

“Michigan” and “Pittsburgh” Fuzzy Classifier Systems for Learning Mobile Robot Control Rules: an Experimental Comparison

Anthony G. Pipe and Brian Carse

Intelligent Autonomous Systems Laboratory
Faculty of Engineering
University of the West of England
Coldharbour Lane
Bristol BS16 1QY
United Kingdom
Email: Anthony.Pipe@uwe.ac.uk, Brian.Carse@uwe.ac.uk
Web Site: <http://www.ias.uwe.ac.uk>

Abstract

We extend our previous work on the artificial evolution of Fuzzy Classifier Systems as reactive controllers for mobile robots, to encompass more versatile genotypic representations and more powerful genetic operators. The results are an improvement on our earlier work; in general, better controllers are evolved in fewer generations. However, the more global evolutionary characteristics of the Pittsburgh approach still bias the overall results heavily in its favour. A major weakness in both approaches is the lack of robustness in retaining crucial, but seldom-active rules in the evolutionary population.

Introduction

The “Michigan” and Pittsburgh” Classifier System structures are both powerful methods by which evolutionary learning and lifetime reinforcement can be combined together in creating entities capable of autonomously acquiring useful rules about a chosen problem domain. *Fuzzy Classifier Systems* widen the scope of these autonomous rule acquisition structures to continuous valued input and output spaces. In the “Pittsburgh” approach evolutionary techniques operate at the level of whole rule sets (Smith, 1980; Carse, Fogarty & Munro, 1996). By contrast in the “Michigan” approach evolutionary techniques operate at the level of individual rules in a set (Booker, Goldberg & Holland, 1989).

A comparative investigation into the characteristics and performance of these techniques in some appropriate shared problem domain is an enlightening and fruitful area for research. The work presented here is part of a

larger programme of research, and follows on from first results published in (Pipe & Carse, 2000). In this paper we widen the scope of (Pipe & Carse, 2000), to more powerful evolutionary operators and more flexible genotypic representations. 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, and the results of the research could have some future use in the real world.

We have chosen fuzzy logic to implement 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 (Wilson, 1987). In order to allow the experiments reported here to be ratified, and perhaps extended, by others – all of the test harness software is available by visiting our web site; the address is given at the head of this paper.

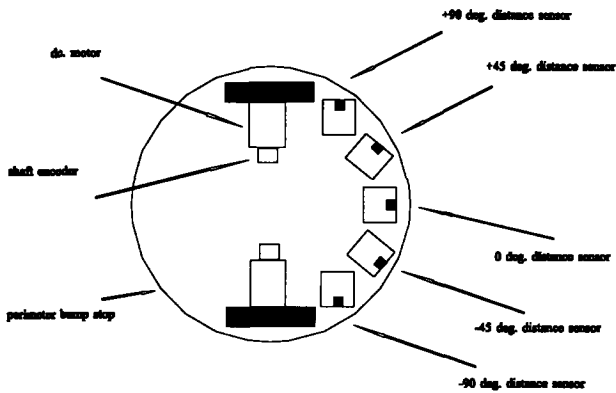
The Application

It is clear from studies in the natural domain that many creatures make use of conscious and sub-conscious cognitive processing for reasoning about the future outcome of planned actions in the environment. It seems clear from recent studies that whilst some reactive behaviours may require “internal state, or *weak internal representations* (Clark & Grush, 1999; Clark & Wheeler, 1998), many others are purely Stimulus-Response (S-R), both being used to good effect in natural and artificial systems.

Our previous paper (Pipe & Carse, 2000) began the comparative work between the two Classifier System approaches by making initial investigations into their abilities to extract a useful S-R behavioural module from environmental experiences. Such a module is an entirely reactive competency, i.e. there is no temporal linkage between the rules. Examples of such competencies are

obstacle avoidance, taking right or left turns at a corridor T-junction with rich sensory feedback, and so on. We have used robot and environmental simulations extensively in our previous research, and continue this approach here, however the test harness is based heavily in real robot experimentation carried out in our laboratory. Details of the harness are given briefly below. However, as mentioned earlier, the C source code is freely available directly from our laboratory's web site.

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-centred "virtual steering wheel".

Figure 1: sensorimotor apparatus of the simulated robot

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°, 90° to the left, 45° to the left, 45° to the right, and 90° to the right, each with a 5 metre maximum sensing range and 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 our web site.

