Multiple Reward Reinforcement learning control of a mobile robot in home network environment

Dong-Oh Kang, and Jeunwoo Lee

Computer and Software Research Laboratory, Electronics and Telecommunications Research Institute, Korea
(Tel : +82-42-860-5012; E-mail: dongoh@etri.re.kr)
(Tel : +82-42-860-5012; E-mail: ljwoo@etri.re.kr)

Abstract: The following paper deals with a control problem of a mobile robot in home network environment. The home network causes the mobile robot to communicate with sensors to get the sensor measurements and to be adapted to the environment changes. To get the improved performance of control of a mobile robot in spite of the change in home network environment, we use the fuzzy inference system with multiple reward reinforcement learning. The multiple reward reinforcement learning enables the mobile robot to consider the multiple control objectives and adapt itself to the change in home network environment. Multiple reward fuzzy Q-learning method is proposed for the multiple reward reinforcement learning. Multiple Q-values are considered and max-min optimization is applied to get the improved fuzzy rule. To show the effectiveness of the proposed method, some simulation results are given, which are performed in home network environment, i.e., LAN, wireless LAN, etc.

Keywords: Mobile robot, home network, multiple reward, reinforcement learning, Q-learning

1. INTRODUCTION

Recently, technology for home network has made remarkable progress in the industries. And, every home is expected to be equipped with home network capability in the recent future as seen in the concept of ubiquitous computing. Such rapid development of the technology for home network will make home smarter by equipping home with intelligent information devices, which include intelligent service robots. Because service robots are aimed at helping humans in daily life, it is inevitable for service robots to operate in home network environment. Especially mobile robots will play an important role among the service robotics.

In home network environment, a mobile robot is equipped with a communication module, and is given some global position information through home network [1]. Although a mobile robot in home network environment has no sensors such as gyroscopes and laser range finders which are expensive and give limited information about environment, the robot can get the global position information through home network, which is provided by some devices such as the home server. The home environment renders various unstructured environment which may be changed by replacing sensor devices, network media, or changing the location of the sensors. The change in the environment deteriorates the performance of control of mobile robots. A mobile robot in home network environment should cope with the problem. Because it is hard to model the changes in home network environment, we apply reinforcement learning to the problem. Furthermore, multiple control objectives should be considered in mobile robot control, for example, the energy consumption, safety, tracking error, etc. Therefore, we propose a multiple reward reinforcement scheme for mobile robot control with multiple objectives in unstructured home network environment.

Reinforcement learning is how to learn the optimal policy based on the reward from environment [2]. Reinforcement learning has been popularly applied to mobile robot control in unstructured environment since it needs no model for environment [3]. Especially, in this paper, a multiple reward reinforcement learning is proposed to deal with mobile robot control problem, because multiple control objectives should be considered in mobile robot control. The method uses the concept of Pareto optimality in optimizing the policy. Among conventional multiobjective optimization methods, the max-min optimization produces one of Pareto optimal solutions, which maximizes the objective with minimum value among the objectives [4]. In this paper, the max-min optimization is applied to reinforcement learning.

We adopt the fuzzy inference system for mobile robot control, which has multiple consequent singletons for the consequent fuzzy set [5]. The fuzzy inference system has the structure similar to the fuzzy controller with inconsistent rule base [6]. Among the multiple consequent singletons, one singleton is selected as the consequent fuzzy set for the fuzzy rule. Multiple reward reinforcement learning is applied to design the fuzzy controller based on rewards corresponding to multiple objectives. We apply multiple reward fuzzy Q-learning for the multiple reward reinforcement learning.

The outline of the paper is as follows. Section 2 reviews several important preliminaries. In Section 3, the multiple reward fuzzy Q-learning is proposed as the multiple reward reinforcement learning for control of the mobile robot. In Section 4, to show the effectiveness of the proposed method, some simulation results are given, which are performed in real home network environment such as LAN, and wireless LAN, etc. Finally, Section 5 concludes the research and provides the discussion about the further research.

2. PRELIMINARY

2.1 Reinforcement Learning

Reinforcement learning is based on Markov decision process where the information available to the agent in the current situation is sufficient to determine the future states of the environment independent of the past information. It is composed of state space *S* and action space *A*.

$$P_{ss'}^{a} = \Pr\{s_{t+1} = s' \mid s_t = s, a_t = a\} \quad (1)$$

where $P_{ss'}^{a}$ is the probability of transition from state *s* to state s' under action *a*. A policy is a mapping from perceived states of the environment to actions to be taken in those states: $p : S \to A, a = p(s)$. In stochastic environment, generally, policies may be stochastic.

