

# Evolving Dynamic Gaits on a Physical Robot

Viktor Zykov

Josh Bongard

Hod Lipson

Computational Synthesis Laboratory  
Sibley School of Mechanical and Aerospace Engineering  
Cornell University, Ithaca, NY 14850  
Email: [VZ25|JB382|HL274]@cornell.edu

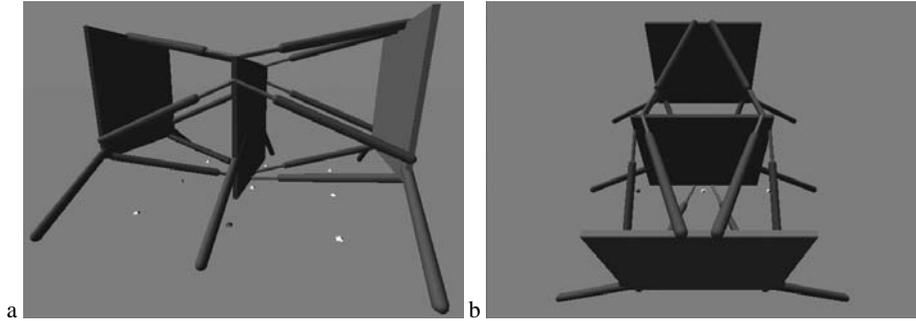
**Abstract.** Here we introduce a method for the evolution of dynamic gaits on a physical robot requiring no prior assumptions about the locomotion pattern beyond the fact that it should be rhythmic. The dynamic gaits were physically evolved in hardware using a parallel-actuated pneumatic robot. We have formulated a genetic algorithm that evolves open-loop controllers; the encoding allows evolution to shape both the speed and pattern of locomotion while ensuring rhythmicity. In future, we plan to evolve closed-loop controllers for the physical robot and integrate our previously developed methods to reduce the number of hardware trials.

## 1 Introduction

The generation of locomotory gaits for legged robots has received great attention in the robotics literature, beginning with the seminal work of Raibert *et al* in the early 1980s: their monopedal [16] and quadrupedal [17] robots were able to exhibit multiple dynamic gaits and were robust against perturbation. Since then, dynamic gaits have been produced for other robots ([10], [3], [11]). However in all of these projects, gait generation is either completely or partially hand-crafted. The closest approach to automated gait generation has been demonstrated in [22], in which a basic alternating tripod gait was set for a hexapedal robot, and the parameters of that gait were then automatically tuned using a machine learning approach.

A different approach to gait generation for legged robots has appeared in the evolutionary robotics literature (refer to [13] for an overview of the field), in which little or no assumptions are made *a priori* regarding the desired gait. However all attempts so far, except for one, have produced static gaits ([12], [9], [7], [1], [5], [15]). Hornby [8] evolved dynamic gaits for the commercial quadruped robot AIBO, but the type of dynamic gait—trot or pace—was fixed in advance and the parameters of that gait were then tuned using artificial evolution.

In this paper we describe the evolution of dynamic gaits for a complex, nine-legged robot (the Nonaped) with 12 actuated degrees of freedom, in which no assumptions are made beforehand regarding the desired gait pattern. Automated gait generation becomes increasingly important as the morphology of the robot increases in complexity. For example, in the field of modular robotics ([19]), no assumptions are made as to the topology of the robot’s body parts at any point in time, so gaits must be generated either on the fly or in response to changes in the robot’s body plan.



**Fig. 1. Two views of the simulated robot.**

Also, our robot is constructed from two serially-attached Stewart platforms (see Figures 1 and 2), in which no actuator drives only one leg, as is the case in most other robots developed to date. As robots become more complex, the relationship between actuators, legs and their overall dynamic behavior becomes increasingly subtle and thus hard to control using traditional analytic methods. In running bipeds, for example, actuators in the upper body are important for stabilizing the gait, even though such actuators are not directly attached to either leg. Thus pre-setting a basic gait before automatically tuning it becomes increasingly impractical for more complex robots, because it is not clear how the actuators should move relative to each other to produce the desired gait pattern. Thus our approach should prove more scalable than other approaches: as robots become more complex, a fully automated approach to behavior generation from scratch is required. This paper presents the first steps in this direction.

