Fuzzy Situation based Navigation of Autonomous Mobile Robot using Reinforcement Learning

Rhea Guanlao, Petr Musilek, Farooq Ahmed, and Armita Kaboli
Department of Electrical and Computer Engineering
University of Alberta, Edmonton, AB, T6G 2V4 Canada
musilek@ece.ualberta.ca

***Abstract*–Design of high-level control systems for autonomous agents, such as mobile robots, is a challenging task. The complexity of robotic tasks, the number of inputs and outputs of such systems and their inherent ambiguity preclude the designer from finding an analytical description of the problem. Using the technology of fuzzy sets, it is possible to use general knowledge and intuition to design a fuzzy control system that encodes the relationships in the control domain into the form of fuzzy rules. However, control systems designed in this way are severely limited in size and are usually far from being optimal.**
In this paper, several techniques are combined to overcome such limitations. The control system is selected in the form of a general fuzzy rule based system. Antecedents of this system correspond to various situations encountered by the robot and are partitioned using a fuzzy clustering approach. Consequents of the rules describe fuzzy sets for change of heading necessary to avoid collisions. While the parameters of input and output fuzzy sets are designed prior to robot engagement in real world, the rules to govern its behaviour are acquired autonomously endowing the robot with the ability to continuously improve its performance and to adapt to changing environment. This process is based on reinforcement learning that is well suited for on-line and real-time learning tasks.

***Index Terms*–Robotics, Navigation, Fuzzy Control, Fuzzy Clustering, Reinforcement Learning.**

I. INTRODUCTION

The navigation task for autonomous mobile robots involves traversing unfamiliar environments while avoiding obstacles and dealing with noisy, imperfect sensor information. While several control schemes have been developed to accomplish the navigation task, the behaviour based control paradigm continues to be an essential element of autonomous mobile robot navigation. Behaviour based controllers often incorporate reactive control schemes that use range sensors such as sonar and ladar to provide responsive control in dynamic and uncertain environments.

Fuzzy logic control is well suited for behaviour based approaches towards navigation as it can be made robust against sensor noise and imperfect odometry, and fuzzy behaviours have been implemented in many successful autonomous navigation systems [10]. However, while fuzzy logic behaviour based design can produce good performance in the navigation task, the design of fuzzy behaviours often involves an ad-hoc trial and error approach that is inflexible and time consuming to implement. In recent research, the adoption of learning strategies such as evolutionary and reinforcement learning has attempted to address this problem by automating the design of low level behaviours and behaviour integration [4], [14]. However, the large state space involved in mobile robot navigation often slows the learning of navigation tasks, and attempts to lower the dimensionality of the state space by grouping input data using expert knowledge just shifts the element of ad-hoc human input from one design stage to another.

Another way to reduce the state space dimensionality is to cluster input data using a fuzzy clustering algorithm, so that the resulting fuzzy clusters can then be considered as fuzzy states by the learning algorithm. Input clustering approach to learning obstacle avoidance behaviours is presented in this paper. First, sonar data from sample indoor environments are partitioned with a fuzzy clustering algorithm. The resulting partitions may not correspond to the state space partitions that a human expert would use, but they would provide a good description of typical situations the robot will encounter in its environment. In the learning stage, each fuzzy state is associated with an action to be taken in that state; and a reinforcement learning algorithm is used to associate punishments generated from obstacle collisions to the state-action pair that caused the collision so that over several iterations an optimal obstacle avoidance policy is learned.

The paper is organized in five sections. Section II summarizes background knowledge of the navigation problem, fuzzy rule based systems, clustering, and reinforcement learning. The proposed fuzzy situation based navigation system is described in Section III and results obtained in simulation are presented and analysed in Section IV. Finally, Section V brings main conclusions and indicates possible directions of future work.

II. BACKGROUND

A. Navigation Problem

The basic task of an autonomous mobile robot is to navigate safely to a specified goal location. The high level problem of goal seeking is that of selecting strategies to specify the goals and of planning routes from a current position to these goals. The other aspect of navigation, safety, can be handled by various methods of obstacle avoidance.

