

# Experiments on a Pittsburgh-style Fuzzy Classifier System for Mobile Robotics

Anthony G. Pipe, Brian Carse, Member, IEEE, Efthimios Malliris

Intelligent Autonomous Systems Laboratory  
Faculty of Computing, Engineering and Mathematical Sciences  
University of the West of England, Bristol, United Kingdom

Web - <http://www.ias.uwe.ac.uk>

Email - [Anthony.Pipe@uwe.ac.uk](mailto:Anthony.Pipe@uwe.ac.uk)

**Abstract**-- We report on experiments designed to highlight the strengths and weaknesses of an autonomous rule acquisition algorithm for the fuzzy controller of a simulated mobile robot. The algorithm is a Pittsburgh-style Fuzzy Classifier System. These experiments are the results of the most recent work forming part of a larger programme, which has the wider scope of investigating, and comparing, a range of algorithms for autonomous acquisition of mobile robot behavior. The Fuzzy Classifier System paradigm is an elegant and versatile combination of evolutionary and lifetime reinforcement learning based on an underlying Fuzzy Logic structure. It possesses a powerful potential to be a general-purpose linguistically interpretable problem-solver for continuous real-valued domains. We have tested performance and robustness of many controllers using this approach, a sample of which, are presented here. We find that, although the robot controllers can often be quite robust to environmental changes after learning, they also sometimes display critical weaknesses in certain scenarios. Normally, we believe this is the result of a learning phase that misses some crucial sensorimotor aspect that is needed during a post-learning experience. We also compare performance of this algorithm with that of a hand-coded fuzzy controller. We find that, at least for the authors, it is more difficult than one would initially expect to derive a set of hand-coded rules that are as versatile as the best of those acquired by the Fuzzy Classifier System. It is worth noting that the task to be learned here is itself quite simple, there are far more complex problems within the mobile robot domain, thus providing a forward impetus to the work.

**Index terms**-- Fuzzy Logic, Classifier Systems, Mobile Robotics, Genetic Algorithms

## I. INTRODUCTION

Evolutionary Computation and Reinforcement Learning are both powerful techniques that can be utilized in creating entities capable of autonomously acquiring useful rules

about a chosen problem domain. Well-established approaches include those that use;

- a) evolutionary techniques operating at the level of whole rule sets [3,14],
- b) evolutionary techniques that operate at the level of individual rules in a set [2],
- c) other "lifetime" reinforcement learning approaches operating within a single rule set [9,10,11,12,13,15,16].

Although each of the three categories listed above are at a quite mature stage within their own fields, the authors believe that a comparative investigation into the characteristics and performance of these techniques in some appropriate shared problem domain could be a very enlightening and fruitful area for research. We chose to conduct such a programme of work in the area of mobile robotics. This application area has characteristics that are complex but easy to visualise, it is widely known, it is a domain with which the authors have considerable experience, and the results of the research could have some future use in the real world. Since this is a real-valued problem domain, we have chosen fuzzy logic to implement local-cued behavioural control of a wheeled robot, the task therefore is to discover good fuzzy rules for implementing a particular competency in an artificial creature, or animat [18]. We have already conducted a considerable amount of work in the area of "lifetime" reinforcement learning applied to the application domain covered by this paper, i.e., category c) above. However, before conducting a thorough comparison we need to adapt existing algorithms from categories a) and b) to our application domain. This work is already under way, but it is still at an early stage [4,8]. This paper focuses on the structure of, and further tests on, an architecture drawn from category a) above. In order to facilitate the future comparative studies between all three architectures, a common testing harness has been developed; it includes the environmental and robot simulations, as well as the fuzzy logic system that controls the robot. In order to allow the experiments to be ratified, and perhaps extended, by others

– all of this software is available by email or by visiting our web site, both addresses are given at the head of this paper.