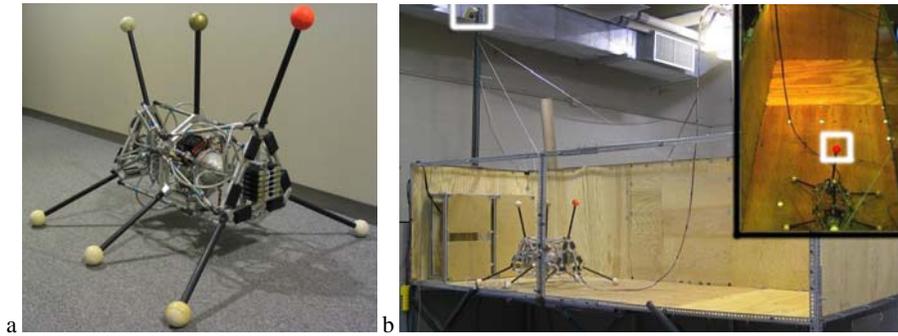
In the next section we describe the evolutionary algorithm we used to generate behavior, and the simulated and physical robot to which we applied it. In section 3 we describe the results of the experiments, and provide some analysis of the evolved gaits. In the final section we conclude and discuss future avenues of study.

## 2 Methods

In order to study the evolution of dynamic gaits from scratch for a complex robot, gaits were evolved for both a simulated and physical robot. The simulated robot is a simplified version of the physical robot, in that the topology and number of degrees of freedom are the same, but the mass distribution, actuator properties and friction coefficients with the ground were not necessarily identical.

### The Simulated Robot

Figure 1 shows a side and top view of the simulated robot. The robot is constructed of two serially-attached Stewart platforms, which allow for parallel actuation of the six legs. The robot is evolved in an open-source physical simulation package [14]. The simulated robot is composed of 27 parts: a front, middle and back plate each with a pair of legs, and 12 linear actuators composed of two rods each. During each time step of simulation the force applied by the actuators—along with the external forces



**Fig. 2. The physical robot and the experimental bay. a:** The physical robot. **b:** The experimental bay. Note the external camera used to compute the fitness of a gait. **Inset:** The view from the external camera. The fitness is determined as the displacement of the upper front foot.

(friction, gravity and momentum) acting on each of the robot's body parts—are used to update the positions, velocities and accelerations of the various body parts to produce behavior. The actuators are attached to the plates with ball and socket joints, and the rods are attached to each other with actuated linear joints.

The actuators can compress and extend up to 20% of the actuator's full length. At each time step, each actuator receives a pair of binary signals: if both values are zero, the joint is passive; if both values are one, the joint stays locked at the current extension; if the first value is one and the second is zero, a force is applied to push the rods apart; if the first value is zero and the second is one a force is applied to pull the rods together. The genetic algorithm described below is used to evolve a cyclical series of these commands for all 12 actuators in order to produce an open-loop controller for the simulated robot.

### The Physical Robot

The physical robot, and its experimental bay, are shown in Figure 2. The three sections of the physical robot are actuated relative to one another with 12 pneumatic pistons, joined to the sections with ball-and-socket joints so that the robot's kinematics correspond to two consecutively connected Stewart platforms. This structure allows the robot 12 non-articulated internal degrees of freedom. Nine legs are rigidly connected to the section frames and do not have any independent degrees of freedom. With respect to the world, the robot has three longitudinal and one transversal planes of symmetry and can achieve static mechanical equilibrium while resting on any of its five sides.

Each driving pneumatic piston is dual-actuated with two 3/2-way solenoid valves mounted with the manifolds on the edge sections of the robot. The solenoid valves are two-state and do not allow continuous control over the air flow rate. Each solenoid valve is controlled by a dedicated digital open collector output of a PC/104 input-output board operated by a PC/104 computer. Both the input-output board, the computer and a pneumatic power source are mounted together on a spring suspension inside of the central section of the robot. Electrical rechargeable batteries supplying power to the

electrical and electronic equipment of the robot are mounted on the robot's edge sections, however for the evolutionary experiments, an external power supply was used. Physical parameters and metrics of the robot are presented in Table 1.

Parameter	Value
Number of internal mechanical degrees of freedom	12
Overall dimensions, all pistons contracted (L/W/H), mm	724/724/635
Central section / Spring suspension weight, kg	5.988/2.064
Edge section / Actuator weight, kg	2.316/0.111
Total weight, kg	8.97
Theoretical actuator force at 6 bar: advance stroke / return stroke, N	47.1/39.6
Actuator stroke length, mm	100

**Table 1. Specifications for the physical robot.**

For conducting the evolutionary runs on the physical robot, an experimental setup was constructed as shown on Figure 2b. Its constituent parts are the cage (3.05m L x 1.22m W x 0.91m H) and the digital camera mounted above the cage with the viewing field shown in the insert of Figure 2b. The digital camera served for the estimation of the evolving gaits' fitness. It reported the position of the upper front ball of the robot before and after the implementation of each motion pattern. The fitness of the gait was computed as the displacement (in camera pixels) of this object. For each evaluation, the robot begins and ends by retracting all of its valves, and evaluation was conducted for five seconds. The timing of the gait was precisely controlled by the time-critical thread running on the robot's on-board controller.

