# Multi-Robot Concurrent Learning of Fuzzy Rules for Cooperation

Zheng Liu
*Electrical & Computer Engineering*
*National University of Singapore*
*10 Kent Ridge Crescent*
*Singapore 119260*
*zhengliu@nus.edu.sg*

Marcelo H. Ang Jr.
*Mechanical Engineering*
*National University of Singapore*
*10 Kent Ridge Crescent*
*Singapore 119260*
*mpeangh@nus.edu.sg*

Winston Khoon Guan Seah
*Institute for Infocomm Research*
*Agency for Science Technology and Research*
*21 Heng Mui Keng Terrace*
*Singapore 119613*
*winston@i2r.a-star.edu.sg*

***Abstract*— For multi-robot systems, how to achieve cooperation is one of the key research issues. In this paper, a reinforcement learning approach based on fuzzy logic is proposed for multi-robot concurrent learning of cooperative behaviors. In contrast to traditional reinforcement learning that assumes discrete and finite state/action space, our fuzzy reinforcement learning controller learns based on fuzzy states and fuzzy actions to find the optimal fuzzy rules. This learning controller can directly retrieve states and then generate corresponding actions, both from continuous and infinite spaces. In addition, to address the problems in multi-robot concurrent learning, a distributed learning control algorithm is proposed to coordinate concurrent learning processes without the need for explicit intercommunications among robots. The distributed fuzzy reinforcement learning controller and the learning control algorithm are applied to multi-robot tracking of multiple moving targets. Simulation results demonstrate the efficacy of our approach.**

***Index Terms*— Reinforcement learning; fuzzy logic; multi-robot cooperation; behavior-based control.**

## I. Introduction

For the research on multi-robot systems, one essential problem is how to achieve cooperation among the robots for accomplishing a common mission [1], especially in a decentralized (distributed) manner. Normally, the desired cooperation is in task level [2], in which the common mission is decomposed into sub-tasks, and robots choose different tasks (roles) according to the state. However, to achieve mission decomposition, task allocation, and conflict avoidance, the designer needs to predict all possible scenarios and preset corresponding actions for each robot to react accordingly and differently. Such development work is undesirable and sometimes extremely difficult. Therefore, in both robotics and artificial intelligence research, machine learning methodologies are studied to enable robots to learn how to cooperate without the need for human hardcoding.

In current research, reinforcement learning is extensively studied for multi-robot concurrent learning of cooperative behaviors. This is mainly because that reinforcement learning can be applied to the behavior based control [3] methodology for generating task level cooperation. In addition, compared with other learning algorithms, reinforcement learning is model free, not strictly supervised, optimal subject to user defined criteria, and practical [4]. However, to apply reinforcement learning to behavior based control, the designer needs to define discrete and finite high level states and actions, e.g., “target is near”, “track the target”. But for most real applications, it is hard to give appropriate and accurate definition to the high level states and actions. Furthermore, even through the states and actions can be discretized and defined, the behaviors are still discrete and finite. At one time, the robot can only perform one action representing one behavior. This contradicts the human reasoning that the optimal solution to accomplish a task might be the concurrent execution of several elementary behaviors. For example, the optimal solution to track a target might be a mixture of the basic behaviors “approach detected targets”, “search other targets” and “avoid obstacles”. Besides, the switching among discrete behaviors usually results in unsmooth control, which is undesirable in most cases.

In addition to the discrete/finite state and action space problem, traditional single agent/robot reinforcement learning may not be valid in multi-robot domain. Some basic assumptions in the single robot domain, e.g., Markov decision process and stationary environment, are not valid in multi-robot domain due to the interaction among concurrent learning robots. To address this problem, the concurrent learning process needs to be carefully controlled and coordinated.

In this paper, we propose fuzzy reinforcement learning to address the limitation of discrete/finite states and actions in traditional reinforcement learning. In our approach, the discrete/finite states and actions are fuzzified as continuous/infinite fuzzy states and actions, and then the reinforcement learning controller learns based on these fuzzy states and actions to find the optimal fuzzy rules. In addition, motivated and inspired by human behaviors, we derive methods for coordinating concurrent learning processes in a distributed manner. The learning controller and the learning control algorithm are applied to multi-robot concurrent tracking of multiple moving targets.

