An Online Hierarchical Fuzzy Rule Based System for Mobile Robot Controllers

Antony Waldock[1], Brian Carse[2], Chris Melhuish[2]

[1]Advanced Technology Centre
BAE SYSTEMS
Bristol, BS34 7QW, UK
antony.waldock@baesystems.com

[2]Intelligent Autonomous Systems Laboratory,
University of the West of England
Bristol, BS 16 1QY
{brian.carse, chris.Melhuish}@uwe.ac.uk

Abstract

The introduction of automated robots has revolutionised the manufacturing industry. The further development of autonomous mobile robots capable of functioning in unstructured and dynamic environments is highly desirable. This paper outlines a novel method for the online development of an interpretable mobile robot controller using supervised learning. An information theoretic approach is used to control the rate of expansion in a Hierarchical Fuzzy Rule Based System (FRBS). Experimental results, on a simulated mobile robot, are provided to demonstrate how the uncertainty tolerated can be used to control the trade-off between accuracy and interpretability.

1 Introduction and Related Work

Over the last 30 years, autonomous robots have revolutionised the manufacturing industry by allowing mass production of a wide range of products from cars to highly complex printed circuit boards. These robots remain relatively unintelligent and operate in sanitised and caged environments. The development of industrial robots capable of coping with unstructured dynamic environments, where humans may be present, could revolutionise the industry again. This paper outlines a novel method for the construction of an interpretable controller for mobile service robots through interaction with a human supervisor.

The construction of a mobile robot controller for a real world application is a complex and time-consuming process. Classical Control Theory (Ogata, 1997) allows the generation of control rules from accurate analytical models of the environment, platform and control laws. The construction of models that accurately represent the dynamics of an uncertain and complex environment can be a long and difficult, if not impossible, process. Soft Computing (Zadeh, 1994) provides a collection of methods that aim to exploit a tolerance for imprecision and approximation to achieve robustness and low cost solutions e.g. Neural Networks and Fuzzy Logic. Fuzzy Logic has been demonstrated to cope well with uncertain and imprecise data when used in a mobile robot controller (Hagras et al., 2001). Using Fuzzy Rule Based Systems (FRBS) to build mobile robot controllers (Tunstel et al., 1996, Hoffman and Pfister, 1997) has become increasing popular due to their ability to automatically generate rules from large data sets while maintaining a degree of human interpretability (Holve and Protzel, 1996).

A FRBS divides the input space into a number of linguistic symbols, each associating a fuzzy set with a natural language meaning (Zadeh, 1998); e.g. very small, small, large etc. A fuzzy set links an input variable with a membership function, to represent its applicability with regard to the current environmental state, and hence, determine its influence in the decision process. The decision space is partitioned using a series of IF...THEN... rules. The combination of meaningful linguistic symbols and well-structured rules provides a high degree of human interpretability. For example, a service robot that is required to avoid collisions might use a fuzzy control rule such as:

IF (wall is close) AND (wall_direction is infront-left) THEN (turn right)

The way in which linguistic symbols are represented can directly affect the accuracy of the approximated function (Bastian, 1994). If a greater degree of accuracy is demanded then the structure (i.e. size, shape and position) of the membership functions, which represent the linguistic symbols, must be altered. Increasing the granularity of the linguistic symbols can facilitate improvements in accuracy, but as the complexity of a system increases, our ability to make precise and yet significant statements about its behaviour diminishes. As the number of linguistic symbols increases human interpretability declines. Hence within this contribution, the number of rules is used as a measure of interpretability. This relationship is referred to as the trade off between accuracy and interpretability (Alcala et al., 2001).

A new variant of a FRBS has been developed in an attempt to improve the accuracy of the approximation and is referred to as an Approximate FRBS (also known as “non-grid-oriented” or “free semantics” FRBS). An approximate FRBS increases the accuracy of the model by manipulating fuzzy sets directly instead of the linguistic symbols. By allowing the fuzzy sets to vary in number, size and position, they can be mapped directly to the data giving significant improvements (Carse et al., 1996). At the end of rule generation, the fuzzy variables are semantically free and can tend to be excessively specialised. This limits the human interpretability of an approximate FRBS. However, it must be noted they often remain more interpretable than other models such as neural networks.