The particular kinematic structure of the robot challenges the genetic algorithm to find a controller that dynamically operates a complex mechanical object characterized by multiple coupled state variables. Twelve internal degrees of freedom contribute 24 state variables to the robot description in terms of the state space. An additional six degrees of freedom relative to the external world require 12 more state variables for the proper characterization of the dynamical processes occurring during robot locomotion.

The problem of dynamic Stewart platform control is actively being investigated in the field of classical control theory. Dasgupta *et al* [4] present an inverse dynamic formulation for a Stewart platform manipulator using the Newton-Euler approach, allowing for the determination of the forces in the linear actuators, given the resulting trajectories. Forward kinematics problem for a class of hexapedal manipulators is addressed in [23]. The solution to this problem has also been sought using the displacements approach in [18], [21], and [20].

Devising a dynamic controller for the presented robot is more challenging than for a single Stewart platform because the two Stewart platforms in the robot are connected consecutively and neither of their sections is statically fixed. Moreover, the distribution of the robot's weight and foot thrust during its motion occurs dynamically, is intrinsically coupled with the mechanics of all parts of the double Stewart platform, and also depends on the frictional properties of the ground. For this reason we have elected to use a genetic algorithm for gait discovery, because it is near impossible to intuit what a

proper dynamic gait could be for such a configuration and then tune the parameters of such a gait. Rather we want to evolve a gait from scratch, with no designer bias beyond the fact that the gait should be rhythmic.

### The Genetic Algorithm

The same genetic algorithm is used to evolve dynamic gaits for both the simulated and physical robot. Each genome of the genetic algorithm encodes an open-loop controller, in which the only assumption made about the gait produced by the controller is that it should be rhythmic. This was achieved using a particular genome encoding in which evolution can modify the cycle period of the 12 actuators and the relative timings of all actuators, while every possible locomotion pattern and speed produces a rhythmic gait.

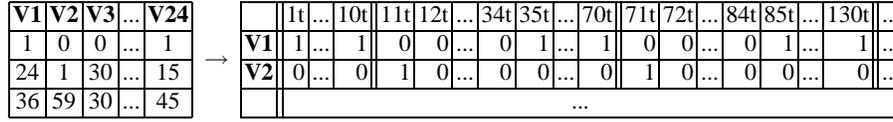
Each genome is represented as a  $3 \times 24$  integer matrix that encodes a specific gait with a particular step length (the time until the motion pattern of an actuator repeats): an example genome is shown in Figure 3. The first row is a binary vector, and corresponds to an initialization phase that occurs prior to evaluation of a particular genome. For the first 10 time steps of a gait evaluation (10 time steps in the simulator, or 10 25ms intervals on the physical robot), the 24 valves are turned on or off corresponding to these values, putting the robot into a starting configuration: for example the example genome turns valve 1 on and valve 2 off for the duration of the initialization phase. At the end of the initialization period, the state of each valve is inverted to initiate motion and begin the gait.

The lower two values of each column indicate the relative timing of each valve. The sum of each of these two values, in each column of a genome, sum to the same value (in the example genome this value is 60), indicating the total cycle period of each actuator. The two values indicate the time during the cycle when the valve changes state: in the example genome, valve 2 switches state after the first time step of each cycle, but valve 1 does not switch state until 24 time steps into the cycle.

Both the simulated and physical robot are evaluated for 200 time steps. In the simulation, each time step represents an integration step in which the new positions, orientations, velocities and accelerations of the robot's body parts are computed based on the forces acting on them. For the physical robot, a time step represents a 25ms period, giving a total evaluation period of  $0.025 \times 200 = 5$  seconds. Each set of 24 valve commands are sent to the robot at the beginning of each time step, and the set of valve commands repeats when the cycle period for that particular gait has elapsed. Note that if a cycle period does not divide evenly into 200, the robot may not terminate part way through a motion pattern. In this way the cycle period is independent of the total evaluation period: evolved genomes prescribing short cycle periods will produce gaits with shorter strides that are repeated more often compared to another gait with a longer cycle period that produces fewer but longer strides.

For each evolutionary run, an initial population of 30 random genomes are generated. Then a cycle period between 10 and 100 time steps is selected (denoted by  $c$ ), and the second row is then filled in with random integers in  $[1, c - 1]$ , and the values of the third row are then set to  $c$  minus the corresponding value in the second row.

