In New Frontiers in Computational Intelligence and its Applications (Book), Eds. M. Mohammadin, IOS Press. pp. 165-174, January 2000

An Embedded-Agent Architecture for Online Learning & Control in Intelligent Machines

Hani Hagras , Victor Callaghan , Martin Colley

Department of Computer Science, University of Essex, Wivenhoe Park Colchester CO43SQ, England (U.K).

Abstract. This paper describes the design of a fuzzy controlled autonomous robot, incorporating Genetic Algorithms (GA) based rule learning, for use in an outdoor agricultural vehicle for path and edge following activities which involve spraying insecticide, distributing fertilisers, ploughing, harvesting, etc. The robot has to navigate under different ground and weather conditions offering complex problems of identification, monitoring and control. This paper addresses the development of an online self-learning system based on a modified version of the Fuzzy Classifier system (FCS) which provides rapid convergence suitable for online learning without the need for simulation. The controller was tested on both an in-door and out-door autonomous vehicle operating with different types of sensors (including a novel wand), propulsion and steering. Experiments include operating the vehicle following irregular crop edges (full of gaps) under different weather and ground conditions within a tolerance of 2 inches.

1. Introduction

Transportation has been a key tool throughout history. The impetus to develop more efficient transportation has driven developments from horse drawn carts to modern cars. Finding more efficient means of transport continues to be the focus of much of the western world's effort. So far, the human driver has had a seemingly irreplaceable role in this. However, it may not be long before it is possible to replace human drivers. Already there are commercial driver-less vehicles operating in structured environments such as rail transportation and some car manufacturers have demonstrated prototype auto- and co-pilots driving road vehicles. The less structured and consistent the environment becomes, the more difficult is the challenge to technology. For example, one of the most difficult technical challenges for vehicle guidance is presented by the agricultural industry due to the inconsistency of the terrain, the irregularity and the open nature of the working environment. This provides numerous difficult challenges to a control system such as dealing with imprecise and inaccurate sensing, and the consequences of autonomous farm vehicles being deeply embedded into a dynamic partly non-deterministic physical world. On farms some of the most important field tasks are those based on crops planted in rows or other geometric patterns frequently involving a vehicle driving in straight lines, turning at row ends and activating machinery at the start and finish runs. Examples of this are in spraying, ploughing and harvesting. In our work we have adopted this challenging environment as a test for the embedded-agent techniques we developed. Our embedded-agent control architecture utilises a much-developed form of fuzzy logic augmented by GA learning that excels in dealing with such imprecise sensors and varying conditions, which characterise these applications.

1.1. Background

AI techniques including expert systems, machine vision, artificial neural networks and fuzzy systems have been applied to intelligent automation of farm machinery. Ziteraya and Yamahaso

In New Frontiers in Computational Intelligence and its Applications (Book), Eds. M. Mohammadin, IOS Press. pp. 165-174, January 2000

[19] showed the pattern recognition of farm products by linguistic description with fuzzy theory was possible. Zhang et al [20] developed a fuzzy control system that could control corn drying. Ollis [13] used machine vision to follow and cut an edge of a hay crop but however he did not address the problem of turning around corners and detection of the end of a crop row. Cho [3] used a simulation of a fuzzy unmanned combine harvester operation but adopted only on-off touch sensors for his fuzzy systems. Cho's fuzzy system was only a simulation and did not in deal with continuous data and thus led the system not having a smooth response and encountering problems when turning around corners. Yamasita [17] tested the practical use of an unmanned vehicle for green house with fuzzy control. Mandow [11] had developed the greenhouse robot Aurora, but, of course, the application and environmental variations in a greenhouse are restricted with respect to outdoor situations.

An agricultural vehicle must react to dynamic events in unstructured agricultural environments, using multiple sensors. For mobile robots, the size of the input space requires a complicated control function. This mapping can be made more manageable by breaking down the space for analysis by multiple sub-agents or processes, each of which responds to specific types of situations, and then integrating the recommendations of these sub-agents. Brooks [2] pioneered such an approach by proposing the subsumption architecture which consisted of many simple co-operating process, which he labelled behaviours, which has proved very successful in the control of mobile robots.