The Hierarchical Fuzzy Rule Based System (HFRBS) has been developed, in recent years, as an attempt to improve accuracy while maintaining interpretability (Cordón et al., 2001b). A HFRBS divides the input space into a fixed number of linguistic symbols each corresponding to a natural language meaning e.g. very small, small etc. Training data is then used to automatically generate rules as in a standard FRBS. The key extension of a HFRBS is the use of an *expansion policy* to determine inaccurate areas of the decision space and the corresponding rules. When an inaccuracy is identified, the rule representing that area is specialised into a set of more specific rules. This involves partitioning the rule’s decision space into smaller areas each represented by a separate rule. This process of specialisation continues until a desired level of accuracy is satisfied. Figure 1 shows an example of a partitioned decision space (a) and its corresponding hierarchical representation (b). This concept of increasing the granularity to fit the underlying decision space has been applied to classifier systems for a number of years (Melhuish and Fogarty, 1994).

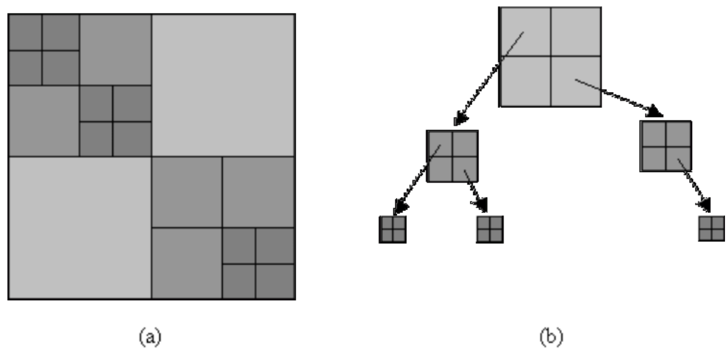

Figure 1: Hierarchical FRBS

This type of linguistic symbol expansion increases the accuracy depending on the complexity of the data modelled and is commonly referred to as "ad-hoc data driven learning" (Alcala et al., 2001). The expansion can be controlled depending on the accuracy desired, human interpretability required and the complexity of the function to be approximated. Some of these properties are mutually exclusive, for example, high human interpretability on a complex, accurate model may be impossible.

Holve outlines a method for specialisation by carefully pre-processing training data such that, when a conflict is encountered, the linguistic symbol is expanded (Holve, 1998). Although demonstrated to approximate complex functions, pre-processing limits its use to applications where all the training data is available and certain. Cordón, Herrera and Zwir (Cordón et al., 2001b) outline a hierarchical FRBS that uses expansion techniques to specialise linguistic symbols with a large degree of error. The error of each rule is calculated by the percentage of the Mean Square Error (MSE) associated with the rule over the MSE for the entire training set. A rule with bad performance is determined by comparing this error with a tuneable parameter α, which dictates the rate of expansion and hence the accuracy of the function approximated. This method relies on the complete training set and test set being available during the expansion phase. For an online application, such as learning behaviour in a mobile robot, access to the entire training set is not possible requiring an Online or Anytime Algorithm (Grefenstette and Ramsey, 1982).

The next section outlines the proposed algorithm with a new method of expansion based on Information Theory. The performance of the proposed algorithm is compared to existing function approximators in section three followed by a simulated mobile robot application in section four. Section five and six discuss the mobile robot results and possible areas of future work.

2 The Information Theoretic Hierarchical Fuzzy Associative Memory (IT-HFAM)

The rule generation algorithm proposed in the current contribution uses a similar method of hierarchical specialisation, but the expansion policy is determined by the amount of uncertainty within a rules decision space. Information Theory, developed by Shannon (Shannon, 1948), was initially concerned with modelling the efficiency of communication systems but has been applied to a multitude of other research areas including decision making (ID3 (Quinlan, 1990)) and fuzzy logic systems (Al-sharhan et al., 2001). Suppose we have a set of *n* possible events whose probabilities of occurrence are $p_1, p_2 ... p_n$. These probabilities are known but that is all we know concerning the event. Shannon defines the amount of 'choice' involved or the uncertainty of the outcome as Entropy. The entropy, $H(p_1, p_2 ... p_n)$ of an event with *n* possible outcomes is defined by Shannon as