Each genome is then evaluated in turn, either on the simulated or physical robot. The resulting fitness is assigned to the genome, where fitness is denoted as either: the forward displacement achieved by the simulated robot after 200 time steps; or the dis-



**Fig. 3. A sample genome and the corresponding expanded valve commands.** Each genome is encoded as an integer matrix, which at the time of evaluation is expanded into a set of 24 valve commands that repeat with a regular period over the length of the evaluation.

tance between the red marker on the top front foot of the physical robot, as measured by two images taken by an external camera before and after evaluation.

When all genomes are evaluated, selection, crossover and mutation is applied. During selection, a pair of genomes is selected at random, and with a 50% probability, the genomes are crossed; otherwise, a copy of the genome with higher fitness replaces the genome with lower fitness. Crossover is accomplished by randomly selecting a column, and all values to the left of, and including the column are copied from the first to the second genome, and the righthand columns are copied from the second to the first genome. The incoming columns are scaled to the cycle period of the host genome.

If a selected pair is not crossed, the genome with the higher fitness replaces the one with lower fitness, and the copied genome undergoes mutation. There are four possible mutation events, each with an equiprobable chance of occurring. Each copied genome undergoes from one to three of these mutations. The first two mutations increase or decrease the cycle period by two time steps, or replace it with a random cycle period in [10, 100]; the third mutation flips one of the bits in the top row; and the last mutation chooses a column at random, and either moves the state change of that valve either one time step earlier or later, or chooses a new state change time from [1,  $c - 1$ ].

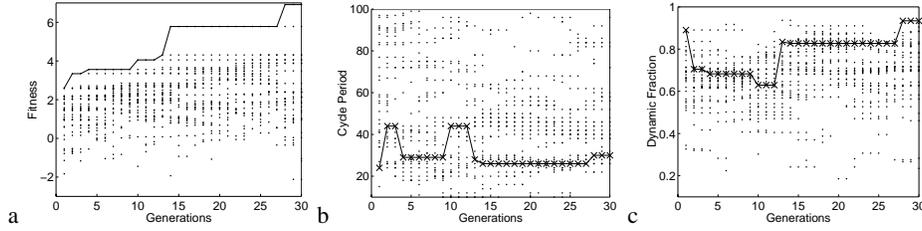
At the end of each generation a total of 22 pairs of genomes are chosen, in turn, to undergo either crossover or mutation. The top five genomes at each generation are shielded against mutation and replacement, ensuring good solutions are not lost. The crossover and mutation parameters were tuned using test evolutionary runs on both the simulated and physical robots. Each evolutionary run on both the simulated and physical robot was conducted for 30 generations.

### 3 Results And Analysis

#### Evolution on the Simulated Robot

Fifty independent evolutionary runs were conducted on the simulated robot, in which each gait was allowed to proceed for 210 time steps (10 initialization time and 200 evaluation time). The evolutionary history of a typical run, along with various properties of each gait, are outlined in Figure 4.

Figure 4c indicates the relationship between the speed of an evolved gait, and the fraction of time for which the robot is statically unstable. To achieve static stability, the projection of the center of mass of the robot must lie within the polygon of support formed by the points of contact of the robot's feet and the ground [6]. Whenever, during its locomotion, this condition is not met, it cannot be characterized as statically stable, and the robot is considered to be undergoing the dynamic transient process.



**Fig. 4. Progress of a typical evolutionary run for the simulated robot.** **a:** The fitness achieved by each evaluated genome. The best genomes are connected by the line. **b:** The cycle period of each genome. Cycle periods are measured in time steps. The cycle periods of the best genome at each generation are connected by the line. **c:** The dynamic fraction (see Equation 1) of each genome; the dynamic fraction of the best genome at each generation are connected by the line.

Because there are no restrictions in our evolutionary algorithm as to the type of gait produced, some gaits may be more dynamic than others. In order to quantify the amount of dynamism in a given gait we have formulated a metric called the dynamic fraction, given as

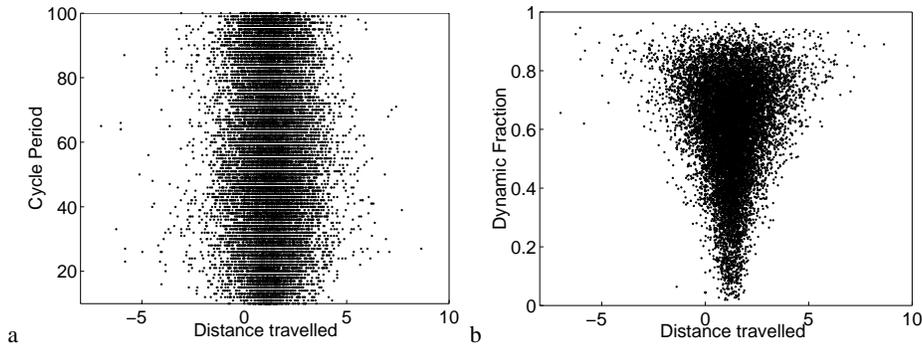
$$d = \frac{\sum_{i=1}^{400} d_i}{400}, \quad (1)$$

