Fuzzy Logic and the Pittsburgh Classifier System for Mobile Robot Control

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

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. The highly cross-coupled and co-operative nature of fuzzy inference systems makes autonomous creation of an optimal rule-base a tough proposition. However, our results show that this architecture can regularly find high-performance solutions that eluded the designers of a hand-coded fuzzy controller.

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

1 Introduction

The research described in this paper was originally motivated by other work being carried out by the authors on environmental cognitive map building in mobile robotics [3]. In this work, a mobile robot builds up a spatial "value" map, via interaction with its environment. Over repeated environmental trials, obstacles and "unpleasant" places acquire low value in such a map, and the locations of rewards acquire high value. This map can subsequently be used for reasoning about future desirable movement sequences in the environment. During acquisition of a map, an ability to carry out latent learning behaviour, i.e., learning in the absence of reward, has been shown to be a worthwhile benefit in both artificial and natural domains [4]. As a part of this larger aspect of our work in mobile robotics, therefore, we decided to equip our artificial creature, or animat [6], with the ability to autonomously acquire such a competency. 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. The findings from an investigation carried out on one type of Fuzzy Classifier System form the focus of this paper.

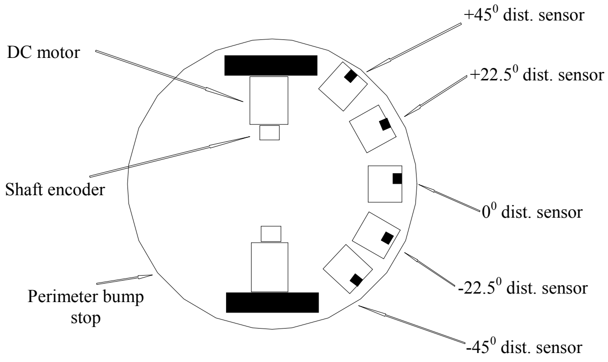

Figure 1: Robot used in experiments

2 The Application

For the work reported on here, 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 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 [5] 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 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. The sensors have a 4-metre maximum sensing range, and are intended for obtaining a local-cued environmental "signature".

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 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, i.e., there is no concept of a "grid".

2.3 Local-Cued S-R Behaviours

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 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.

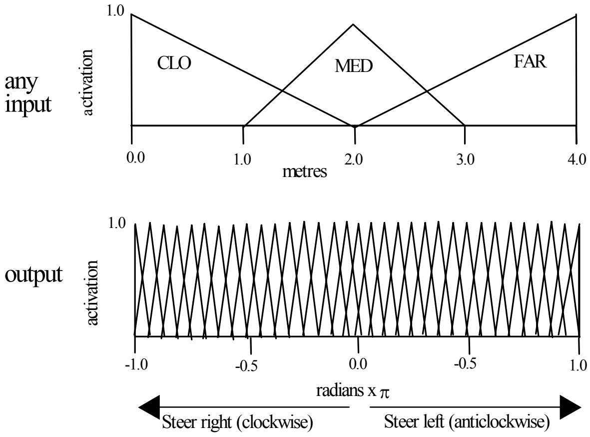

Figure 2: Membership functions

The FLS is a "Mamdani"-style system [1]. 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 33 at the output was established during previous research as being appropriate for this type of fuzzy controller in this application (see [2,4] for example) 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).

3 Hand-Coded Fuzzy-Controller

To set the scene, and allow for some investigations into the nature of this problem, some hand-coded fuzzy controllers were designed. We began with very simple systems, with only a handful of rules. Rule bases such as this *could* operate robustly, in terms of obstacle avoidance. However, they would always produce strongly localised cyclic behaviour, and therefore did not fulfil our requirement for an “investigative” competency.

Table 2: More complex hand-coded fuzzy rule-base. Each input entry shows the centre position of a specified membership function in metres, whilst the output entries give steering angle in units defined in figure 2.