The remaining parts of this paper are organized as follows. Section 2 introduces the background and related work. Section 3 presents the basic idea and the concept of our approach. Then, the implementation of our fuzzy reinforcement learning controller for multi-robot tracking of multiple moving targets is introduced in Section 4. The simulation results and discussion are presented in Section 5. Finally, Section 6 concludes this paper and introduces our future work.

## II. RELATED WORK

### A. Reinforcement Learning in Continuous Space

As mentioned previously, traditional reinforcement learning assumes discrete and finite state and action space. But for real applications, the input and output space are usually continuous and infinite. To address this problem, the most popular solution is discretization [5]. However, if the discretization is too coarse, some states may be hidden therefore the optimal control policy can not be found; if the discretization is too fine, the states cannot be generalized and the huge state/action space will badly affect the learning speed. Some methods are proposed to enable reinforcement learning in continuous space without the need for discretization. Function approximation approach [6] and HEDGER [7] can apply a generalizing function approximator to estimate the state-action value instead of using discrete lookup table. References [8][9] propose reinforcement learning to derive optimal feedback control law for linear/nonlinear systems. However, these approaches usually assume the environment model is known, and have heavy computational burden if the training data set is large.

Another class of solutions is to integrate reinforcement learning with Fuzzy Inference Systems (FIS). The idea is to let the reinforcement learning module learn/tune the fuzzy rules for the FIS, therefore the FIS can retrieve continuous and infinite states and then perform corresponding actions. In [10], Jouffe proposes a dynamic programming algorithm that is applied in a four layer FIS scheme for online tuning the conclusion part of the FIS. In [11], Yan et al introduce a reinforcement learning algorithm for learning the fuzzy rules of a Takagi-Sugeno type FIS. In [12], reinforcement learning methods are applied to maintain the correctness, consistency and completeness of the fuzzy rules. These deliberatively designed approaches can tune the fuzzy inference systems to achieve satisfying performance; however, the control architecture and learning algorithm are usually complex and the applications are mostly for the low level control regarding simple task and mission e.g., approaching target with obstacle avoidance.

In this paper, the proposed learning controller is also based on the integration of reinforcement learning and fuzzy inference system; however, this fuzzy learning controller is different in the definition of fuzzy states and actions. Based on simple and “fuzzier” states and actions, this fuzzy learning controller can effectively find the optimal fuzzy rules thus achieve desired cooperation among robots. The details are to be introduced in next section.

### B. Multi-Robot Concurrent Learning

Reinforcement learning and most other machine learning algorithms assume the learning process is Markovian and the learning environment is stationary [13]. These two assumptions both require the full/sufficient observation of the environment. However, limited by sensor ability, robots cannot have a complete and accurate view of the environment. Furthermore, if all robots learn concurrently, the learning process of each robot will interfere with the others. Then, during multi-robot concurrent learning, in the view of an individual robot, the process and environment are neither Markovian nor stationary; therefore the learning may result in local maxima or the undesired cyclic switching of control policies. This is usually termed as the convergence or stability problems in multi-robot (agent) learning.

One class of solutions to address this problem is to estimate the influence of other robots, thus make the process semi-Markovian and pseudo-stationary for an individual learning robot [14]. Another class of solutions is to coordinate or schedule the distributed learning processes to reduce the interference. References [15][16] propose the global scheduling method that limits the number of learning robots to reduce the mutual interference. Reference [17] proposes a distributed learning control algorithm that can enable multiple robots learning concurrently. However, the coordination and scheduling of learning processes need to be deliberatively designed and usually require explicit intercommunications among the robots.

In this paper, a distributed learning coordination algorithm is proposed. The basic idea is to let the robot stop learning regarding one state when it has learned enough in this state. This algorithm is suitable for generating cooperative behaviors; the details are to be introduced in next section.

## III. OUR APPROACH

As introduced previously, the objectives of our research are as follows:

- Enable the robots to learn by retrieving low level input and generating low level output.
- Let the robots learn cooperative control policy by distributed (local) learning processes.
- Coordinate the distributed learning processes to solve the problems of concurrent learning. Try to avoid the undesired learning results (local maxima or cyclic switching of control policies) by minimal intercommunications among the robots.

To achieve above objectives, a learning controller integrating reinforcement learning and fuzzy logic is proposed as depicted in figure 1. This controller (inside the dotted rectangle) includes five main modules: 1) Fuzzifier; 2) Defuzzifier; 3) Fuzzy Inference System (FIS); 4) Reward Generator; and 5) Reinforcement Learning Module. In the following of this section, we will introduce our fuzzy reinforcement learning controller from three aspects: 1) Fuzzy Inference System (FIS); 2) Reinforcement Learning of Fuzzy Rules; and 3) Coordination of Concurrent Learning Processes.