$$H(p_1, p_2 .. p_n) = \sum_{i=1}^{n} (p_i \times \log(p_i))$$

This measure of information can be used to calculate the uncertainty within each rule (Zadeh, 1965) and to determine the expansion of a HFRBS. A FRBS can be viewed as a Fuzzy Associative Memory (FAM) where the linguistic symbols M_i, for each input *n*, produce an *n*-dimensional decision space partitioned by an *i*-dimensional grid. Each cell represents an IF-THEN rule with *n* linguistic inputs corresponding to a single output linguistic symbol. The proposed Information Theoretic Hierarchical Fuzzy Associative Memory (IT-HFAM) extends the traditional FAM approach by including an applicability distribution over all the output linguistic symbols. Figure 2 displays a FAM partitioned into four cells (rules) each with an applicability distribution over all possible output symbols.

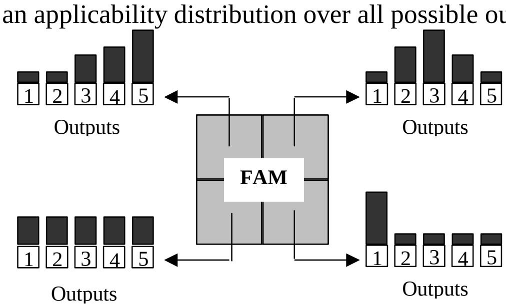

Figure 2: FAM with Output Applicability Distributions

Using Shannon's measure of uncertainty on the output applicability distribution gives the amount of choice experienced by this cell. If the applicability distribution is flat i.e. A_i are equal, (Figure 2: Bottom Left Rule) then the uncertainty of the cell is at a maximum. This indicates that the cell at this granularity cannot effectively model the decision space it represents. In order to model this area of the decision space more effectively it is necessary to divide the cell into a number of smaller cells. The algorithm can be initialised with a single linguistic symbol covering each input or with a set of linguistic symbols if input space partitions are known.

Training of an IT-HFAM consists of updating the applicability distribution when exposed to a new training pattern. T_p. Each pattern is a *(n+1)* dimensional vector, $T_p = <x_1, x_2, x_3 ... x_n, y>$ where n is the number of inputs; x is the current input values and y the target output value. The algorithm is trained on all patterns until the training set has been exhausted. The algorithm is trained by first identifying all the cells with an activation greater than zero when presented with the inputs ($x_1..x_n$) from the training pattern. The applicability distribution for each active cell is updated depending on the target value y. For each output membership function, M_j^y, the applicability of the output symbol $a(M_j^y)$ is increased by the level of activation for the target value y (see Figure 3).

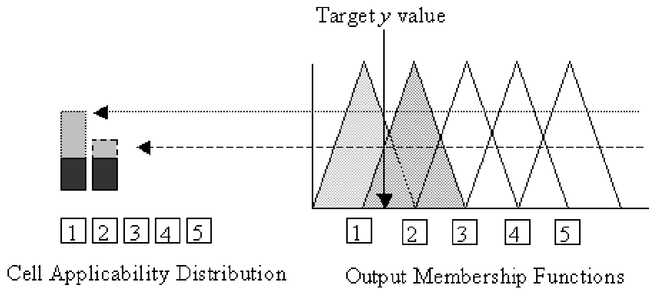

Figure 3: Cell applicability update procedure

Centre of Sums is used on the applicability distribution to calculate the current output symbol for this cell in the FAM. After each training pattern, the entropy (uncertainty) of each cell is compared with a tuneable parameter $E_{\max}$, which represents the maximum amount of uncertainty tolerated within each cell. If the uncertainty of the cell exceeds $E_{\max}$ and the applicability distribution covers more than two consecutive output symbols then the cell is divided into four smaller specialised cells (assuming the FAM consists of only two dimensions – see Figure 4).

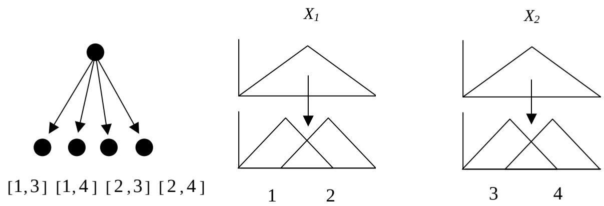

Figure 4: Membership Function Specialisation

The cell expansion is halted when only two consecutive output symbols are active in the applicability distribution. A final stage of rule reduction is applied which merges neighbouring cells with the same output fuzzy set. Rule reduction can be performed after a number of training phases or after training has been completed.

