current individual is given a score of zero, then the robot gets up by itself (using a hand-coded behavior) and the next individual is tried. Otherwise, if the robot did not fall, the trial ends successfully and the robot pans its head until it finds the color strip. Head panning uses the MCU in a way similar to the centering behavior, only in this case the robot's body remains fixed and the head turns. Once the color strip is detected in the appropriate CDT, the robot calculates straightness based on the average horizontal location of the color strip and the current angle of the head and uses these values to compute the offset angle between its forward direction and the color strip. With the robot's head centered on the color strip, it uses its PSD sensor to find its distance to the color strip. The stop distance is determined by averaging seven consecutive PSD sensor readings. Using the starting and stopping distances from the color strip, as well as the time it traveled, the robot calculates its average speed.

To simplify optimizing both velocity and straightness the score of a trial is the product of its velocity and straightness scores. 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, (figure 3). 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 – since 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:

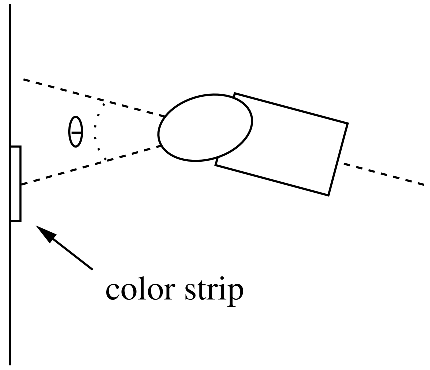


Fig. 3. Robot position at the end of an evaluation trial.

TABLE V
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

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

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

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

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

For the function $s()$, 80 and 10 are used as the constants because they are the PSD sensor's maximum and minimum measurable distances. Table V 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 zero for the trial, the same score it would receive if θ is greater than, or equal to, $90°$.

With early versions of our system the experimenter evaluated the performance of each gait and entered the fitness score in the same way as [16] and [9]. In performing these experiments it was noticed that a single trial with a particular set of parameters is a poor measure of the gait's quality since the same set of parameters can receive a moderately wide range of fitness scores. Consequently, to achieve a better measure of performance a single individual is evaluated with three trials and its fitness is the average of the three scores.

D. Evolutionary Algorithm

The search algorithm that we use for optimizing gait parameters is an evolutionary algorithm (EA). EAs are a family of population-based stochastic search algorithms that include genetic algorithms [11], evolutionary strategies [2], evolutionary programming [5] and genetic programming [15]. An EA operates by creating an initial population of candidate solutions, called *individuals*, which it optimizes by iteratively using better individuals to create new ones and discarding the poor individuals. Because they maintain a population of candidate solutions, EAs can be speeded up by evaluating individuals in parallel and are more robust to optimization in the presence of noise than optimization strategies such as hill climbing which only operate on a single candidate solution [1].

In these experiments each individual in the population is a set of gait parameters. Before optimization begins, an initial population of randomly generated gait parameters must be created. An individual in the initial population is created by setting each of its parameters to a random value with a uniform distribution over that parameter's minimum and maximum initial search range. These initial search ranges were determined from experience in developing gaits by hand and are listed for the OPEN-R Prototype in table I and for the ERS-110 in table II. To produce an initial population of minimally viable gait parameters, each new set of gait parameters is tested to determine whether or not it will cause the robot to fall over. If a set of gait parameters does cause the robot to fall over, it is replaced by another randomly generated individual. When all individuals in the initial population are non-falling, evolution begins.

The particular EA that we use is a steady-state evolutionary algorithm [22] with a population of thirty individuals. A steady-state EA works by iteratively selecting individuals from the population to act as parents and then using them to create a new individual. The number of individuals selected is determined by whether mutation or recombination is used to produce the new individual, with an equal probability of choosing either. If the mutation operator is used, then two individuals are selected from the population. The individual with the higher fitness value is chosen as the parent and its offspring replaces the other individual. If the recombination operator is used, then three individuals are selected from the population. The two individuals with higher fitness are the parents and their offspring replaces the individual with the lowest fitness.

