An Autonomous Legged Robot that Learns to Walk Through Simulated Evolution

Sadayoshi MIKAMI, Hiroaki TANO and Yukinori KAKAZU[1]

[1]Faculty of Engineering, Hokkaido University,
Kita-13, Nishi-8, Sapporo 060, JAPAN. E-mail: mikami@hupe.hokudai.ac.jp
Tel: +81-11-736-3818, Fax: +81-11-758-1619,

Abstract

A method of acquiring the skill of walking for a legged robot by on-line learning is proposed that mimics both the individual learning and the evolutionary learning of living creatures, which we should like to refer to as an A-Life approach. We propose a distributed controller associated with each leg that learns whether to trigger the stride of its leg or not according to the observed states. One difficulty in taking such a state-oriented approach to the learning mechanism is that we have to reduce the explosively huge state spaces. We demonstrate how Genetic Algorithms are incorporated with the acquisition of state compression functions.

Introduction

Planning a gait sequence for a legged robot so that it can walk stably in a variety of terrain is a hard problem, especially when it is applied to a real world environment. When precise models of both the dynamics of the robot and its environment are available, we can obtain analytical solutions to give stable gait sequences. Much study has already been done on robots with 4 or 6 legs under such conditions (Todd 1985) However, adaptive gait planning has only been studied for cases where a robot walks in modeled terrain, and where it causes malfunction of its own legs (Adachi, et. al. 1988, Hirose and Knieda 1991, Kumar and Waldron 1989).

In recent years, a turning point has been reached in walking robot research, which we would like to refer to as the *A-life approach* (e.g. Maes and Brooks 1991, Snaith and Holland 1991). This tries to find a gait through learning methods inspired by nature. What we expect from this is a gait generation method without any environmental or internal models, and that is robust to any kind of dynamic change in the environment; one like those found in animals that develop their gait control mechanism through evolution and individual learning, without knowing any models of themselves.

How do animals achieve this ability to walk? Whether it holds true in real animals or not we don't know, but we think that this process of acquiring the ability to walk may be realized by the following two processes of self-organization and the evolution of legs and their controllers:

- (*Individual learning*) First, each of the legs is associated with its own controller. The controller learns plans for successfully moving its leg to achieve walking; it corresponds an action to an observed state that is received through sensors, and rewards the sequences of these action-state pairs when they succeed in walking through reinforcement learning.
- (*Evolutionary learning*) These observed states are high dimensional spaces where trial and error learning, actually, a reinforcement type learning, may not easily achieve convergence without state compression technology. Here, we suppose that animals have acquired their sensory integration functions through 'evolution.' This is based on the assumption that there are 'principal pairs' of sensory input. A principal pair is the subset of a sensory input. When a principal pair is detected in an observed state, the

pair works as a "wild-card"; any sensory information that does not belong to a principal pair is simply neglected and categorized to a single state. For any set of states that belongs to a principal pair, only one action is appropriate. For example, if we are faced with the situation of avoiding a stone flying toward us, the principal pair is the set of sensor inputs that distinguishes flying stones from others. The action to be taken (bend the body, in this case) should not be affected by other sensory input such as sound. This would be realized by a table-lookup type of state compression function, thus, avoiding a state explosion. Animals are assumed to have acquired these principal pairs through the process of evolution, and are therefore able to avoid the redundant classification of state spaces.

In this paper, we artificially implement this individual and evolutional learning to control a walking robot that adaptively walks through unpredictable terrain, that acquires robustness to malfunctions of the legs, and that may be of a variety of physical configurations. This method is realized by stochastic learning automata (*SLA*) theories (Narendra and Thathachar 1989) and *Genetic Algorithms* (e.g. Goldberg 1989), and its effect is empirically demonstrated through simulated robots. We have developed experimental hardware to test the effectiveness of this method in a real-world environment (**Fig.1**). Although the hardware has not yet been completed at this point in time, the results of the computer simulations shown in this paper assure that the method can be expected to perform with high adaptability to unknown physical configurations and environments in the real world.