A reward function maps each perceived state (or state-action pair) of the environment to a single number, a reward, indicating the intrinsic desirability of the state.

$$r_{t+1} = \Re_{t+1}(s,a) : S \times A \to R \,. \quad (2)$$

Just as the policy, it may be stochastic.

A state-value function $V^p(s)$ of the policy p specifies what is good in the long run when the system starts at the state s and adopts the policy p.

In case of an infinite-horizon model, the expected discounted return is used for the state-value function as follow:

$$V^p(s) = E_p(R_t \mid s_t = s) = E_p(\sum_{k=0}^{\infty} g^k r_{t+k+1} \mid s_t = s)$$
$$= E_p(\sum_{k=0}^{\infty} g^k \Re_{t+1}(s_{t+k}, p(s_{t+k})) \mid s_t = s) \quad (3)$$
$$= \sum_a p(s,a) \sum_{s'} P^a_{ss'}(\Re^a_{ss'} + gV^p(s')),$$

where s is an initial state, s_t is a state at time t after starting from the initial state s, r_{t+1} is a reward at time t given that the agent follows the policy p, and $0 \le g < 1$ is the discount rate. The state-value function defines a partial ordering over policies in the set $\prod$ of the possible policies p's, in the sense that a policy p is better than or equal to a policy p' iff $V^p(s) \ge V^{p'}(s)$ for all $s \in S$.

If the policy p is optimal, it satisfies the following relation called Bellman optimality equation for V^*:

$$V^*(s) = \max_{a \in A(s)} Q^*(s,a)$$
$$= \max_a \sum_{s'} P^a_{ss'}(\Re^a_{ss'} + gV^*(s')). \quad (4)$$

Therefore, reinforcement learning is how to get the optimal policy which gives the optimal value function. Adaptive heuristic critic and Q-learning are two major reinforcement learning methods [2].

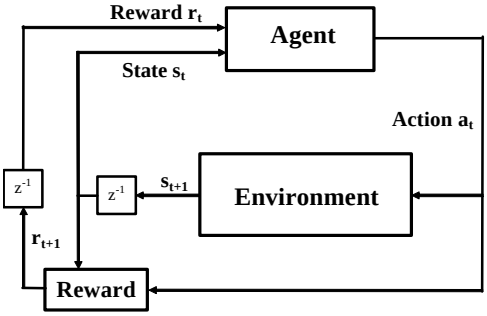


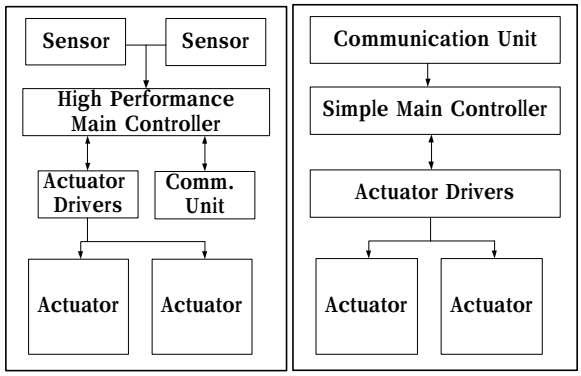
Fig. 1. General structure of reinforcement learning

Q-learning learns the action-state value function Q rather than state value function in order to get an optimal policy. Using the Q value, the optimal policy can be obtained. Q value is learned by the following rule:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + a[r_{t+1} + g \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (5)$$

2.2 Mobile Robot in home network environment

In general, a mobile robot has some sensors to detect its location and posture, whereas sensors are equipped within home network in home network environment. Therefore, a mobile robot in home network environment communicates with sensors through home network. Therefore, a mobile robot in home network environment has three parts: main controller, actuators, and communication units, while the conventional robot has four parts: main controller, actuators, sensors, and communication units [1].



(a) General mobile robot architecture.
(b) Mobile robot architecture in home network.
Fig. 2 Sensing architecture of mobile robots in the home network environment [1].

Although the mobile robot in home network environment has no sensors such as gyroscopes and laser range finders which are expensive and give limited information about environment, the robot can get the global information through home network, which is provided by some devices such as the home server. The home server provides the global map by gathering sensor information from various sensors pervaded in home through home network, and the middleware can give interoperability among heterogeneous devices.

