Available online at www.sciencedirect.com
ScienceDirect
Journal of Bionic Engineering 4 (2007) 255−263

Extended Evolutionary Fast Learn-to-Walk Approach for Four-Legged Robots

Muh. Anshar, Mary-Anne Williams

Innovation and Technology Research Laboratory, Faculty of Information Technology, University of Technology, Sydney, City Campus 15 *Broadway, Ultimo, NSW* 2007, *Australia*

Abstract

Robot locomotion is an active research area. In this paper we focus on the locomotion of quadruped robots. An effective walking gait of quadruped robots is mainly concerned with two key aspects, namely speed and stability. The large search space of potential parameter settings for leg joints means that hand tuning is not feasible in general. As a result walking parameters are typically determined using machine learning techniques. A major shortcoming of using machine learning techniques is the significant wear and tear of robots since many parameter combinations need to be evaluated before an optimal solution is found. This paper proposes a direct walking gait learning approach, which is specifically designed to reduce wear and tear of robot motors, joints and other hardware. In essence we provide an effective learning mechanism that leads to a solution in a faster convergence time than previous algorithms. The results demonstrate that the new learning algorithm obtains a faster convergence to the best solutions in a short run. This approach is significant in obtaining faster walking gaits which will be useful for a wide range of applications where speed and stability are important. Future work will extend our methods so that the faster convergence algorithm can be applied to a two legged humanoid and lead to less wear and tear whilst still developing a fast and stable gait.

Keywords: legged-robots, locomotion, learning, genetic, convergence, walking gaits

1 Introduction

1.1 Background

Since Sony introduced the so-called AIBO, a four-legged entertainment robot, with all features embedded in its mechanisms, many researchers have used it and found it to be a good research test bed. In 1997 the international RoboCup initiative commenced using soccer as the benchmark problem. Two years later the Four-Legged League using AIBO robots as the platform was introduced.

In a robot soccer game, apart from having an effective team work, strategies and collaboration, locomotion plays an important role in an individual or team performance. Walking speed is a key factor in producing an internationally competitive team since it determines how quickly a team can gain possession of the ball or to move to a desired attacking or defensive position. Many individual researchers and four-legged RoboCup teams have studied and developed fast and stable gaits[1]. The most recent method is through biological learning approaches, i.e. evolutionary computation[2,3]. Murata *et al.* developed a genetic network programming combined with automatic defined rules to achieve comprehensible control policies[4]. Zhou *et al.* extended their genetic algorithm approach aided by the Elman neural network[5] in order to optimize the adaptation capability to dynamic environments. Golubovic and Hu proposed a hybrid evolutionary algorithm approach to generate stable and fast gaits[6]. Besides the evolutionary algorithm approach, there are also some researchers employing reinforcement learning as a means to optimize the speed. Kohl and Stone presents a policy gradient reinforcement learning algorithm to optimize the team locomotion

Corresponding author: Muh. Anshar
E-mail: muh.anshar@gmail.com

speed[7]. Cheng *et al.* implemented a genetic algorithm based on roulette selection and were able to obtain faster speeds than that of their previous approach, hill-climbing search[8]. Other researchers employed two different learning approaches to develop walking while moving an object. Kamio *et al.* proposed an integration approach that fuses genetic programming and reinforcement learning to enhance the box moving task[9,10]. If one only considers speed as the goal then it is plausible to achieve quickly. However the body balance and the robot head can be negatively affected and hence vision accuracy and performance will be reduced[8]. A literature review shows that none of the aforementioned research projects addresses the need to focus on speed and stability whilst paying appropriate attention to undue wear and tear of the robots. Robots, like all hardwares, typically have a lifetime, and AIBOs were designed to be entertainment robots and not soccer players that spend much of their time fighting each other for the ball. In addition, other factors beyond speed are important. For example, in a typical robot environment robots must avoid obstacles and therefore they may need to walk slowly so as to have sufficient time to detect and respond to, e.g. dodge, obstacles in their path. Clearly, there are many trade-offs that need to be taken into account when designing robot walks. Furthermore, sudden movements will affect the momentum on the robot joints as this will contribute to wear and tear issues. Apart from that, the learning process requires a huge amount of time due the large number of runs before converging to satisfactory solutions. This has made the problems more complicated and challenging to tackle. In Ref. [1], the authors employed manual hand tuning to generate the omnidirectional walking gaits which has to tackle with the large number of parameters. Through this paper, we focus on the aforesaid problems on generating walking gaits based on learning approaches. We propose a more sophisticated approach and develop a direct learning based genetic algorithm which mimics genetic evolution[11] and minimizes robot hardware wear and tear.