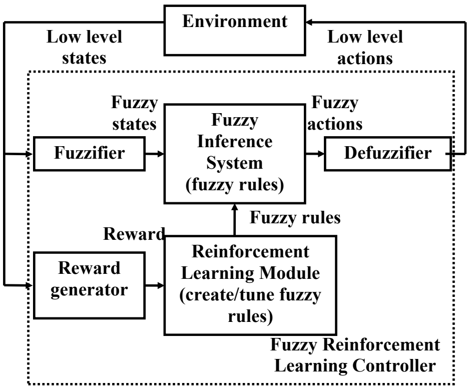
Environment
Low level states
Low level actions
Fuzzy states
Fuzzy actions
Fuzzifier
Fuzzy Inference System (fuzzy rules)
Defuzzifier
Fuzzy rules
Reward
Reward generator
Reinforcement Learning Module (create/tune fuzzy rules)
Fuzzy Reinforcement Learning Controller

Fig. 1. Fuzzy Reinforcement Learning Controller

### A. Fuzzy Inference System

Fuzzy inference system works based on the concept of fuzzy set. Fuzzy set is a kind of set that does not have a crisp, clearly defined boundary as the classical set. This property is very useful for handling continuous and infinite real sensor readings and thus generating corresponding actuator commands. In our approach, we fuzzify the low level environment information to fuzzy states and fuzzify the low level actuator commands to fuzzy actions. The fuzzification we proposed has following properties:

- Regarding the environment input, one kind of information (or sensor reading) is represented by one fuzzy state. This fuzzy state (action) covers the whole data range.
- Regarding the actuator output, one kind of commands is represented by one fuzzy action. This fuzzy action is actually a group of commands that has common properties.
- The design of fuzzy states and actions is based on *a prior* knowledge of the mission, robot and environment.

In contrast to common fuzzy inference systems, the definition of the fuzzy states in our approach is "fuzzier". This is due to following reasons:

- In our approach, one fuzzy state represents the whole data range of one kind of environment information. It does not describe the "fuzzy value" of the environment variable (e.g., sensor input); instead it provides membership values for the entire range of the environment variable. For example, in Faria and Remero's approach [20], there are four fuzzy states "nearest", "near", "far", and "farthest" used to describe the levels of distance to the target (Figure 2-a); however, in our approach, there is only one fuzzy state "target is found" to represent the information of distance to the target (Figure 2-b). This fuzzy state covers the whole data range of distance, and its membership degree (value) is given based on human knowledge: if the target is near, this target may have more influence to the robot, therefore the level of "target is found" should be high.
- In our fuzzy reference system, at one time, it is possible that several fuzzy states are activated concurrently. This can enable the robot to learn several basic behaviors for each fuzzy state concurrently
- By this fuzzification methodology, less fuzzy states will be defined and thus used in learning. This may avoid the curse of dimension for reinforcement learning [13].

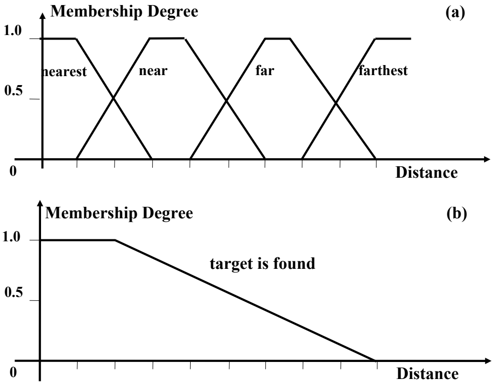

Fig. 2. Definition of Fuzzy State

In the proposed fuzzy inference systems, the definition of fuzzy actions is also based on human knowledge. One fuzzy action is defined to represent one kind of behavior. The membership function demonstrates the relationship between behavior level (strength) and the environment status. For example, regarding fuzzy action "track target", the membership function may be defined as shown in Figure 3. When the target is near (distance is small), the robot does not need to put much emphasis on tracking it; therefore the level (strength) of behavior "track target", i.e., the membership value, is low.