3 Comparison with Existing Function Approximations

This section describes the performance of the IT-HFAM algorithm, on an intermediate complexity function, compared with existing function approximators. Within the experiment below, the accuracy is determined using the Root Mean Squared Error (RMSE) equation using a test set of 100 evenly distributed points over the input space.

Cordón, Herrera and Zwir (Cordón et al., 2001a) demonstrate the ability to train a hierarchical FRBS using a Mean Squared Error (MSE) expansion policy on an intermediate complexity function defined as:

$$f(x_1, x_2) = e^{x_1} \times \sin^2(x_2) + e^{x_2} \times \sin^2(x_1)$$
$$x_1, x_2 \in [-8,8],\ f(x_1, x_2) \in [0,5836]$$

The system was trained using a training set of 1156 training patterns, which were evenly distributed over the function, compared to Cordón's 1089 training patterns. The RMSE was calculated after exposure to the complete training set and then after a second iteration. Table 1 gives a comparison between the IT-FAM algorithm and existing function approximators.

Table 1: Comparison with existing function approximators

Function Approximator	No Rules	RMSE
Static Weight Counting Algorithm (S-WCA) (Bardossy and Duckstein, 1995)	9	0.0868
Hierarchical S-WCA (Cordón et al., 2001b)	316	0.0134
Fuzzy C-Means (FCM) (Bezdek, 1992)	6	0.1124
Hierarchical FCM (Cordón et al., 2001b)	9	0.0403
IT-HFAM (E_{max} 100%)	7	0.0880
IT-HFAM (E_{max} 45%)	336	0.0291

The results shown in Table 1 demonstrate that the on-line IT-HFAM performance is comparable to the other off-line methods.

4 Learning Behaviour on a Mobile Robot

In this section the IT-HFAM algorithm will be used to construct a set of control rules for a mobile robot though interaction with a supervisor. The supervisor presents examples of correct operation (training patterns) to the algorithm when the robot's behaviour is incorrect. The task taught is inspired by a collision avoidance problem demonstrated by Furuhashi, Nakaoka and Uchikawa (Furuhashi et al., 1996). The implementation of the task differs from that addressed by Furuhashi et al. by using simulated robot sensors to detect obstacles instead of the exact range and bearing information. The performance and rules generated by the IT-HFAM algorithm over a number of E_{max} values are compared.

The task involves navigating a mobile robot to reach a goal while avoiding two other robots (Figure 5). The controlled robot moves with a constant speed with the two crossing robots moving at half the speed of the controlled robot. Varying the angular velocity, ω, alters the direction of the controlled robot. The controlled robot starts from the bottom line and attempts to navigate towards the goal at the top. The two other robots appear from the left and right hand side and cross between the robot and the goal. The aim of the task is to reach the goal without hitting either of the two crossing robots. The task is performed using the Pioneer simulation environment, Stage[1]. The controlled, and crossing robots, are simulated Pioneer robots, while a red box is used to represent the goal. The controlled robot can detect an obstacle using eight sonar sensors evenly distributed over the front of the robot. Each sonar sensor returns a value indicating the distance to the nearest obstacle. The sensors are divided into two sets, left and right, and the minimum distance for each set is normalised ([0,1]) before being used as an input to the controller (Figure 6).

