First Results from Further Experimental Comparisons Between Pittsburgh and Michigan Fuzzy Classifier Systems for Mobile Robotics

A. G. Pipe, B. Carse
Intelligent Autonomous Systems Laboratory
Faculty of Computing, Engineering and Mathematical Sciences
University of the West of England, Bristol, United Kingdom
http://www.ias.uwe.ac.uk
Anthony.Pipe@uwe.ac.uk, Brian.Carse@uwe.ac.uk

Abstract

This paper presents new work carried out in simulation on a performance comparison between the Michigan and Pittsburgh Fuzzy Classifier System approaches to a control problem in mobile robotics. First results from this work, combined with thoughts on our previous work, indicate that changes in the problem formulation can swing the balance between whether the Michigan or Pittsburgh approaches give the best results, in terms of fast and robust convergence on high performance solutions. In the context of the evolutionary and learning tasks overall, the changes could appear minor, in this case modifying the sensor range of the robot. However, such changes can fundamentally modify the processes of a fuzzy controller in a given environment and, thereby, the characteristics of fitness evaluations. However, these early results are not yet statistically significant, a larger experimental programme is required to achieve this.

Keywords: Fuzzy Logic, Learning Classifier Systems, Mobile Robotics, Genetic Algorithms

1 Introduction

This paper presents new work on a performance comparison between the Michigan [7] and Pittsburgh [1,5] Fuzzy Classifier System approaches to a problem in mobile robotics. It is a companion to that reported on in [4] and an extension to that reported on in [3]. Although essential descriptive details of the experimental set up are given here, the reader is referred to [3] for a fuller treatment of this background information, since only that which is specifically mentioned in this paper has been altered.

In that article, a number of different architectures were proposed for the Michigan approach, principally to help counteract difficulties created by the 'co-operation/competition' problem, to which the Michigan Fuzzy Classifier System seems more susceptible than its Pittsburgh counterpart.

The 'co-operation/competition' problem for Fuzzy Classifier Systems can be summarised by the statement that high-strength rules, which co-operate in forming the output during the normal feedforward processes of fuzzy implication, go on to compete for survival under the action of the evolutionary algorithm, thus putting future favourable co-operative rule linkages at risk. In the Michigan case, where the evolutionary competition takes place at the individual rule level, this situation is normally worse than in the Pittsburgh case.

In the series of experiments described here, we shifted focus to the problem itself; autonomous acquisition of an 'investigative' obstacle avoidance competency for a mobile robot in a simulated environment.

2 Application

As in our previous recent work, we have imposed some restrictions on its scope. 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 optimisation. Second, we have chosen an 'investigative' obstacle avoidance competency for these experiments, and for this task we have used only 'Stimulus-Response' (S-R) fuzzy systems, i.e., there is no internal memory. Third, although environmental reinforcement is temporally linked, it is not delayed, i.e., Temporal Difference learning [6] is not used. Details of the test harness are freely available on request to the email address above, or directly from our laboratory's web site.

2.1 The Simulated Robot

The following is a general description of the simulated twin-wheeled differential drive robot and

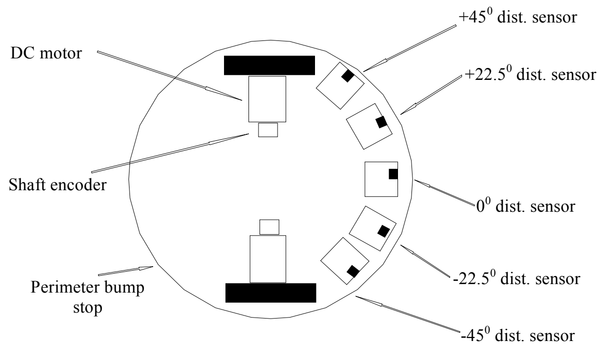

Figure 1: Robot used in experiments

its sensorimotor apparatus, illustrated in figure 1. The simulated environment assumes that control can be effected by a ‘steering angle’ and forward velocity of 0.1 metres/second. The maximum continuously variable turning speed is 0.5 radians/second. The set of ultrasonic distance measuring sensors form a five-element array, set at the following angles from the ‘straight ahead’ position: 0^o, 22.5^o to the left, 45^o to the left, 22.5^o to the right, and 45^o to the right.