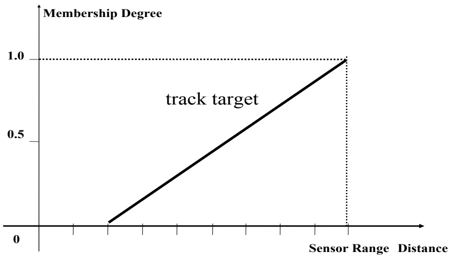

Fig. 3. Definition of Fuzzy Action

As the fuzzy states and actions are defined, the fuzzy inference system can make decision based on fuzzy rules. In our approach, the format of the fuzzy rules is defined as (1), in which $r_{s,a}$, $s$, and $a$ means fuzzy rule, fuzzy state, and fuzzy action respectively.

$$\text{Rule } r_{s,a}\text{: IF } s \text{ THEN } a \tag{1}$$

For the fuzzy inference system, the total number of fuzzy rules to be tested equals $m$ times $n$; $m$ is the number of fuzzy states, $n$ is the number of fuzzy actions. The aim of the reinforcement learning module is to find the optimal fuzzy rules regarding each input fuzzy state; the results are $m$ fuzzy rules. For example, if the fuzzy inference system has four fuzzy states, and three possible fuzzy actions; then the aim of reinforcement learning is to find four optimal fuzzy rules each for one fuzzy state. As shown in Figure 4, regarding "fuzzy state 1", the robot learns the fuzzy rule "IF *fuzzy state 1* THEN *fuzzy action 1*".

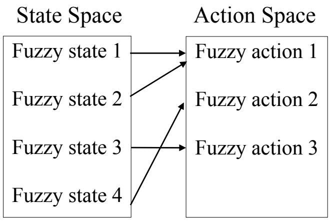

Fig. 4. Learning of Fuzzy Rules

During learning, the robot will select fuzzy actions with regard to current fuzzy states according to fuzzy rules being learned (or already learned). Since the fuzzy states may happen concurrently, several corresponding fuzzy actions may thus be activated together. In our approach, the output of the fuzzy inference system (*FIS output*) is the summation of the fuzzy actions activated by fuzzy states as shown in (2). In this equation, *mvfs* and *mvfa* means the membership value of activated fuzzy state and corresponding action respectively.

$$\textit{FIS output} = \sum_{\text{activated state i}} (mvfs \cdot mvfa)_i \qquad (2)$$

### B. Reinforcement Learning of Fuzzy Rules

As introduced previously, the fuzzy inference system makes decision based on fuzzy rules. While in most fuzzy inference systems the fuzzy rules are deliberatively designed by human, our learning controller aims to learn the optimal fuzzy rules by reinforcement learning. In our approach, Regarding each fuzzy rule $r_{s,a}$, we define $V(r_{s,a})$ to indicate the result of applying it. Then, to find the optimal rules becomes to find the $V$ values of all fuzzy rules. The meaning of $V(r_{s,a})$ is similar to $Q(s,a)$ as in traditional reinforcement learning [4]. However, the Q-function used to update $Q(s,a)$ cannot be applied for updating the $V(r_{s,a})$. This is because the states and actions in Q-function (3) should be discrete and exclusive. In (3), $s$, $a$, $r$, $\alpha$, $\gamma$, $s'$, and $a'$ means state, action, reward, learning rate, discount rate, next state, and next action respectively.

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma max_{a'} Q(s',a') - Q(s,a)) \quad (3)$$

To solve this problem, we first define the triggers for updating $V(r_{s,a})$. This is because in our approach, it is hard to find "sharp" fuzzy state transition time point. For example, when the target is 0.7 meters away, the fuzzy state "target found" is 0.1; when the target is 0.2 meters away, the fuzzy state "target found" is 0.9. In both cases the fuzzy state "target found" is activated (non-zero). Therefore, we set following two triggers to update/reselect fuzzy rules:

- The fuzzy state has a zero/non-zero change; or
- The fuzzy state has been activated (non-zero) for a long period of time ($N$ simulation steps).

When one of above two conditions is activated, the $V(r_{s,a})$ will be updated by following equation (4). In this equation, $\alpha$ is the learning rate; $reward$ is the feedback regarding the progress of the mission. Comparing (3) and (4), we may find that the new state, $s'$ in (3), does not appear in our algorithm (4). This is due to the fact that the "next" fuzzy state is usually the same as the previous one, but different in membership value, e.g., the fuzzy state "target found = 0.1" changes to fuzzy state "target found = 0.9". Therefore, in (4) it is not necessary to add the item referring to "next state".

$$V(r_{s,a}) \leftarrow (1-\alpha)V(r_{s,a}) + \alpha(reward) \qquad (4)$$

In the fuzzy inference system, at one time, more than one fuzzy state may be activated; thus the output of the fuzzy inference system is the combination of the fuzzy actions corresponding to these fuzzy states (2). In this case, the robot will learn more than one fuzzy rules concurrently. The update of each fuzzy rule's value is also according to (4).