where  $d_i = 1$  if at time step  $i$  the robot is statically unstable, and  $d_i = 0$  if it is statically stable. The robot is statically unstable when any of the following conditions are met: there are two or fewer feet on the ground; only the left three or the right three feet are on the ground; or only three feet forming a corner (ie., the two front feet and the middle left foot) are on the ground. In all other cases, the robot is statically stable. The dynamic fraction then indicates the fraction of the evaluation period for which the robot is in a dynamic state (ie., is statically unstable). Highly dynamic gaits such as running and bounding exhibit large periods of static instability. As can be seen in Figure 4c, the highly fit gaits tend to obtain high dynamic fractions, and are thus highly dynamic.

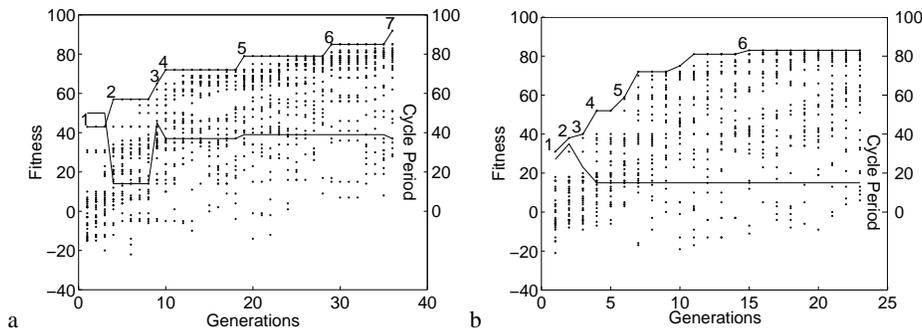
Figure 5 reports the cycle periods and dynamic fractions for every gait from all 50 runs using the simulated robot. As can clearly be seen, highly fit gaits exhibit a wide range of cycle periods, but fit gaits tend to possess higher dynamic fractions than less fit gaits. Gaits that do not move the robot a significant distance forward (or backward) exhibit a wide range of dynamic fractions: those with low dynamic fractions move a little or are completely still; and those with high dynamic fractions thrash wildly in place without producing systematic forward motion.

### Evolution on the Physical Robot

Two independent runs on the physical robot were then conducted. Both runs used the same genetic algorithm as was used for the simulated robot, with two modifications. The first modification was that fitness was not measured directly in meters, but rather in the dimensionless displacement of an object (a red ball placed on the upper front leg of the robot, see Figure 2b inset) inferred from two images taken by the external camera before and after evaluation. The second modification was that the first run on



**Fig. 5. Properties of the gaits evolved on the simulated robot.** The cycle period and dynamic fraction for every gait produced by the 50 evolutionary runs was computed. **a:** The cycle period of each gait plotted against its fitness (total displacement, measured in meters). **b:** The dynamic fraction for each gait plotted against fitness.