Let us focus on category a) and category b) for a moment. The Fuzzy Classifier System brings together a number of powerful modern Soft Computing paradigms. This architecture allows for a rule-based system to interact with a continuous real-valued environment via fuzzy logic. It also allows both Evolutionary and Lifetime Reinforcement learning algorithms to work together in creating and tuning the system. It is an architecture that has great potential for building autonomous self-adaptive systems. Although Classifier System research is in a quite mature state, for Fuzzy Classifier Systems there are still deep underlying issues to be settled for a given class of application. Here we report on our recent investigations to date in using a Pittsburgh-style Fuzzy Classifier System, i.e., an algorithm in category a).

Since we are still in the early stages of this study, we have imposed some restrictions on its scope, in order to focus the experimental work on some specific topics. First, we allow modification of the fuzzy-rule base only, i.e., the membership function details are presumed already to be set by hand *a priori*, and are *not* the subject of tuning or optimization. Second, we have looked at “Stimulus-Response” fuzzy systems only, i.e., there is no internal memory. Third, although environmental reinforcement is temporally linked, it is not delayed.

## II. PITTSBURGH AND MICHIGAN CLASSIFIER SYSTEMS

A major distinction among Classifier Systems is the way that the Evolutionary Algorithm (EA) is applied. With the so-called “Michigan” approach, the individual, as far as the EA is concerned, is a single rule or classifier. An alternative approach, called the “Pittsburgh” approach, maintains a population of rule-sets: each individual as far as the EA is concerned is a complete assembly of rules encoded on an appropriate genotype. Clearly the role of the EA in the two approaches is different, as are the known difficulties. In the case of Michigan-style systems a careful balance must be set between co-operation and competition between individual rules. In the case of Pittsburgh-style systems reinforcement bandwidth is usually smaller and, although it is normally less damaging, genetic crossover can still be a cause of disruption for emerging co-operative rule groups. Indicative works using the Michigan approach include [1,7,17] and works using the Pittsburgh approach include [3,5].

## III. THE APPLICATION

The work described below concentrates on making initial investigations into the abilities of a Fuzzy Classifier System

to extract useful Stimulus-Response (S-R) behavior from environmental experiences. Such a controller must encapsulate an environmentally reactive competency. We have chosen an “investigative” obstacle avoidance competency for these experiments. Because the behaviors are to be S-R, any linkages between rules are made via the environment itself; there is no need to build internally linked behavioral sequences, and therefore the optimization and/or learning tasks are simplified. The current test harness is based heavily in real robot experimentation carried out in our laboratory. Details of the harness are given briefly below. However, the C source code is freely available on request to the email address above, or directly from our laboratory’s web site.

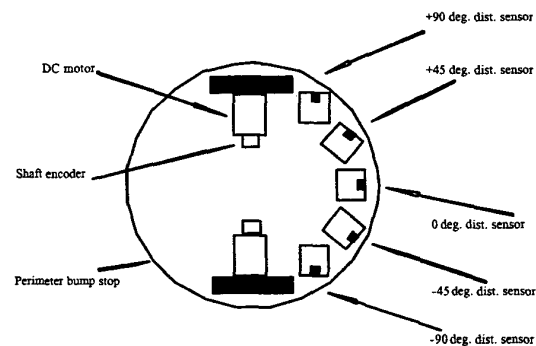


Figure 1: Robot used in experiments

### A. The Simulated Robot

The following is a general description of the simulated twin-wheeled differential drive robot and its sensorimotor apparatus, illustrated in figure 1. The real robots in our laboratory possess two geared d.c. motors with an incremental shaft encoder on each. They are used in a low-level feedback loop to provide position and velocity control. These controllers are coupled through a kinematic algorithm to give a body-centered “virtual steering wheel”. The simulated environment therefore assumes that such a low-level control system is present, allowing control to be effected by an equivalent steering angle and forward velocity. In this work the robot travels through its environment with a constant forward speed of 0.1 m/s and a maximum continuously variable turning speed of 0.5 rad/s. The robot has an array of five distance sensors. The simulation supports a simple point-to-point measurement, to which noise and bias errors may be added if required, these are based upon ultrasonic sensors used on our real robots. The set of distance measuring sensors form a five element array, set at the following angles from the “straight ahead” position: 0°, 45° to the left, 90° to the left, 45° to the right, and 90° to the right. The sensors have an 8-metre maximum sensing range, and are intended for obtaining a local-cued environmental “signature”. A fuller description of the kinematic details used to generate the simulation of

movement and of the type of distance sensors are also available via email or at our web site.