The home environment renders various unstructured environment which may be changed by replacing sensor devices, network media, or changing the location of the sensors. The change in the environment deteriorates the performance of control of mobile robots. In this paper, the change in home network environment is assumed as the change in time delay of sensor values and in noise level.

3. MULTIPLE REWARD REINFORCEMENT LEARNING FUZZY CONTROL

3.1 Fuzzy Inference System

For the fuzzy controller to be designed, the fuzzy inference system is used in the paper, which has multiple consequent singletons for the consequent fuzzy set [5]. The fuzzy inference system has the structure similar to the fuzzy controller with inconsistent rule base [6]. Among the multiple consequent singletons, one singleton is selected as the consequent fuzzy set for the fuzzy rule. This is the process of the design of the fuzzy controller of the multiobjective control problem.

The fuzzy inference system is composed of four layers as depicted in Fig. 3. Each layer of the fuzzy inference systems is as follows:

Layer 1) Input layer.
Layer 2) Membership calculation of each antecedent term.
Layer 3) Rule base node. It connects the input from the layer 2 to the output.
Layer 4) In the node, the defuzzification is performed and the final output of the fuzzy controller is determined. In the paper, output fuzzy sets are singleton, and the center-average defuzzification is adopted as:

$$y = \frac{\sum_{i=1}^{N} y_i m_i(\mathbf{x})}{\sum_{i=1}^{N} m_i(\mathbf{x})} \quad (6)$$

where y_i is the singleton consequent singleton of the rule *i*, and $\mu_i(\mathbf{x})$ is the firing strength of the rule *i*.

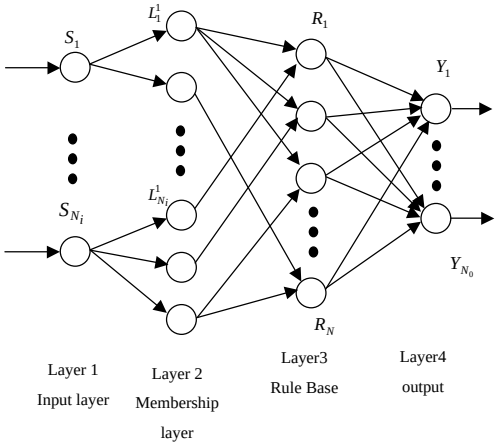


Fig. 3 Adaptive Fuzzy Inference System [5], where N_i: number of the input variables, N: number of the rules, N_o: number of the outputs

In the fuzzy controller, multiple consequent singletons for each rule may be used. The output consequent set is singleton but has many possible candidates. The rule is in the form of the following:

$$R_i: \text{If } x_1 \text{ is } L_1^i \text{ and } \ldots \text{and } x_N \text{ is } L_1^N, \text{ then } u \text{ is } U^i$$
$$U^i \in \{U^{i,1}, U^{i,2} \cdots U^{i,p}\} \quad (7)$$

At any instance, only one output consequent singleton should be selected in the scheme.

$$y = \frac{\sum_{i=1}^{N} U^i m_i(\mathbf{x})}{\sum_{i=1}^{N} m_i(\mathbf{x})} \quad (8)$$

where U^i is the selected output consequent term of the rule *i* among $\{U^{i,1}, U^{i,2} \cdots, U^{i,p}\}$.

3.2 Multiple Reward Reinforcement Learning

For multiple reward reinforcement learning, we extend the fuzzy Q-learning structure [5]. The method uses the concept of Pareto optimality in optimizing the policy. Among conventional multiobjective optimization methods, the max-min optimization produces one of Pareto optimal solutions, which maximizes the objective with minimum value among the objectives [4]. Therefore, we apply the concept of max-min optimization for multiple objective optimization to fuzzy Q-learning for multiple reward reinforcement learning.

To implement the max-min policy in fuzzy Q-learning, we use the multiple action-state value functions of each action as follows: Q_j^i where *i* is the index for rule and *j* is the index for objective.

In case of fuzzy inference system, for each consequent part of each fuzzy rule, multiple action-state value functions are assigned and the minimum value among the action-state value functions of multiple objectives is considered as its action-value. And, if we take an ordinary greedy policy, we have taken the very max-min optimization.