Mutation and recombination work as follows. Mutation takes one parent individual and perturbs a few genes (one to eight, determined at random) by a small amount to generate a child individual. The genes to be mutated are selected randomly and the mutated value is, $c_i = p_i + \delta_i m_i$; where δ_i is a uniform random value in the range of -1 to 1. Values for m_i are set to 5% of the size of a parameter's initial search range. 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.

A problem experienced in initial experiments was that sometimes an individual would receive a significantly higher fitness score than it deserved, such as through an inaccurate measure of distance from the PSD sensor. This resulted in pulling the search toward 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 and when an individual reaches the age of four it is re-evaluated and its age is reset to zero. Since an individual is only replaced by another one with a higher fitness, this re-evaluation of an individual is the only way in which the the best fitness in the population can decrease.

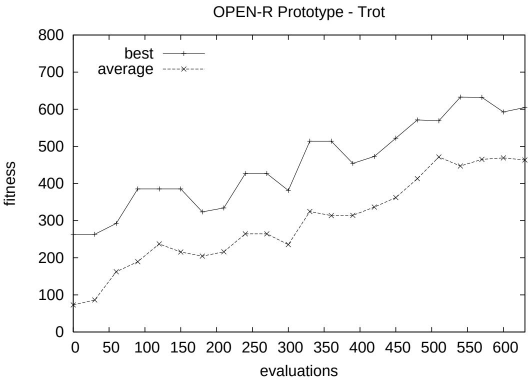
III. Evolving Pace and Trot Gaits with the OPEN-R Prototype

We performed two different experiments with our robots, first with the OPEN-R Prototype and then with the ERS-110. Since previous work to manually develop a dynamic gait for the OPEN-R Prototype met with poor results, our initial objective was to achieve either a pace or a trot gait through the use of an on board evolutionary algorithm. After successfully evolving both a pace and a trot gait with the OPEN-R Prototype, and with the launch of Sony's entertainment robot AIBO approaching, we became interested in evolving a dynamic gait that performed robustly on a variety of different surface types. In this section we describe the results of evolving gaits for the OPEN-R Prototype, and describe the results with the ERS-110 in the next section.

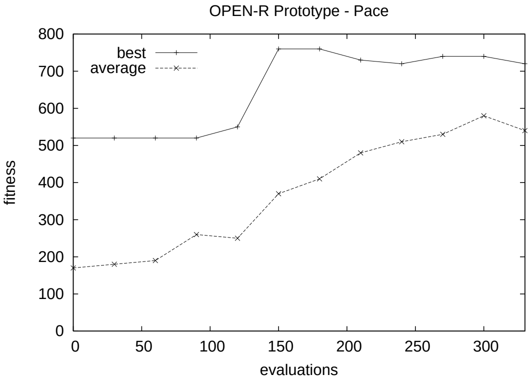
Before taking an evolutionary approach to the development of gaits our lab created gaits by hand. The two best hand-developed gaits were a crawl gait of 5m/min. and a fast-crawl gait (halfway between a crawl and a trot) of 6m/min. A pace gait was also developed by hand, but it was not very good and would at times move the robot backward, and we were not successful in developing a trot gait.

Unlike our attempts to hand-craft a dynamic gait for the OPEN-R Prototype, the evolutionary algorithm was able to evolve both a pace and a trot gait. Evaluating a single set of gait parameters with three trials takes approximately two minutes and good gaits are achieved after a couple hundred evaluations. Figure 4 shows the average and best fitness scores for individuals in the population for one run of both the trot and pace gaits. From these graphs it appears that the best pace parameters from the initial random population are almost as good as the best hand tailored controllers. In fact, running the best individual from the initial population shows that while the robot does move at almost 5m/min., a little less than its fitness score would indicate, it does not move smoothly but stutters, bounces in-place, and frequently turns. The average fitness score of the population is a better indicator of performance.

In evolving a trot gait, the first thirty individuals were all non-falling individuals and formed the initial population. Most of these individuals did not move well, with some moving backward and the best moving only 26cm in seven seconds.