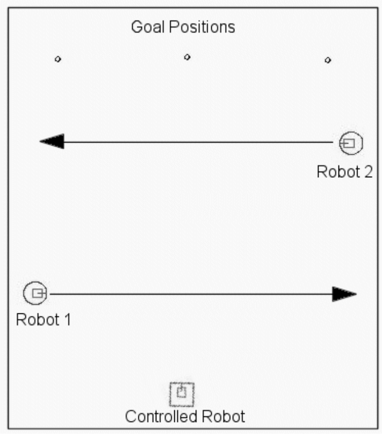

Figure 5: Robot Task

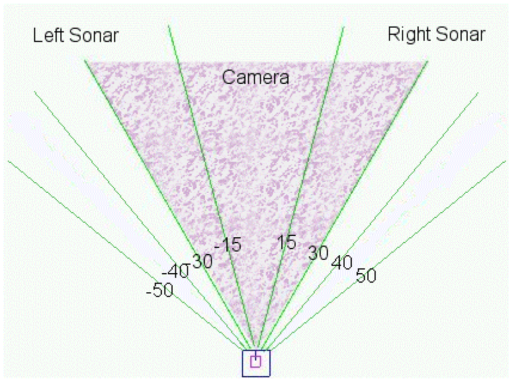

Figure 6: Robot Platform

A bearing to the goal is acquired by simulating a front mounted camera to identify the position of the red box representing the goal. The goal is detected in the camera's field of the view by extracting the red areas of the image (red blob detection). The pixel column with the highest concentration of red within the image is marked as the centre of the goal. A normalised bearing to the goal is found by dividing the pixel column with the highest concentration by the total pixel width of the image. The input to the controller is a continuous value in the range [0,1] indicating the bearing to the goal. In summary, the inputs to the controller are a bearing to the centre of the goal and the distance to the nearest object on the front left and right of the robot.

The controlled robot is trained using a human supervisor to present examples of correct operation during three scenarios. The starting position of the controlled robot is varied in each scenario giving the controlled robot exposure to a

[1] http://playerstage.sourceforge.net

large proportion of the decision space. The supervisor uses a joystick to indicate the correct angular velocity (continuous value over the range [0,1]) by monitoring the inputs to the algorithm (left/right sonar and the position of the goal). The training examples are only provided when the supervisor deems the robot's behaviour is incorrect.

The IT-HFAM algorithm is initialised with three membership functions evenly distributed over each of the input variables and five membership functions over the output variable. The task performance and characteristics of the IT-HFAM algorithm are compared over a number of E_{max} values: 100, 90, 75, 50. Three test scenarios, where the position of the goal and crossing robots are varied, are used to determine the performance of the trained controller. Task performance is obtained by studying the amount of deviation from the shortest path to the goal. The three test scenarios are displayed in Figure 7. The shortest path for each scenario is a straight line between the start position and the goal. The amount of deviation is calculated by comparing the distance travelled by the controlled robot during the test scenario and the shortest path.

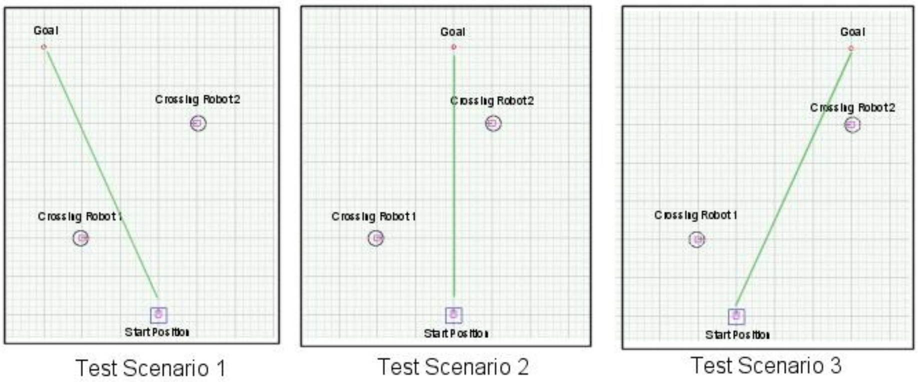

Figure 7: Test Scenarios

4 Robot Results and Discussion

Table 2 outlines the average deviation for three test runs on each scenario (D1, D2, D3 in millimetres) and the number of rules (C_r) created for values of E_{max}.

Table 2: IT-HFAM Simulated Robot Task Results

E_{max}	C_r	D_1	D_2	D_3	D_{total}
100	15	171	249	190.5	610.5
90	33	156	223.5	157	536.5
75	183	165.5	193.5	143	502
50	609	141.5	170	62.5	374

Table 2 demonstrates that E_{max} can be used to control the trade off between the number of rules generated (C_r) and the accuracy of the solution (D_{total}). With no specialisation ($E_{max} = 100$), the controlled robot was capable of navigating to the target in all test scenarios but deviated from the optimal path by 203mm on average. The algorithm produced only 15 rules that remain highly interpretable to the supervisor. Some of the rules are listed in Table 3. The selection of rules presented resembles control rules a human may have used to solve the problem and hence demonstrates that the algorithm can produce interpretable solutions. Only 6 out of the 15 rules are displayed helping to demonstrate that a human designer may not have designed a rule base that captures all situations and the arbitration between the two tasks.