The expected action-state value of each action of each rule is updated by the temporal difference of the expected overall action-state value function corresponding objective as expressed in Eq. (9).

$$\Delta Q_j^i(t) = a[r_t^j + g \max_a Q_{j,t-1}(s_t, a) - Q_{j,t-1}(s_{t-1}, a_{t-1})]$$
$$\text{for all } i = 1, \cdots, N \quad (9)$$

where *t* is the time index, a is the learning rate, g is the discount rate, r_t^j is the reward corresponding the *j*th objective, *i* is the index for rule, *N* is the total number of rules, and *j* is the index for objective.

Therefore, learning algorithm for multiple reward fuzzy Q-learning as follows:

Step1) Calculate the maximum action-state value.

$$Q_{j,t}^*(s_t) = \sum_{R_i \in A} \max_{U_{i,k} \in U(A)} Q_{j,t}^{U_{i,k}}(U_{i,k}) f_{R_i}(s_t) \quad (10)$$

where $f_{R_i}(s_t) = \frac{m_i(s_t)}{\sum_{j=1}^{N} m_j(s_t)}, i = 1, \cdots, N$ is the truth value of each rule, *i* is the index for fuzzy rule, and *j* is index for the objective.

Step 2) Compute temporal differences for each objective.

$$d_t^j = r_t^j + g Q_{j,t-1}^*(s_t) - Q_{j,t-1}(s_{t-1}, U_{t-1}) \quad (11)$$

where g is the discount rate, and $Q_{j,t}(s_t, U_t) = \sum_{R_i \in A} Q_{j,t}^{U_t^i} f_{R_i}(s_t)$.

Step 3) Update the quality vector, that is, utility vector of each rule output.

$$\bar{v}_{j,t} = \bar{v}_{j,t-1} + b d_{j,t} \bar{\mathbf{F}}_{t-1}, j = 1, \cdots, M$$
$$Q_t^{U_{i,k}} = Q_{t-1}^{U_{i,k}} + J d_t^p e_{t-1}^{U_{i,k}}, i = 1, \cdots, N, k = 1, \cdots, q, \quad (12)$$

where J is a learning rate, e_t^i is the eligibility trace.

Step 4) Eligibility Update

$$e_t^{U_{i,k}} = \begin{cases} l' e_{t-1}^{U_{i,k}} + f_{R_i}(s_t) & (U_t^i = U^{i,k}) \\ l' e_t^{U_{i,k}} & \text{otherwise} \end{cases}, \quad (13)$$

where l' is the actor recency factor.

Step 5) Select a new action U_t^i where U_t^i is the e-Greedy action of rule *i* at time step *t* [5]. The utility of an action is determined by the minimum utility of the action among the multiple utilities corresponding to multiple objectives. Maximization process is performed stochastically by e-Greedy selection.

Step 6) Calculate the value of the action-state value function.

$$Q_{j,t}(s_t, U_t) = \sum_{R_i \in A} Q_{j,t}^{U_t^i} f_{R_i}(s_t) \quad (14)$$

where U_t is the global action.

Step 7) Take the action $U_t = \frac{\sum_{i=1}^{N} U_t^i \mu_i(s_t)}{\sum_{i=1}^{N} \mu_i(s_t)}$.

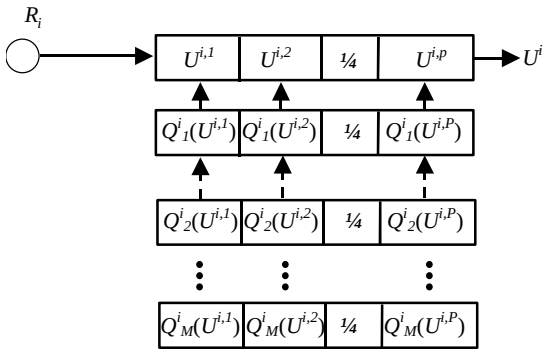


Fig. 4 Selection of the consequent term for each rule.