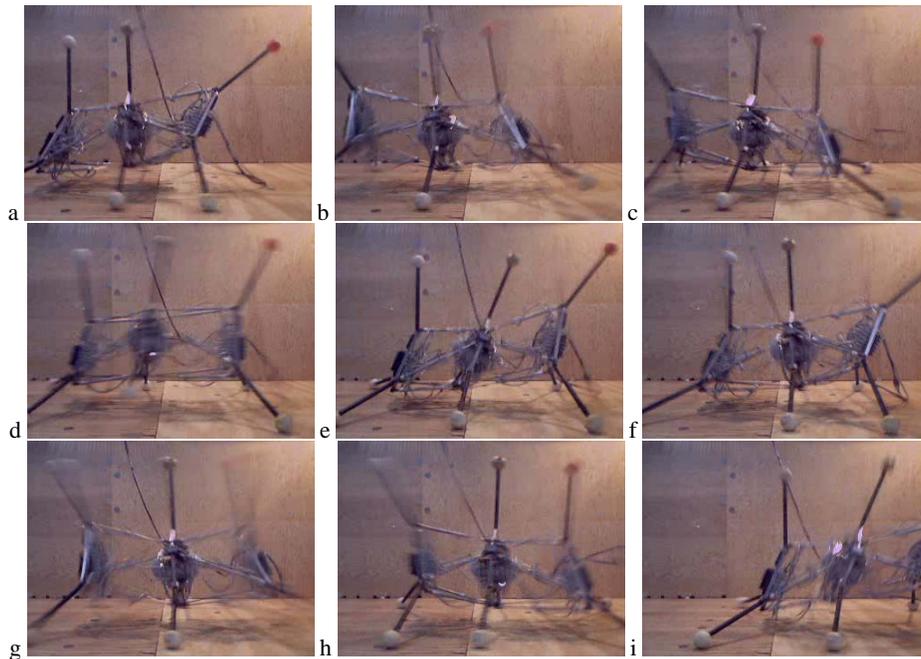


**Fig. 6. Evolutionary progress of two runs conducted on the physical robot.** The upper line shows the fitness of the best genome at each generation. The lower line indicates the cycle period of the most fit genome at each generation. **a:** The first run. **b:** The second run. The footprint patterns produced by the numbered gaits are plotted in Figure 8.

the physical robot was conducted for 36 instead of 30 runs, because a new best gait was discovered in generation 29. The second run was conducted for only 23 generations, as evolution seemed to have reached a local optima around generation 15. The first run took  $2\frac{1}{2}$  hours to complete, required a total of 750 hardware trials; the second run took 2 hours to complete, required a total of 490 hardware trials. The results from both runs on the physical robot are reported in Figure 6.

Figure 7 shows the progress of the best gait evolved during the first run on the physical robot. The gait has two distinct stages. During the first stage, the robot pushes with its two middle legs and thrusts its rear and front sections forward. During the second stage, the robot, relatively slowly, repositions the middle section into an initial pre-thrust configuration, resting on the four legs of the front and rear sections.

The robot initiates movement by leaning its front and rear sections forward simultaneously while pulling the two feet of its central section forward. At the completion



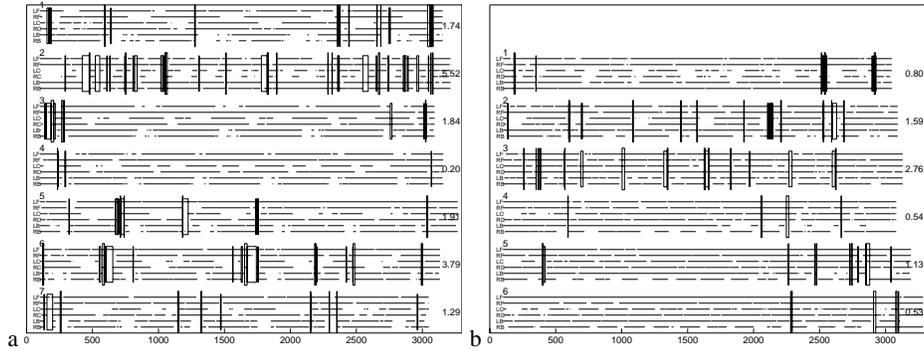
**Fig. 7.** Two consecutive cycles of the fastest moving gait evolved during the first run on the physical robot. (Gait 7 in Figure 6a). This gait achieves a speed of 28cm/sec.

of this move the driving actuators reach their extreme amplitudes: the upper front and the lower rear pneumatic pistons are fully extended; the lower front and the upper rear pneumatic pistons are fully retracted (see Figure 7a).

Next, using the inertia accumulated during the first phase of the movement, the robot thrusts its rear and front sections forward, resting mainly on the two feet of its central section. The front section is lifted in the air and the rear section is pulled behind the robot over the ground (Figure 7b). As shown in Figure 7c, this stage of the gait ends when the actuators approach their extreme amplitude: the upper front and lower rear pistons retract in a counter phase with the extension of the lower front and the upper rear cylinders. During this stage, most of the robot's weight rests on the two feet of its central section. As can be better seen in Figure 7g, at that stage of the gait, the front and rear feet are moving relative to the ground. This is the most dynamic stage of the gait, as most of the weight of the robot is resting on the central two legs.

Then, as shown in Figures 7d and e, the robot's rear upper and lower cylinders retract while its front upper and lower cylinders extend. This position is not appropriate for locomotion continuation unless a new set of resting points is established. This is done during the last phase of the gait, when the central section of the robot is brought into the upright position by extending the rear lower and middle actuators and retracting the front rear actuator, as can be seen in 7f, repeating the position seen in 7a. Figures 7g-i show the next gait cycle.