### B. The Simulated Environment

The environmental mazes are set on rectangles of any size although, for the experiments reported on in this paper, they are square, being 10metres on each side. Any number of rectangular obstacles, of any dimension, may be placed in a maze. The start position may also be anywhere inside the maze. It should be stressed that choosing rectangular shapes for the obstacles and the maze was purely an expedient in generating the maze simulation. The robot itself has no such restrictions in its sensory or motor parts. All measurements made and movements executed by the robot are continuous real valued, so for this simulation there is no concept of a “grid” or discretised state space.

### C. Local-Cued S-R Behaviors using Fuzzy Logic

In the work presented in this paper, the fuzzy membership functions are fixed beforehand for both the input and output spaces, rule acquisition is limited to the creation and deletion of rules. When active as the robot's controller the Fuzzy Logic System (FLS) is run through one forward pass every simulated 100ms clock cycle, providing an updated steering angle for that period. The fuzzy controller has five inputs, one from each of the distance sensors and a single output defining steering angle. If fuzzy rule strength falls below a minimum threshold, then motion continues on a “straight-ahead” setting, so that minimally-active rules are not able to influence the steering control.

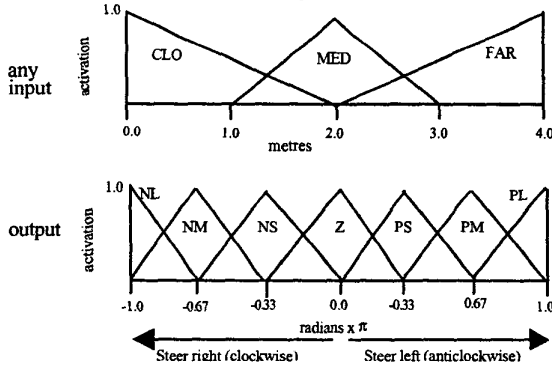


Figure 2: Membership functions

The FLS is a “Mamdani”-style system [6]. A conventional distribution of unit-height triangular membership functions was chosen. All functions were identical and equally spaced, with the exception of each function placed at the end of the range of an input or output, as shown in figure 2. For fuzzy AND a product of membership function activations was used for a given rule as opposed to the simpler MIN operator, since it requires little extra processing and is known to produce superior interpolation properties.

Defuzzification was performed by conventional centre of gravity calculations. The use of 3 membership functions at each input and 7 at the output was established during previous research as being appropriate for this type of fuzzy controller in this application [12,13] and incorporated into this test harness.

Each fuzzy rule was of the form shown in table 1. Each of the six fields is an integer specifying a fuzzy membership function to use for that input or the output in forming a rule - counting from left to right on each graph shown in figure 2 (i.e. the interval (1-3) for each input and (1-7) for the output).

Table 1: format for a fuzzy rule

0	45L	90L	45R	90R	OUT
---	-----	-----	-----	-----	-----

All rules in our previously published early experiments [8] used an alphabet in which every rule had to specify a membership function for each input, i.e., no “don’t care” symbol was used. This was done specifically to enable an investigation to be undertaken into the possibly beneficial generalization effects of allowing “don’t cares”. Results from these investigations were reported on in [4]. Indeed, for the problem domain as it is presently formulated, these beneficial generalization effects certainly outweighed any problems of “over generalization”. Consequently, “don’t cares” were included in all of the experiments reported here.

## IV. FURTHER CLASSIFIER SYSTEM DETAILS

In the “Pittsburgh”-style approach, an evolutionary algorithm acts upon whole sets of rules. The rule sets are evaluated for fitness by running a trial of the robot through a chosen simulated environment for each rule set in the population. When all rule sets have been evaluated in this way, a conventional Genetic Algorithm (GA) applies its operators to produce the next generation. This continues until, either the process is halted by the designer, or the maximum number of generations is reached. The rule set size was 50 for each individual, with a population size of 40 rule sets. Crossover was single-point, with a probability of 0.9. Mutation was two-point, one in the input space, and the other in the output space. The Selection operator was “Roulette” wheel with “elitism”.