Related Works

The idea of applying Genetic Algorithms to gait acquisition has also been proposed by Snaith (Snaith and Holland 1991), and de Garis (de Garis 1990). Snaith's approach, as well as de Garis's ideas, are based on the development of appropriate neural network controllers for gait control which eventually generate reflective leg performances to give a stable walk. The controller itself is almost the same as our SLA based learning controller, except that the SLA approach is based purely on the association of a state to the probability of a selection among actions. This means that the SLA approach is more extendible thus incorporating a variety of sensory input enabling it to realize the smooth adaptation to a dynamically changing environment. One of the drawback is, as Snaith's paper has pointed out, the explosion of search spaces. Our Genetic approach is to realize state compression function. Thereby, it will be effective in improving the convergence and robustness, not only of our SLA controller, but of any kind of learning gait controller that to some extent observes sensory input.

Brooks and Maes (Brooks 1989, Brooks and Maes 1992) have been actively studying a multi-legged robot that can learn to walk under a dynamic environment through trials. Their robot is controlled by behavior-based distributed hierarchical control units (Brooks 1989). The robot is given correction of behavior - a set of commands that are triggered by sensory condition - and this behavior is coordinated to achieve a given task - "walk", in this case - through reinforcement type learning. Our SLA based controller is close to their controller in the sense that the appropriate behavior (in the behavior-based approach), or the associations of action-state pairs (in our SLA approach), are both given through stochastic reinforcement learning. The difference is that our approach produces the reflection of legs, while the behavior-based approach acquires the correction of a feasible set of actions. The reflection level description is a more precise and universal description so that it is more powerful in adapting to a dynamic environment. For example, this approach could easily be applied to a situation where a number of multi-legged robots segment themselves or join together dynamically. One drawback is again the explosion of state spaces as Brooks

and Maes have pointed out. This paper is intended to explore to what extent this simple approach can achieve a practical and endurable convergence using the application of a state compression mechanism.

Gait Acquisition Through Individual Learning

Simple legged robot:

The legged robot considered in this paper is assumed to be controlled only by the trigger signals that sequentially invoke one of the following three movements of the legs (**Fig.2**): (m_1) lift and recover the leg to a set-point near the body, (m_2) lower the leg and support the body, and (m_3) drive the body forward. Note that these three actions are only assumptions to be held here, and that a greater variety of movement is possible. The fundamental point is that each movement is invoked only by a trigger signal. This means that, at each unit time, a leg is given only one of the following two commands: an a_1 command, that causes a movement, or an a_2 command, that specifies remaining in the current posture. Each movement will be controlled by a simple scheme, such as an open loop control with limit switches; no model of dynamics is necessary.

Each leg is separately controlled by its own controller; there are no restrictions as to the number of legs and their configuration. The controller consists of the following three parts (**Fig.3**):

1. A *state observer*, that integrates sensory input and corresponds it to two types of states, s and t. As described later, s and t contribute to the determination of the actions and the teaching signals, respectively. A sufficient variety of sensors should be prepared as they dominate the accuracy of the control. In the rest of the paper, we prepare the following minimum variation of sensors: (1) leg state sensors, that detect the actions carried out by each leg, (2) a landing sensor, that detects whether the robot body touches the ground or not, and (3) a velocity sensor, that observes whether the robot moves forward or not. The leg state sensors are corresponded to s, while the landing and velocity sensors are combined into t to give a teaching signal.

2. A *decision maker*, that determines an action a_i according to the current state s and applies that action to the leg. This is done by using stochastic learning automata as will be shown in the next section.

3. A *critic unit*, that gives a binary teaching signal b which specifies whether the robot has succeeded to walk (b=1) or not (b=0) by referring to the state t. The design of this critic unit is essential to the behavior of the walking robot because the objective of the learning is to maximize the expectation of b. A simple example of this critic unit should be a combination of the following two decision criteria (**Fig.4**): whether the robot touches the ground (C1=1) or not (C1=0), and whether the robot succeeds in moving forward (C2=1) or not (C2=0). Here, the condition C2=1 means that a leg has performed the movement m_2 while one of the other legs remains in a posture between m_1 and m_2, or m_2 and m_3. One possible combination is to let $b = C1 \wedge C2$.

From the above assumptions, there is no difficulty in extending the above mentioned configuration to a more complicated real mobile robot; only the actions, states, and the evaluation of the results of the actions would need to be considered in designing such a robot.

Learning to Walk:

A controller corresponds a state s to an action a_i, i=1 or 2. For each controller, the objective in learning to walk is to acquire the set of these state-action pairs that minimizes the expectation E(b) of failure. Here, by letting $p_{si} = \Pr(a_i|s)$ be the

probability of selecting the action a_i at the state s, this learning process results in acquiring these probabilities for all the possible states s.