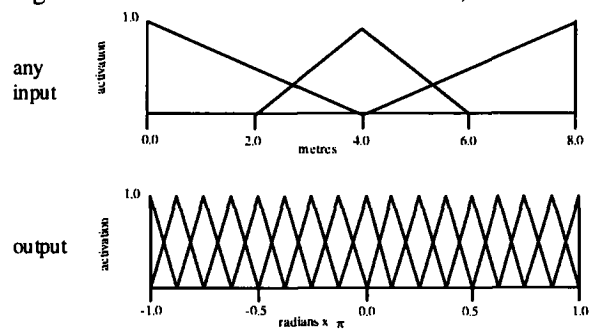
The Simulated Environment

The environmental mazes are set on rectangles of any size, although for the experiments reported 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. If there are start and goal positions, they may also be placed anywhere. It should be stressed that choosing rectangular shapes for the obstacles and the maze was purely an expedient in generating the maze simulation. The animat 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.

When operated in normal mode, simulated animats sense and act in real time; for example velocities and sensory sampling intervals, established from observing actual vehicles in the laboratory, are tied to a real time clock with a period of 100ms.

Implementing Behaviours using Fuzzy Logic

In the work presented in this paper, we focus on rule generation, and therefore the fuzzy membership functions are fixed beforehand for both the input and output spaces. When active as the robot's controller the Fuzzy Logic System (FLS) is run through one forward pass every 100ms simulation 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: fuzzy membership function distributions

The FLS is a "Mamdani"-style system (Mamdani & Assilian, 1975). 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 (Harris, 1992). Defuzzification was performed by conventional centre of gravity calculations. The use of 3 membership functions at each input and 17 at the output was established during previous research as being appropriate for this type of fuzzy controller in this application (Pipe & Winfield, 1996) and incorporated into this test harness. The reasons for choosing these parameters are given in that paper.

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

Table 1: format for a fuzzy rule

Each fuzzy rule was of the form shown in table 1, where each of the six fields is a name, coded as an integer ID specifying a fuzzy membership function (MF) to use for that input or the output in forming a rule. The counting is done from left to right on each graph shown in figure 2 (i.e. the interval (1-3) for each input and (1-17) for the output),

- 0 MF name for front pointing distance sensor,
- 45L MF name for sensor at 45° to the left of front,
- 90L MF name for sensor at 90° to the left of front,
- 45R MF name for sensor at 45° to the right of front,
- 90R MF name for sensor at 90° to the right of front,
- OUT MF name for output angle in radians $\times \pi$ – where positive values indicate a clockwise turning angle from the current orientation

As an extension of our previous work, the (1-3) interval of each input field is augmented by a fourth “don’t care” symbol, that allows more general rules to be created that use a subset of the input data.

A “Pittsburgh”-style Fuzzy Classifier System

An evolutionary algorithm operating at this population based level, is analogous to the well known natural processes of evolution. The rule sets are evaluated for fitness by running a trial of the animat through a chosen simulated environment for each rule set in the population. There is no credit assignment for individual rules in the basic “Pittsburgh” structure. Here, the fitness of each rule set is derived from a fitness function composed of components that deal with final proximity to the goal, length of route taken, and generality in the rule set (related to the number of active rules during the trial).

Implicit in this fitness measure, and the characteristics of the problem to be solved, is reward for those rule groups that are successfully temporally linked internally via the message list of the Classifier System. Therefore, in part, overall strength is based on its ability to link its rules together in useful chains. When all rule sets have been evaluated in this way, the GA applies its operators to produce the next generation of rule sets. These processes carry on until, either the process is halted by the designer, or the maximum number of GA generations is reached. In the experiments reported on in this paper, an attempt is made to modify this basic architecture to reduce the disruptive effects of coarse-grained crossover using individual rule credit assignment. This allows high strength rules to be gathered together on the genome, thus reducing the tendency for them to be split up during creation of the next generation. It is based on the approach described by Grefenstette in (Grefenstette, 1987).

A “Michigan”-style Fuzzy Classifier System

In our “Michigan”-style approach to this problem, an evolutionary algorithm acts upon some subset of a single set of rules. The elements of the evolutionary algorithm’s population are therefore rules of a single rule set, rather than a group of rule sets as in the previous architecture. Again, for this early work, a simple system was created. A GA applies its operators to create a new single rule set at each generation. A group of the highest fitness scoring rules are used as parents for creating a new generation. An “elitism” operator retains a subset of this group into that next generation, but with fitness re-evaluated at that time. Fitness evaluation also now operates at the level of individual rules, carried out during a single simulation trial of the animat in a maze. Each rule’s fitness is evaluated during this trial, the GA then produces the next generation, and so on.