(a)



(b)

Fig. 4. Results with the OPEN-R Prototype: (a) trot gait; and (b) a pace gait.

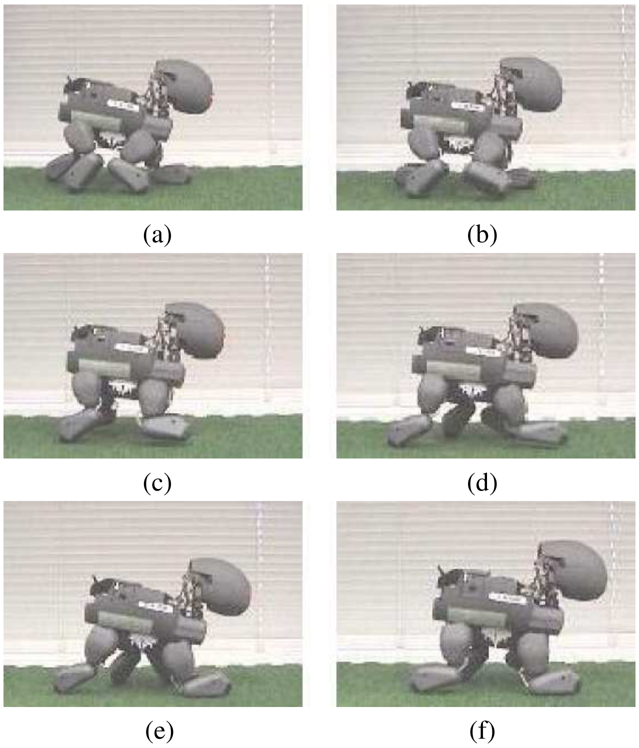


Fig. 5. A sequence of images showing the best evolved trot gait.

Early individuals tended to move too slowly to have a dynamic gait and moved in a curved path because a third leg was always dragging on the ground. Individuals which always leaned to one side would turn to that side while moving and had a poor straightness score. Over the course of evolution individuals in the population became better at alternating between resting on one side and then the other so that by the end of the evolution they were trotting in a nearly straight line. While the best individual that was evolved did not move in a true dynamic gait, it had a fitness score of 630 and moved 650cm in a one minute trial (approximately twenty-six body-lengths/min.). Figure 5 shows a sequence of images from one cycle of the best evolved trot gait and the evolved parameters for this individual are listed in table I. In this sequence the robot starts with its right legs at their closest point together and left legs furthest apart in (*a*), followed by the right legs moving apart and the left legs moving together in (*b*)-(*e*), and then reversing direction (*f*). A digital video of the best evolved trot gait is available online [12].

Unlike randomly generated trot gaits, which tended to be stable, randomly generated pace gaits tended to make the robot fall over and it took eighty-four randomly generated sets of gait parameters to create the initial population of thirty individuals. Like with the trot gait, the initial population had a couple of good individuals that moved quickly but did so awkwardly, and most individuals had a fitness score less than 150. After eleven generations of evolution (330 evaluations) the best individuals could move 1020cm/min., which is approximately forty body-lengths/min. At this point the maximum evolvable speed seemed to be limited by variation in the starting angle of the robot. After centering on the colored strip the robot spends one second switching gait parameters to those of the current individual. In doing so it would often turn slightly and then the robot would receive a poor straightness score even though it ran straight. As a result the difference in fitness scores between the top individuals was mostly a matter of luck. With little selective pressure on these individuals the population ceased to improve. Table I contains the parameter values for the best evolved pace gait and a sequence of images from one cycle of this gait are in figure 6. This sequence starts with the robot balanced on its left legs with its right legs off the ground (*a*) and then shifting its weight (*b-c*) until it is balanced on its right legs with its left legs off the ground (*d*), and then shifting its balance back to its left side (*e-f*). A video of this evolved pace gait is available online [12].