One reinforcement learning scheme, specifically, stochastic learning automata (SLA) theory, is able to let a controller acquire optimal probabilities under the assumption that the probability of b=1 is stationary at each of the states (Narendra and Thathachar 1989.) Although this assumption is not proven to hold for the gait acquisition problem, the computer simulations in a later section demonstrate that optimal actions are given applying this scheme. The optimal gait is given because all the legs satisfy the condition for walking, specifically, the condition of moving forward without touching the ground.

The updating equation used in this paper is the L_{RP} (Linear Reward-Penalize) scheme, which is one of the major schemes of SLA. Let p_{si} be the feasibility (meaning probability) of selecting the i-th action under the state s. Suppose the i-th action has been performed under that state s. Suppose the evaluation b (b=1 when the robot can successfully step to walk and b=0 otherwise) is given after the performance of that action. The feasibility of selection of that action-state pair should be linearly encouraged (or penalized when b=0). Also, the feasibility of other actions under the same state s should be penalized since they do not contribute to walking (or, if b=0, all of them should be encouraged since they seem to have equal potential for contributing to successful walking.) This is summarized in the following formulae:

$$\text{If } b=1, \quad p_{si} \leftarrow p_{si} + \alpha(1-p_{si}),\ p_{sj\neq i} \leftarrow (1-\alpha)p_{sj}, \tag{1}$$

$$\text{otherwise}, \quad p_{si} \leftarrow (1-\beta)p_{si},\ p_{sj\neq i} \leftarrow (1-\beta)p_{sj} + \frac{\beta}{N-1}. \tag{2}$$

where α *and* $\beta \in [0,1]$ are the learning factors for reward and penalty, and N represents the number of actions. One problem with a reinforcement learning scheme is that it takes many trials before achieving skilled action-state pairs. This convergence deteriorates as the rate of failures increases during trials. Thus, we should avoid settling an objective that is hard to achieve, especially for cases where the teaching signal is given by the intersection among a number of criteria.

So far, the critic has determined a teaching signal b by taking an intersection between C1 (for standing) and C2 (for moving). We extend it to achieve a greater number of rewarded (b=1) cases as follows: from the definitions of C1 and C2, the condition C1 must hold true (C1=1) when C2 is true. Thus, by letting $(Q_{11}, Q_{10}) = (1,0)$, and $(Q_{21}, Q_{20}) = (1,0.5)$, and by letting $Q = Q_{1C1} \times Q_{2C2}$ be the probability of determining b as 1, the rewarded conditions are extended while the objective condition $C_1 \wedge C_2$ is retained, because $Q_{11} \times Q_{12} = 1$. By using this probabilistic evaluation of the conditions, we can introduce a number of criteria, such as the restriction that no more than two legs are allowed to land together, without deteriorating convergence.

Evolutional Learning for Sensor Integration

Despite this method of improving convergence, the state spaces for controlling walking are still huge. This is because the number of spaces is the multiplication of the states for all the legs, and a walking robot may have any number of legs. After acquiring a gait under a certain environment, if this environment changes, then, the

learning process must work again to acquire another gait that can adapt to the new environment. Much learning time may be required for the acquisition of the gait, and this deteriorates the adaptability of the robot.

If the number of states is reduced into a smaller set, then the convergence of learning will be improved; but, in some cases, the gait will never be found by such compressed states. Thus, a compression method for states that well achieves both adaptability and ability of learning needs to be found. The following describes a method for acquiring such compressed states (referred to as *principal pairs* in section 1) using a simulated evolution method. This method realizes the evolution of walking robots where optimal sensory integration is acquired through generation.

Let the number of compressed states be M, and let the compression function be $m = \{(s,s'), s = 1,\ldots,N\}$ where s is an original state and s' is the state that corresponds to s after compression. The problem of acquiring optimally compressed states is to find a function m that maximizes the mean expectation of the teaching signals E(b).

This is a search problem through unknown huge spaces for maximizing an objective function. Genetic Algorithms (e.g. Goldberg 1989) are suitable for this type of search problem since the function m can directly be corresponded to a chromosome of the GA. This is done by the following processes:

1. Prepare a population of compression functions $\{m(i), i = 1,\ldots,N_p\}$ that contain randomly generated elements (s,s') ranging from 1 to N for s and from 1 to M for s', respectively. Let us indicate the robot that uses $m(i)$ as R_i.