1.2 Four-legged robot overview

At the hardware level, an AIBO ERS-7 consists of several sensors, joints and motors. For the purpose of walking it has twelve joints across its front and back legs. It also comes with an internal processor, an external memory stick and a wireless communication card which support a programming environment. Fig. 1 illustrates the physical appearance of the robot and the positions of the sensors.

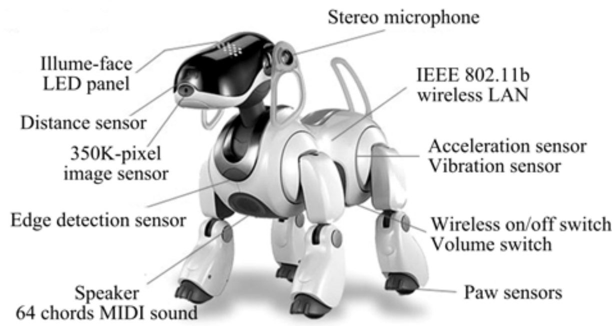

Fig. 1 ERS-7 features.

The features of this robot are listed as follows.

- 576 MHz MIPS R7000
- 64 MB RAM
- 802.11b wireless ethernet (standard)
- Memory stick reader/writer (in dog)
- 18 PID joints, each with force sensing
  - 4 legs
    - 3 joints each (elevate, rotate, knee)
    - 1 paw button each
  - 3 joints on neck (tilt, pan, nod)
  - 2 joints on tail (tilt, pan)
  - 1 joint on mouth
- 2 ears, 1 boolean joint each (flick up or down)
- 26 independent LEDs
- Video camera
  - 56.9° wide and 45.2° high
  - Resolutions: 208×160, 104×80, 52×40
  - 30 frames per second
- Stereo microphones
- 3 IR distance sensors
- *X*, *Y*, and *Z* accelerometers
- 4 pressure sensitive buttons (one on head, three on back)
- 1 boolean button under mouth
- Sensor updates every 32 ms, with 4 samples per update.

For best result, higher level planning and control of walking can parameterize joints onto accurate loci.

Some soccer teams have developed different methods to achieve a faster walking gait. For example, the University of Texas team, UT Austin, developed a policy gradient reinforcement learning algorithm to generate fast quadruped locomotion[7]. The University of New South Wales team developed fast walking gaits using a rectangular locus[1]. The UTS Unleashed! team at the university of Technology Sydney, the code employs separate binary code which supports the interrupt ability of actions between the locomotion and other behaviours, including the learning code. This feature allows the learning to occur without causing physical damage to the robot. In addition, the UTS Unleashed! code protects the robots via other mechanisms too, such as using a hierarchy of actions where for example if the robot falls over during a trial, the learning will commence after the robot gets up and positions itself in the home position[8]. On the other hand, damage still may occur to the robots if the wrong parameters are sent to the joints that jerk the hardware and there is no guard against or detect wear and tear.

1.3 Learning to walk

The first attempt by the UTS Unleashed! team to improve the walking capability was using learning through a fitness driven approach, where the search was guided via a learning rate coefficient. This learning process requires an amount of time to converge to a solution and typically leads to unacceptable levels of wear and tear of robot hardware. With a wide range of populations to search, the learning will stop before convergence condition to the best solutions occurs because there is limit on the number of trials due to the need to protect the robot hardware by reducing the number of trials. As the learning result is achieved slowly, a biological approach seems appropriate and leads to promising initial results. The use of three genetic operators, selection, crossover and mutation, increases the speed of the learning to walk process. However, with the original operators, similarity problems in the population reduce the degree of the fitness solution. Also the body stability decreases as the learning process may result in parameters exceeding the normal thresholds (The range should be in this area as to prevent body shaking and wrong parameters). In this paper, we will draw attention to four important aspects:

(1) With the capability of a fast learning genetic algorithm, we would be able to achieve the optimal fitness solution in fewer iterations and trials. This will reduce the on-line run experiment on the robot as to avoid wear and tear issue.