## V. EXPERIMENTAL EVALUATIONS

In all of the experiments below, the acquisition of an obstacle avoidance competency was the overall aim. We desired this competency to include a tendency to explore environments, since we have wider aspirations for it as part of a latent learning behavior in the future. In this context, investigation of the environment must be encouraged for

such a competency to be useful otherwise a stationary robot could be deemed highly fit for the purpose. Therefore the fitness functions for both architectures included a measure of the maximum "straight-line" distance traveled by the robot from the start location during a trial in the environment, combined with a measure of traveled distance over a route. These two factors were simply combined by multiplying them together.

Any environmental trial was terminated under either of two conditions, if a maximum time allocation of 200 simulated seconds was reached, or there was an environmental collision before this time.

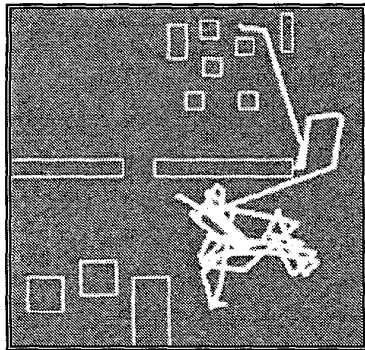


Figure 3: a typical tuned fuzzy controller

The positioning of the obstacles in the maze space that was used for learning purposes (as opposed to robustness testing after learning) in all cases reported here was intended to

represent a "warehouse"-style environment (see figure 3). There is an area at the top-right that is quite "closed-in", a pair of parallel walls in the center with two smaller openings, a collection of larger objects at the bottom-left, and some quite large "open" spaces. This environment was chosen for these experiments because it was found that it has some interesting characteristics that make obstacle-avoiding rule acquisition harder than for, say, "corridor"-style mazes. Different categories of rules are required in the "closed-in" areas, compared with the "open" spaces. This means that the architectures must be more efficient in storing knowledge in their rule sets, in order to have adequate behavioral coverage within the fuzzy logic system.

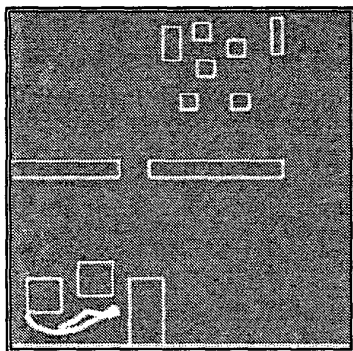


Figure 4: Learned system, bottom-left start

There is only space below to describe some selected results, which illustrate a noteworthy strength or weakness, from a range of experiments that have been conducted.

#### A. Performance in the Maze used for Learning

Figure 3 shows the final maze trial during learning, for a typical fuzzy controller. It was derived by selecting the 4<sup>th</sup> population member at generation 79. The robot trajectory starts at the top right and only stopped when the maximum trial-time was reached, near the center of the maze.

Learning was now turned off, and this controller used as an example for further experiments. First, the robustness of the controller to different start positions was tested. Two examples of this are shown in figures 4 and 5. In figure 4 the trajectory begins traversing from left to right of the figure and then doubles back to the left, where collision occurs with the bottom right obstacle. In figure 5 the trajectory traverses down the figure until collision occurs in the central region of the maze.

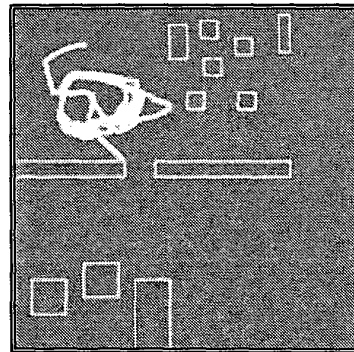


Figure 5: Learned system, top-left start

Figure 4 clearly demonstrates that this controller is not robust to start position in some parts of the maze, although one can see from examination of figure 3 that the robot never traversed this area during learning.