2. For all R_i, evaluate the mean expectation $E^m(b(i))$ of teaching signals through on-line trials for a given number of intervals. These trials are carried out by changing the compression functions sequentially one after another, without taking any break. Let the *fitness value* for $m(i)$ be $E^m(b(i))$. $E^m(b(i))$ is given by taking a moving average of b over a certain interval.

3. Generate the next population of compression functions $\{m(i)\}$ that improve $E^m(b(i))$ through Genetic Operators such as Mutation, Crossover, and Reproduction. In the computer experiments described in the next chapter, a Traditional GA (Davis 1990) is used.

4. When the whole population converges in nearly the same function m^{opt}, then the optimal state compression is said to be found.

These processes are outlined as follows: let the number of the candidates for the compression function be M. The robot uses a set of candidates for the compression function successively for N unit times of trials for each; After all the candidates have been used (after N*M unit times of trials), the next candidates for the compression function are prepared according to the performances of each candidate logged during the trials. Then, the successive trials begin with these new candidates. The candidates are equivalent to the strings of GAs, the evaluation of the fitness function is the trials, and the preparation of new candidates corresponds to the genetic operations of GAs.

Computer Simulations and Discussions

Simple computer simulations have been carried out taking a 1-Dimensional 4-legged mobile robot as our example (**Fig.5**). The states $s \in \{1,\ldots,81\}$ are generated by letting $s = \sum_{k=1}^{4} g_k 3^{k-1}$ where g_k takes the value 0, 1, and 2 when the previous movement of the number j leg was m_3, m_1, and m_2, respectively. For the sake of simplicity, the dynamics of the robot are not included in the simulations, and the ground is assumed to

be flat; only the inclination of the ground is considered. These experiments started with a uniform probability table; thus, the robot began with random strides and thereby converged into regular gaits.

Experiment 1: The first simulation is to observe whether the learning procedure can adaptively generate gaits under a dynamically changing environment. In this simulation, each of the controllers does not use the states s; s is fixed to 1 throughout the simulation. Learning factors are set to α=0.5 and β=0.01. Reward ratio is observed by taking a moving average over 30 unit times.

Figure 6 shows the change of the reward ratio plotted against learning time. The reward ratio corresponds to the rate of gaits that successfully drive the body without interlocking its strides. The simulation starts with the ground at a -15 degree inclination, indicated as (a) in the figure. In this inclination, No.3 leg strides across the point of intersection between the gravity vector from the center of gravity and the ground surface(**Fig.5**). The same situation occurs with No.2 leg when the ground is inclined to a +15 degree. Thus, the gait under a -15 degree inclination must differ from that under a +15 degree inclination.

After 230 unit times, an optimal gait for a -15 degree inclination is given. The gait diagram of the acquired gait is shown in **Fig.7**. At 270 unit times, the inclination is changed to 0 degree, which is indicated as (b) in Fig.6. After 100 leaning steps from there, another optimal gait is given as shown in **Fig.7**. The point (d) in Fig.6 shows that the inclination is changed from -15 degrees to 15 degrees. Again, the optimal gait is given, but it costs more learning time than the others. Thus, the robustness of the proposed learning acquisition of gaits has been demonstrated.

Experiment 2: The previous experiment is for a controller with one state, and this can only acquire a gait that is in a constant state of movement at each unit time. To realize a gait that contains a period of suspended movement, specifically, a gait that contains the action a_2, the controller must observe the states for all the other legs. One typical situation where a gait must contain the action a_2 where only a combination of two legs are allowed to be in contact with the ground at a time. We refer to this condition as C3. Experiment 2 was carried out to observe the effect of the introduction of state observation. Here, the critic unit is designed by letting $(Q_{31}, Q_{30}) = (1, 0.5)$ and then, by letting $b = Q_{1C1} \wedge Q_{2C2} \wedge Q_{3C3}$. Controllers without state observation were prepared for comparison. Simulation parameters are the same as for experiment 1. **Figure 8** illustrates that, although the controllers with only one state could get a higher reward ratio in the earlier steps of the simulations, they did not converge into an optimal gait; while the controllers that observed 81 states could acquire the solution. **Figure 7** shows the diagram of the gait obtained.