In the experiments reported on in this paper, an attempt is made to enhance this architecture to reduce the conflict between competition for selection and co-operation to form useful rule-chains. The method proposed by Wilson and Goldberg (Wilson & Goldberg, 1989) is used to gather rule groups into “corporations”.

Example Experiments & Discussion

Many experiments have been carried out, unfortunately however, there is not space within the format of this paper to present details of the many evolutionary and fuzzy parameters used in carrying them out, or indeed to present a large number of the test results themselves. For the former, the reader is referred to our earlier paper on this topic; it provides more detail of the parameters used (Pipe & Carse, 2000). With respect to the latter, the reader is

encouraged to visit our website, download the C source files, and conduct experiments of their own devising in order to confirm, or refute, the general tenor of the discussions below.

The main changes to the architectures, relative to our previous paper are;

- inclusion of a “don’t care” state in the inputs space for both algorithms so that general rules using only some of the sensory inputs can be evolved,
- using our “Michigan”-style individual rule fitness evaluation mechanism within the “Pittsburgh” algorithm to allow gathering together of fit rules before genetic crossover is applied,
- extension of the “elitism” operator in the “Michigan” algorithm.

The 2nd and 3rd of the changes above are both efforts to reduce the, sometimes disruptive, effects of genetic crossover. The inclusion of a “don’t care” state in the inputs space gave a general improvement in the robustness of evolved fuzzy controllers for both approaches. For example, a typical controller evolved after only two generations of the Pittsburgh approach is illustrated in figure 3, where the robot starts at the top of the figure.

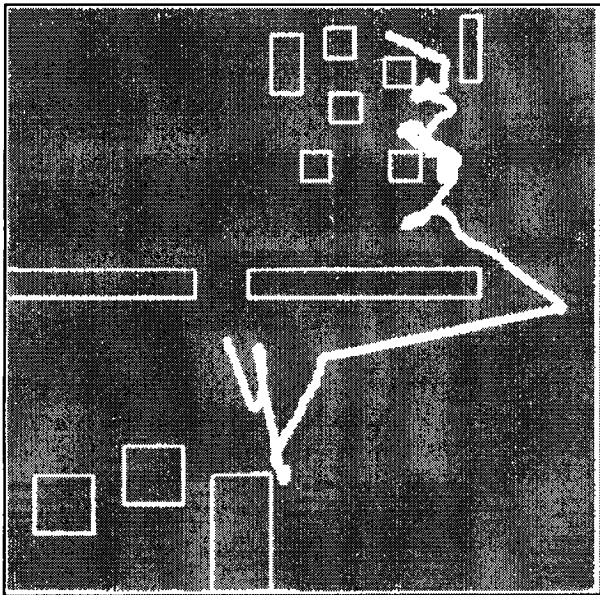


Figure 3: Typical Pitt 2 best controller after 2 generations

The same algorithm without “don’t cares” would typically take 4 to 8 generations to evolve an individual of similar performance. Analysis of the rule structures themselves showed that this modification allowed the rule set to be typically about one quarter of the size for similar performance. The modifications outlined above, that were made to each of the algorithms in an attempt to reduce the

disruptive effects of genetic crossover, did not produce significant difference in performance for either algorithm. However, there may have been a much more disruptive effect at work in each of the algorithms. The generally chaotic behaviour of the evolutionary process, which is more apparent in the Michigan approach but distinctively present in both, is very obvious when tracking the progress of fitness. The Michigan approach, in particular, suffers from a “sawtooth” style progression over generations. The rules set gets gradually better, and then there is a sudden drop to a much lower fitness (i.e. distance travelled without collision in this application). Following the structure of the rule set for examples of this behaviour in the Michigan algorithm reveal that there are two main phases of development that give rise to this characteristic. In the first phase the general fitness of the rule set increases as it becomes more cohesive as a group. Usually there is at least one seldom-active, but nonetheless crucial, rule in this group. Because it is seldom-active it does not accrue a high fitness in the conventional methods for fitness evaluation used in traditional Classifier Systems. In the second phase the rule replacement policy therefore eventually deletes one of these rules and the overall fitness of the controller drops suddenly. For illustrative purposes figure 4 shows the Michigan algorithm at one of the peaks of performance that, in this case, was immediately followed by virtually stationary circulatory behaviour in the next generation.