Figure 5, however, shows a more robust behavior. In both of these experiments, small changes in the start position or angle did not affect these results significantly.

Table2: Hand-coded fuzzy rule-base. Each input entry shows the center position of a specified membership function in meters, whilst the output entries give steering angle in units defined in figure 2

INPUTS					
0°	45°L	90°L	45°R	90°R	OUT
0.0	#	#	#	#	1.0
#	0.0	#	#	#	-0.33
#	#	0.0	#	#	-0.33
#	0.0	#	0.0	#	0.33
#	0.0	#	#	0.0	0.33
4.0	#	4.0	#	4.0	0.0
4.0	#	2.0	#	2.0	0.0
4.0	#	2.0	#	4.0	0.0
4.0	#	4.0	#	2.0	0.0
2.0	#	4.0	#	4.0	0.0
2.0	#	2.0	#	2.0	0.0
2.0	#	2.0	#	4.0	0.0
2.0	#	4.0	#	2.0	0.0

Figure 5 also illustrates another aspect to this problem domain, which was evident in a large number of trials. This type of collision finished a very large proportion of trials during learning, and could be a facet of a sensorimotor problem. With the sensory equipment provided to the robot, it seems quite difficult to formulate robust behavior close to the edges of obstacles.

A hand-coded fuzzy controller was designed, after a number of iterations. It finally only consisted of 13 rules, as shown in table 2. This took some time to develop, and the authors "cheated" by inspecting a number of successful learned rule

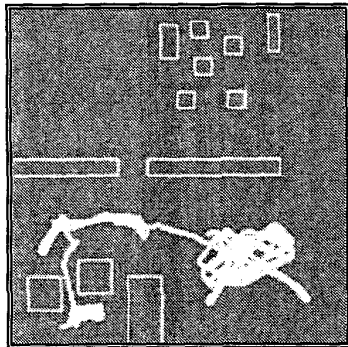


Figure 6: Hand-coded system, bottom-left start

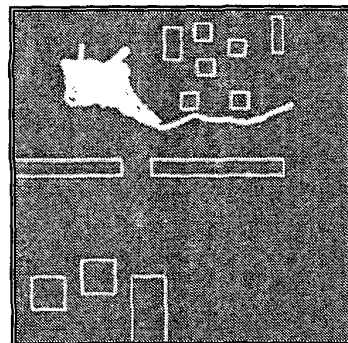


Figure 7: Hand-coded system, top-left start

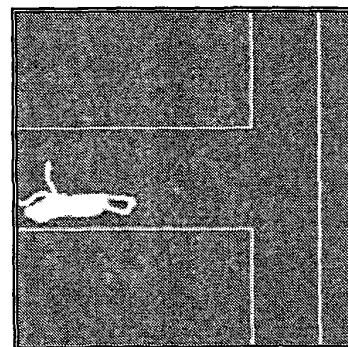


Figure 8: Learned system, start toward left-wall

bases for common rule types. It worked very well in the "open" spaces of the original "warehouse" maze, as shown by figures 6 and 7 where, like figure 3, the simulation was stopped because of maximum time rather than because of collision.

However, despite these apparently robust results, if the hand-coded controller was started at the same position as that used during learning for the Fuzzy Classifier System, results were very poor. In fact the robot always collided quickly with any obstacle in the local area of the top-right corner of the maze when started anywhere in this "closed-in" region. After many attempts, with much larger hand-coded rule bases, the authors gave up on trying to develop a set of rules that would operate successfully and robustly in both regions of the maze.

Some tentative observations can be made about this. Although the learned controller was less robust in "open" spaces, it could be that a large proportion of its rules were being used in subtle co-operative ways for operating in "closed-in" spaces. Further, perhaps there is conflict between rules established for this purpose, and those established for "open" space operation, making learning harder. This is clearly an area for future work.