Experiment 3: From the result of the previous simulation, controllers with large state observation should be able to adapt to more complicated conditions, but their rate of convergence is slow. Experiment 3 was carried out to acquire a state compression function under the same conditions as in experiment 2; and to demonstrate the effectiveness of introducing this compression function in improving the time needed to achieve convergence. Parameters for the learning scheme are set to $\alpha = 0.8$. The traditional Genetic Algorithm (Davis 1990) is applied by setting the parameters to 50% crossover ratio, 1% mutation ratio, 6 cutting positions, 12 chromosomes per

population, and 3000 unit times of trials for each of the simulations. The number of the states after compression is set at 8; 81 states are compressed into 8 states.

Figure 9 illustrates that the reward ratio is gradually increased through the generations, and that the optimal gait is given after 6 generations. The improvement of convergence by the introduction of the compression function is demonstrated in **Fig.10**. In this experiment, the inclination of the ground is changed from +15 degrees to -15 degrees during the trials. Plot (A) in this figure shows the result for the controller that used the state compression function acquired by the Genetic Algorithm, and plot (B) shows that for the controller that observed the whole 81 states without compression. The results show that the state compression function is effective in reducing subsequent learning time caused by changes in the environment.

Conclusion

A method for acquiring the skill of walking for a legged robot by on-line learning is proposed that mimics both the individual learning and the evolutional learning of life creatures, which we should like to refer to as an A-Life approach. We propose a distributed controller associated with each leg that learns whether to trigger the stride of its leg or not according to the observed states. One difficulty in taking such a state-oriented approach to the learning mechanism is that we have to reduce the explosively huge state spaces. We demonstrate how Genetic Algorithms are incorporated with the acquisition of state compression functions. From the universal and adaptive nature of this approach, we expect that it can be applied to a wide variety of legged robots under uncertain environments or configurations: for example, legged mobile robots that are connected with each other; massively legged machines; or robots that walk unfamiliar environments such as the inside of narrow pipes. Practical testing by experimental hardware and proof of convergence of the proposed methods remain for further research.

References

Adachi, H., Koyachi, N., and Nakano, E. 1988. Mechanism and control of quadruped walking robot. *IEEE control system magazine*. 8, 5, 14-19.

Brooks, R.A. 1989. A robot that walks; emergent behaviors from a carefully evolved network. *Neural Compt.* 1, 2, 253-262.

Davis, L., ed. 1991. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold.

de Garis, H. 1990. Genetic programming. Evolution of a time dependent neural network module which teaches a pair of stick legs to walk. *ECAL 90*. 204-206.

Goldberg, D.E. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley.

Hirose, S., and Knieda, O. 1991. Generalized Standard Foot Trajectory for a Quadruped Walking Vehicle. *J. Robotics Research*. 10, 1, 3-12.

Kumar, V., and Waldron, K.J. 1989. Adaptive gait control for a walking robot. *J. Robotics Systems*. 6, 1, 49-76.

Maes, P., and Brooks, R.A. 1990. Learning to coordinate behaviors. *AAAI-90*, 2, 796-802.

Narendra, K., and Thathachar, M.A.L. 1989. *Learning Automata*. Addison Wesley.

Snaith, M. and Holland, O. 1991. Quadrupendal walking using trained and untrained neural models. *IJCNN-91*. 2, 715-720.

Todd, D.J. 1985. *Walking Machines - An Introduction to Legged Robots*. Chapman and Hall.

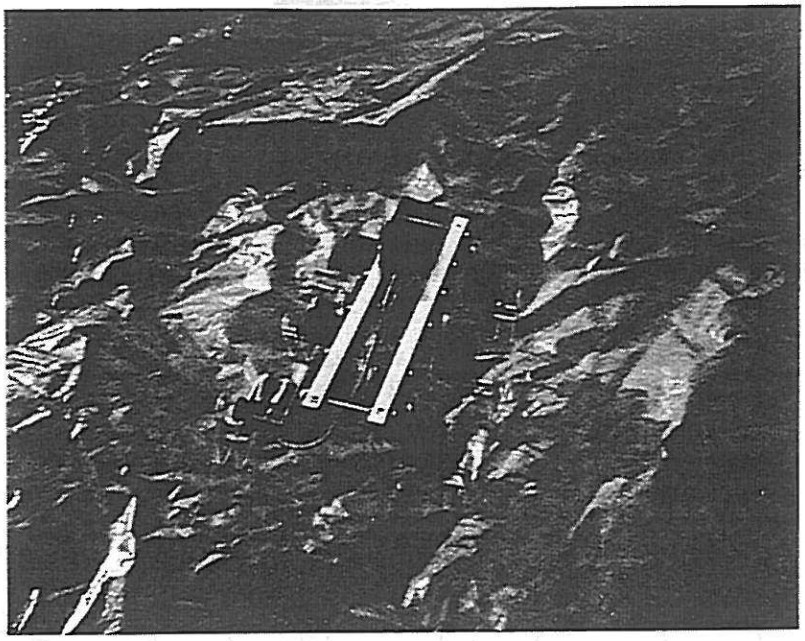

Fig.1 Experimental hardware of a 6 legged mobile robot. Each leg has 2 motor driven joints that realize the vertical movement of the leg. This machine is currently under construction.