2.2 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 10-metres 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, i.e., there is no concept of a “grid”.

2.3 Details of the Fuzzy Controller

In the work presented in this paper, the fuzzy membership functions are fixed beforehand for both 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 to zero, then motion continues on a “straight-ahead” setting. This characteristic of the system is successfully exploited by the Classifier System, as will be illustrated later in the paper.

The FLS is a “Mamdani”-style system [2]. 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.

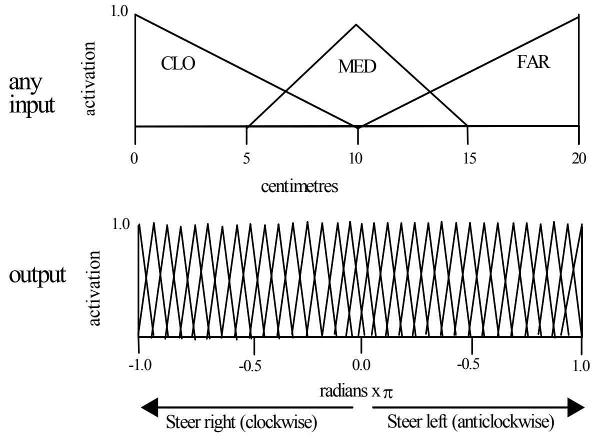

Figure 2: Membership functions

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 33 at the output was established during previous research as being appropriate for this type of fuzzy controller in this application and incorporated into this test harness.

Table 1: format for a fuzzy rule

0	22.5L	45L	22.5R	45R	OUT

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 (1-33) for the output).

2.4 Discussion

The principal change in the problem domain presented in this paper is in the sensor range and its concomitant effect on the input fuzzy membership function placements. To best understand the important point detailed below, it is appropriate to view the form of the environment first, illustrated in figure 3, along with a robot trajectory traversed under control of one of the fuzzy systems detailed in previous work (which starts at the top of the figure) [4].

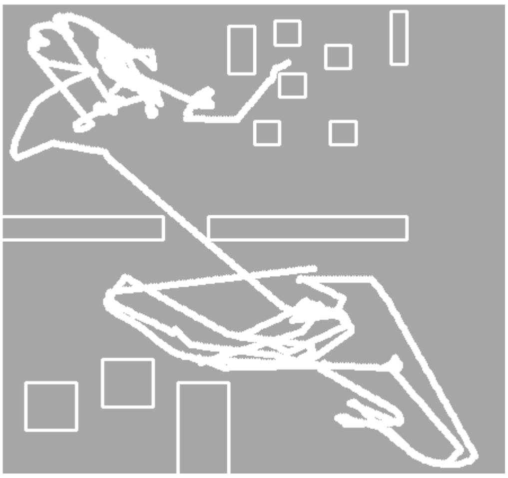

Figure 3: Pittsburgh derived fuzzy controller with 4-metre U/S sensor range

Previously, we have used a 4-metre maximum sensing range. Originally, this value was deliberately chosen so as to allow for sensing in the open areas of the environment, which is set on a 10-metre square. Indeed, one of the characteristics features of these controllers is the often smoothly curving trajectories in the open spaces of the environment. However, as previously reported experimentation progressed [3,4], which included comparisons between hand-coded fuzzy controllers, Pittsburgh and Michigan Fuzzy LCSs, it became clear that this also led to an interesting characteristic to the problem for these architectures, especially in the context of the 'co-operation/competition' problem described above. Because there were only three, equally distributed, fuzzy membership functions across the 4-metre input domain of discourse, the level of individual rule activation in the closed-in areas of the environment was usually very low; effective control of the robot being achieved through subtly shifting low-activity co-operative interactions between rules. As far as the learning problem posed for the Fuzzy Classifier System is concerned, reducing the maximum sensing range, and thereby the width and separation of the fuzzy membership functions, turns out to greatly change the nature of the problem and its solutions. For all experiments reported on here, the sensor range was reduced greatly, from 4-metres to 20-centimetres, with a resulting membership function distribution in the input spaces as shown in figure 2. Now, the radius of the simulated robot is 5-centimetres, so it is clear that this distribution is oriented strongly towards control in the closed-in parts of the environment.