For the robot, after updating $V(r_{s,a})$, the learning controller needs to reselect fuzzy rules (fuzzy actions) to perform and test. To both explore and exploit the possible fuzzy rules (actions), the controller adds an exploration factor to each fuzzy rules' $V$ value, and then the fuzzy rules having highest resultant values will be chosen. It should be noted that this random factor is only used for fuzzy rules selection; it will not affect the fuzzy rules' real $V$ values.

### C. Coordination of Concurrent Learning Processes

Besides the discrete and finite state/action space limitation of traditional reinforcement learning, another critical research issue is to coordinate concurrent learning processes to avoid the undesired learning results as the local maxima or cyclic switching of control policies. For this purpose, a solution inspired by natural human behaviors is proposed. Assuming two humans are approaching in the corridor and they want to avoid the collision, what will they do? If they are both trying, they may "struggle" several rounds to find the best. So, in real life, usually one of them (say $A$) will fix his policy first, e.g., keeping left, then the other one (say $B$) can choose another side. In this encounter case, the optimal cooperation is that the two people choose opposite sides. Whatever $A$ chooses initially, finally $B$ can learn to choose another side, and the resultant control policy is optimal. Many real world applications have the same property: even if the learning process of one robot stops very early, the resultant control policy of the whole team can still be optimal because other learning robots can eventually find appropriate control policy to respond to the former one.

Our distributed learning coordination algorithm is proposed based on above considerations. For a robot, if regarding one fuzzy state $i$, the best fuzzy action $j$'s value is much larger than others, i.e., above a given threshold over the average, the robot will fix the control fuzzy rule "IF fuzzy state $i$, THEN fuzzy action $j$" for this fuzzy state $i$. But the robot will still learn the optimal fuzzy rules for other fuzzy states unless all the fuzzy states have got a fixed control fuzzy rule. For example, after a period of learning, the robot is in fuzzy states $a$ and $b$, and the best fuzzy rule $r$ regarding state $b$ is already much better than other fuzzy rules for state $b$, then the robot will always choose rule $r$ for state $b$, but still select and test fuzzy rules regarding state $a$. Till all the fuzzy states have fixed fuzzy rules, the learning of the robot stops.

By this method, a robot will fix its control policy (partially or entirely) when it feels that it has learned enough; and the future improvement (learning) is thus left to other robots. This learning control algorithm is entirely distributed and does not need explicit communications among robots. It should be noted that for this learning coordination algorithm, the threshold is critical for the robot to decide when to stop learning. If this threshold is too high, the robot will in fact never stop learning, or if this threshold is too low, the robot may be easy to give up learning. The influence of the threshold is tested and discussed in Section 5.

## IV. APPLICATION TO MULTI-ROBOT TRACKING OF MULTIPLE MOVING TARGETS

### *A. Museum Problem: Multi-Robot Tracking of Multiple Moving Targets*

In robotics research, multi-robot tracking of multiple moving targets is also referred to as the "museum problem" or "art gallery problem". The assumptions and descriptions of the problem are as follows:

- The environment is a large bounded plain area including some mobile targets and robots.
- The targets are moving in the environment. The number, distribution, and the motion pattern of targets are unknown.
- Each robot has a 360 degree view within a certain range. When an object is inside this range, the robot can differentiate this object as obstacle, target, or robot, and detect the distance and orientation toward it. The summation of the sensible area of all robots is far less then the size of the environment.
- The robot needs to track (move together with) the targets to maintain observation. For one target, only one robot is needed for observation.
- The robots do not know the size and map of the environment, and cannot localize themselves in the environment. Besides, there are no explicit inter-robot communications available, e.g., wireless communication.
- The objective is to maximize the number of targets being simultaneously observed (detected within the robot's sensing range)

In current research for the museum problem, Artificial Potential Field (APF) based control is mostly used. The idea is to map the targets (or robots and obstacles) as attractive (or repulsive) force sources, and then let the robot move under the vector sum of these forces. However, purely summing these forces (pure APF) may not achieve desired cooperation. For example, if two robots detect each other and a same target, both of them will be repulsed by the neighbor robot and attracted by this target. In most cases, none of them will give up this "shared" target. Obviously, this is not the optimal cooperation because one of robots can leave and search for other targets.