Obstacle avoidance typically uses information about robot's immediate surroundings to choose action (e.g. turning) appropriate to avoid obstacles in the environment. Such information is provided by the means of robot's sensory subsystem, equipped e.g. with proximity sensing devices. Pioneer 2DX robotic platform considered in this paper has 8 sonar sensors spaced nearly evenly from -90 degrees to 90 degrees around its front side (cf. Figure 1).

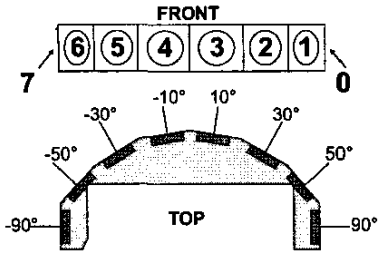

Fig. 1 Sonar configuration on mobile robotic platform Pioneer 2DX

B. Fuzzy Clustering

Clustering, or cluster analysis, is a technique for partitioning and finding structures in data [6]. By partitioning a data set into clusters, similar data are assigned to the same cluster whereas different data should belong to different clusters. While it is possible to assign each data-point strictly to only one cluster, such crisp assignment rarely captures the actual relationship among the data, i.e. real data-points can to some degree simultaneously belong to several clusters. This leads to the formulation of fuzzy clustering [2], where membership degrees between zero and one are used instead of crisp assignment of the data to clusters. In addition to better conformity, fuzzy clustering provides ability to interpolate between the cluster prototypes. This latter property is very important when considering clustering in connection with rule-based systems as described in the following section.

Fuzzy c-means clustering algorithm [2] is a commonly used method of fuzzy clustering. It is based on the minimization of an objective function J with respect to U, a fuzzy partition of the data set, and V, a set of c prototypes

$$J_m(U,V) = \sum_{j=1}^{n}\sum_{i=1}^{c} u_{ij}^{m}\left\|X_j - V_i\right\|^2, \qquad (1)$$

where u_{ij} denotes membership of data-point X_j, $j\in[1,n]$ in cluster V_i, $i\in[1,c]$, $m\geq 1$ is partition order, and $\|*\|$ is any norm expressing the similarity between measured data and prototypes (e.g. Euclidean distance). Fuzzy partition is obtained through iterative optimization of (1) with the gradual update of memberships u_{ij} and cluster centres V_i.

C. Fuzzy Rule Base Systems

Fuzzy rule based systems (FRBS) can be used to extend conventional approach to mobile robot navigation using the notion of fuzzy behaviours [14]. Fuzzy behaviours associate a soft condition with an appropriate fuzzy action. For example, a fuzzy behaviour for avoiding an obstacle in front of the robot could be described using the following rule:

IF **there is an obstacle** *close* to front-left of the robot,
THEN **turn** right *slightly*.

The labels *close* and *slightly* make the behaviours fuzzy and endow the underlying fuzzy control system with the ability to interpolate between the discrete cases described by a limited number of rules.

D. Reinforcement Learning

Reinforcement learning problems require online learning and dynamic adjustments in order to search for optimal methods and actions. Some examples include adaptive control systems, game playing machines, an autonomous robot navigation and exploration. Search heuristics such as genetic algorithms, genetic programming, or simulated annealing have been previously applied in solving such problems. However, they lack in the sense that they cannot learn while interacting with its environment and update itself accordingly. Instead a more general machine learning technique based on positive/negative reinforcement and trial and error interactions with the environment is efficient. It is favoured for its generality and similarity to how humans think and learn on a higher level – through experience. Sutton describes reinforcement learning as a computational approach to understanding and automating goal-directed learning and decision-making [10]. The learning agent determines which actions yield the most rewards in a particular environment by exploring and performing several different actions repeatedly thus learning which actions to exploit. Reinforcement learning is a continual combination of exploration and exploitation. A progressive combination of the two should be performed to continually evaluate and improve the agent.

A reinforced learning environment consists of a set of defined attributes. The *actions* are the varying methods the learning agent explores in order to determine the optimal actions. The *reward* function defines the desired goal of the learning problem. The *value* function defines which actions produce the highest rewards in the long run or after several states, as opposed to which actions produce the highest rewards in the immediate state. The *policies* determine how the learning agent evaluates and responds to the rewards and values measured. It may define, perhaps with a set of stimulus-response rules, when and how to explore and exploit actions. Finally, the *model* is used simply to mimic the behaviour of the environment.