3 Further Classifier System Details

3.1 The Pittsburgh System

In the Pittsburgh-style approach, an evolutionary algorithm acts upon whole sets of rules. In all these experiments, including the Michigan-style experiments, the rule sets were evaluated for fitness by running a trial of the robot five times through the simulated environment from a different start position for each rule set in the population. These start locations were selected to represent the interior of each 'closed-in' region, the 'open' regions, and the corners of the environment close to walls.
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. For the experiments reported on here, the rule set and population sizes were each set to 20. Crossover was single-point, with a probability of 0.9, and respecting rule boundaries. Mutation was two-point, one in the input space, and the other in the output space. If input space mutation was to take place, first a rule was selected randomly, then one of the input components with equal probability, and finally its specified membership function was modified with equal probability of selecting any membership function including the #-value. Independently, mutation in the output space was evaluated. If this was to occur then one of the output membership functions was selected, with equal probability, to replace the existing one. With this quite small population size, a quite large mutation rate of 0.1 was required, and this value was used throughout the experiments reported, evaluated separately for each point.

3.2 The Michigan-Style Systems

In a Michigan-style Classifier System each rule has to have its own fitness value, and therefore the controller-level fitness factors were scaled for each classifier according to an additional factor related to the average activation of the rule during an environmental trial. Since a measure of rule activity is computed as part of the fuzzy inference process anyway, this did not incur a significant processing overhead. The rule activity was simply averaged over the trial and then multiplied by the same factors used as the fitness function for the Pittsburgh architecture.

3.2.1 Sub-Population Niches

An evolutionary algorithm acts upon some subset of a single set of rules in the Michigan-style approach. The elements of the evolutionary algorithm's population are therefore rules of one or more rule sets, rather than a group of rule sets. In order to help alleviate many of the co-operation/competition problems described above, a population 'niching' approach was adopted. In niching schemes, the rules are divided into sub-populations, where each classifier in each sub-population has the same antecedent (including don't cares), so there is a tendency to gather classifiers that would naturally compete together and separate them from rules with which they are likely to co-operate.

In these experiments, there were 243 sub-populations, with 20 rules in each. In all of these experiments we are interested in investigating the ability of the learning algorithm to derive versatile rule-bases, rather than its ability to tackle very large search spaces. For the problem as it is presently formulated, 243 sub-populations (5 inputs each specifying one of 3 fuzzy membership functions $\rightarrow$ 3^5 input states = 243) can cover the entire input search space. However, this still leaves the problems of searching the output space for each rule, the issue of generalisation across the input space, and the subtle problems, mentioned previously, of interplay between rules. All 20 rules in a sub-population begin with identical antecedents, and an output membership function selected randomly from the 33 possibilities. Each rule then has each of its antecedent components potentially modified by randomly changing it to a # "don't care" with 1/5 (i.e. 1/number-of-inputs) probability. The "don't care" policy described above meant that, for these Michigan-style approaches, identical rules could be created in different sub-populations. However, in these experiments this was not monitored. Mutation was carried out in the same way as that described above for the Pittsburgh approach. In all experiments reported below, it was necessary to set the mutation rate to 0.1, i.e., the same as that used for the Pittsburgh experiments.

For the work reported on here, the following approach was adopted for forming fuzzy controllers from a rule-base of this type. A rule was selected from each sub-population in a deterministic manner before each environmental trial, thus making a 243-rule fuzzy controller each time. The robot was then run through five trials of the maze environment (from the same start locations used for the Pittsburgh experiments), acquiring reinforcement. The next set of rules was then formed by choosing again from each sub-population and again the robot was run through the maze. This process continued 20 times, i.e., until all rules had been used once as part of a fuzzy controller, and acquired some environmental feedback. The GA was then run within each sub-population, i.e., across these fuzzy-controller rule-sets, and then the whole process was repeated until some stopping condition was reached.