Controllers for each of the legs

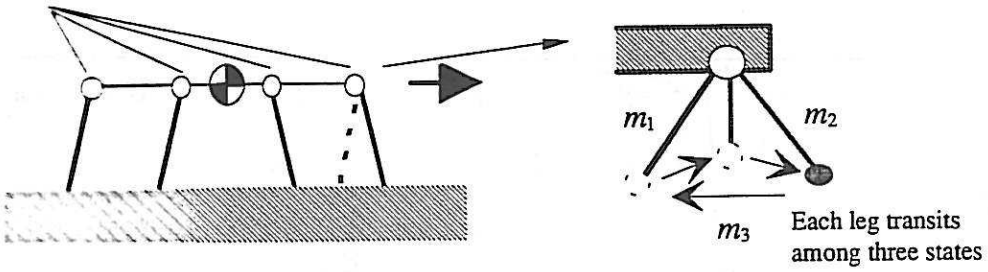

Fig.2 A legged robot of which the legs are triggerd by autonomous controllers. Each leg transits among three states. The transition m3 causes the actual traction force needed for walking.

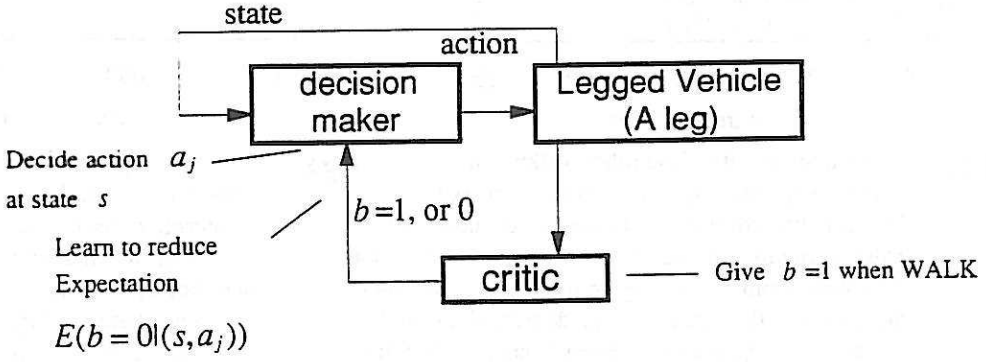

Fig.3 Block diagram of a leg controller. Each controller independently observes sensory input. The controller has its own critic. Therefore, any number of these controllers, that is, any number of legs are possible for such a robot.

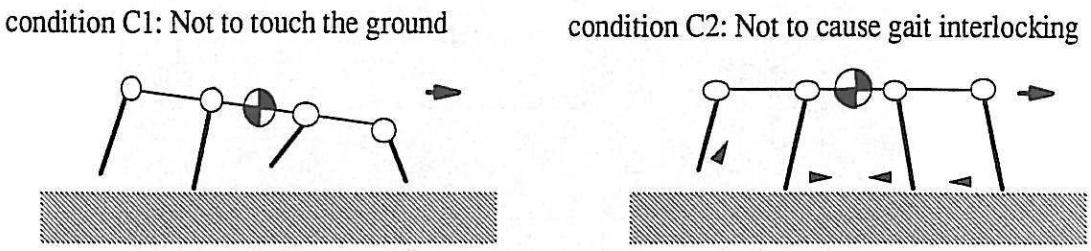

Fig.4 Two decision criteria for walking. C1 means that the robot did not maintain static stability; and C2 means that ambulation did not cause traction force because of interlocking.