(2) In the current method, “similar” parents in the population will not be executed since it reduces the productivity of the learning. Through the extended learning approach, all similar parents will be forced to produce more offspring so as to increase productivity.

(3) In order to maintain convergence quickly we introduce a “threshold value” by which all candidate solutions will be measured and a decision will be made as to whether they will be chosen for further generation or not.

(4) Wrong parameters, which might result from the large space of learning solutions, would affect the physical aspects of the robot, such as motors at leg joints would move beyond the safe motion region. By addressing this factor, the learning process will guarantee the generated parameter gaits will not introduce any damage to robot joints’ actuators. Details about wrong or invalid parameters will be explained in the next section.

2 Proposed learning approach

2.1 Problems with current approach

The current genetic algorithm approach reaches optimal solutions slowly after many trials and iterations, each of which requires the robot to walk several meters. As a result it leads to unacceptable levels of wear and tear. Designing learn to walk algorithms that wear out a robot’s hardware during the process is not an acceptable solution. The current algorithms employ three original genetic operators, i.e. selection, crossover and mutation. Whilst they improve with each iteration, it turns out that each one requires the robot to walk approximately 2.5 m, almost 10 body lengths for an AIBO. A typical solution takes 100 iterations which means the robots must walk 200 m, which can cause significant damage to the motors.

Since the learning is based on probabilistic methods, the generation can vary dramatically and the speed can degrade greatly. As a result, the robot body can become shaky and unstable whilst walking.

The learning process runs with no any measurement to determine for future revision in order to optimize the learning results. In other words, there is no a clear rule which would direct the searching approaches toward best solutions.

2.2 The proposed approach

Our extended Genetic Algorithm, Directed Gait Generation, finds optimal solutions faster than standard Genetic Algorithm (GA). By using the history of parameters we can use the probability theorem to ignore some of the iterations. This means that we will find the optimal parameters faster with less distance travels by the robot, thereby producing a good solution at less cost. Besides that, there are some addition parameters which will also lead the learning towards the upward-learning trend. However, by ignoring some candidate solutions in the population can reduce the variety of the searching space, which may lead to a local minima problem. To avoid this, we choose the fittest solutions as the candidate solution referenced for every generation. Also the threshold value for every candidate solution will be minimized to ensure an optimal solution is found.

We designed the following experiment that shows the extension leads to a solution in a faster way, and hence with less wear and tear.

2.3 Proposed methodology

Fig. 2 illustrates the design of our proposed method which is implemented on a robot, in this case an AIBO. It is based on grounding representation diagram[12] where each box describes a robot skill. We use two terms which are “threshold values” and “a threshold value”. The first term, threshold values or reference values, refer to all values originated from deviation values of seven best populations in database history. The second term, a threshold value, refers to the total number of diversity resulted from a comparison between the reference values and deviation values of the new generated population. The comparison process, which occurs in the domain range prover, will be explained further in the subsection below.

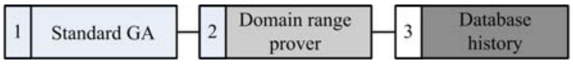

Fig. 2 Representation of extended GA.

2.3.1 Standard GA

The initialization process takes place in the candidate solution which applies genetic operators, i.e. selection, crossover and mutation. In order to solve similarity problems in candidate parents, the generator will be forced to pick up another pair of candidate parents by iterating the selection process, i.e. tournament selection. We choose this approach as to avoid additional time for introducing other operators, e.g. AddAlternative[13].

As illustrated in Fig. 3, the selection process occurs in the first operator, i.e. selection, of the standard GA approach. The selection process employs the tournament selection which picks the best candidate populations among the proposed populations. The selection process produces two parents which generate an offspring. The mutation process takes place using a mutation rate less than one.

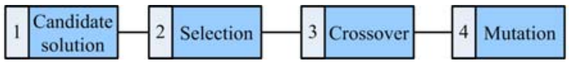

Fig. 3 Standard GA.