There are many forms of behaviour co-ordination. Classical robot architectures such as the subsumption architecture [2] use an on-off switching schema: in each situation, one behaviour is selected and is given complete control of the effectors. This simple scheme may be inadequate in some situations [15] and later proposals included dynamic arbitration policies, where the decision of which behaviour to activate depends on both the current (sub-goal), given by the planner and the environmental conditions. Both fixed and dynamic arbitration policies can be implemented using the mechanisms of fuzzy logic. The two main advantages in doing so are the ability to express partial and concurrent activation of behaviours and the smooth transition between behaviours.

Broadly speaking, our work situates itself in the recent line of research that concentrates on the realisation of real-time embedded-agents. A fundamental aspect of such agents is that they are grounded in the physical world (situated, embodied and operating in real time). As such adaptive behaviour cannot be considered as a product of an agent in isolation from the world, but can only emerge from strong coupling of the agent with its environment. Despite the validity of computer simulations to build autonomous robots being much criticised [12] many projects still use simulations to test or train their models.

Evolutionary robotics approaches are based on genetic algorithm techniques. In this approach an initial populations of different "genotypes" each codifying the control system (and possibly the morphology) of a robot are created randomly. Each robot is evaluated in the environment and to each robot is assigned a score ("fitness") corresponding to the ability of the robot to perform some desired task. Then the robots that have obtained the highest fitness are allowed to reproduce by generating copies of their genotypes with the addition of random changes ("mutations"). The process is repeated for a certain number of generations until, hopefully, the desired performances are achieved [5]. Several researchers have used modified version of GA mechanisms [7], [9] to learn robot behaviours but based on offline simulation, the results of which they then download into real robots. One of the long term goals of our research is to develop an autonomous robot for the outdoor agricultural domain; an open and varying environment which is particularly difficult to simulate. Thus, the agricultural domain is a particularly appropriate and challenging environment for our online behaviour learning techniques.

In New Frontiers in Computational Intelligence and its Applications (Book), Eds. M. Mohammadin, IOS Press. pp. 165-174, January 2000

1.2. Fuzzy Classifier Systems

There are two main approaches in GA based fuzzy control learning. The first is the so called "Michigan" approach [5] in which the "population" consists of fuzzy rules. A "fitness" is assigned to individual rules which compete amongst each other in an evolution process. This approach is appropriate for on-line learning because the fuzzy controller is itself built from a population of such rules which can be improved constantly as part of the evolution process [10]. Methods such as the "bucket-brigade" can be used to distribute credit reinforcement among fuzzy rules activated sequentially in time. The second approach, the so called "Pitts" methodology, uses a population of fuzzy controllers. Each individual alone is a candidate solution to the optimisation problem. It is only possible to learn off-line because in each generation a population of solutions has to be tested [10]. The fitness function evaluates the performance of the entire fuzzy controller and whilst apportioning credit is easier it has the drawback that poor rules sometimes benefit from good ones.

A classifier system is an adaptive, general purpose machine learning system which is designed to operate in noisy environments with infrequent and often incomplete feedback. Classifiers simply are if-then rules. The name Learning Classifier Systems (LCS) comes from the capability of rules to classify messages into arbitrary message sets [8]. However, this is only one facet of rules. In classifier systems rules or productions have the same role as instructions in ordinary programs. Such production systems are computationally complete [8] and therefore as powerful as any other Turing-equivalent programming language. A FCS is a genetic based machine learning system whose classifier list is a fuzzy rule base. They learn by creating fuzzy rules which relate the values of the input variables to internal or output variables. Valezuela Rendon [14] gave the first description of the fuzzy classifier system. Each classifier contains the actual description of the membership functions that correspond to each input and output variable, which consists of parameters that define the associated fuzzy set. The few examples of systems which exist are generally simple applications of on-line learning in robotics. Notable amongst these is Bonarini [1] in which he suggests a hybrid method for solving the co-operation versus competition problem. He uses sub `populations of similar fuzzy rules, which are undergoing local competition. Co-operation of fuzzy rules is achieved by composing each of the best local solutions into an entire fuzzy controller. He applied this method for simple behaviours such as following another robot or moving in a corridor. However his controllers are developed via simulation before being downloaded and applied to the real robots, so it is not pure on-line learning.