It is important to note here that, although this approach is clearly *not* a standard Michigan Classifier System one, *operation of the GA and reinforcement are both still at the level of individual rules*. Rules compete within sub-populations and co-operate across sub-populations.

4 Experiments

In all of the work in this type of application domain, and this is true even in simulation, fitness evaluation takes a *very* large proportional of the time, compared with that used by the evolutionary and learning processes themselves. There is, therefore, a justifiable (in the authors' opinions) focus on the number of fitness evaluations required by this or that algorithm in such experimental work; this is the case in the results presented below.

It is worth noting at the outset that, in each of the Pittsburgh and Michigan cases, we have 20 controllers to be evaluated at each generation of the GA. In the Michigan case, they represent a controller composed from a rule drawn from each subpopulation niche; there are 20 controllers to be tested because there are 20 rules in each niche. In the Pittsburgh case, 20 is simply the size of the population of rule-sets.

So, the number of maze evaluations per evolutionary generation has been set to be equal. The principle difference between the two cases is the size of rule-set used to form a controller. In the Pittsburgh case this was set to 20, whereas in the Michigan case there are 243 rules, one drawn from each niche. However, at the end of each evolutionary trial for the Michigan case, as a separate test, we always cropped the rule-size of a selection of controllers to the 20 strongest rules. This very seldom produced any significant variation in performance compared with using all 243 rules, i.e., after the processes of evolution and reinforcement a much smaller number of rules had always been chosen for active use in the controller.

4.1 Pittsburgh System Results

This algorithm was able to find acceptable results in a small number of generations, typically in the first 10 (10 generations x 20 population members x 5 maze trials per fitness evaluation = 1000 maze trials). However, this was quite variable from run to run because of the high mutation rate required to find solutions in a small number of fitness evaluations. The trajectories produced by these controllers were constituted of more precisely controlled arcs in the closed-in regions of the environment, but largely linear motion resulting from zero rule activation in any open-space, until a

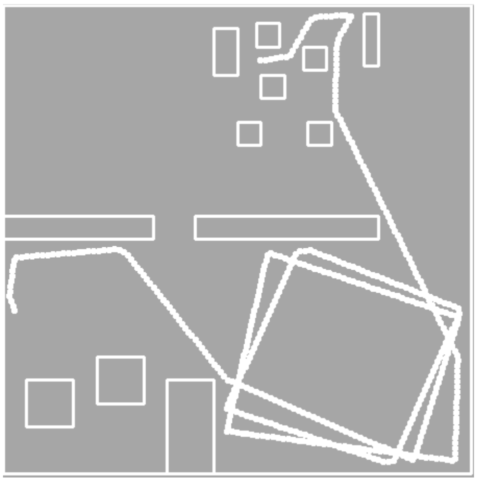

Figure 4: A Pittsburgh derived fuzzy controller with 20cm U/S sensor range

wall became very close. This can be observed in the example result shown in figure 4, which is one created in the third generation after 295 environmental trials (19th population member in the second generation, i.e., 59 fitness evaluations with 5 trials per evaluation).

This trajectory starts in the top-centre of the figure and stops, due to maximum run time for a maze trial, in the centre-left. Note the straight-line motion in the open areas. This was a very robust controller, which we could not induce to fail within the time constraints of a fitness trial, despite many different start positions. However, more generally, the number of maze trials for this architecture to achieve comparable performance in the best controller averaged across ten runs with different random seeds was 637 maze trials.

4.2 Michigan System Results

With a niche size of 20, similar performance was attained in a comparable number of maze trails to those generated by the Pittsburgh style experiments described above. However, the average number of maze trials across ten runs was slightly lower at 532. However, this difference was not statistically significant, since the standard deviation, in each case, was very large.

