Automation and Robotics in Construction XII
E.Budny, A.McCrea, K.Szymanski (Editors)
© 1995 IMBiGS. All Rights reserved.

Adaptive Gait Acquisition using Multi-agent Learning for Wall Climbing Robots

Lawrence Bull[a], Terence C Fogarty[a] and Sadayoshi Mikami[a], James G Thomas[b]

[a]Faculty of Computer Studies and Mathematics

[b]Faculty of the Built Environment
University of the West of England, Bristol, BS16 1QY, England

Abstract

In this paper we present work in progress to examine the use of two machine learning techniques to determine the gait of a wall climbing robot. We describe the use of the genetic algorithm and then that of the reinforcement learning technique Q-learning, within a multiple-agent framework, for this task. We assert that there is one agent responsible for the control of each leg of the robot, where each agent is represented by a rule-based controller. It is shown that it is possible to use these techniques to control the gait of the basic robot.

1. Introduction

Nature is full of examples of both inter and intraspecies cooperation; from the workings of ant colonies to the cleaning symbiosis seen between the Pederson shrimp and the fish of the Bahamas. The fields of Artificial Intelligence and Artificial Life have subsequently focused on this phenomenon as an approach to complex systems in which agents must cooperate to achieve certain goals. In this paper we examine the use of the genetic algorithm (GA) [5] and the reinforcement learning architecture Q-learning [9] when applied to a system of this type; we develop the gait of a wall climbing robot by learning the rules of rule-based controllers for each leg.

The paper is arranged as follows: the next section details the robot platform onto which we aim to implement the approaches. Section 3 firstly introduces the genetic algorithm and describes the computer simulation of the wall climber. Results from using the approach are then presented. Section 4 introduces Q-learning and then presents the results from its application to the task. Finally we discuss our findings.

2. The Wall Climbing Robot

We are developing the control system of a wall climbing robot for the inspection and maintanence of buildings, in an extension to previous work [e.g. 2, 3]. The robot is built on a framework of steel tube and measures approximately 400mm x 400mm x 150mm. It consists of four independently controllable vaccuum feet, one on each corner of its chasis, and a central assembly, also consisting of four vaccuum feet. The corner feet are each connected to a leg consisting of two pneumatic cylinders allowing vertical and lateral movement, i.e. each foot is connected to a leg consisting of two pistons which enables overall backwards/forwards movement by the robot. The central assembly of four feet is able to move forwards and backwards with a larger stroke (200mm). This assembly is also able to rotate, thus enabling the robot to alter its path. The vaccuum feet used are silicon rubber (54mm diameter) with pressure sensors to allow the amount of contact with a surface to be determined. The robot also carries a distance sensor which can return the robot's distance to some fixed point (e.g. the ground); forward/backward movement is detectable. For this paper we are concerned with the developement of a basic gait in the robot using the four corner feet - developement of a complete control system will be presented in the future.

3. The Genetic Algorithm Approach

The genetic algorithm is a population-based search technique based on the principles of Darwinian evolution. Here an initial random population of individuals is created and evaluated on a given task, with each individual receiving some measure of their utility - their fitness. On succeeding generations individuals are chosen to reproduce proportional to their fitness, using genetics-based operators recombination and mutation. The created offspring then share attributes with their (fit) parents, but with the likelyhood of some novel characterstics. In this way the population begins to store information about the search space, eventually converging on individuals with the required traits for the given task.

Genetic algorithms have been applied to a number of areas but it is known that, while it may work for simple systems, the method does not scale up to more complex problems[4]. We address this problem by dividing a system into a number of simpler sub-systems and evolving each sub-system separately. The application of this approach to the evolution of gait in a quadraped is now presented in which we use the GA to evolve the complete rule sets of classifier systems.