2. The Target Environment

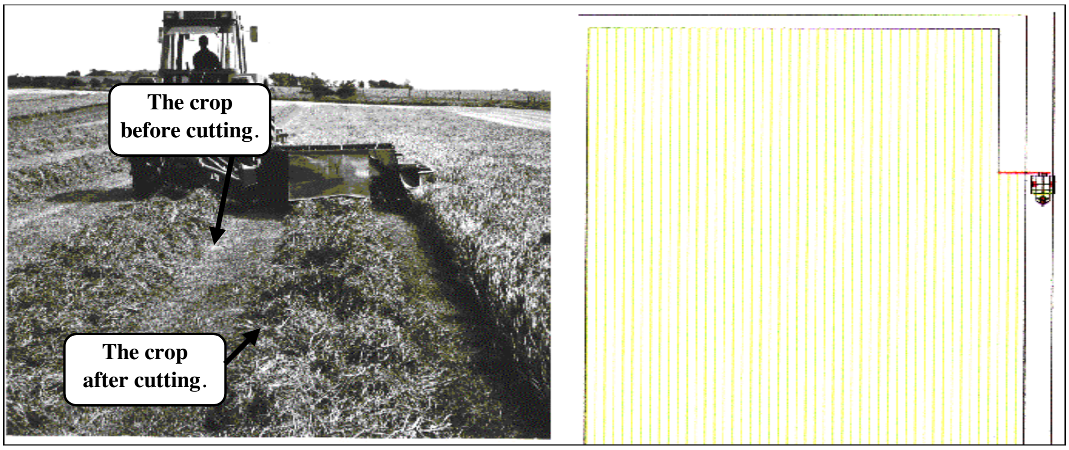
The particular target application of the work described in this paper is a vehicle traction and steering controller for crop harvesting. In earlier work [6] we developed a hierarchical fuzzy logic controller that had many advantages including greatly reducing the number of rules needed and facilitating better behaviour arbitration. In this paper we describe how we have added genetic mechanisms to provide rule learning in which reinforcement is given as actions are performed. We use a modified version of the Fuzzy Classifier system (FCS) equipped with a rule-cache making it possible for learnt expertise to be applied to future situations thus allowing learning to start the search from the best point found so far. The system uses sensory information in-order to narrow the search space for the GA. This process can be viewed as a hierarchy. The proposed techniques have resulted in rapid convergence suitable for learning individual behaviours online without need for simulation. In this paper we focus on the GA learning aspects of the controller.

The robot is designed to harvest a crop by following its edge while maintaining a safe distance, in this case 45 cm from the vehicle, while at the same time allowing the cutter (fixed to the side

In New Frontiers in Computational Intelligence and its Applications (Book), Eds. M. Mohammadin, IOS Press. pp. 165-174, January 2000

of the vehicle) to cut the crop. Figure (1a) shows a hay harvester with the associated cutting technique being depicted in Figure (1b). The robot can also follow lines (real or virtual) for other purposes such as spraying insecticide, distributing fertilisers, ploughing, harvesting, etc.

Initially developed our design using an indoor mobile robot, introducing to it all the hard conditions that it might encounter in a real field. Although there are clearly significant differences between the indoor environment and that of a farm, we did what we could to make the experiments more realistic such as using noisy and imprecise sensors, irregular geometrical shapes and fences constructed from hay (in baled form). However, it is self evident that ultimate test of a farm robot is on a real farm and thus a large part of the evaluation has been based around our outdoor electric and diesel vehicles. We feel that this hybrid-environment approach involving real robots is better than a computer simulation which suffers from well known modelling difficulties (especially when trying to model the physical environment comprising varying ground and weather conditions and objects such as trees and hay).



(a) (b)

Figure 1: a) A manned harvester cutting hay, b) The harvesting technique.

3. Overview of the proposed on-line algorithm

In this work, we target learning individual behaviours where reinforcement can be given as an action is undertaken (immediate reinforcement) such as edge following and goal seeking. These behaviours can be co-ordinated to achieve more complex capabilities such as getting out of a tight corridor and navigating towards a certain goal. In the following design each behaviour is treated as an independent fuzzy controller and then using fuzzy behaviour combination we obtain a collective fuzzy output which is then deffuzzified to obtain a final crisp output as shown in Figure (2). Each behaviour uses a singleton fuzzifier, triangular membership function, product inference, max-product composition and height defuzzification. The selected techniques were chosen due to their computational simplicity. The equation that maps the system input to output is given by:

$$\frac{\sum_{p=1}^{M} y_p \prod_{i=1}^{G} \alpha_{Aip}}{\sum_{p=1}^{M} \prod_{i=1}^{G} \alpha_{Aip}} \qquad (1)$$