Table 3: Rules Generated for $E_{max} = 100$

Task	*Sonar Left*	*Sonar Right*	*Vision*	*Action*
Track	*Far*	*Far*	*Left*	*Left*
Track	*Far*	*Far*	*Middle*	*Forward*
Track	*Far*	*Far*	*Right*	*Right*
Avoid	*Close*	*Far*	*Left OR Middle OR Right*	*Left*
Avoid	*Inrange*	*Far*	*Left OR Middle OR Right*	*Forward*
Avoid	*Close OR Inrange OR Far*	*Close*	*Left OR Middle OR Right*	*Right*

As the amount of uncertainty tolerated (E_{max}) is decreased, the deviation from the optimal path is reduced with the side affect that the number of rules generated increases significantly. For example when E_{max} is 50, the deviation from the optimal path is 38.7% better than without specialisation while the number of rules generated has increased to 609. The hierarchical structure used to perform specialisation can now be used to increase the interpretability of a complex rule base containing 609 rules. As the hierarchical structure is maintained, a representation containing the initial 15 rules can be extracted. For example when $E_{max} = 50$, a representation containing the initial rules (as in Table 3 when $E_{max} = 100$) can be extracted. This indicates that this hierarchical structure could provide a means of increasing the interpretability of complex rule bases. Work is continuing on how this hierarchical structure could allow a supervisor to explore and understand a complex collection of rules.

These results demonstrate that adjusting the amount of uncertainty tolerated within the decision making process can be used as a trade off between the interpretability and the performance of the solution. Figure 8 demonstrates how the deviation from the optimal path is reduced in test scenario 3 over E_{max} values 100, 75 and 50. The reduction in the deviation from the optimal path is achieved through specialisation of the decision space by more accurately modelling the behaviour demanded by the supervisor. For example, specialisation adjusts the distance at which the controlled robot performs obstacle avoidance. These results also indicate that specialising the input space in all dimensions limits the scalability of the proposed algorithm. A method of specialising the most uncertain dimension could be employed to improve scalability.

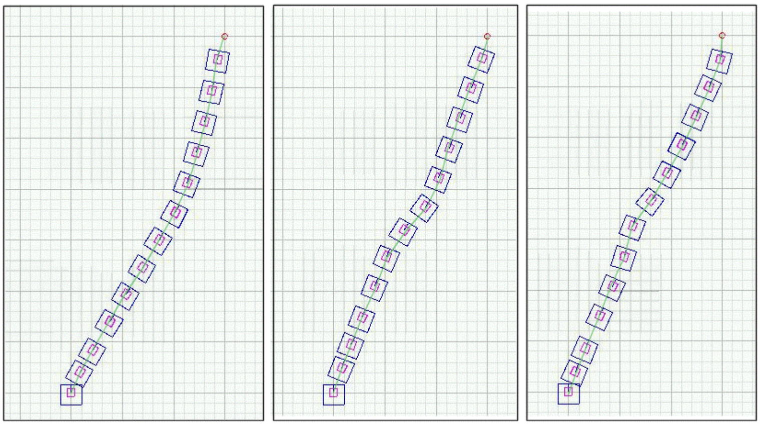

Figure 8: Robot path in test scenario 3 for Emax = 100,75,50

5 Conclusions and Further Work

This paper has demonstrated a novel online method to construct a hierarchical Fuzzy Rule Based System that can achieve levels of accuracy comparative with previous offline methods. The algorithm uses an information theoretic approach to determine if any rule does not effectively model the underlying decision space within a specified degree of uncertainty. The algorithm has been demonstrated to approximate an intermediate complex function and learn a simulated mobile robot task. The results demonstrate that accuracy and interpretability can be controlled using a single tuneable parameter (E_{max}) but more intelligent specialisation is required to improve scalability.

Future work will focus on investigating limitations of the algorithm when applied to a real mobile robot using a human supervisor. Another important aspect of constructing a human interpretable controller is the visualisation of complex controllers with large numbers of rules. When a high degree of certainty in the decision is required the algorithm described within this paper produces a solution with a large number of rules. A possible method of interpreting a large number of rules could arise from the hierarchical structure of the rule base. This may provide a means of allowing the supervisor to explore different parts of the decision space (rules) at varying levels of complexity.