In this first set of experiments the evolutionary algorithm produced pace gaits that were faster than the best trot gaits. This may be because it is easier to perform a true pace gait with this hardware than a true trot gait. Unlike a pace gait, in which legs on the same side of the body move forward and backward together, the trot gait has legs on diagonally opposing sides of the body moving forward and backward together. Without a torso that can twist in the middle, it is more difficult to lift both forward-moving legs high enough off the ground in the trot gait than it is with a pace gait. This is supported by the results of our evolutionary system in which we evolved a truly dynamic pace gait, but the evolved trot gaits dragged a third leg along the ground.

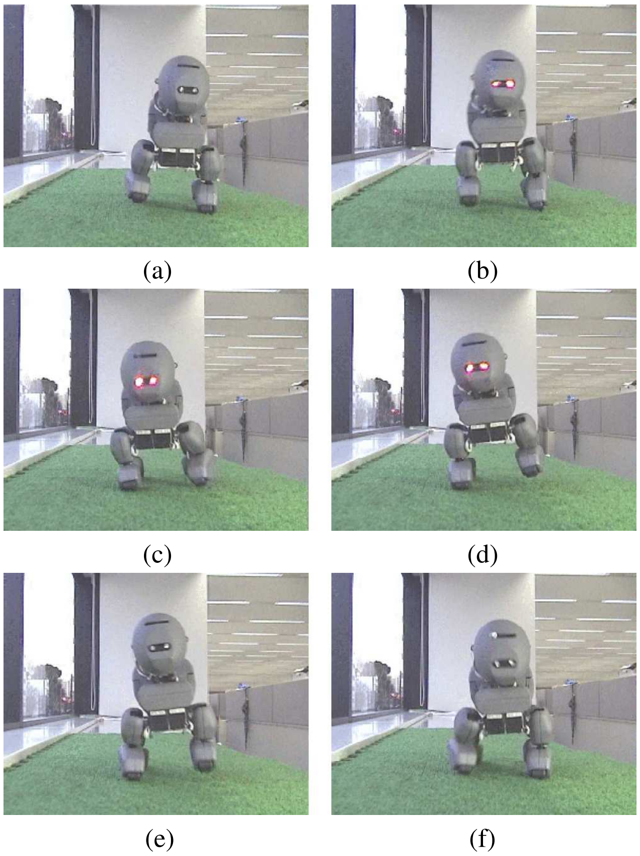


Fig. 6. A sequence of images showing the best evolved pace gait.

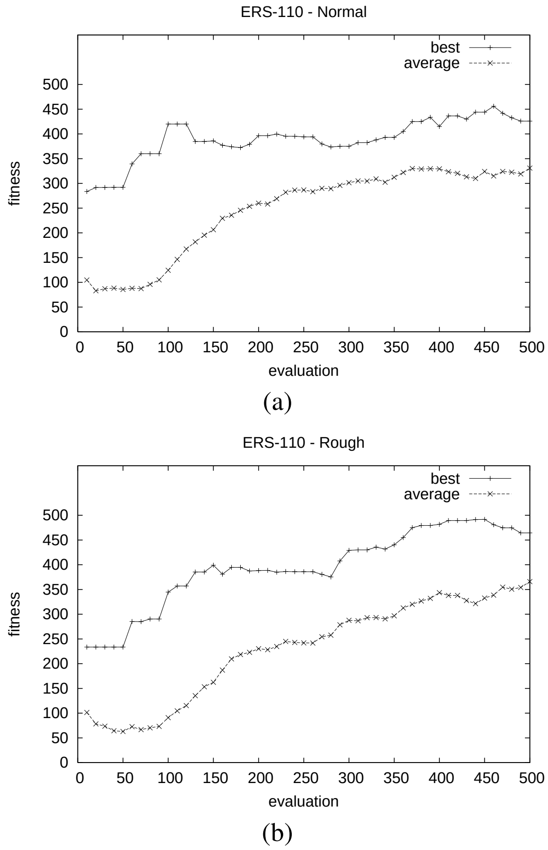


Fig. 7. Results with the ERS-110: (a) on a normal surface; and (b) on the obstructed surface.