4.3 Discussion

We were interested in investigating whether, for the problem as it is now formulated, reduction in the number of niches could give an advantage to the Michigan approach, in terms of required fitness evaluations. Reducing the number of niches to 1, i.e., a single 243 rule controller, then disabling crossover and increasing mutation probability to 0.9, generates a form of single-agent random search for a solution. Of course, the same situation could have been created in the Pittsburgh case by setting the population size to 1. By re-starting the algorithm with different random seeds, it was occasionally possible to produce acceptable results within a hundred maze trials, however, this was not a robust scenario, it was more likely, in a given run of the algorithm, that no acceptable solution would be found after several thousand maze trials.

Early results from setting the niche size for the Michigan approach, and the population size for the Pittsburgh approach each to only 4, produce some results which, whilst still not yet statistically significant, indicate that there could be an advantage to small sized Michigan architectures relative to small size Pittsburgh architectures for this problem. In fact, to attain high performance controllers from the Pittsburgh style system with these parameter values it was necessary to increase the mutation rate significantly (to 0.5 in the experiments conducted to test this so far). This caused the convergence dynamics to be quite chaotic, reminiscent of the single-niche Michigan system, where it would occasionally find high performance solutions in a small number of fitness evaluations, but commonly become stuck in a fitness plateau for many hundreds of maze trials. There was no indication of this for the problem as it was previously formulated, where the

Pittsburgh approach regularly outperformed the Michigan approach in terms of fitness evaluations. However, it should be noted that, for those experiments, the greater difficulty of the problem precluded obtaining reasonable results from the Michigan approach using a small number of niches.

5 Conclusions and Further Work

The first results of the work presented here indicate that, for some problems in this area of application, a Michigan Fuzzy Classifier System approach could be competitive with, and perhaps better than, a Pittsburgh Fuzzy Classifier System in terms of robust convergence on high performance fuzzy controllers for a mobile robotics problem in a minimal number of fitness evaluations. Since time spent in fitness evaluations is so dominant for problems such as these, this is a prime focus of this work.

We could tentatively suggest that these results could be due to the reduced level of complex low activation interactions in the closed in areas, relative to the maze-controller problem as it was previously formulated. The case for this suggestion could be supported by the relative ease with which we were able to create a hand-crafted fuzzy controller for this task, compared with the extreme difficulty in the case of the former task, especially in handling the closed-in areas. If this suggestion turns out, after more experimentation, to be true, then one could further suggest that the current problem is somewhat simpler, in a global searching sense, and this could account for the different balance in performance between the two approaches. That is, the Pittsburgh approach *should* do better in a problem that requires greater strength in the global aspects of the search.

However, these are the first experiments resulting from a change in the problem domain. Clearly, a great deal remains to be done. Although some indications are given by the work so far, the principal proposal for further work must be a larger experimental programme that might produce a statistically significant result in this comparative venture between two approaches to the autonomous acquisition of a fuzzy rule-base.

Another important piece of further work in this problem domain is to allow for an uneven distribution of membership functions. The need for this is especially stark in the input spaces, given the discussions in this paper about the previous and current problem domains. However, the output space might also greatly benefit, since it seems intuitively likely that a close distribution of membership functions might only be beneficial in the robot turn directions that are close to zero. In fact, allowing these to be evolvable parameters would be an interesting way to take this work forward.

References

[1] 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.

[2] 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.

[3] Pipe A G, Carse B (2005). Fuzzy Classifier System Architectures for Mobile Robotics: An Experimental Comparison. International Journal of Intelligent Systems, in print.

[4] Pipe A G, Carse B (2003). Fuzzy Logic and the Pittsburgh Classifier System for Mobile Robot Control. Proceedings of EUSFLAT '03, Zittau, Germany.

[5] Smith S F (1980). A learning system based on genetic adaptive algorithms. PhD thesis, Univ. Pittsburgh.

[6] Sutton R S (1984). PhD thesis `Temporal Credit Assignment in Reinforcement Learning', University of Massachusetts, Dept. of computer and Information Science.

[7] Wilson S W (1987). Classifier Systems and the Animat Problem. Machine Learning 2 (3), pp.199-228.