2.3.2 Database history

The theory uses a similar approach as the schema theorem[10], however they are different in principle. The schema theorem uses an equation to represent populations which have a high fitness level, while the database history of our approach contains all candidate solutions with a fitness level that satisfies a minimum value. Let $g_i(v, f(x_i))$ denotes a generation of each member of population in the database history per run (see Fig. 4), where x_i takes values from 1 to 7 such that 1 represents the initial population which is set first through a pre-designing stage, v is the solution (i.e. speed value in $\mathrm{cm \cdot s^{-1}}$) and $f(x_i)$ is the population function.

Fig. 4 Database history architecture.

These seven best solutions will remain constants as the reference values used to determine the learning process.

2.3.3 Domain range prover

Fig. 5 depicts the internal mechanisms of the domain range prover, which manipulates the current proposed solution and determines whether it will be processed further or rejected.

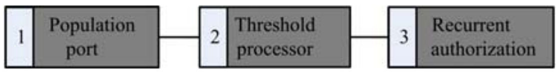

Fig. 5 Mechanisms of domain range prover.

We employ the standard deviation of the seven best populations generated from 106 training cycles in the pre-design stage.

$$\sigma = \left[\sum_{i=1}^{N} \left(f\left(x_i\right) - f\left(\overline{x}_i\right) \right)^2 / N \right]^{1/2}, \qquad (1)$$

where σ is the standand deviation of the populations, x_i the ith member of the best fitness of candidate population, $\overline{x}_i$ the mean of the seventh best fitness population.

This standand deviation σ is the threshold for determining a new population generated by the standard GA. The new members of population along with the seventh best fitness populations in the database history will be re-calculated to find a new standard deviation. If this standard deviation is below the threshold, it is executed, otherwise it is rejected. This threshold is obtained from the total diversity among the standard deviation values of reference (seven best populations in the database history) and the standard deviation values of the new population generated from Eq. (1). This means that the proposed population is not sent to the engine of the robot but it is probably chosen for the next regeneration process (Regeneration process takes place in the standard GA, which uses the tournament selection property to obtain offsprings). At this stage, some of the proposed walking parameters are evaluated according to whether they are in the safe region or not. If the parameters are not in this range, then they are replaced by certain values which are taken from the most fitness solution. This process assures that the walking parameters are not the wrong parameters. The pseudo-code of this property is depicted as follows.

```
IF parameter_1 LESS THAN safe region
   THEN parameter_1 EQUAL TO safe region value_1
IF parameter_2 LESS THAN safe region
   THEN parameter_2 EQUAL TO safe region value_2
IF parameter_3 LESS THAN safe region
   THEN parameter_3 EQUAL TO safe region value_3
```

As a result, the robot joints, especially the leg joints be protected from disruptive movements.

There are 20 parameters that will take part in the learning process as shown in Table 1 and the position for each parameter is illustrated in Fig. 6. More details can be found in Ref. [8].

Table 1 Parameterized walking gaits

No.	Parameters
1	Front right shoulder height
	Front left shoulder height
2	Back right shoulder height
	Back left shoulder height
3	Front right step height
	Front left step height
4	Back right step height
	Back left step height
5	Front right step width
	Front left step width
6	Back right step width
	Back left step width
7	Front right step length
	Front left step length
8	Back right step length
	Back left step length
9	Step per second
10	Maximum forward speed
11	Front leg locus points x_1
12	Front leg locus points y_1
13	Front leg locus points x_2
14	Front leg locus points y_2
15	Back leg locus points x_3
16	Back leg locus points y_3
17	Back leg locus points x_4
18	Back leg locus points y_4
19	Front right forward factor
	Front left forward factor
20	Back right forward factor
	Back left forward factor

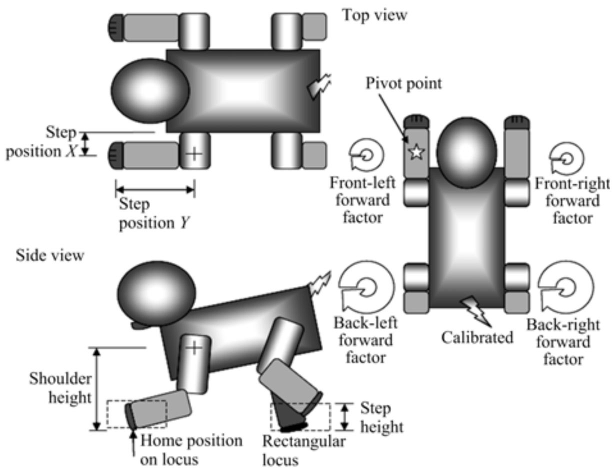

Fig. 6 Position for each parameter.