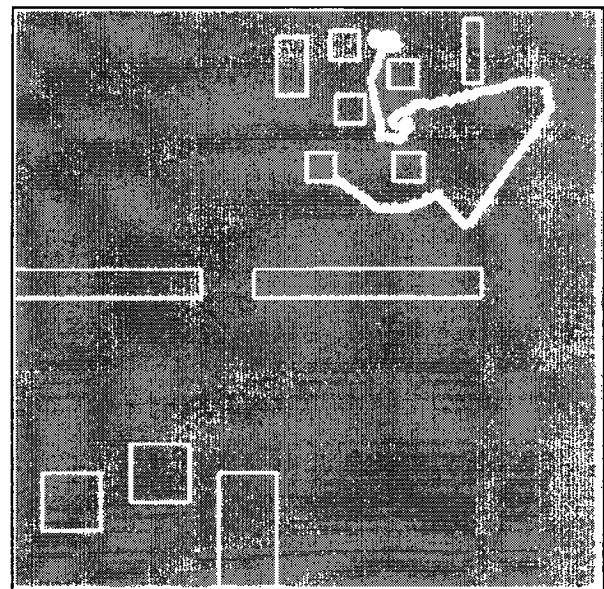


Figure 4: Typical Mich 2 controller at generation 40

Conclusions & Further Work

The main objective of the work presented in this paper was to extend the preliminary results and analysis presented in (Pipe & Carse, 2000) to the use of more powerful evolutionary operators and more flexible genotypic representations. These modifications were intended to confirm the authors' suspicions that the conventional fitness evaluation processes of traditional Classifier Systems do not work well for applications like these. Although more work is to be carried out, the work *has* confirmed these suspicions as far as it has gone.

There are two approaches to be pursued in further work. First, for the Michigan approach, a Temporal Difference reinforcement learning algorithm (Sutton, 1984) should be brought to bear on the single rule set. Its credit assignment policy would help to reinforce seldom-active rules that are crucial to a long trajectory. Secondly, for both approaches, an *accuracy* based fitness evaluation process like that adopted in XCS (Wilson, 1995) needs to be fully investigated to ascertain whether this would be a better method for rating fitness of individuals in the population.

References

- Booker L B, Goldberg D E & Holland J H (1989) Classifier Systems and Genetic Algorithms, *AI* 40, pp.235-282.
- Carse B, Fogarty T C & Munro A (1996) Evolving fuzzy rule based controllers using genetic algorithms, *Fuzzy Sets and Systems* 80, pp.273-293.
- Clark A & Grush R (1999) Towards a Cognitive Robotics, *Journal of Adaptive Behavior*, 7 (1), International Society for Adaptive Behavior, pp.5-16.
- Clark A & Wheeler M (1998) Bringing Representation Back to Life, *From Animals to Animats 5*, Proceedings of fifth International Conference on Simulation of Adaptive Behavior, pp.3-12.
- Grefenstette J J (1987) Multilevel credit assignment in a genetic learning system, in *Genetic Algorithms and their applications: Proc. 2nd Int. Conf. On Genetic Algorithms*, pp.202-209.
- Harris C J (1992) Comparative aspects of neural networks and fuzzy logic for real time control, in *Neural Networks for Control and Systems*, IEE Control Eng. Series #46, chap. 5, Peter Peregrinus, pp.72-93.
- 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
- Pipe A G & Winfield A (1996) An Autonomous System for Extracting Fuzzy Behavioural Rules in Mobile Robotics, *From Animals to Animats 4*, Cape Cod, USA, MIT Press, ISBN 0-262-63178-4, pp.233-244.
- Pipe A G & Carse B, 2000, Autonomous Acquisition of Fuzzy Rules for Mobile Robot Control: First Results from two Evolutionary Computation Approaches, *Procs. Genetic and Evolutionary Computation GECCO 2000*, pp.849-856.
- Smith S F (1980) A learning system based on genetic adaptive algorithms, PhD thesis, University of Pittsburgh.
- Sutton R S (1984) PhD thesis 'Temporal Credit Assignment in Reinforcement Learning', University of Massachusetts, Dept. of computer and Information Science.
- Wilson S W (1987) Classifier Systems and the Animat Problem. *Machine Learning* 2 (3), pp.199-228.
- Wilson S W & Goldberg D E (1989) A critical review of Classifier Systems, in *Proc. 3rd Int. Conf. on Genetic Algorithms*, pp.244-255.
- Wilson S W (1995) Classifier fitness based accuracy, *Evolutionary Computation*, 3(2), pp.149-175.