	INPUTS					OUT
Rule	0°	22.5°L	45°L	22.5°R	45°R	
1	0.0	#	#	#	#	1.0
2	#	0.0	#	#	#	-0.1
3	#	#	0.0	#	#	-0.1
4	#	0.0	#	0.0	#	0.1
5	#	0.0	#	#	0.0	0.1
6	4.0	#	4.0	#	4.0	0.0
7	4.0	#	2.0	#	2.0	0.0
8	4.0	#	2.0	#	4.0	0.0
9	4.0	#	4.0	#	2.0	0.0
10	2.0	#	4.0	#	4.0	0.0
11	2.0	#	2.0	#	2.0	0.0
12	2.0	#	2.0	#	4.0	0.0
13	2.0	#	4.0	#	2.0	0.0

A more competent hand-coded rule base is shown above in table 2, consisting of 13 rules. This took some time to develop, and the authors “cheated” by inspecting a number of successful learned rule bases from previous work, looking for common rule types. It worked well, as shown by figure 3, where the simulation starts in the bottom-left “closed-in” corner of the maze and was stopped because of maximum time rather than because of collision.

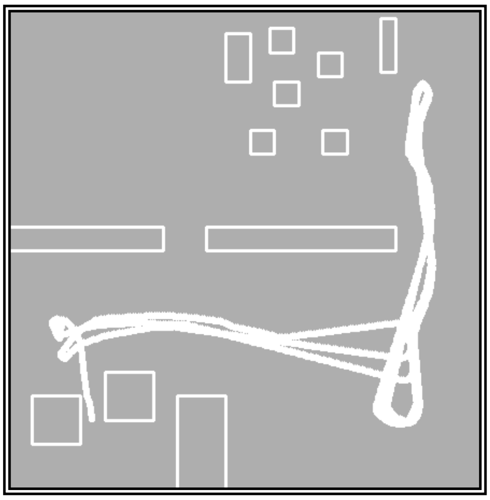

Figure 3: Hand-coded system, top-left start

Despite the apparently robust results, in fact the robot always collided quickly with any obstacle in the more severely “closed-in” local area of the *top-right* of the maze when started inside this region. After observing a number of trajectories like that shown in figure 3, we gained the impression that the “table 2” rule-base tended to perceive the *top-right* area of the environment as a single large obstacle. In fact, if we had only used these results to assess performance of the control system, we would have assumed that the quite coarse sensory resolution, in terms of the input-space membership function size and spread, was preventing successful operation in this area. However, by contrast, all of the highly evolved rule bases *were* able to achieve robust behaviour in *both* of the “closed-in” *and* in the “open” regions using the same basic structure for fuzzy inferencing. Some tentative observations can be made about this. The evolved controllers’ rule-bases were larger than this hand-coded example, and it could be that some rules were being used in subtle co-operative ways for operating in “closed-in” spaces. Further, there may sometimes be conflict between rules established for this purpose, and those established for “open” space operation, thus increasing problem complexity. However, these are only broad observations about the nature of the problem, this is clearly an area for future work.

For fuzzy controllers, co-operation between rules is a fundamental and crucial feature. For a reactive system, the details of the co-operation are highly specific to the current sensory input. At some times, perhaps only a single rule is active, at others some large subset of the rule base may be responsible for determining the system output. This can be seen in