To solve this problem, two heuristics of pure APF are proposed. Both of them add a weight factor $w_{R_i}^{T_j}$ to the attractive force as shown in (5). In this equation, $\vec{F}_{R_i}$ means the summation of the attractive and repulsive forces for $R_i$; $\vec{T}_{R_i,T_j}$ means the attractive force from robot $R_i$ to target $T_j$; $w_{R_i}^{T_j}$ means the weight of $\vec{T}_{R_i,T_j}$; $\vec{R}_{R_i,R_j}$ means the repulsive force from neighbor robot $R_l$ to $R_i$; $dt$ means the set of detected targets; $dr$ means the set of the detected neighbor robots.

$$\vec{F}_{R_i} = \sum_{j\in dt} w_{R_i}^{T_j} \cdot \vec{T}_{R_i,T_j} + \sum_{l\in dr} \vec{R}_{R_i,R_j} \qquad (5)$$

The all-adjust heuristic [18] lets the robot decrease the weight when the robot find another robot(s) tracking this target; while the selective-adjust heuristic [19] only lets the robot decrease the weight when it is not the nearest robot to this target. Both heuristics are proved effective; however, to make them work, the designer needs to carefully select appropriate parameter, e.g., weight decrease ratio, for each robot, especially when the scenario is complex and the robot team is heterogeneous.

Examining (5), we can find the weight of the attractive force affects the behavior of the robot regarding the target. If the weight is low, the robot will leave the target; if the weight is high, the robot will track the target. Changing the value of the weight means changing the preference to the two basic behaviors "track target" and "leave target". Therefore, for our fuzzy reinforcement learning controller, the two fuzzy actions "track target" and "leave target" can be represented by this weight value.

### *B. Applying Our Learning Controller to Museum Problem*

As introduced previously, we plan to implement our distributed learning controller for multi-robot tracking of multiple moving targets. For this learning controller, we define two fuzzy states "target found" and "target tracked by others", and two fuzzy actions "track target", "leave target", as shown in Figures 5 and 6. For the fuzzy states, the membership degree (value) indicates the degree of the state regarding the distance to the target (or the target to other robots). For fuzzy action "track target", the membership degree (value) is the weight of the attractive forces to the target, i.e., $w_{R_i}^{T_j}$ in (5). The higher membership degree means stronger preference to approach to the targets. For fuzzy action "leave target", the membership degree is the inverse of the weight of the attractive forces. The higher membership degree means stronger preference to leave the targets.

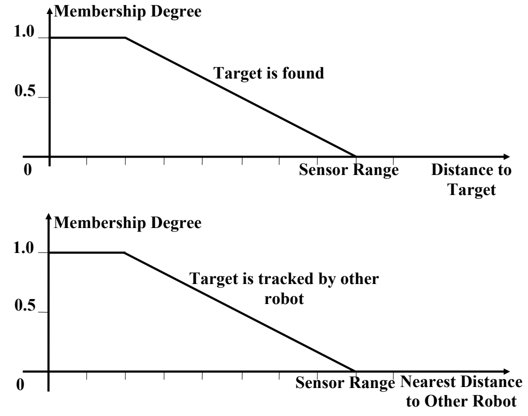

Fig. 5. Fuzzy States: "Target is Found", "Target is Tracked"

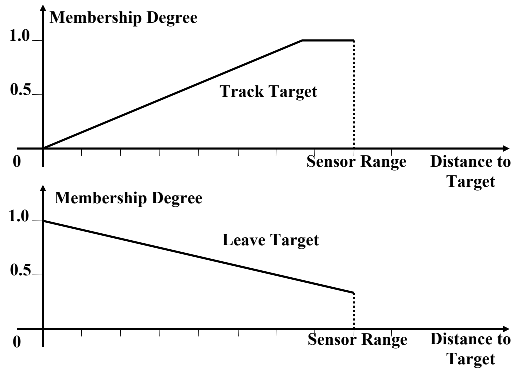

Fig. 6. Fuzzy Actions: "Track Target", "Leave Target"

The design of the fuzzy states and fuzzy actions are based on human experience. When the target is near to the robot, the degree of fuzzy state "target is found" is large; when the target is neighbored by another robot, if they are near, the degree of fuzzy state "target is tracked" is large. Also, when the target is far, the degree of fuzzy action "track target" is large because if the tracking action is weak under this condition, the robot will lose the target; when the target is quite near, the degree of fuzzy action "leave target" is large because if the leaving action is weak under this condition, the robot may hit the target.