IV. Evolving a Robust Dynamic Gait with the ERS-110

One shortcoming that we noticed with the results of the first set of experiments is that a particular set of gait parameters will often perform quite differently with different robots of the same type or on a different surface. This suggests that evolved individuals are somewhat specialized to the environment in which they are evolved. While adding sensor feedback might reduce these effects, sensor feedback was not incorporated into the locomotion module. This leads to our second set of experiments in which our objective is to evolve a set of gait parameters that are sufficiently robust to be used on the consumer version of AIBO.

One reason why gaits evolved on the OPEN-R Prototype were not very robust may be because of the environment in which the trials took place. Since the carpet on which trials took place is relatively smooth, evolved gaits were not adapted to rough surfaces: behaviors evolved for one environment will work on that environment and should work on easier environments, but will not necessarily work on harder environments. Similarly, switching robots is a form of changing the environment because of differences in manufacture and calibration. To achieve robust gaits that work well on a wide variety of environments, we hypothesized that evolution should take place in a difficult environment. In these second set of experiments, this hypothesis is tested by creating a more difficult environment for gaits to evolve and comparing gaits evolved in this environment to those evolved in the original environment.

To create a more difficult environment for evolution, plastic rods were placed on the ground perpendicular to the direction in which the ERS-110 would run. In setting up this environment we tried a number of different configurations of the plastic rods. When moving across this surface, the ERS-110 would often step on top of a rod and sometimes this would cause it to turn and change direction. As a result, this set of parameters would have a poor straightness score and a low fitness score even though it may have been a good individual. With too many rods the ERS-110 would step on rods too frequently and with too few rods, or with poorly placed rods, the ERS-110 would have little interaction with them and the rods would not influence evolution. The configuration we settled on uses six rods (as shown in figure 2.b) which requires both the front and rear legs of the ERS-110 to move over at least two rods in a typical trial.

Three runs of evolution on the obstructed carpet were performed and then compared against three runs of evolution on the normal carpeted surface. Each evolutionary run was for five-hundred evaluations, which took approximately twenty-five hours. In the early generations of these experiments, most individuals would drag their feet along the carpet. With the obstructed carpet, individuals with low steps would have a foot catch on a rod and fall over. These individuals would receive a low fitness score and be replaced by individuals with higher

(a)

(b)

(c)

Fig. 8. The gait sequence for an evolved gait on the ERS-110.

TABLE VI
EVOLVED GAITS TESTED ON DIFFERENT SURFACES

Surface	Regular Carpet	Obstructed Carpet
office carpet	387 cm/min.	600 cm/min.
Robocup	369 cm/min.	598 cm/min.
tatami	369 cm/min.	576 cm/min.
wood	387 cm/min.	443 cm/min.

fitness scores. By the end of the evolutionary run individuals evolved steps high enough to move over the rods. In contrast, individuals evolved on the unobstructed carpet received no such pressure to use high steps. Evolved gaits were halfway between a crawl and a trot gait. The graphs in figure 7 are plots of both the highest and average fitness scores of the population, averaged over the three trials. Figure 8 shows a sequence of images of an evolved trot-like gait. Here the robot starts with its right legs in the process of coming together (*a*) until they are at their closest point (*b*), and then extending until the right legs are furthest apart from each other (*c*). Videos of this experiment and the resulting gait are available online [12].

To determine if evolution on the obstructed carpet did result in a more robust gait, the best individual evolved for each surface type was then run using four different surface types and three different ERS-110s. Table VI lists the results of these trials, with each entry being the speed averaged over the three robots. On all four surfaces, the individual evolved on the obstructed carpet outperformed the individual evolved on the unobstructed carpet.