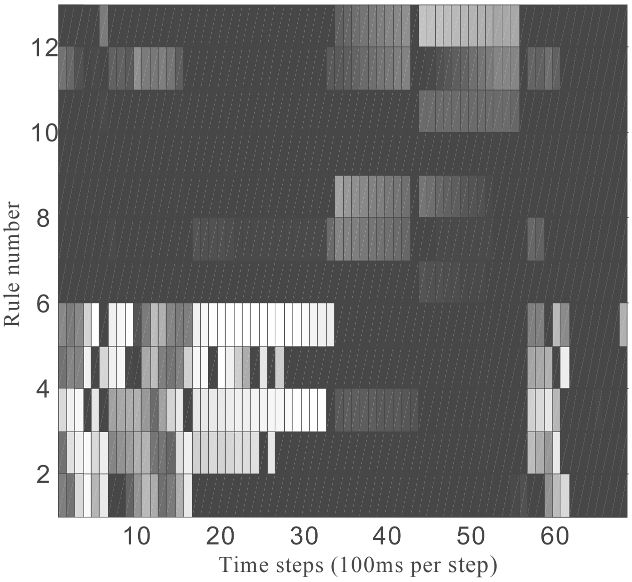

Figure 4: Rule activity plot for the 13-rule hand-coded controller

the "rule activity" plot shown in figure 4. This plot shows activity for the 13-rule hand-coded system specified in table 2, over the first 7 seconds of the trajectory shown in figure 3. Highly active rules are shown in light shading, whilst inactive rules are darkly shaded. Rule numbers are the same as those used in table 2. One can see that the first five rules are highly active together whilst the robot negotiates its exit from the bottom-left region, from steps 0 to 34. The region of the plot from time steps 35 to 58 is dominated by a co-operation between the other rules. This corresponds to the period in which the robot is "emerging" from between the two bottom-left boxes in the environment. The re-activation of the first five rules around time step 60 is coincident with a small radius "360^0 + 90^0 anti-clockwise turnaround" (hidden by subsequent parts of the trajectory) that occurs as the robot heads for the bottom-right of the figure. However, rule interaction is not as simple as it first appears. In the few periods surrounding time steps 15 and 60, there is strong interaction involving rules that straddle the two broad groupings identified above. In fact, if one artificially "disables" active rules in the higher group during these times, the robot very quickly collides with an obstacle. One can observe from this that, even if a rule is a relatively inactive "outlier" involved in determining the output, its contribution can often be crucial to successful behaviour. Of course, this "outlier" rule may be much more strongly active at other times and in different sensory scenarios, but then it may not. Later in the same environmental trial (not shown), in the very open parts of the maze, one could frequently observe partial activity of a gradually shifting majority of the rules in determining the system output. Of course the two main groupings inherent in this hand-crafted system are more easily observed than in the autonomously created highly evolved rule-sets, an example of which is presented later. It is clear from inspecting table 2, that its human creator formulated rules 1 – 5 according to one set of design ideas and the rest of the rules from another. However, this general kind of interaction was observed in every useful hand-coded *and* autonomously acquired rule base that we have looked at so far. The only way to reduce this strongly co-operative effect in a functional system is to make the size of the rule base very small, perhaps only two or three rules. Under these circumstances, the fuzzy controller is always dysfunctional, in the sense of the competency requirements of this application. One can conclude from these observations that there are some complex issues of co-operative rule interaction to be handled by any algorithm that attempts to discover a good rule base autonomously for this application.

4 Further 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 20 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".

As stated earlier, we desired this competency to include a tendency to explore environments. 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 used for all the architectures discussed below included a combination of two "distance travelled" metrics. Firstly, the maximum distance from the start location ever achieved during a trial of the environment was retained. Second, a measure of "journey length" over the route was also accumulated during a trial. These two factors were simply combined by multiplying them together to

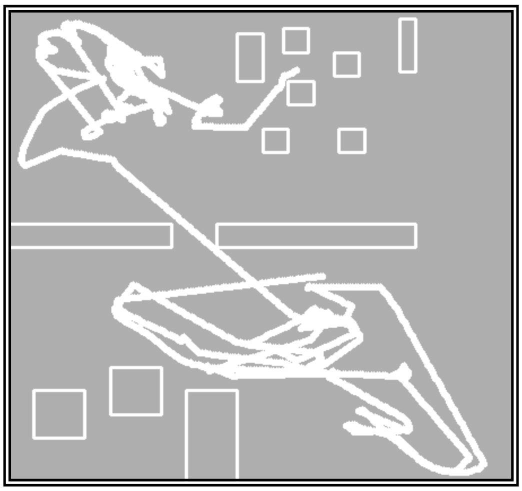

Figure 5: Pittsburgh derived fuzzy controller

form a part of the fitness measure. In all experiments presented below, these values were averaged over five trials with different start locations. 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.

Any environmental trial was terminated under either of two conditions, if a maximum time allocation of 200 simulated seconds was reached (i.e. 2000 time steps), or there was an environmental collision before this time.

5 Fuzzy Classifier System Experiments

To save space, we show only a single typical maze-trajectory from an environment, figure 5. This figure is actually showing a controller derived by selecting the 14th population member at generation 15. At generation 15, the population had still not converged, but this controller was representative of the type to which the Pittsburgh approach generally converged later in a run. The robot trajectory started at the top-right and only stopped when the maximum time of 200 seconds was reached. For this experiment, learning had been turned off; with the start location different to any of the five used for fitness evaluation trials. The rule base is shown in table 3.

Table 3: Pittsburgh derived fuzzy rule-base. Details as given in table 2.