III. Fuzzy Situation Based Navigation

The proposed navigation system is based on FRBS that performs mapping from the space of situations that the robot can encounter to the space of actions the robot can perform. In system described here, the antecedents of the rules correspond to situations obtained through the fuzzy clustering of robot's sensory space, while the consequents express the turning angle corresponding to the change of robot's current heading. The rules themselves are obtained autonomously through the process of reinforcement learning. The overall structure of the navigation system is illustrated in Figure 2.

The system is composed of four major subsystems: fuzzification, rule base, defuzzification, and reinforcement learning. Although the system is currently using off-line clustering, it is proposed to increase its autonomy and adaptability through an on-line incremental clustering as indicated by the dashed arrow in the block diagram.

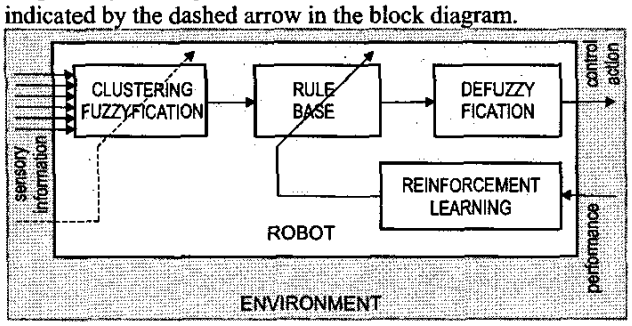

Fig. 2 Architecture of the navigation system

A. Clustering: Finding Situations

As mentioned in the previous section, antecedents of the FBRS describe situations the robot encounters. These situations are in turn defined by the sensory signals. In order to collect sonar signals corresponding to different situations, experiments have been performed by placing the robot into simulated environments containing commonly encountered situations, cf. Figure 3.

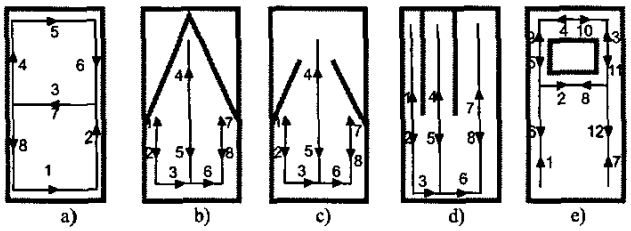

Fig. 3 Room setup and robot trajectories used for data collection

Acquired data has been used to obtain a partition of the 8-dimensional sonar space to the space of fuzzy situations, e.g. "open space", "right corner", etc. Although the situations can be labelled and interpreted this way, these labels only serve to gain insight into the results of clustering and are not relevant for the robot and its navigation system. Clustering has been performed using fuzzy c-means clustering algorithm briefly discussed in Section II.

In practical applications, there are several parameters of the clustering algorithm to be chosen for (1) that have a profound impact on quality of resulting partition, namely partition order m, and number of clusters c. There are several cluster validity measures [6], [9], [12], [13] that can be used to evaluate quality of the partition and to aid the process of selecting particular values of m and c.

Bezdek [2] defined partition coefficient V_{PC}

$$V_{PC}(U)=\frac{\sum_{k=1}^{n}\sum_{i=1}^{c}u_{ik}^2}{n}, \tag{2}$$

and partition entropy V_{PE}

$$V_{PE}=-\frac{1}{n}\{\sum_{k=1}^{n}\sum_{i=1}^{c}u_{ik}^2\log_a(u_{ik})\},\ a\in(1,\infty). \tag{3}$$

V_{PC} takes its maximum (and V_{PE} takes its minimum) on every hard c-partition for which $u_{ik}(x)\in\{0,1\}$ and V_{PC} takes its minimum (and V_{PE} takes its maximum) on the fuzziest possible partition for which $u_{ik}(x)=1/c$. Therefore, to achieve good inter-cluster separation, V_{PC} should be maximized while V_{PE} should be minimized.

Non-fuzzy index [13] is defined as

$$NFI(c)=\frac{c[\sum_{k=1}^{K}\sum_{i=1}^{n}u_{ik}^2(x)]-n}{n(c-1)} \tag{4}$$

This index provides another measure of how fuzzy a c-partition is: it takes its maximum value for crisp partitions and its lowest value in the case of fuzziest clustering. Analogously to V_{PC}, NFI should be maximized.