In the experiments presented so far, a large search space was used to create an initial population of random gait parameters. Once a good set of parameters has been found, these can be further refined by using them to seed a second evolutionary run. For this second evolutionary run, the initial population is created by adding small random values to each of the parameters of the seed individual so as to start with a population focused on a small part of the entire parameter space. Using one of the better individuals from an evolutionary run as the seed individual for further evolution, we evolved a trot gait on the rough surface that moves at 900cm/min. This gait is much faster than any of the individuals evolved in the first set of experiments with the ERS-110 and was sufficiently robust that it was approved by Sony's quality assurance department and is used on the consumer version of AIBO.

V. CONCLUSION

The primary interest in this project was to develop a system that would learn dynamic gaits for the OPEN-R Prototype, the ERS-110 and future robots with minimal human attention. The autonomous gait acquisition system described in this paper achieves this goal through the use of an evolutionary algorithm which runs on both robots and uses the robots' onboard sensors to evaluate its own performance. Using this system we evolved both pace and trot gaits on the OPEN-R Prototype, of which the fastest gait moved at over 10m/min., which is approximately forty body-lengths/min. One shortcoming we found with evolved gaits is their sensitivity to the specific robot or environment in which they were evolved. By roughening the floor of our experimental environment to create a more challenging surface we were able to evolve robust trot-like gaits on the ERS-110, one of which was approved by our quality assurance department and is included in the consumer version of AIBO.

A secondary interest of this project was to investigate the viability of an evolutionary algorithm running on a real robot. One limitation on our current system is that cables are used to transmit power to the robot and receive data from it. But this constraint is not necessary for future versions of our system since newer hardware for AIBO includes a wireless communication device and a base-station at which AIBO can dock to recharge its batteries. While the amount of time required to learn/evolve a task in real time with a real robot may preclude its use in some situations, a robot that evolves behaviors over time through its reactions in the real world has the potential to be a valuable developmental behavior for an entertainment robot.

ACKNOWLEDGEMENT

This work was conducted at Sony Digital Creatures Lab (DCL). The authors would like to thank J. Yokono and the

members of Sony DCL Group 1 and Sony ER. Images are copyright Sony Corporation.

REFERENCES

[1] D. V. Arnold and H.-G. Beyer. A comparison of evolution strategies with other direct search methods in the presence of noise. *Computational Optimization and Applications*, 24(1):135–159, 2003.

[2] T. Bäck, F. Hoffmeister, and H.-P. Schwefel. A survey of evolution strategies. In R. K. Belew and L. B. Booker, editors, *Proc. of the Fourth Intl. Conf. on Genetic Algorithms*, pages 2–9. Morgan Kaufmann, 1991.

[3] H. Benbrahim and J. A. Franklin. Biped dynamic walking using reinforcement learning. *Robotics and Autonomous Systems*, 22:283–302, 1997.

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

[5] L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence Through Simulated Evolution*. Wiley Publishing, New York, 1966.

[6] M. Fujita and H. Kitano. Development of an autonomous quadruped robot for robot entertainment. *Autonomous Robotics*, 5:1–14, 1998.

[7] J. C. Gallagher and R. D. Beer. A qualitative dynamical analysis of evolved locomotion controllers. In J.-A. Meyer, Herbert L. Roitblat, and S. W. Wilson, editors, *From Animals to Animats 2*, pages 71–80, 1992.

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

[9] F. Gruau and K. Quatramaran. Cellular encoding for interactive evolutionary robotics. Technical Report 425, University of Sussex, 1996.

[10] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of honda humanoid robot. In *Proc. of Intl. Conf. on Robotics and Automation*, pages 1321–1326, 1998.

[11] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, 1975.

[12] G. S. Hornby. Autonomous evolution of dynamic gaits. http://ic.arc.nasa.gov/ic/people/hornby/evo_gaits/evo_gaits.html.

[13] G. S. Hornby, H. Lipson, and J. B. Pollack. Generative representations for the automatic design of modular physical robots. *IEEE Transactions on Robotics and Automation*, 19(4):703–719, 2003.

[14] N. Jakobi. Running across the reality gap: Octopod locomotion evolved in a minimal simulation. In P. Husbands and J-A Meyer, editors, *Proceedings of Evorob 98*, 1998.