Where M is the total number of rules, y is the crisp output for each rule, αAi is the product of the membership functions of each rule inputs, G is the number of inputs. In this hierarchical architecture we use a fuzzy operator to combine the preferences of different behaviour into a collective preference. According to this view, command fusion is decomposed into two steps: preference combination and decision. In the case of using fuzzy numbers for preferences, product-sum combination and height defuzzifcation the final output equation is [15]:

$$\mathbf{C=}\quad \frac{\sum_i (BW_i * C_i)}{\sum_i BW_i} \qquad (2)$$

Where i = right behaviour, left behaviour, goal seeking, C_i is the behaviour command output (left and right velocity in our case). BW_i is the behaviour weight. The behaviour weights are calculated dynamically taking into account the environmental context of the mobile robot. By doing this there is no need to pre-plan, as the system effectively plans for itself depending on context.

In a real-time GA, it is desirable to achieve a high level of online performance while, at the same time being capable of reacting rapidly to changes requiring new actions. Hence it is not necessary to achieve a total convergence of the population to a single string, but rather to maintain a limited amount of exploration and diversity in the population. Incidentally, near-convergence can be achieved in terms of fitness, with diverse structures. These requirements mean that the population size should be kept sufficiently small, so that progression towards near-convergence can be achieved within a relatively short time. Similarly the genetic operators should be used in a way that achieve high-fitness individuals rapidly [10]. Figure (3) introduces a block diagram of the operation of the proposed on-line algorithm. The rule base for the behaviour being learnt is initialised randomly. In the following sections we will introduce the various algorithmic steps involved.

3.1. Identifying Poor Rules

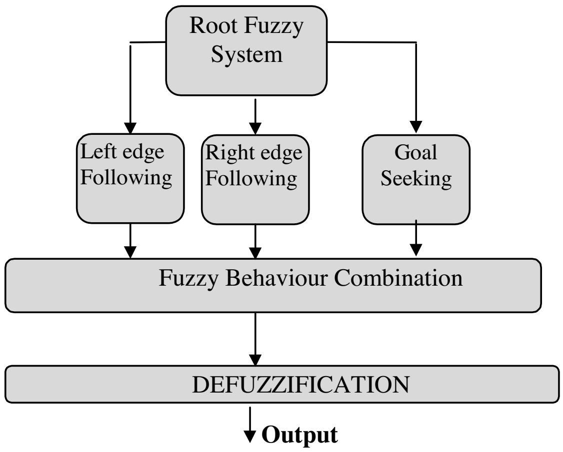
After the rule base has been initialised, the robot starts moving. If the rules are poor then it will begin deviating from its objective (e.g. not maintaining a constant distance from an edge, or drifting from the direct direction of a goal etc). In this case the on-line algorithm is fired to generate new set of rules to correct this deviation. At this stage the GA population will consist of all the rules contributing in the action (which is usually a small number; for example our behaviours contain as little as 25 rules). As in other classifier systems, in order to preserve the system performance the GA is allowed to replace a subset of classifiers (the rules in our case). The GA creates replaces the worst m classifiers by new ones [4]. New rules are tested by the combined action of the performance and credit assignment algorithms. In our case, only two rules actions will be replaced (those already identified with being predominantly responsible for the error).

3.2. Fitness Determination and Credit Assignment

The system fitness is evaluated by how much it reduces the absolute deviation (d) from the nominal value, which is given by:

$$\mathrm{d} = \frac{|\,nomial.value - deviated.value|}{\max.deviation} \qquad (3)$$

Where the nominal value will correspond to the value that gives maximum normal membership function (40 c.m in case of wall following and zero degrees in case of goal seeking). The deviated value is any value deviating from the nominal value. The maximum deviation corresponds to the maximum deviation that can occur (which is equal to 80-40= 40 c.m in case of wall following and 180-0=180 degrees in case of goal seeking). So the fitness of the solution is given by d1-d2 where d2 is the absolute deviation before introducing a new solution and d1 is the absolute deviation following the new solution. The deviation is measured using the robot's physical sensors (the sonar in case of the wall following and the infra-red scanner in case of goal seeking), which gives the robot the ability to adapt to the imprecision and the noise found in the real sensors rather than relying on estimates from simulations.