For parameters 1 to 8 and 19 to 20, the values of the right and left legs are set the same. For parameters 9 to 18, they are a stand-alone parameter which represents different values. Parameter loci adjustment (11 to 18) will follow the type of the locus type, rectangular locus.

Invalid parameters may lead to the following problems:

(1) Front right and left step widths are excessive small to create friction with the robot body. Similarly, this will occur if the back right and left step widths are small.

(2) Front right and left shoulder heights are excessive high which will produce a jumping effect on the walking style. The same problem exists whenever the back shoulder heights are excessive high. The opposite values will introduce higher friction between the legs and the ground surface.

(3) Maximum forward speed and number of steps per second can also produce invalid walking gaits because the actuators for each leg joint have a limited torque. The higher the speed, the lower the torque whereas the body mass of the robot remains constant. As a result, the robots will vibrate during the walk.

(4) Speed synchronization between the front legs and the back legs.

Problems may also be raised by other parameters during the learning process, but they will not contribute any significant damage to the robot joints, such as locus points and forward parameters. These parameters will affect the speed of the walk, since the walking direction will not be fully a straight line due to the non-centred location of the AIBO battery which is situated on the left side, which results in a non-uniform weight across the robot body.

3 Experiments

3.1 Pre-stage experiment

In order to provide an effective comparison, we ran the experiment using a fitness-driven search. After several hand-tunings on the initial walking parameters, the learning process based on standard GA commenced several runs. The most productive run, which obtained faster and more stable gaits, was chosen and seven of them were calculated for the standard deviation. This value was then used as the threshold.

The robot walks between two points on the soccer field as shown in Fig. 7. Each turning point is marked by a 15 cm × 25 cm × 40 cm box and the first box is painted purple-green and the second one is green-purple. This allows the AIBO to detect the end of the track over which the walk is timed. One lap starts at the first box and finishes at the second one. The distance between the two boxes is 2 m (see Fig. 7). The vision based on color recognition guides the walking pathway and the infrared sensors detect whether the robot has reached an end point, i.e. box. The gait generations can be monitored through the wireless communication as the learning to walk algorithm commences.

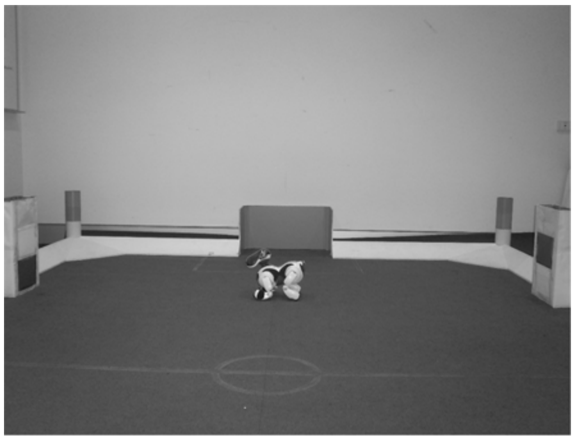

Fig. 7 Learning to walk experiment.

3.2 Extended GA experiment

The data was collected from several runs under the same conditions as the previous pre-stage experiment. After several runs, the proposed method would produce better results by maintaining diversity above 12 values (threshold value). This means that there are at least 12 variables of generated parameters which should be different from the threshold values (reference values).

4 Results

The results of the experiments are divided into two stages, i.e. pre-design stage and implementation stage.

4.1 Pre-design stage

In this stage, we employ another learning method using a searching method based on fitness-driven and the results can be seen in Fig. 8. We also run several tests based on the standard GA and the results can be seen in Fig. 9.

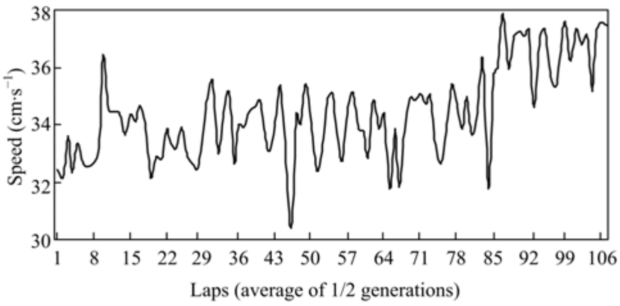

Fig. 8 Speed vs generation based on fitness-driven approach.