3.1 Classifier Systems

Classifier Systems[6] are parallel, message passing, rule-based systems running in discrete time steps. The rules are in the form of condition(s)/action sets, usually specified over the alphabet {1,0,#} (#represents a wildcard), with an associated strength. All external input and internal message passing is done via a blackboard-like message list, the contents of which must satisfy a given rule's condition string(s) before its action can become active on the next time step. Output from a rule-base is generated by all satisfied rules bidding against each other, with the strongest rule's action string turned into an effector to alter the classifier system's environment. All, or some fixed number, of the active rule's action strings are posted onto the message list for the next time step. The classifier system receives feedback from its environment, indicating the effectiveness of its rule-set, which is then used to discover new rules from the most effective current rules.

Within this rule-based framework then, the associations and predictions required for the system to operate effectively in a given environment are produced by the associations and interactions between the rules; the rules must couple together to form chains which together perform a given computation. One rule is said to be coupled to another in the action string of the first satisfies one or more of the condtion parts of the other. Therefore when the first rule is active on a given time step, the second will (probably) become active on the next. These coupled rule sequences allow the classifier system to generate plans/strategies (potentially many in parallel) along side any stimulus-response rules it may require.

3.2 The Simulation

To implement our approach on this system we use classifier systems to control the movement of each leg and coordination is achieved via the use of passed information is between the classifier systems - we allow each leg to know what the other legs are doing on each time step. Such an approach is closely analogous to the biological control mechanism - "Coordination of the legs is not determined by a hierarchically superior control system ... the gait pattern emerges from the cooperation of the separate control systems ... [this] cooperation is based on different types of signals which convey information on the actual state of the sender to the control systems of the neighbourng legs" [1].

The robot's pneumatic leg joints lend themselves nicely to digital control, as they represent an almost discrete system. Therefore in the simulation each leg controller can return one of three actions for each of the two pistons - move forward or backward, or do nothing. Actions such as move forward when a piston is already in the forward position are simply said to have no effect. For this paper we assume a stable gait to consist of at least two opposite corner feet being down. Feet in contact with a surface are said to have enough contact for a vaccuum to have (automatically) formed. Failure to satisfy this stability constraint on a given time step means that all outputs are ignore and the system remains in its previous state. The legs also fail to execute their outputs if any two or more legs try to give a power stroke in opposing directions, or if any legs would effectively be dragged, since they d are attatched to the surface by their vaccuum feet.

Each leg receives its own environmental input message (detector) which contains information as to whether or not the robot moved on the previous time step, which would be available from the distance sensor.

The robot scores on every one of the twenty discrete time step it moves forward, that is every time at least two legs give the correct power stroke (front to back) in a stable gait. All agents receive this fitness. We have also found it neccessary to reward individual legs for executing either a power or return stroke to obtain walking systems from experiments with populations of reasonable size. The maximum achievable fitness for each leg is therefore 40.

We use roulette wheel selection, mutation (0.001 per bit) and multi-point recombination (0.01 per bit) with the GA operating over the genomes of concatonated rules. That is we turn a normal fixed length GA string into a classifier system for evaluation. All classifier systems contain six bi-conditional rules.

The simulation starts with all four feet being on the ground in the back position, but as Cruse et al. [1] point out "the start is particulary difficult when [opposite corner] legs begin from the same x-position". However for the purposes of this paper we have not experimented with varying the starting configuration (as Cruse et al. [1] discuss).

3.3 Results

Figure 1 shows the typical evolutionary progress of 2000 individuals in the developement of gait in a wall climbing quadraped.