4. SIMULATION

4.1 Mobile robot

For simulation, the kinematics of a mobile robot is used as Eq. (15). Input variables are the velocity values of both wheels. We assume there is a kind of damping factor when we drive the wheels, therefore, the first order dynamic equation as Eq. (16) is used.

$$\dot{P} = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{q}_c \end{bmatrix} = \begin{bmatrix} \cos q_c & -h\sin q_c \\ \sin q_c & h\cos q_c \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{15}$$

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} (v_R + v_L)/2 \\ (v_R - v_L)/L \end{bmatrix}$$

$$\begin{aligned} \dot{v}_R &= -0.8 v_R + 0.8 u_R \\ \dot{v}_L &= -0.8 v_L + 0.8 u_L \end{aligned} \tag{16}$$

where v_R is the right wheel velocity value, v_L is the left wheel velocity value, h is the displacement between the center of the robot and the wheel axis, and L is the distance between two wheels. For simulation, we use h=0 (m), L=0.3 (m).

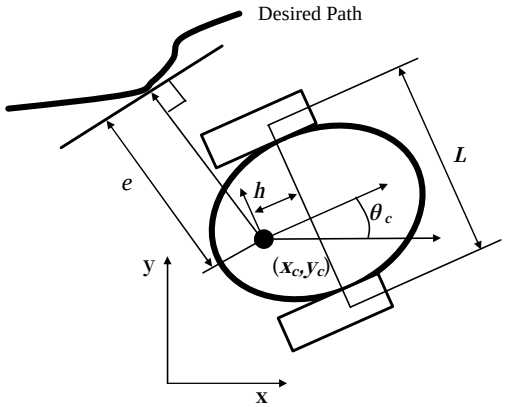


Fig. 5 Kinematics of a mobile robot.

The multiple rewards for reinforcement learning are given as follows:

$$\begin{cases} r_1 = 1. - e/0.7 \\ r_2 = 1. - |q|/1.7 \end{cases} \tag{17}$$

where e is the distance between the desired path and the center of the mobile robot. The position error and the posture error are considered as control objectives.

In simulation, we use two personal computers. One personal computer emulates a mobile robot, and the other emulates the sensors. Two computers communicate through home network just like a mobile robot and sensors in home network. Fuzzy controller uses the data from remote computer which emulate sensors in home network, whereas the data are originated from the computer where the fuzzy controller and mobile robot kinematics are emulated. The data rebound to the computer which originates the data. The change in home network environment is assumed as the change in time delay of sensor values and in noise level.

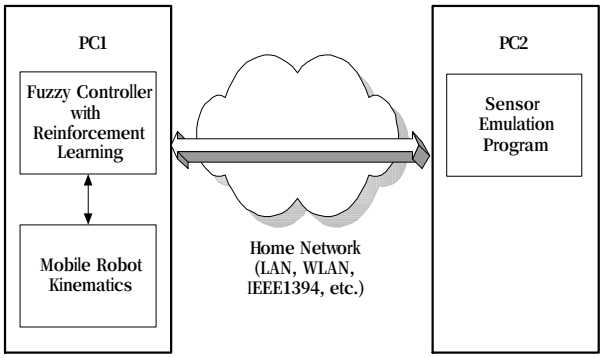


Fig. 6 Simulation Environment.

4.2 Simulation results

We assume the sensor noises have Gaussian random distribution with zero mean and standard deviation (0.001, 0.001, 0.001) for the posture variable (x, y, q). Initial posture of the mobile robot is set as (0,-0.4,0) and the desired path is set to the x-axis. Therefore, the distance between the desired path and the center of the mobile robot, that is, e is the same as $|y|$. The parameters for reinforcement learning scheme are as follows:

$$\begin{cases} g = 0.1 \\ b = 0.1 \\ J = 0.1 \\ l' = 0. \end{cases} \tag{18}$$

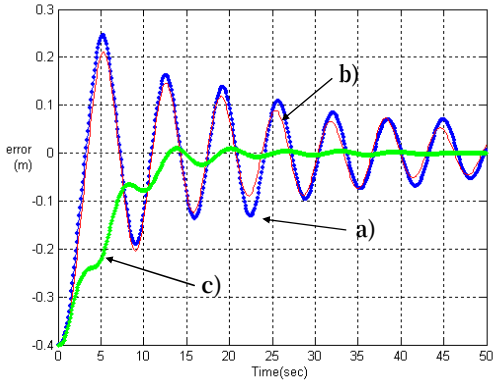
The trials are performed for 10 times where each trial is composed of ten learning periods. Both average squared error sums of e and q among trials are used as the performance indices.