Minimum and Mean Hard Tendency [12]

$$\mathrm{MinHT}=\max\{-\log_{10}(T_s)\}\ \ 1\le s\le c, \tag{5}$$

$$\mathrm{MeanHT}=\frac{1}{c}\sum_{s=1}^{c}-\log(T_s), \tag{6}$$

with hard tendency $T_s\to 0$ defined as

$$T_s=\frac{\sum_{x_i\in X_s}r_i}{\mathrm{card}(X_s)};\ r_i=\frac{u_{ki}}{u_{ji}}, \tag{7}$$

where $\mathrm{card}(X_s)$ is cardinality of the input data set, and

$$u_{ji}=\max_{t}\{u_{ti}\}\ \text{and}\ u_{ji}=\max_{t\neq j}\{u_{ti}\};\ 1\le t\le c \tag{8}$$

are the first and second maxima of the elements of U_i which determine the membership of the element x_i in all clusters. Therefore, MinHT extracts the least favourable hard tendency of the set of clusters while MeanHT determines the average of the hard tendencies of all clusters.

All functionals (1-6) have been examined to determine the optimal values of the clustering parameters. The result of clustering process with the determined values of m=1.2 and c=8 is depicted in Figure 4 in form of polar plots of sonar signals corresponding to the determined cluster prototypes.

B. The Rule Base

For the purpose of the robot navigation problem, the fuzzy inputs are derived from the sensory information of the robot's front eight sonars (cf. Figure 1). The fuzzified input provides an indication of the robot's position relative to the walls and obstacles in the environment. The defuzzified output should give constructive directions for the robot to effectively stay clear from collisions while moving at a constant velocity. The number of input fuzzy sets and the output fuzzy sets should be minimized to reduce complexity of the search space for fuzzy rule base learning. In order to minimize the input state space, input generalization is performed using fuzzy clustering as described in the previous section. A single input variable, labelled *situation*, is used to represent the situational

environment of the robot. Its fuzzy sets' membership functions are derived from the results of fuzzy c-means clustering used to transform the robot's eight sonar readings into clusters that make clear definitions of the robot's situations. Through inspection of the eight cluster centers that were derived, their cluster centers were extracted and the situations are qualitatively described as *front obstacle, right corner, right obstacle, open space, narrow corridor, left obstacle, left corner,* and *wide corridor* (corresponding, respectively, to sonar configurations a-h in Figure 4).

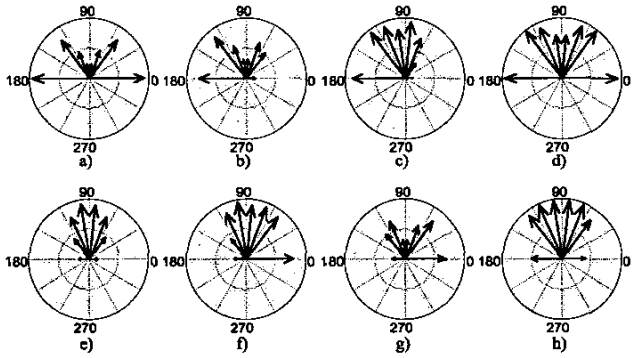

Fig. 4 Cluster prototypes of fuzzy situations

Likewise, there is one output state variable that dictates the relative heading the robot should rotate by. The linguistic label for this output is *turn amount.* The fuzzy set labels and their respective headings for *turn amount* are: *complete left* (90°), *very left* (60°), *left* (30°), *straight* (0°), *right* (-30°), *very right* (-60°), *complete right* (-90°). The surface structure of a fuzzy linguistic rule showing the relationship between the input and output linguistic variables takes the following form

Rule 1: IF **situation** is *front wall*
THEN **turn amount** is *complete right*

The database pf a fuzzy system defines the membership functions of the input and output fuzzy sets for fuzzification and defuzzification. Since only the cluster centres of the input fuzzy sets are known, the degrees of membership for fuzzification is determined using the distance between the actual sonar values and centre of each fuzzy cluster. This is accomplished using the same expression figuring in the objective function of fuzzy c-means clustering algorithm (1).

$$u_{lk} = \left[\sum_{j=1}^{c} \left(\frac{D_{lkA}}{D_{jkA}} \right)^{\frac{2}{m-1}} \right]^{-1}, \qquad (9)$$

where u_{lk} is the degree of membership of the current data point k to cluster l, c is the number of clusters, and D_{lkA} is the distance of the data point k to cluster l measured using Euclidian distance A.