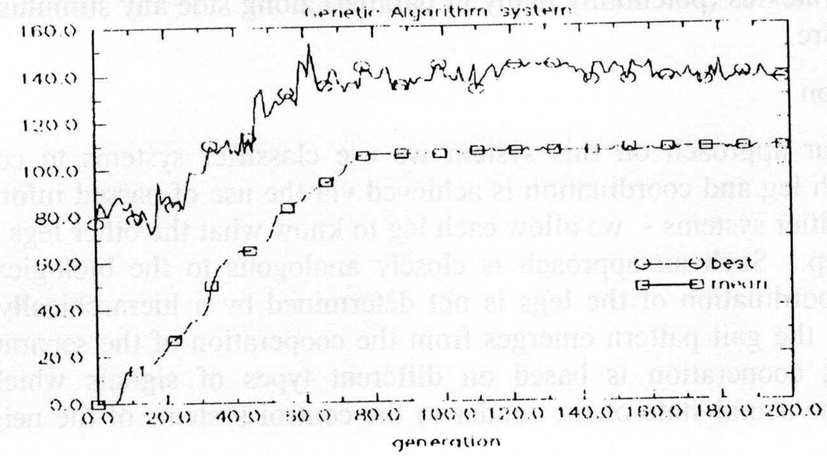

Figure 1.

The results show that on average the GA comes close to generating a group of controllers capable of constantly walking forward and systems have been evolved capable of walking on every time step; we have successfully evolved the "oppostite stance gait", whereby the legs on opposing corners are synchronous.

4. The Q-Learning Approach

Reinforcement learning is the name given to the machine learning technique which is basically learning by trial and error - feedback is given to indicate the degree of utility for asserting an action in a given state. Q-learning learns to select appropriate actions using a Gaussian probability and each potential action's Q-value. These Q-values are learned from the system's environmental feedback. Therefore over time optimal actions are learned through the repetition of state observation, stochastic selection, and the evaluation of the improvement to the system's situation (if any).

Q-learning has been applied to a number of problems but, just as with the genetic algorithm, it suffers from the combinatorial explosion of using it on increasingly complex problems. Therefore we now show that the same multiple-agent approach described above can be used with the reinforcement learning technique also.

4.1 The Simulation

The simulation details for this implementation are largely the same as those in section 3.2, except with regards to the reward mechanism. Here agents (legs) receive a positive reward when the robot as a whole moves foward (0.5) and negative reward (-0.3) when individually their action would result in an unstable gait for the robot. Again the legs are mutually connected.

4.2 Results

Figure 2 shows the average progress of the four legs using the Q-learning reinforcement technique. The results show that no systems capable of walking have been discovered.

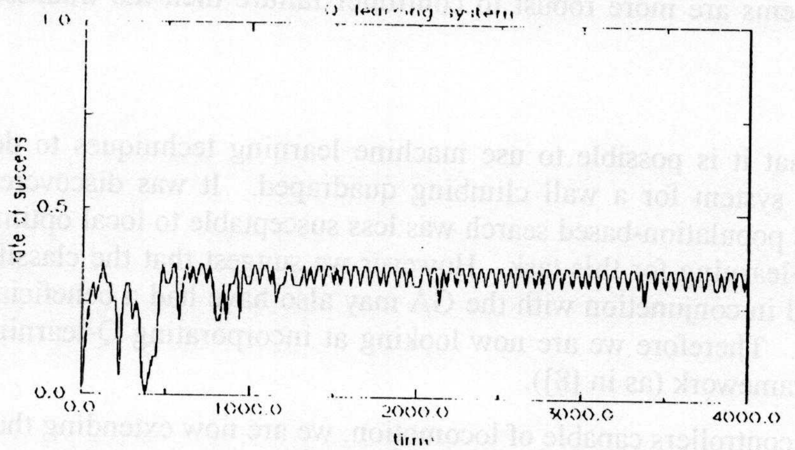

Figure 2.

This is suspected to be due to the way in which Q-learning does its searching, in that the technique uses hill climbing. Therefore the search has become stuck in a local minima. Various parameters were altered with no significant improvement until the number of legs was increased to six or eight, as shown in figure 3 (six legs).