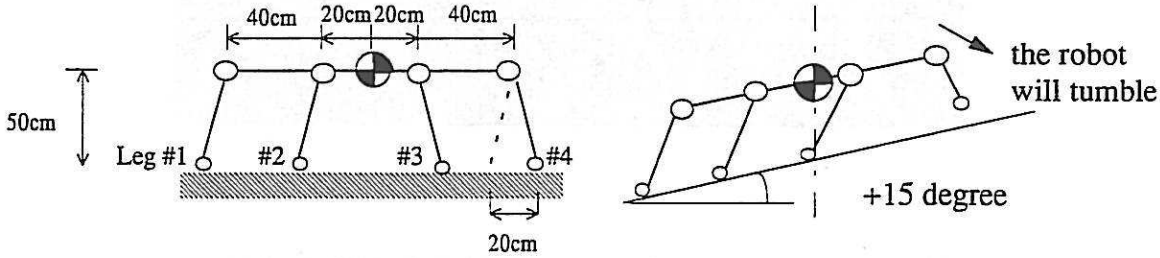

Fig.5 Left hand of the figure shows the configuration of the simulated robot. Only 1-dimensiond strides are considered in this simulation. Right hand of the figure illustrates this robot walking a slope inclined at 15 degrees. The gait shown in this figure is not stable even though it was stable when the robot walked on a flat surface. Thus, a robot walking between flat and sloping should adaptively change its gait.

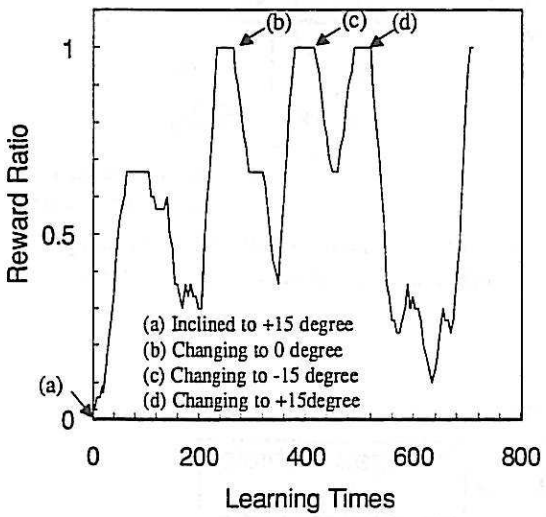

Fig.6 Simulation results of the robot walking in a dynamically changing environments. The axis 'Reward ratio' corresponds to the rate of success of walking. In this simulation, the inclination of the groud is changed 4 times: Starting from a +15 degrees slope, changing to a flat surface, a -15 degrees slope, and finally a +15 degrees slope. Even though the learning agents did not observe sufficient sensory inputs, this figure shows that successful gaits were given after only short periods of trials.

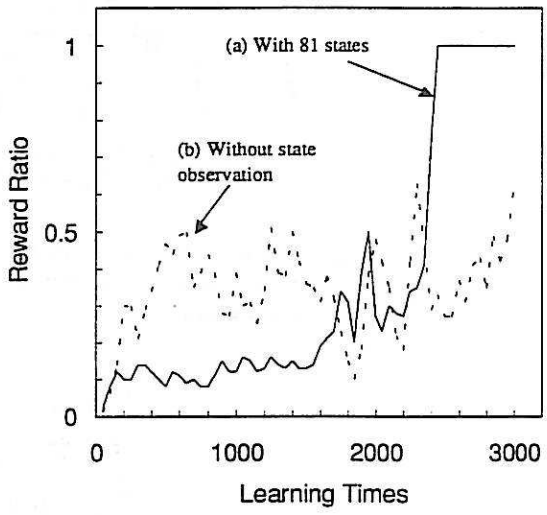

Fig.8 The comparison of the rate of success between two simulations: in simulation (a), a controller observes the states of all the other legs; in simulation (b) the controller does not observe these states. The major difference between Fig.6 and Fig.8 is that, in this simulation, the number of llegs allowed to touch the ground at the same time is restricted to 2. The results show that state observation is necessary for gait acquisition under complex walking conditions.