The next step in the fuzzy inference model [6] is rule evaluation. Since there is only one input variable, there is always one antecedent in each rule. Thus, the rule strength, or the numerical minimum value of the antecedents in a rule, is strictly the membership value of the current situation to the rule's single antecedent. Consequently, the numerical strength for each output fuzzy variable is the maximum value for that consequent derived among all the rules in the rule base after rule evaluation.

Finally, defuzzification is performed to convert the output variable strengths into crisp system outputs. Singleton membership functions are used to define each *turning amount* output fuzzy set. The value of each singleton is equal to the turning amount in degrees that the output variable refers to. The crisp output is determined via the commonly used Center of Gravity Algorithm for defuzzification (COG). COG defuzzification is determined by formula

$$y = \frac{\sum_{k=1}^{n} (w_k \cdot x_k)}{\sum_{k=1}^{n} (w_k)}, \qquad (10)$$

where y is the crisp output, w_k is the weight of the fuzzy rule k, x_k is the position of the singleton k in output domain, and n is the number of output fuzzy sets.

C. Reinforcement Learning

Initially, the rule base of the fuzzy system is not known. Reinforcement learning is used to discover the rule base sufficient for obstacle avoidance. The problem definition of reinforcement learning is to determine the optimal policy or mapping between situations and actions (or antecedents and consequents) in order to maximize rewards over time and thus, in this case, maintain repeatable obstacle avoiding performance.

The main challenge that arises in applying reinforcement learning to robot navigation is assigning appropriate immediate and delayed credit to actions. Reinforcement, induced by the robot's immediate and/or past actions, is very particular to locality. Therefore, optimal long term actions are difficult to learn. Q-learning, an off policy [10] control in combination with a temporal difference (TD) algorithm is used to address this challenge and eventually find the optimal policy. A Q-value can be defined as a prediction of the sum of the reinforcements the agent will receive when performing the associated action and following given policy [1]. A Q-value is assigned for each situation-action pair in the problem space. Initially, these are randomly assigned. In this case, the dimension of the lookup table holding the Q-value predictions corresponds to the number of clusters by the number of turning options, respectively. Thus, each situation-action pair has a Q-value or a utility associated with it. After several iterations, the knowledge gained from updating the Q-values is propagated backwards through a lookup table from later states to earlier states in time until it eventually predict the optimal Q-value function.

Each situation-action pair in the lookup table can be regarded as a possible rule in the fuzzy rule base. A rule base is formed

from all possible rules in the fuzzy knowledge base. The number of rules in the rule base at any time is equal to the number of clusters and each rule in the rule base provides an action for each situation. Initially, a rule is added for each situation and the action associated with that situation is chosen randomly.

With each iteration, the robot assesses its situation and fuzzifies the values of its sonar readings. Through rule evaluation, it determines the strength s_i of each rule i. Defuzzification of the rule consequents then produces the final (crisp) change of heading direction for the robot to perform. Therefore, instead of identifying discrete single situations and performing discrete single actions as in traditional reinforcement learning, fuzzified combinations of situations and actions are considered. After the action is taken, the next state is observed and the Q-values in the lookup table are updated. Only the rules which were activated in the rule evaluation step are updated in the lookup table according to

$$Q(s_t,a_t) = Q(s_t,a_t) + \alpha[r(s_t,a_t)+\gamma \max(Q(s_{t+1},a) - Q(s_t,a_t)] \quad (11)$$

where $Q(s_t,a_t)$ is the utility value of situation s and action a at time t, and $r(s,a)$ is the reinforcement received after performing the action a in situation s. The effect of the learning rate α and the discount factor γ on learning is further discussed in [10].