[15] J. R. Koza. *Genetic Programming: on the programming of computers by means of natural selection*. MIT Press, Cambridge, Mass., 1992.

[16] M. A. Lewis, A. H. Fagg, and A. Solidum. Genetic programming approach to the construction of a neural network for control of a walking robot. In *Proc. of Intl. Conf. on Robotics and Automation*, pages 2618–2623, May 1992.

[17] H. Lipson and J. B. Pollack. Automatic design and manufacture of robotic lifeforms. *Nature*, 406:974–978, 2000.

[18] P. Maes and R. A. Brooks. Learning to coordinate behaviors. In *National Conference on Artificial Intelligence*, pages 796–802, Menlo Park, CA, 1990. AAAI Press.

[19] G. B. Parker, D. W. Braun, and I. Cyliax. Learning gaits for the stiquito. In P. Husbands and J-A Meyer, editors, *Proc. of 8th Intl. Conf. on Advanced Robotics*, pages 285–290, 1997.

[20] C. Ridderstrom. Legged locomotion control — a literature survey. Technical Report TRITA-MMK 1999:27, Dept. of Machine Design, Royal Institute of Technology, S-100 44 Stockholm, Sweden, November 1999. ISSN 1400-1179.

[21] D. Wettergreen and C. Thorpe. Gait generation for legged robots. In *IEEE International Conference on Intelligent Robots and Systems.*, pages 1413–1420, 1992.

[22] D. Whitley. The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In J. David Schaffer, editor, *Proc. of the Third Intl. Conf. on Genetic Algorithms*, pages 116–121. Morgan Kaufmann, 1989.

[23] J. Yamaguchi, S. Inoue, D. Nishio, and A. Takanishi. 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*, pages 96–101, 1998.

Gregory S. Hornby graduated with a B.Sc. in Computer Science from the University of British Columbia in 1994, an M.Sc. in Computing Science from the University of Alberta in 1996 and received his Ph.D. in Computer Science from Brandeis University in 2002 for his work using Generative Representations in Evolutionary Design.

From 1998 to 1999 he was a visiting researcher at Sony's Digital Creatures Laboratory, where he evolved various behaviors for Sony's entertainment robots, such as the dynamic gait used on the first consumer version of AIBO. Currently he is a Computer Scientist with QSS Group Inc. working at the Computational Sciences Division of NASA Ames Research Center. His research consists of using evolutionary algorithms for optimization and design, such as designing antennas for spacecraft and controllers for robots.

Seiichi Takamura attended the Mechanical Engineering Division of Osaka University and received his B.Sc. in 1996 and M.Sc. in 1998.

He is now engaged in software development and design with Sony's Entertainment Robot "AIBO" at Sony Corporation. He is especially interested in image recognition and artificial intelligence.

Takashi Yamamoto attended Tohoku University and graduated with a B.Sc. in Mechanical Aeronautical Engineering in 1995 and an M.Sc. in Information Science in 1997.

Previously he worked in Technology Development Department 2, Entertainment Robot Company at Sony Corporation where he did research in dynamic control of robots for actions such as running and hopping. Currently he is working in the Design Section of the Camera Products Department of Sony Corporation Professional Solutions Network Company.

Masahiro Fujita received a B.A. degree in Electronics and Communications from Waseda University, Tokyo. He joined Sony Corporation in 1981 and worked on the development of a spread spectrum communication system, which was used for the global positioning system in a car navigation system. In 1988, he became a graduate student at the University of California, Irvine and recieved his M.S. in Electrical Engineering in 1989.

He started the Robot Entertainment project in 1993 and developed the entertainment robot AIBO, which came on sale in 1999. Currently he is a General Manager and Chief Researcher at the Intelligent Systems Laboratory, Information Technologies Laboratories, and a Research Director at the Life Dynamics Laboratory Preparatory Office in Sony Corporation. More recently he is in charge of development for the cognitive part of a small humanoid robot QRIO. His research interests include self-developmental systems, visual and audio perception, and emotional systems.