### B. Performance in a T-Junction Maze

Further tests were carried out on a maze different from that used for learning. It was a "T-junction" maze. It was also set on a 10metre square base, so this was a junction on a quite large scale compared with the size of the robot and the "closed-in" spaces of the "warehouse" maze. Performance of the same learned fuzzy controller as that used above, from the same start position, but different angles are shown in figures 8 and 9. In both cases the trajectory starts on the centre-left of the figure, and ends in collision with the left maze wall.

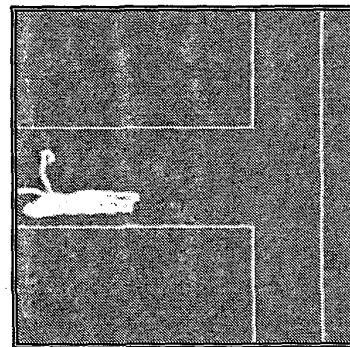


Figure 9: Learned system, start away from wall

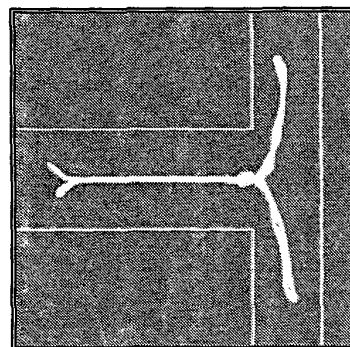


Figure 10: Hand-coded system in T-junction

Unsurprisingly, in this "open" environment, the hand-coded rule base performs well, as shown in figure 10. The figure shows several overlaid trajectories as the robot moves up and down all corridor spaces until the maximum time is reached.

In both cases it is apparent that, although the robot spends a long time moving around in the left corridor before collision, investigation of the large open space represented by the T-junction itself is not attempted. In fact this apparent resistance to investigating very open spaces was quite common, and may have been a side-product of the learning process on the "warehouse" maze. It is also noteworthy that there is no knowledge in the rule base for handling "cul-de-sacs", since this scenario does not occur in the

## VI. CONCLUSIONS AND FURTHER WORK

The Pittsburgh-style Fuzzy Classifier System was able to derive versatile rule bases, which the authors were not able to match across the whole problem domain of the “warehouse” maze. However, it was possible to hand-code fuzzy rule bases that were superior in the “open” space part of this maze, and in other “open” space mazes. This may indicate that it would be beneficial to separate the “closed-in” and “open” parts of the maze into different fuzzy controller learning problems. The two controllers could then be combined at a higher level by a behavioral module selection mechanism. Some further tentative conclusions can be drawn from these experiments, linked with the discussion above. One must begin by considering that, for this work, the distance sensing membership functions have equally spaced centers at 2-meter intervals. When one also notes that the “warehouse” maze used for learning is set on a square 10-meters on each side, it could be claimed that the dimensions of the “closed-in” area in the top-right are really below the resolution of the sensory system. However, part of the power of a fuzzy system, is its ability to make use of the interplay between multiple partially active rules. This means that, to move around this part of the environment without collision, successful fuzzy controllers are likely to have to make quite subtle use of quite low-activity rule groups. If further experiments confirm these tentative conclusions, then it would shed some light on the reasons why the authors found it so difficult to derive a hand-coded fuzzy controller that would perform well in both the “closed-in” and “open” parts of this maze. Of course another alternative would be to change the distribution of membership functions across their domain of discourse, and this will certainly be considered in future work. For other future work, it is important that we test the learning architecture on its ability to extract more complex competencies. In many cases this will require a solution to the problem of building internally linked behavioral sequences; i.e. a fuzzy rule-base that is able to encapsulate internal state in some form. For other more complex competencies, there will be a need for a larger number of states for each rule antecedent and possibly a larger number of inputs. This would be a worthwhile test of this learning architecture on larger search spaces. Finally, future work will include carrying out further comparisons between this Pittsburgh approach and a Michigan-style Fuzzy Classifier System [8].