Reinforcement is given in terms of reward or penalty. A reward of +1 is achieved when the robot has maintained the desirable performance of not hitting an obstacle. Alternatively, a penalty of -1 is received when the robot hits an obstacle. The maximum utility possible observed in the next state, $\max(Q(s_{t+1},a))$ can be represented as

$$\max(Q(s_{t+1},a) = \sum_{j=1}^{c} u_{jk_{t+1}} \max(Q(j,a)), \quad (12)$$

where $\max(Q(j,a))$ is the maximum utility in the lookup table possible for cluster or situation j. Since the next state is not a single discrete state, $\max(Q(s_{t+1},a_{t+1}))$ is calculated as the weighted average of the membership degrees to each situation and the maximum utility in the lookup table for that situation. Likewise, since each rule in the rule base is responsible only for a portion of the crisp output heading, the reinforcement and the contribution of $\max(Q(s_{t+1},a_{t+1}))$ should be proportional with respect to the rule strength, s_i, of the corresponding situation-action pair. This leads to reformulation of the update rule to

$$Q(s_t,a_t)_i = Q(s_t,a_t)_i + \alpha[w_{i_{at}} r(s_t,a_t)+\gamma w_{i_{at}} \max(Q(s_{t+1},a) - Q(s_t,a_t)_i] \quad (13)$$

where $w_{i_{at}}$ is the strength of rule i that corresponds to the situation-action pair. The values in the lookup table encode the *estimation policy* that dictates the optimal situation-action pairs. Obviously, the estimation policy favours maximum Q-values. Thus learning ends not when the Q-value ceases to change, but when the estimation policy, or maximum Q-values for each situation, ceases to change.

In order for Q-learning to converge to the optimal policy the robot must visit every situation and execute each possible action in the situation several times. Therefore, a behaviour policy must be chosen that will ensure a sufficient exploration of the search space. Softmax policy [3] has been chosen, which assigns a probability to each action proportional to the situation-action utility value in the lookup table. It is the probability that the rule will be added to the fuzzy rule base: the higher utility value the action has in that situation, the more likely it will be chosen. The softmax policy overcomes the drawbacks of the ε-greedy and ε-soft policies [10], that both select random actions uniformly, by still selecting action stochastically but with favouritism of high utility situation-action pairs. In addition, softmax provides a simple compromise between exploration and exploitation of the reinforcement learning problem.

IV. Results

Convergence is achieved when the estimation policy ceases to change. The Q-values may modulate, however the relative Q-values among the actions for each situation remain constant. After convergence, the robot agent was indeed able to learn the rule base sufficient to solve the obstacle avoidance problem.

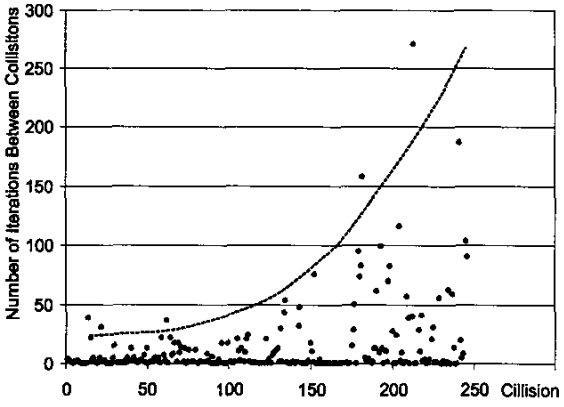

Figure 5 Number of iterations between collisions during the learning process

The graph in Figure 5 illustrates how learning progressed with iterations. It is evident that as the number of iterations increases, the robot's performance appears to degrade at certain points. This demonstrates the exploration aspect of reinforcement learning which continues to experiment with different actions in search for a better performance. The main characteristic to notice is the increasing number of iterations between collisions indicated by the dashed trend line.

The Q-values in the lookup table are essentially predictions of the cumulative sum of reinforcements the agent will receive

by performing the action in the current situation and following the policy thereafter [1]. Therefore, the lookup table should converge such that the most desired actions will have greater positive Q-values (utility values). Consequently, actions with greater utility values have a greater probability of being added to the fuzzy rule base. Table I shows the complete rule base derived from the lookup table in this manner. The agent was able to solve the fuzzy rule base autonomously.