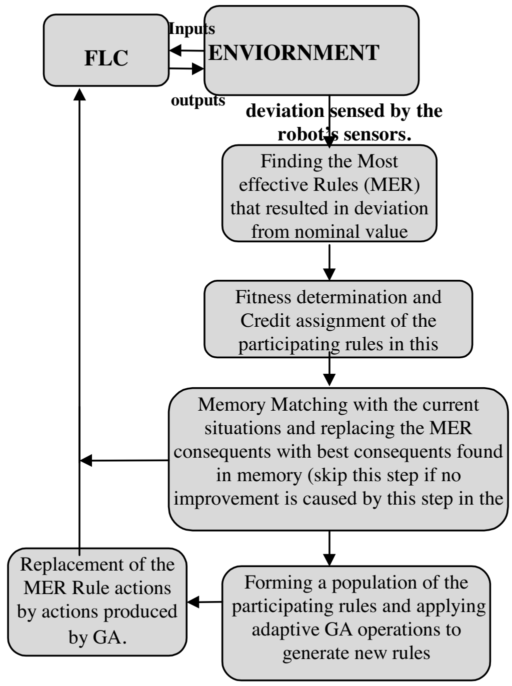


Figure 2: The behaviour co-ordinated system.

Figure 3: Block diagram of the Proposed algorithm.

The fitness of each rule at a given situation is calculated as follows. We can write the crisp output Y_t as in (1). As, in our applications, we have two output variables (left and the right wheel speeds), then we have Y_{t1} and Y_{t2}. The contribution of each rule p output (Y_{p1}, Y_{p2}) to the total output Y_{t1} and Y_{t2} is denoted by S_{r1}, S_{r2} where S_{r1} and S_{r2} is given by:

$$\mathbf{S_{r1}} = \frac{Y_{t1} - \dfrac{Y_{p1}\prod_{i=1}^{G}\alpha_{Aip}}{\prod_{i=1}^{G}\alpha_{Aip}}}{Y_{t1}}, \quad \mathbf{S_{r2}} = \frac{Y_{t2} - \dfrac{Y_{p2}\prod_{i=1}^{G}\alpha_{Aip}}{\prod_{i=1}^{G}\alpha_{Aip}}}{Y_{t2}} \qquad (4)$$

We then calculate each rule's contribution to the final action S_c by $S_c = \frac{S_{r1} + S_{r2}}{2}$. The two most effective rules are those which have the two greatest value of S_c. We use mutation only to generate new solutions because of the small population formed by the fired rules.

3.3. Memory Application

After determining the rules actions to be replaced, the robot then matches the current rules to sets of rules stored in a memory (containing each rule and its best fitness value to date). The fitness of the rule in a given solution is given by **:**

$$\mathbf{S_{rt}} = \text{Constant} + (d_1 - d_2)\, S_c \qquad (5)$$

d_1-d_2 is the absolute deviation improvement or degradation caused by the adjusted rule base produced by the algorithm. If there is an improvement in the deviation, then the rules that have contributed most will be given more fitness to boost their actions. If there is degradation then the rules that contributed most must be punished by reducing their fitness, repeating the process for the next most responsible rule etc. For every rule action to be replaced the best fitness rule will replace the current action in the behaviour rule base. If the deviation decreases, then the robots will keep the best rules in the behaviour rule base. If the deviation remains the same, or increases, the robot fires the GA to produce new solutions by mutating these best rules until the deviation begins decreasing or the rule is proved ineffective when the robot is moving, thus indicating another rule might be more effective. This mechanism is used to speed up the GA search as it starts the GA from the best known point in the solution space instead of starting from a random point. This then becomes a solution for the current situation with the rule fitness being calculated and compared to the maximum fitness rule. If its fitness is greater than the best previous one then replaces it, otherwise the existing best rule still is retained in memory. Clearly, when the algorithm is first fired, there is no memory and the parents of the new solutions are by necessity randomly generated rules.

3.4. Using GA to Produce New Solutions

On being fired the GA begins its search for a new rule actions to replace those identified with poor performance. New solutions are generated by mutating the two most effective rules. Figure (4) records the results of experiments measuring the effect of differing mutation rates on learning performance (in our case, time the robot needs to achieve its purpose such as reaching its goal or following a wall). As can be observed, for mutation values less than 0.3 there is nearly no convergence as the population size and the chromosome size is small, and the low mutation rates does not introduce a lot of new genetic materials to introduce new solutions. The same occurs for high mutation rates (higher than 0.7) as the mutation rate reaches 1.0 the genetic materials available are the primary chromosomes (e.g. 0101) and its inversion (1010) which is not enough for introducing new solutions. Clearly 0.5 is the optimum value, in our case producing solutions after 96 seconds (on average). The robot also uses its sensory information to narrow up the search space of the GA and thus reduces the learning time. For example, if the robot is implementing left wall following and it is moving towards the wall, then any action that suggests going to the left will be a bad action. Hence, if we use the front left side sensor and it senses that we are going towards a wall, then the GA solutions will have a constraint not to go left, the same applies for any behaviour.