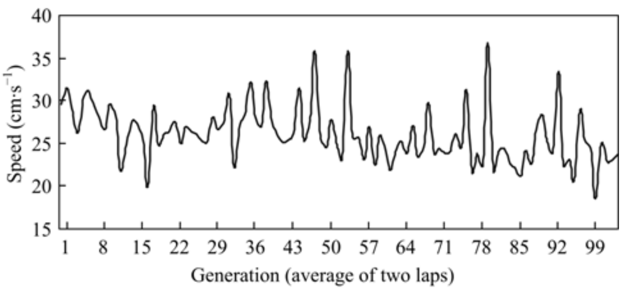

Fig. 9 Speed vs generation based on standard GA.

We determine seven peaks of speed generation (Table 2) and calculate the standard deviation for each member of the population (one population consists of 20 members), which refers to Eq. (1). These values become default values or threshold values which are shown in Table 3.

Table 2 Seven best fitness populations in the database history

g_i	v	$f(x_1)$	$f(x_2)$	⋯	$f(x_{19})$	$f(x_{20})$
2	56.1	98.1	118.1	⋯	1.9	3.0
20	53.0	98.1	118.1	⋯	1.9	3.0
47	50.8	98.1	118.1	⋯	1.9	3.0
50	55.1	98.1	121.4	⋯	1.9	3.0
56	54.8	98.1	118.1	⋯	1.9	3.0
82	54.3	98.1	115.1	⋯	1.8	3.0
95	51.9	98.1	118.1	⋯	1.9	3.0

Table 3 Threshold values for each member of the population

$f(x_i)$	$f(x_1)$	$f(x_2)$	⋯	$f(x_8)$	$f(x_9)$	$f(x_{10})$
σ	0.0	1.7	⋯	0.9	0.1	0.1
$f(x_i)$	$f(x_{11})$	$f(x_{12})$	⋯	$f(x_{18})$	$f(x_{19})$	$f(x_{20})$
σ	0.8	0.9	⋯	1.0	0.1	0.0

4.2 Implementation of extended GA

After running the experiment, the results show that gait generations generated by the extended GA are faster and the best solutions success to converge before the robot shuts down due to wear and tear issue (Fig. 10). The figure illustrates a twelve-diversity run, which are obtained through four different scenarios of test cases. The first case used a ten-diversity run for the threshold values, the second with eleven diversity values and the last utilized twelve diversity values.

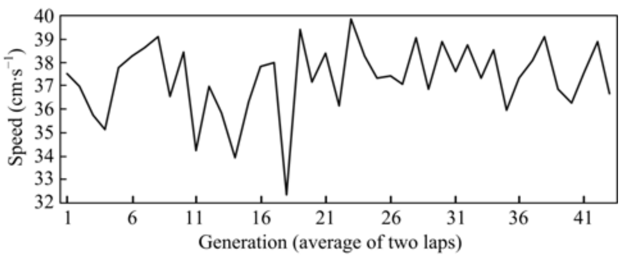

Fig. 10 Speed generation based on directed gait generation.

5 Discussions

Developing gaits for robots is a complex and challenging task. This research is aimed at developing a methodology and techniques to improve the learning process for developing gaits. Our primary objective is to develop an effective gait for a robot at minimal cost to the robot hardware. Our approach is based on the idea of filtering candidate solutions before testing them on the robot, thus significantly reducing the number of trails and consequentially reducing the amount of wear and tear on robot hardware.

It can be seen that the convergence to the best solutions in fitness-driven search occurs above 50 generations while the standard GA requires 80 generations. In the directed gait generation approach, the solution converges at the 24th generation. This is significantly faster compared to other approaches, and gives a favourable outcome, i.e. less wear and tear.

Different issues arise in the standard GA, where the learning produces some incorrect parameters, which lead impairment problems at the robot joints. However, the extended GA generates better solutions in a faster way compared to the previous approach.