TABLE I
LEANED FUZZY RULE BASE FOR OBSTACLE AVOIDANCE

IF **situation** is	THEN **action** is *Turn*
Front Obstacle	Compete Left
Right Corner	Complete Left
Right Obstacle	Very Left
No Obstacles	Straight
Narrow Corridor	Straight
Left Obstacle	Complete Right
Left Corner	Complete Right
Wide Corridor	Straight

Figure 6 shows a simulated trajectory of the fuzzy controlled robot after learning the fuzzy rule base in Table 1. It is evident that the robot's path is such that it avoids collisions with any of the polygons present in the environment.

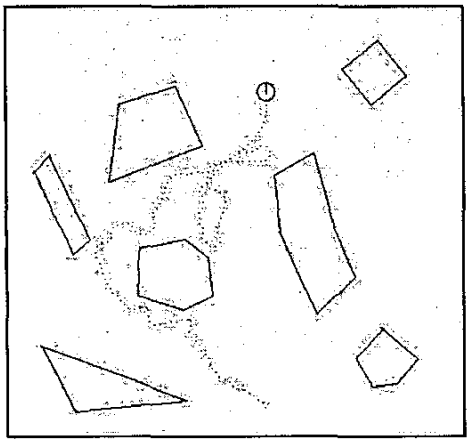

Figure 6. Sample trajectory of robot using the learned fuzzy rules

V. CONCLUSION

In this paper, a new approach for robot navigation has been presented combining favourable properties of several existing technologies: fuzzy clustering for classification of situations in robot's environment, fuzzy rule based systems for inferring actions suitable in different situations, and reinforcement learning to set up the rules of inference in autonomous manner.

To further increase the degree of autonomy of such navigation system, the current off-line clustering of situations could be replaced by an on-line incremental clustering system that would create models of new situations as they arise during the robot's interaction with the world.

The results presented in this paper demonstrate the success of the new approach in solving the problem of obstacle avoidance. The same methodology could be adapted for other types of behaviours taking part in robot navigation, such as wall following, goal seeking, or foraging.

ACKNOWLEDGMENT

Support provided by the Natural Sciences and Engineering Research Council (NSERC) and Canada Foundation for Innovation (CFI) is gratefully acknowledged.

REFERENCES

[1] R.C. Arkin, Behaviour-based Robotics, MIT Press, 1998.
[2] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, New York: Plenum, 1981
[3] J. S. Bridle, Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimates of parameters. In Touretzky, D. S., ed., *Advances in Neural Information Processing Systems*, Morgan Kaufmann, pp. 211—217, 1990
[4] G. Dongbing, H. Huosheng. Reinforcement Learning of Fuzzy Logic Controllers for Quadruped Walking Robots. *Proceedings of 15th IFAC World Congress*, 2002
[5] M. E. Harmon, S. S. Harmon, *Reinforcement Learning: A Tutorial*, 1996
[6] F. Höppner, F. Klawonn, R. Kruse and T. Runkler, *Fuzzy Cluster Analysis*, Wiley, 1999
[7] G. Klir and B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice Hall, 1995
[8] Likhachev, M. and Arkin, R.C., "Spatio-Temporal Case-Based Reasoning for Behavioral Selection," Proc. 2001 IEEE International Conference on Robotics and Automation, pp. 1627–1634. 2001.
[9] N. R. Pal and J. Bezdek, "On Cluster validity for the fuzzy c-Means Model", IEEE Transaction on Fuzzy systems, VOL.3. No.3, August 1995
[10]A. Saffiotti. The Uses of Fuzzy Logic in Autonomous Robot Navigation. *Soft Computing* 1(4):180-197, 1997.
[11]R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998.
[12]F.F. Rivera, E.L. Zapata, and J.M. Carazo, "Cluster validity based on the hard tendency of the fuzzy classification", Pattern Recognition Letters, 11:7-12, 1990.
[13]M. Roubens, "Pattern Classification Problems and Fuzzy Sets", Fuzzy Sets and Systems, 1:239-253, 1978.
[14]K. Ward, A. Zelinsky, P. McKerrow. Learning Robot Behaviours by Extracting Fuzzy Rules From Demonstrated Actions. *AJIPS*, Vol 6, No 3, pp. 154-163, 2001
[15]J. Yen and R. Langari, *Fuzzy Logic: Intelligence, Control, and Information*, Prentice Hall, 1999