We then attempted to measure the dynamic fraction for the seven best evolved gaits from the first run (Figure 6a), and the six best evolved gaits from the second run (Figure



**Fig. 8. Dynamic properties of the best evolved gaits obtained from the physical robot. a:** The footprint patterns for the seven best gaits from the first evolutionary run. **b:** The footprint patterns for the six best gaits from the first evolutionary run. Each band is composed of six lines corresponding to the six feet of the robot. Each horizontal line indicates the time period for when that foot was on the ground; blank areas indicate when the foot was off the ground (from top to bottom: LF=left front; RF=right front; LC=left center; RC=right center; LB=left back; RB=right back). The numbers in the upper left of each band indicate which gait is plotted (see Figure 6; improvement goes from top to bottom). The boxes indicate periods of static instability; the short boxes indicate times for which two or fewer feet are on the ground; taller boxes indicate times for which only triplets of feet that cause static instability are on the ground. The right-hand numbers indicate the percentage of time for which that gait causes the robot to be statically unstable.

6b). To estimate this, the best gaits were replayed on the physical robot, and the number of feet contacting the ground during the robot's locomotion was recorded over time. Six optical proximity sensors were mounted on the robot, one on each of the robot's six feet to track the foot-to-ground contact.

The sensors were individually calibrated such that the foot-to-ground contact was reliably recorded for all possible configurations and section inclinations of the robot. Reliable ground contact records for inclined foot positions resulted in the sensing distance increase for the cases when the robot's foot was standing straight. Therefore, for non-inclined feet, the ground contact was falsely detected even when the foot was lifted up to 6 millimeters above the ground. This measurement technique ensured a conservative, error-proof record of foot-to-ground contact which provides the lower-bounded estimate of the fraction of time during the robot's locomotion when it was statically unstable. The results of these measurements are shown in Figure 8. As can be seen, relatively little time is spent in a statically unstable configuration. The most dynamic gait using this metric was the best gait that appeared in the first run at generation 4 (gait 2 in Figures 6 and 8). However our metric is very conservative in that it does not take into account other situations for which the robot may be in a dynamic state, such as when legs are only dragging passively along the ground and are not providing support. To this end we plan to replace the optical sensors with strain gauges on each of the legs in order to refine our measurements regarding a gait's degree of dynamism.

## 4 Conclusions

Here we have introduced a genetic algorithm that was used to evolve dynamic gaits for both a simulated and physical robot, with no preconceptions about the desired gait beyond the fact that it should be rhythmic (ie. the robot should return to a particular configuration after some time period has elapsed).

Unlike other approaches to automated gait generation, we did not confine our search to static gaits (such as walking or crawling), or predefine what particular gait should be produced. Few attempts have been made to evolve controllers directly on real robots; fewer still have been performed on legged robots; and only one attempt has been made to evolve dynamic gaits for a legged robot, but in that case the type of dynamic gait desired (trot or pace) was pre-determined, and evolution refined the gait [8].

We began with a complex robot with 12 actuated DOFs, in which the actuators act in parallel as opposed to serially, and thus there is a non-intuitive relationship between a given actuator and a given leg. Here we have introduced a genetic encoding that allows us to evolve arbitrary gait patterns and speeds in which the only constraint is that the gait should be rhythmic.

We presented evidence using a simulated robot in which evolution favored dynamic gaits: gaits in which the robot's center of gravity does not lie within the polygon defined by its foot-to-ground contacts. In other words, the most successful gaits were those that were most dynamic. Using a physical robot that is capable of both static and dynamic motion, we have shown that most of the highly fit gaits exhibited periods of static instability, and were thus at least partially dynamic.

The two evolutionary runs conducted in hardware required a total of  $4\frac{1}{2}$  hours of time and 1240 physical motion pattern evaluations. Both runs yielded qualitatively different dynamic behaviors, characterized by different motion cycle periods and locomotion strategies. These results suggest that the search space has not yet been fully spanned by the genetic algorithm and that additional hardware runs can be expected to produce other types of successful dynamic gaits using various locomotion strategies. We also plan to include sensor-driven control to the robot, as well as to use a co-evolutionary algorithm we have already devised [2] to reduce the number of required hardware trials.

Physical evolution yielded open-loop controllers capable of dynamically operating a very complex mechanical structure requiring at least 36 state variables for its analytical characterization in the state space. Approached from the standpoint of classical control theory, the problem of dynamic controller design for such an intrinsically non-linear system with numerous inter-coupled state variables is far from trivial. The obtained results inspire the research team to seek ways to solve even more challenging control problems and, at a certain point, devise such controller design methods that could surpass their analytical counterparts.

## Acknowledgments

This work was supported in part by a Microsoft Research University Relations grant for Embedded Systems. The authors would like to thank Evan Malone for his advice, help, and support throughout the project.