Acknowledgements

This work was funded by the Advanced Technology Centre Filton, BAE SYSTEMS Ltd.

References

Alcala, R., Casillas, J., Cordón, O. and Herrera, F. (2001) *Building Fuzzy Graphs: Features and Taxonomy of Learning for Non-grid-oriented Fuzzy Rule-based Systems, Journal of Intelligent and Fuzzy Systems,* 11**,** pp 99-119.

Al-sharhan, S., Karray, F., Gueaieb, W. and Basir, O. (2001) In Fuzzy Entropy: A Brief Survey at The 10th IEEE International Fuzzy Systems Conference, 2001, pp 1135-1139

Bardossy, A. and Duckstein, L. (1995) *Fuzzy Rule-based Modelling with Application to Geophysical, Biological and Engineering Systems,* CRC Press.

Bastian, A. (1994) *How to handle the flexibility of linguistic variables with applications, Intl. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* 2**,** pp 463-484.

Bezdek, J. C. (1992) *Fuzzy Models For Pattern Recognition: Methods that search for structures in Data,* IEEE Press.

Carse, B., Fogarty, T. C. and Munro, A. (1996) *Evolutionary Learning of Fuzzy Rule Based Controllers using Genetic Algorithms, Fuzzy Sets and Systems,* 80**,** pp 273-293.

Cordón, O., Herrera, F., Hoffmann, F. and Magdalena, L. (2001a) *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases,* World Scientific.

Cordón, O., Herrera, F. and Zwir, I. (2001b) *Fuzzy Modeling by Hierarchically built Fuzzy Rule Bases, International Journal of Approximate Reasoning,* 27**,** pp 61-93.

Furuhashi, T., Nakaoka, K. and Uchikawa, Y. (1996) *A Study on Fuzzy Classifier Systems For Finding a Control Knowledge of Multi-input Systems, Genetic Algorithms and Soft Computing***,** pp 489-502.

Grefenstette, J. J. and Ramsey, C. L. (1982) In An Approach to Anytime Learning at Ninth International Machine Learning Workshop, San Mateo, CA, Morgan Kaufmann, pp 189--195

Hagras, H., Callaghan, V. and Colley, M. (2001) *Outdoor Mobile Robot Learning and Adapation, IEEE Robotics and Automation Magazine,* 8**,** pp 53-69.

Hoffman, F. and Pfister, G. (1997) *Evolutionary Design of a Fuzzy Knowledge Base for a Mobile Robot, International Journal of Approximate Reasoning,* 17**,** pp 447-469.

Holve, R. (1998) In Investigation of Automatic Rule Generation for Hierarchical Fuzzy Systems at IEEE World Congress on Computational Intelligence, FUZZ IEEE, Anchorage, Alaska, May 4-9 1998, pp 971-978

Holve, R. and Protzel, P. (1996) In Generating Fuzzy Rules by Learning from Examples at Biennial Conference of the North American Fuzzy Information Processing Society - NAFIPS, Berkeley, CA, USA, June 19th-22nd 1996, pp 451-455

Melhuish, C. and Fogarty, T. (1994) In Applying Restricted Mating Policy to Determine State Space Niches Using Immediate and Delay Reinforcement at Evolutionary Computing, AISB Workshop, Leeds, UK, 1994, pp 224-237

Ogata, K. (1997) *Morden Control Engineering*, Tim Robbins.

Quinlan, J. R. (1990) *Induction of Decision Trees,* Morgan Kaufmann.

Shannon, C. E. (1948) *A mathematical theory of communication, Bell System Technical Journal,* 27**,** pp 379-423 and 623-656.

Tunstel, E., Lippincott, T. and Jamshidi, M. (1996) In Introduction to Fuzzy Logic with Application to Mobile Robotics at First National Students Conference of the National Alliance of NASA University Research Centres, NC A&T State Univ, Greensboro, NC, Marc 1996, pp

Zadeh, L. A. (1965) *Probability measures of Fuzzy Events, Journal Math. Anal. Appl.*, 23**,** pp 421-427.

Zadeh, L. A. (1994) *Soft Computing and Fuzzy Logic, IEEE Software***,** pp 48-56.

Zadeh, L. A. (1998) *Fuzzy Logic, Computer,* 21**,** pp 83-93.