3.5. The Learning Length Criteria

The robot assumes it had learnt the required behaviour if it succeeds in maintaining the nominal value for the behaviour for a distance sufficient to prove that the learnt rule base is adequate. The optimal learning distance has been related to units of length of the robot, so that the algorithm can be applied in an invariant manner to different size robots. In order to determine the optimal learning distance we have conducted numerous experiments evaluating performance relative to the robot's length (e.g. 1x robot' s length, 2x robot's length, etc.). We then followed the same track that was used during learning to determine the absolute deviation at each control cycle from the optimum value (ie maintaining a constant distance from a wall in case of edge following and going toward the goal in case of goal seeking). Then we calculated the average and standard deviation of this error and compared learning distances to settle on a suitable criteria (i.e. as short as possible whilst producing a stable rule base). From Figure (5) it is obvious that the average and standard error for the wall following stabilises at three times the robot's length at average value of 2cm and standard deviation of 1cm. Thus we use three times the robot the length as our learning length criteria.

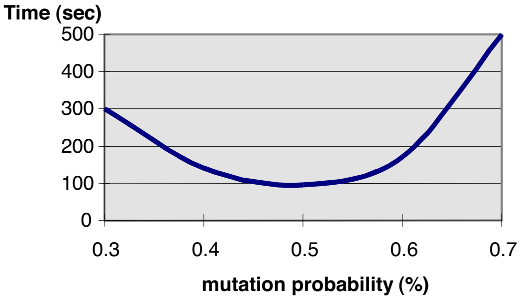


Figure 4: Mutation rate versus time

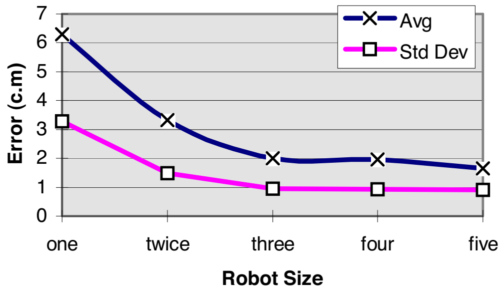
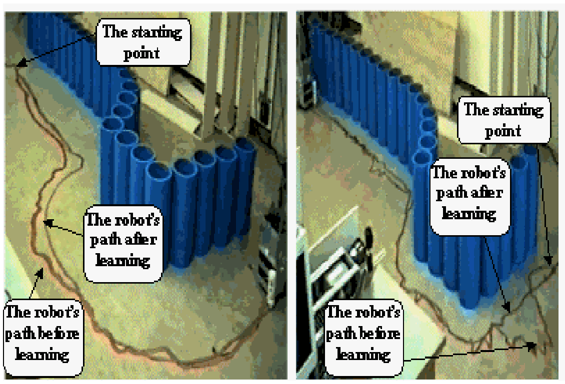


Figure 5: The robot size plotted against the error

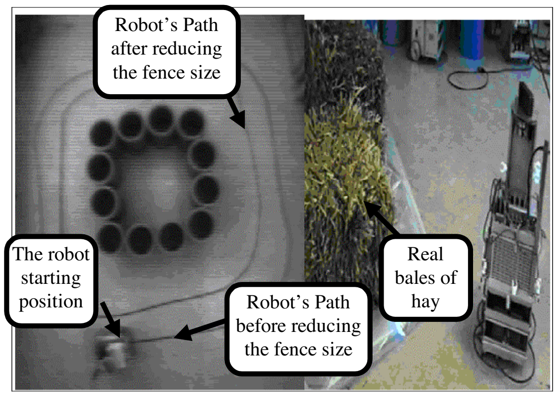
4. Experimental Results

The performance of the architecture has been assessed in two main ways. First, we physically emulated the crop-following process by using real physical robots in a fabricated environment whilst, not farm based, still contained more non-deterministic elements of the problem than would be provided in a simulation. For example, the robots were situated, embodied and interacting with real irregular objects, such as hay bales, using imperfect sonar sensing. In the next phase we tested the same control architecture but outdoors in a real farm environment, tracking fences and crop edges. All experiments were repeated 5 times to test the system repeatability and stability for different weather and ground conditions (e.g. rain, wind, holes in the ground, going up and down hill etc).