In the directed gait generations, there are two important features, which will supervise the learning process. The first is that the threshold values are originated from the standard deviation of the history populations, and the second is that the threshold values are used for guaranteeing the safety of the gait's parameters. The first threshold values should be minimized in order to avoid poor diversity in the search space. The higher these values the more candidate solutions will be pruned and as a result, there is a possibility of pruning best solutions in one run.

It is concluded that the extended GA is better than fitness-driven search and standard GA. It can prevent incorrect parameters, which will only decrease the productivity of the learning process. Indeed, the learning only requires a short time to converge to the best solutions, which will reduce the effect of wear and tear.

6 Conclusions

Walking skills play a crucial role for legged robot mobility, particularly for robot soccer players, since they enable the robot to perform a wide range of tasks, such as dribbling, dodging, ball grabing and kicking. The development of effective walking gaits focuses on two important aspects, stability and speed. Both of these aspects are very difficult to achieve at the same time without making a trade-off. The search space of potential parameters is vast, and as a result hand-tuning to find the optimal combination is not feasible. Therefore, learning approaches have become powerful tools in the determination of the best solutions. The experiments described in this paper show that the proposed directed gait approach based on genetic algorithms is a more acceptable learning process since it leads to a solution in a shorter period compared to the fitness-driven search and the standard GA. Consequently there is less wear and tear on the robots. Besides, the approach is capable of maintaining a monotonic convergence towards a solution which can help to prevent damage to the robot joints due to invalid walking parameters. Our extended GA approach is applicable to a wide range of problems in robot system design including vision calibration. Future work will use the approach to develop effective gaits for two-legged humanoid robots.

Acknowledgement

We would like to thank Christopher Stanton and Fedja Hadzic for their contributions to the completion of this paper.

References

[1] Hengst B, Ibbotson D, Pham S B, Sammut C. Omnidirectional locomotion for quadruped robots. *RoboCup* 2001: *Robot Soccer World Cup V*, *Lecture Notes in Computer Science*, 2002, **2377**, 368–373.

[2] Juan T J, Carlos R, Jorge R. Automatic behavior generation in a multi-agent system through evolutionary programming. *IEEE* 3*rd Latin American Robotics Symposium*, 2006, 2–9.

[3] Reeve R, Hallam J. An analysis of neural models for walking control. *IEEE Transaction on Neural Networks*, 2005, **16**, 733–742.

[4] Murata T, Okada D. Using genetic network programming to get comprehensible control rules for real robots. *IEEE Congress on Evolutionary Computation*, 2006, 1983–1988.

[5] Zhou H, Guo G, Liu M. GA-aided Elman neural network controller for behavior-based robot. *The Sixth World Congress on Intelligent Control and Automation*, 2006, **2**, 9068–9072.

[6] Golubovic D, Hu H. A hybrid evolutionary algorithm for gait generation of Sony legged robots. *IEEE* 28*th Annual Conference of the IECON* 02 [*Industrial Electronics Society*], 2002, **4**, 2593–2598.

[7] Kohl N, Stone P. Policy gradient reinforcement learning for fast quadrupedal locomotion. *IEEE International Conference on Robotics and Automation*, 2004, **3**, 2619–2624.

[8] Cheng M, Dissanayake G, Gurram D, Hadzic F, Healy G, Karol A, Nguyen H S, Stanton C, Trieu M, Williams M, Zeman A. *Robot World Cup Soccer* 2004: *The Magic of UTS Unleashed*! *Team Report*, Innovation and Technology Research Laboratory, 2004, 1–22.

[9] Kamio S, Iba H. Adaptation technique for integrating genetic programming and reinforcement learning for real robots. *IEEE Transaction on Evolutionary Computation*, 2005, **9**, 318–333.

[10] Kamio S, Liu H, Mitsuhasi H, Iba H. Researches on ingeniously behaving agents. *Proceedings of IEEE NASA/DoD Conference on Evolvable Hardware*, 2003, 208–217.

[11] Mitchell M. *An Introduction to Genetic Algorithms*, The MIT Press, Cambridge, Massachusetts, England, 1998.

[12] Stanton C. A methodology for grounding representations. *Proceedings of the* 2006 *International Symposium on Practical Cognitive Agents and Robots*, 2006, 73–80.

[13] Mitchell T. *Machine Learning*, McGraw-Hill, Singapore, 1997.