Table 1 Simulation results

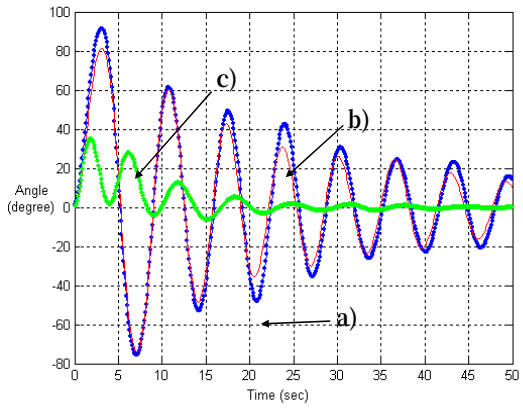
Simulation type	Performance Index of e	Performance Index of ?
Random delay with no learning	70.01	2088
Random delay with learning	66.523	1902
Random delay with multiple reward reinforcement learning	50.97	786.9
LAN with no learning	64.6	1737.3

LAN with learning	62.73	1651.85
LAN with multiple reward reinforcement learning	60.25	1548.6
WLAN with no learning	61.072	1682.1
WLAN with learning	60.1	1624.71
WLAN with multiple reward reinforcement learning	58.9	1532.9

Table 1 shows the simulation results. The first row is the simulation with random delay 0.2~0.7 (sec.) and no learning performed. The second row is the simulation with reinforcement learning. The third row is the simulation with the proposed multiple reward reinforcement learning. The performance is improved by 27% reduction of the average squared error sum of *e* and 62% reduction of the average squared error sum of *?*. Fig. 7 and Fig. 8 show one example of comparison among three cases.



a) Random delay with no learning
b) Random delay with learning
c) Random delay with multiple reward reinforcement learning
Fig. 7 Simulation Result.



a) Random delay with no learning
b) Random delay with learning
c) Random delay with multiple reward reinforcement learning
Fig. 8 Simulation Result.

In real LAN environment, the value in the fourth row is derived without learning. With reinforcement learning, the performance is improved as the value in the fifth row. The proposed multiple reward reinforcement learning produces the value in the sixth row. In WLAN environment, the values in the seventh, eighth and ninth rows show the improvement in the performance.

5. CONCLUDING REMARKS

In this paper, the multiple reward reinforcement learning scheme is proposed as a solution to the control problem of a mobile robot in home network environment. Multiple reward fuzzy Q-learning is proposed for the multiple reward reinforcement learning.

Some simulation results are given to show the effectiveness of the proposed scheme, which is performed in real home network environment. The experiment with a real mobile robot and convergence issue of the reinforcement learning remain for the future research.

.

REFERENCES

[1] Byoung-Ju Lee, Hyun-Gu Lee, Joo-Ho Lee and Gwi-Tae Park "New architecture for mobile robots in home network environment using Jini," Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation, 2001, pp.471–pp.476, vol.1, 2001.

[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning An Introduction*, MIT Press, 1998.

[3] H. R. Beom and H. S. Cho, "A Sensor-Based Navigation for a Mobile Robot Using Fuzzy Logic and Reinforcement Learning," IEEE Transactions on Systems, Man, and Cybernetics 25(3): pp.464-pp.477, 1995.

[4] M. Sakawa, *Fuzzy Sets and Interactive Multiobjective Optimization*, New York, Plenum Press , 1993.

[5] L. Jouffe, "Fuzzy Inference System Learning by Reinforcement Methods," *IEEE Transaction on Systems, Man, and Cybernetics*, Part C, vol. 28, no. 3, pp. 338 –355, 1998.

[6] Z. Bien and W. Yu, "Extracting core information from inconsistent fuzzy control rules," Fuzzy Sets and System, vol. 71, no. 1, pp. 95-111, April. 1995.

[7] Chin-Teng Lin and I-Fang Chung, "A reinforcement neuro-fuzzy combiner for multiobjective control," *IEEE Transactions on Systems, Man and Cybernetics*, Part B, vol. 29, no. 6, pp. 726 –744, 1999.

[8] S. G. Tzafestas and G. G. Rigatos, "Fuzzy Reinforcement Learning Control for Compliance Tasks of Robotic Manipulator," *IEEE Transactions on Systems, Man and Cybernetics*, Part B, vol. 32, no. 1, pp. 107 –113, 2002.

[9] M. Wargui, K. Hentabli, M. Tadjine, and A. Rachid, "Effect of network induced delay on the stability of an autonomous mobile robot," 23rd International Conference on IECON 97, vol.3, pp. 1187 -1191, 1997.