(a) (b)

Figure 6: a) Learning left wall following.
b) Learning right wall following.



(a) (b)

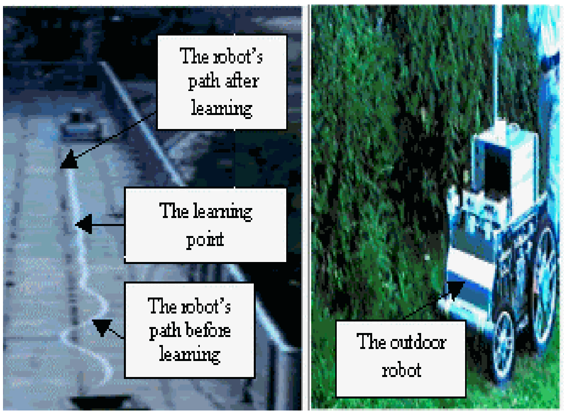
Figure 7: a) The robot emulating the harvesting operation.
b) The robot following fences of bales of hay.

Figure (6-a) & (6-b) shows the robot learning left-wall following whilst dealing with an irregular edge and imprecise ultra sound sensing in an average of 96 seconds. Note the straightness of the path. All of the learnt behaviours were tested in different (and difficult) terrain from those in which they were originally trained. In these tests the robots experienced an average deviation of 2cm and standard deviation of 1cm which, given the irregularity of the terrain, and the imprecision of the sensors, was a particularly good result.

Figure (7-a) shows the robot emulating the crop cutting operation. Here it continues going inwards to complete the harvesting operations (the cutting action was simulated by reducing the size of the fence). The same experiment was repeated with real bales of hay, Figure (7-b). Note that the response is smooth especially when the robot turns which is due to the smooth transition between rules and the smooth interpolation between different actions, which characterises fuzzy logic.

To show that the proposed system can deal with an open outdoor environment we used a different outdoor robot (with 68040 microprocessor and 0.5 m/s maximum speed) to learn right edge-following behaviour on an irregular metallic fence. Figure (8-a) shows the robot reducing its deviation rapidly to almost zero deviation after 79 secs and a distance equal to three times the robot length. Note that the robot's path shows only a small deviation despite the irregularity of the fence and the high imprecision of the sensors. Figure (8-b) shows the electric robot on a real farm following a plant edge (characterised by highly irregular gaps in plants) and over varying ground conditions (e.g. slopes and holes). It used two ultra sound sensors to sense the crop edge. Again the robot performed well following the crop at a safe distance from the edge. Although we currently have no quantitative means for evaluating the precision of the crop following, we estimate that the crop edge was tracked successfully within a tolerance of 2 inches.

In Figure (9-a) we tested a diesel-powered robot in a hay field using a mechanical wand sensor system following an irregular crop edge. The robot gave stable, repeatable and robust response as shown in Figure (9-b), tracking the edge of the crop successfully within a tolerance of 2 inches and turning smoothly around poorly defined hay crop corners.



(a) (b)

Figure 8: a) The outdoor robot learning to follow an irregular fence in outdoor environment.
b) The electrical robot following out door irregular tree hedges.

(a) (b)

Figure 9: a) The diesel robot following irregular hay crop edge using mechanical wands.
b) The robot path.

5. Conclusions

In this paper we have developed an embedded-agent capable of teaching itself how to undertake complex vehicle steering and traction control task. We have demonstrated this agent architecture successfully learning online how to follow crop-edges. We have shown how it is possible for such learning to be achieved on-line in less than 2 minutes. We believe this is a particularly notable achievement as to our knowledge, all previous reported work has required many hours of pre-implementation simulation and computation to learn these control behaviours. As part of this work we have developed a novel sensor (mechanical wands) which has advantages in detecting edges of certain kinds of crops. To the authors' knowledge, the work described in this paper is the only system which has successfully guided a diesel tractor in outdoor environments following real crop edges (including irregular edges which include gaps) and turning around corners with a high degree of repeatability to a tolerance of two inches. The control and learning techniques described in this paper are applicable to a wider set of problems than the agricultural and vehicle guidance applications addressed in this paper. Many of the techniques disclosed in this paper are protected by patents. Whilst we have chosen what is generally agreed to be a most demanding test of this control architecture, autonomous farm vehicles, the invention is equally capable of effecting control in other areas such as factory machinery, telecommunications, medical instrumentation, engines, weapons and emerging areas such as intelligent-buildings, underwater vehicles and embedded-agents in general. Current work is focusing on both refining the techniques and expanding the range of control applications.