## REFERENCES

- [1] Bonarini A (2000). An Introduction to Learning Fuzzy Classifier Systems. In P.L. Lanzi, W. Stolzmann and S.W. Wilson (Eds.), *Learning Classifier Systems- from Foundations to Applications*, Lec. Notes in AI, pp.83-104. Springer Verlag Berlin Heidelberg, Germany.
- [2] Booker L B, Goldberg D E & Holland J H (1989). Classifier Systems and Genetic Algorithms, *AI* 40, pp.235-282.
- [3] Carse B, Fogarty T C & Munro A (1996). Evolving Fuzzy Rule-based Controllers using Genetic Algorithms. *Fuzzy Sets and Systems* 80(3), pp.273-293.
- [4] Carse B & Pipe A G. (2001). X-FCS: a fuzzy classifier systems using accuracy based fitness – first results. In *Procs. Int. Conf. In Fuzzy Logic and Technology, EUSFLAT*, pp.195-198.
- [5] Hwang W & Thompson W (1994). Design of Fuzzy Logic Controllers using Genetic Algorithms. In *Procs. 3rd IEEE International Conference on Fuzzy Systems*, pp.1383-1388, Piscataway, NJ: IEEE Computer Press.
- [6] Mamdani E H & Assilian S (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp.1-13.
- [7] Parodi A & Bonelli P (1993). A New approach to fuzzy classifier systems. In *Procs. of 5th International Conference on Genetic Algorithms*, pp.223-230. San Mateo, CA:Morgan Kaufman.
- [8] Pipe A G & Carse B (2000). Autonomous Acquisition of Fuzzy Rules for Mobile Robot Control: First Results from two Evolutionary Computation Approaches. In *Procs. of Genetic & Evolutionary Comp. GECCO 2000*, pp.849-856.
- [9] Pipe A G & Carse B (1994). A Comparison between two Architectures for Searching & Learning in Maze Problems, *Lecture Notes Comp. Science* #865, Springer-Verlag, Ed. T C Fogarty, pp.238-249.
- [10] Pipe A G, Fogarty T C & Winfield A (1994A). Hybrid Adaptive Heuristic Critic Architectures for Learning in Mazes with Continuous Search Spaces, *Lecture Notes in Computer Science* #866 *Parallel Problem Solving from Nature III*, Springer-Verlag, Ed. Schwefel D, pp.482-491.
- [11] Pipe A G, Fogarty T C & Winfield A (1994B). A Hybrid Architecture for Learning in Continuous Environmental Models in Maze Problems, *From Animals to Animats 3*, Ed. Cliff, Husbands, Meyer & Wilson, MIT Press, pp.198-205.
- [12] Pipe A G, Fogarty T C & Winfield A (1996). An Experiment in Knowledge Abstraction - From Cognitive Maps to Behaviours, *Procs. 2nd Int. Conf. Adaptive Computing in Engineering Design & Control*, pp.65-70.
- [13] Pipe A G & Winfield A (1996). An Autonomous System for Extracting Fuzzy Behavioural Rules in Mobile Robotics, *From Animals to Animats 4*, USA, MIT Press, ISBN 0-262-63178-4, pp.233-244.
- [14] Smith S F (1980). A learning system based on genetic adaptive algorithms, PhD thesis, University of Pittsburgh.
- [15] Sullivan J C W & Pipe A G (1996). Efficient Evolution Strategies for Exploration in Mobile Robotics, *Procs. AISB workshop of Evolutionary Computation*, Lec. Notes in Computing #1143, Springer Verlag, pp.147-161.
- [16] Sutton R S (1984). PhD thesis ‘Temporal Credit Assignment in Reinforcement Learning’, University of Massachusetts, Dept. of computer and Information Science.
- [17] Valenzuela-Rendon M (1991). The Fuzzy Classifier System: a Classifier System for Continuously Varying Variables. In *Proceedings of the Fourth Int. Conference on Genetic Algorithms*, pp.346-353, San Mateo, CA:Morgan Kaufman.
- [18] Wilson S W (1987). Classifier Systems and the Animat Problem. *Machine Learning* 2 (3), pp.199-228.
- [19] Wilson S W (1995). Classifier Fitness based on Accuracy. *Evolutionary Computing* 3(2), pp.149-175.