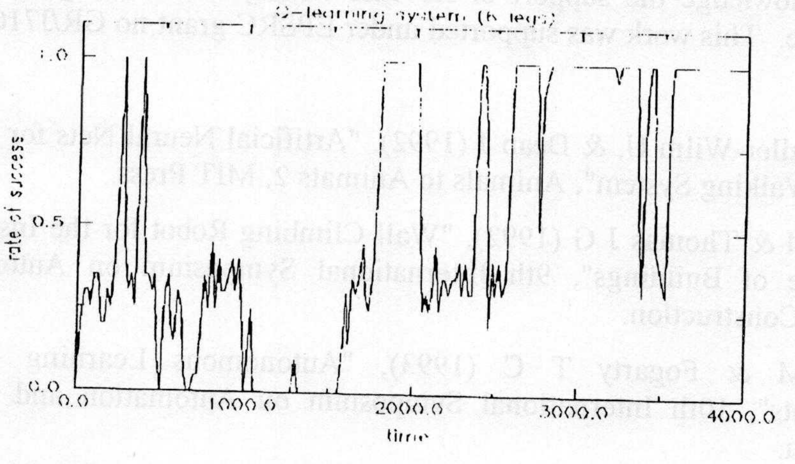

Figure 3.

In these cases the chances of discovering a stable gait are much higher and so the technique is able to successfully produce a walking system. A stable gait here is taken to consist of the two end legs on one side being in contact with the surface, whilst the middle leg on the opposite side is also down. The drift seen, whereby the gait is temporarily lost (e.g at time = 2700), is due to the stochastic element of the technique and can be simply removed using something like a synchronous gait sequence detector, with learning then being turned off.

In the eight legged simulations it was also found that (results not shown) the system could compensate when a randomly chosen leg was disabled once a gait had been discovered. This is obviously significant in terms of the robustness of the control system and demonstrates another advantage to the multiple-agent approach; distributed multiple-agent systems are more robust to controller failure then the traditional single controller model.

5. Conclusion

We have shown that it is possible to use machine learning techniques to develop the distributed control system for a wall climbing quadraped. It was discovered that the genetic algorithm's population-based search was less susceptable to local optima than the hill climbing of Q-learning for this task. However we suggest that the classifier system representation used in conjunction with the GA may also have had a beneficial effect on the search process. Therefore we are now looking at incorporating Q-learning into the classifier system framework (as in [8]).

Having developed controllers capable of locomotion, we are now extending the work in a number of directions. Firstly we are looking at intergrating the central foot assembly which allows the robot to change direction. We are also looking at incorporating noise into the simulations such that perfect contact between the vaccuum feet and the surface are not assumed and hence foot placing strategies will have to be learned. We are also in the process of implementing these controllers onto the real-world platform, with all the caveats that entails for robots[7].

6. Acknowledgements

The authors acknowledge the support of HP labs through their European Equipment grants programme. This work was supported under EPSRC grant no GR/J7168.

7. References

[1] Cruse H, Muller-Wilm U, & Dean J (1992), "Artificial Neural Nets for Controlling a 6-legged Walking System", Animals to Animats 2, MIT Press.

[2] Cusack M M & Thomas J G (1992), "Wall Climbing Robot for the Inspection and Maintenance of Buildings", 9th International Symposium on Automation and Robotics in Construction.

[3] Cusack MM & Fogarty T C (1993), "Autonomous Learning in Vertical Environments", 10th International Symposium on Automation and Robotics in Construction.

[4] Fogarty T C (1992), "Evolving Controllers", IEE Digest, No 106.

[5] Holland J H (1975), "Adaption in Natural and Artificial Systems", Ann Arbor, University of Michigan Press.

[6] Holland J H (1986), "Escaping Brittleness: the possibilities of a general purpose machine learning algorithm applied to parallel rule-based systems", Machine Learning II, Los Altos.

[7] Snaith M & Holland O (1991), "Quadrapedal Walking Using Trained and Untrained Neural Models", International Conference on Artificial Neural Networks (ICANN '91).

[8] Twardowski K (1993), "Credit Assignment for Pole Balancing with Learning Classifier Systems", 5th International Conference on Genetic Algorithms, Morgan Kaufmann.

[9] Watkins C J C H (1989), "Learning with Delayed Rewards", PhD Dissertation, Cambridge University.