References

[1] A. Bonarini ,“ Delayed Reinforcement, Fuzzy Q-Learning and Fuzzy Logic Controllers”, In Herrera, F., Verdegay, J. L. (Eds.) Genetic

[2] R. Brooks, A robust layered control system for a mobile robot , IEEE Journal of Robotics and Automation, pp. 14-23,1986.

[3] S. Cho, H. Ki , “Unmanned Combine Operation Using Fuzzy Control – Graphic Simulation”,Applied Engineering in Agricultural , pp. 247-251, 1996.

[4] M.Dorigo, “ Genetics-based machine learning and behaviour based robotics: A new synthesis”, IEEE transactions on system, man and cybernetics, pp 141-154, 1993.

[5] D. Goldberg , “ Genetic Algorithms in search, optimization and machine learning”, Addison-Wesley, Reading , MA , 1989.

[6] H.Hagras, V. Callaghan, M.Colley and M. Carr-West, “ A behaviour based hierarchical fuzzy control architecture for agricultural autonomous mobile robots”, The International Conference on Computational Intelligence for Modelling, Control and Automation, pp 172-177, Vienna-Austria 1999.

[7] F. Hoffmann , “Incremental tuning of fuzzy controllers by means of evolution strategy”, GP-98 Conference, Madison, Wisconsin, 1998.

[8] J. Holland , “Genetic Algorithms and classifier systems : foundations and future directions”, The MIT Press, Cambridge , Massachusetts, The MIT Press edition ,1992.

[9] D. Leitch, “A new genetic algorithm for the evolution of fuzzy systems, PhD thesis, University of Oxford, 1995.

[10] G. Linkens, O. Nyongeso, “GA for fuzzy control, Online system development and application “, IEE proc control theory appl, pp. 177-185, 1995.

[11] M. Mandow et-al, “The Autonomous Mobile Robot Aurora for Greenhouse Operation”, IEEE Robotics and Automation Magazine, Vol. 3, pp. 18-28, 1996.

[12] O. Miglino , H. Lund, S. Nolfi., “ Evolving Mobile Robots in Simulated and Real Enviroments. Artificial Life” Technical Report NSAL-95007, Reparto di Sistemi Neurali e Vita Artificiale, Istituto di Psicologia, Consiglio Nazionale delle Ricerche.

[13] M. Ollis , A. Stentz , “ First results in vision-based crop line tracking”, Proceedings of the 1996 IEEE International Conference on Robotics and Automation, pp. 951-956, 1996.

[14] V. Rendon ,. “ A classifier system for continously varying variables”, Proc of the Fourth Int. Conf. On GA ,pp. 346-353, 1991.

[15] A. Saffiotti, Fuzzy logic in autonomous robotics: behaviour co-ordination, Fuzz-IEEE, pp. 573-578, Spain 1997.

[16] S. Schallter, “Object Recognition In Agricultural Environment”, MSc , Essex University, 1996.

[17] S. Yamasita, “Examination Of Induction Method Of Unmanned Vehicle For Greenhouse”, in Proc.49th Congress in Soc. Of Farm Machinery pp. 273-274, Japan, 1990.

[18] Zadeh L A, “Fuzzy Sets”, Informat. Conf, Vol. 8, pp. 338-353, 1965.

[19] Z. Ziteraya, K. yamahoso, “Pattern Recognition Of Farm Products By Linguistic Description With Fuzzy Theory”, in proc .3rd fuzzy Fuzzy system symp of IFSA Brach, pp. 127-132, Japan, 1987.

[20] Q. Zhang, J. Letchfield , J. Bentsman, “Fuzzy Predictive Control Systems For Corn Quality Control Drying Food Processing Automation”, in proceedings of the 1990 conference, pp. 313-320 , ASAE Pub.2, St Joseph, Mich.: ASAE. 1990.