## References

1. Chocron, O., Bidaud, P.: Evolving walking robots for global task based design. In: *IEEE Congress on Evolutionary Computation*, 405–412 (1999)

2. Bongard, J. C., Lipson, H.: Automated robot function recovery after unanticipated failure or environmental change using a minimum of hardware trials. In: *Procs. of the NASA/DoD Conference on Evolvable Hardware*, Seattle, WA (2004)
3. Clark, J. E., Cham, J. G., Bailey, S. A., Froehlich, E. M., Nahata, P. K., Cutkosky, M. R.: Biomimetic design and fabrication of a hexapedal running robot. In: *IEEE Intl. Conf. on Robotics and Automation*, Seoul, Korea (2001)
4. Dasgupta, B., Mruthyunjaya, T. S.: A Newton-Euler formulation for the inverse dynamics of the Stewart platform manipulator. In: *Mechanism and Machine Theory* **33**(8): 1135–1152 (1998)
5. Earon, E. J. P., Barfoot, T. D., D’Eleuterio, G. M. T.: From the sea to the sidewalk: the evolution of hexapod walking gaits by a genetic algorithm. In: *Intl. Conf. on Evolvable Systems*, Edinburgh, Scotland (2000)
6. Full, R. J., Kubow, T., Schmitt, J., Holmes, P., Koditschek, D.: Quantifying dynamic stability and maneuverability in legged locomotion. In: *Integrative and Comparative Biology* **42**: 149–157 (2002)
7. Gallagher, J. C., Beer, R. D., Espenshied, K., Quinn, R. D.: Application of evolved locomotion controllers to a hexapod robot. In: *Robotics and Autonomous Systems* **19**: 95–103 (1996)
8. Hornby, G. S., Takamura, S., Yokono, J., Hanagata, O., Yamamoto, T., Fujita, M.: Evolving robust gaits with AIBO. In: *IEEE Intl. Conf. on Robotics and Automation*, 3040–3045 (2000)
9. Gruau, F., Quatramaran, K.: Cellular encoding for interactive evolutionary robotics. In: *Technical Report, University of Sussex, School of Cognitive Sciences*, Brighton, UK (1996)
10. Kimura, H., Sakurama, K., Seiichi, A.: Dynamic walking and running of the quadruped using neural oscillator. In: *IEEE Intl. Conf. on Robotics and Automation*, 50–57 (1998)
11. Komsuoglu, H., Moore, E. W., Saranli, U., Brown, B., McMordie, D., Buehler, M., Full, R. J., Koditschek, D. E.: Dynamical gaits and energy efficiency in a hexapod robot. In: *IEEE Intl. Conf. on Robotics and Automation*, Washington, DC (2002)
12. Lewis, M. A., Fagg, A. H., Bekey, G. A.: Genetic algorithms for gait synthesis in a hexapod robot. In Zheng, ed.: *Recent Trends in Mobile Robots*, World Scientific, NJ, 317–331 (1994)
13. Nolfi, S., Floreano, D.: *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press, Boston, MA (2000)
14. [opende.sourceforge.net](http://opende.sourceforge.net)
15. Parker, G. B.: The incremental evolution of gaits for hexapod robots. In: *Procs. of the Genetic and Evolutionary Computation Conference*, 1114–1121 (2001)
16. Raibert, M. H., Brown, H. B., Chepponis, M. : Experiments in balance with a 3D one-legged hopping machine. In: *Intl. J. Robotics Research* **3**: 75–92 (1984)
17. Raibert, M. H., Chepponis, M., Brown, H. B. : Running on four legs as though they were one. In: *IEEE J. Robotics and Automation* **2**: 70–82 (1986)
18. Sreenivasan, S. V., Waldron, K. J., Nanua, P.: Closed-form direct displacement analysis of a 6-6 Stewart platform. In: *Mechanism and Machine Theory* **29**(6): 855–864 (1994)
19. Stoy, K., Shen, W.-M., Will, P. M.: A simple approach to the control of locomotion in self-reconfigurable robots. In: *Robotics and Autonomous Systems* **44**(3-4): 191–200 (2003)
20. Wampler, C. W.: Forward displacement analysis of general six-in parallel SPS (Stewart) platform manipulators using SOMA coordinates. In: *Mechan. Mach. Theory* **31**(3): 331–337 (1996)
21. Wang, G.: Forward displacement analysis of a class of the 6-6 Stewart platforms. In: *Robot., Spatial Mechan. Mechan. Syst.* **45**: 113–117 (1992)
22. Weingarten, J. D., Lopes, G., Groff, R. E., Buehler, M., Koditschek, D. E.: Automated gait adaptation for legged robots. In: *IEEE Intl. Conf. on Robotics and Automation*, New Orleans, LA, 2153–2158 (2004)
23. Yang, J., Geng, Z. J.: Closed form forward kinematics solution to a class of hexapod robots. In: *IEEE Trans. Robot. Automat.* **14**: 503–508 (1998)