	INPUTS					OUT
Rule	0°	22.5°L	45°L	22.5°R	45°R	
1	2.0	2.0	2.0	2.0	4.0	-0.31
2	0.0	2.0	4.0	0.0	2.0	0.38
3	4.0	0.0	0.0	4.0	#	0.00
4	4.0	0.0	2.0	#	0.0	0.12
5	2.0	4.0	2.0	0.0	#	0.69
6	2.0	2.0	#	4.0	2.0	-0.88
7	2.0	0.0	2.0	0.0	0.0	-0.56
8	0.0	0.0	#	#	0.0	0.94
9	2.0	2.0	4.0	4.0	2.0	0.69
10	#	2.0	0.0	0.0	2.0	-0.81
11	4.0	0.0	#	2.0	4.0	-0.50
12	2.0	0.0	2.0	#	2.0	-0.81
13	2.0	#	4.0	0.0	2.0	0.38
14	4.0	#	#	2.0	2.0	0.12
15	2.0	2.0	#	#	2.0	-0.06
16	2.0	#	2.0	2.0	0.0	-0.69
17	#	2.0	4.0	4.0	2.0	-0.81
18	2.0	4.0	2.0	2.0	4.0	-0.19
19	0.0	#	0.0	2.0	4.0	-0.50
20	#	2.0	2.0	2.0	0.0	0.38

A “rule activity” plot (not included due to space restrictions), for the trajectory of figure 5, showed that there were significant periods of “zero” activity in the rule base. One must remember, that if all rules return zero activity for a given sensory input, then the low-level robot controller simply travels in a straight line at the current orientation for this time step. In fact, this was a very commonly observed effect, and corresponds to the evolutionary component of the Classifier System exploiting a useful characteristic of the problem-space, thus allowing a policy that favours rules that are active close to obstacles, and inactive in the open spaces. This would seem to be an efficient use of the available number of rules. As in the case of the hand-coded rule base, the presence of a highly active rule group, in this case rules 14 – 16 (numbers as in table 3), should not preclude one from appreciating the subtle importance of the other rules. In many cases, it was weak firing, with and without the activity of rules 14 – 16 that resulted in collision-free and investigative movement; as we found by selectively deleting these rules.

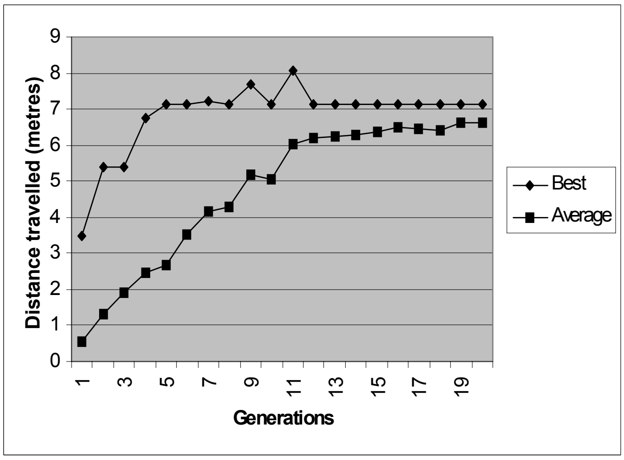

Figure 7: a typical convergence path for the Pittsburgh System

Figure 7 shows a typical convergence characteristic for this architecture. Data has been averaged over 5 separate runs of the system, with the population initialised using different random seeds each time. For these runs, the population size was 40 and the experiment was run for 20 generations, thus resulting in a total of 800 sets of fitness evaluations for each run (remember that a fitness evaluation set is 5 maze-trials from different start locations). In subsequent generations, the population converged fully on a single solution, with only minor disruption from the low mutation rate.

6 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. 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 behavioural 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 centres at 2 metre intervals. When one also notes that the "warehouse" maze used for learning is set on a square 10 metres 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. However, these results as they stand are a testament to the powerful combination of fuzzy logic and evolutionary computation.

Future work will include carrying out further comparisons between this Pittsburgh approach and a Michigan-style Fuzzy Classifier System (see [2] for our work to date).

References

[1] 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.

[2] Pipe A G, Carse B (2002). First Results from Experiments in Fuzzy Classifier System Architectures for Mobile Robotics, Procs. Parallel Problem Solving from Nature VII. Lecture Notes in Computer Science #2439, Springer Verlag, Granada, Spain, pp.578-587.

[3] Pipe A G (2001). An architecture for learning "potential field" cognitive maps with an application to mobile robot navigation, Journal of Adaptive Behavior 8(2), pp.173-204.

[4] Pipe A G (1998). An Architecture for Building "Potential Field" Cognitive Maps in Mobile Robot Navigation, Procs. IEEE Int. Conf. on Systems, Man & Cybernetics, San Diego, pp.2413-2417.

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

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