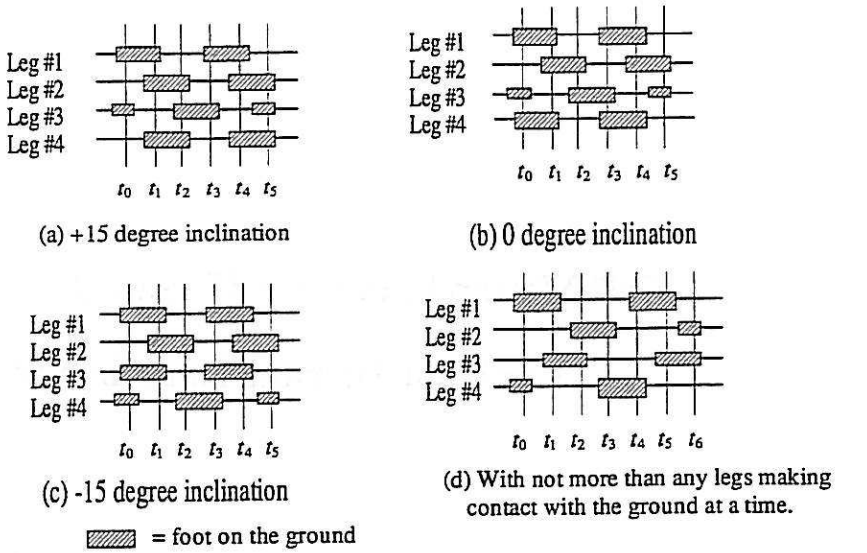

Fig.7 Gait diagrams of the acquired gaits for figures 6 and 8. A horizontal line indicates whether a leg is making contact with the ground (indicated by hatched block) or not (simple line segment) against time passed. The leg numbers correspond to those specified in figure 5. A vertical line corresponds to each of the unit times. According to the gait on figure (a), the robot was succeeding in acquiring static stability by allocating the two of the legs landing phases backward of the body. In figure (c), the same was done but with the landing phases forward of the body. Different from (a) to (c), figure (d) was generated under the condition that not more than two legs were to make contact with the gound at a time. As in (d), the gait should contain the state 'hold the leg off the groud'. This means that, without state observation, this type of gait would not be acquired. This is illustrated by the result shown in figure 8.

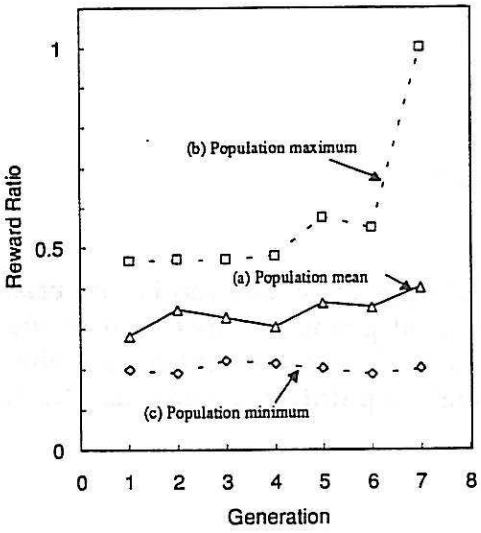

Fig.9 Change in fitness values plotted against generations. The fitness values correspond to the reward ratio after 2000 unit times of learning by the robot that used a chromosome as its state compression function. After 7 generations, an optimal state compression function that could achieve a successfull gait after 2000 unit times was found. This means that this type of state compression function could be successfully introduced to a walking machine using the Genetic search method.

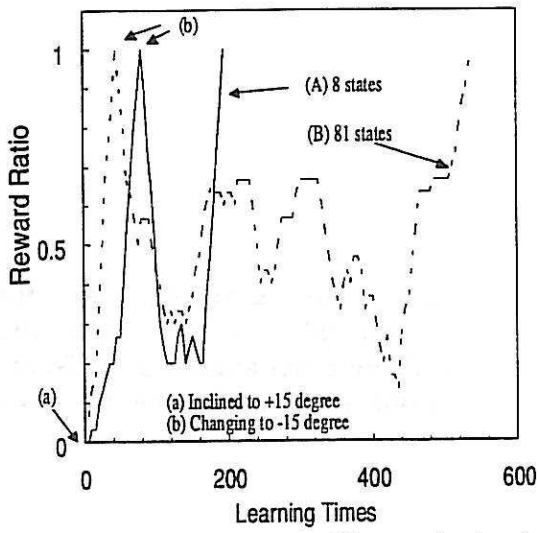

Fig.10 Comparison of the convergence of the two simulated robots. Each of the leg controllers observes the state of all the other legs in the robot. In simulation (A), the robot uses the state compression function generated through the simulation in figure 9 (thus distinguishing 8 states); while (B) did not use the compression function (thus 81 types of states were distinguished). After the robot had acquired its regular gait, the inclination of the ground wassuddenly changed to -15 degrees. The results in this figure show that the robot with the state compressin function is more robust than that without. It seems that the principal pairs under the two inclinations are similar to each other so that the post-learning process terminates within a short period.