For reinforcement learning, one important issue is the generation of rewards because reward represents the objective of the human and thus can directly affect the learning results. Since task level cooperation is desired, following behaviors should be encouraged: 1) track target; 2) leave the target being tracked by other robots. For this purpose, we define three kinds of rewards:

- $Reward\_TT/Reward\_LT$: track/lose target reward (positive/negative) - if tracks/loses targets.
- $Reward\_WT$: waste time reward (negative) - if tracks a target being tracked by others.
- $Reward\_PT$: pass target reward (positive) - if passes the target to other robot(s) to track.

For the distributed learning controller in each individual robot, these rewards are generated by its local sensing. For example, if both robot $A$ and robot $B$ are tracking the same target, in the view point of robot $A$, unless $B$ is detected, it cannot generate the $Reward\_WT$.

Other important implementation issues, including the updating and reselection of fuzzy rules and the coordination of concurrent learning processes, are according to the methods introduced in Section 3C.

## V. SIMULATION AND DISCUSSION

### A. Simulation Methodology

The aim of our research is to let the mobile robots learn how to cooperatively work without the need for human hardcoding. This research aim includes two main aspects:

- The learning approach can generate cooperative behaviors.
- The performance of the learning system should be comparable to other approaches that have been deliberatively hardcoded and tuned.

To justify the efficacy of our approach, we simulate four control modes as follows:

- Pure Artificial Potential Field (APF) based control.
- All-adjust heuristics to pure APF.
- Selective-adjust heuristics to pure APF.
- Robot learning controller: different threshold for stopping learning from small to infinite (never stop learning).

### B. Simulation Settings

The parameters and settings of the environment are as follows:

- The simulations are run on Webots, a 2D differential-wheel robot simulator.
- Museum: 4m x 4m to 6m x 6m square plain area with no obstacles inside. The simulated robot and target are smaller then 0.1m in diameter. The sensor range is 0.8 meter.
- For each control mode, run about 15 episodes to get the average. Each episode is 15000 simulation step long. One simulation step is about 0.1s long in real time.

For the all-adjust heuristics of pure potential field based control, if two or more robots find the same target, and they find each other, they will all decrease the weight of the attractive force to target. In the simulation, we test two all weight decrease ratio ($AWDR$): 0.80 and 0.95. For the selective-adjust heuristics of pure potential field based control, if two or more robots find the same target, and they find each other, the further robot(s) will decrease the weight of the attractive force to the target. In the simulation, we test two selective weight decrease ratio ($SWDR$): 0.5 and 0.1.

- The initial values of all fuzzy rules are 10.
- $Reward\_TT/Reward\_LT$ = 0.8/- 0.8; $Reward\_WT$ = -1.5; $Reward\_PT$ = 2.0.
- Learning stop threshold is set 1.0, 1.5, 3.0, 10.0, and infinite (never stop learning). During learning, if regarding one state, one action's $V$ value is above the threshold over average of all actions' $V$ values for this state, the robot will stop learning for this state.
- If the fuzzy state is unchanged (non-zero) for $N$ = 50 simulation steps, the robot updates and reselects fuzzy rules.
- When selecting fuzzy rule (action), a number uniformly distributed in [-1, 1] is added to the real fuzzy rule's value as the exploration factor. This exploration factor is only for fuzzy rule selection, not for updating the fuzzy rule's value.

### C. Simulation Results and Discussion

*1) Performance Comparison:*
The performance of different controllers is evaluated by the average tracked target: in average, how many targets are tracked simultaneously. The higher this number, the better the performance is. Figure 7 compares average tracked target of learning controller and other human designed controller. In this figure, the performance of the learning controller is represented by the one with stop learning threshold of "1.5". It should be noted that the results of learning controller is the performance after all robots stop learning. This is because that during the initial part of learning, the robots' behaviors are far from optimal. Only after all the robots stop learning, the behaviors of the robots are fixed and thus the performance is stable. Another problem worth noting is that the learning performance presented here are the average of all the episodes, including the failed cases that do not learn optimal fuzzy rules. (The success rate of learning optimal rules is to be discussed later)

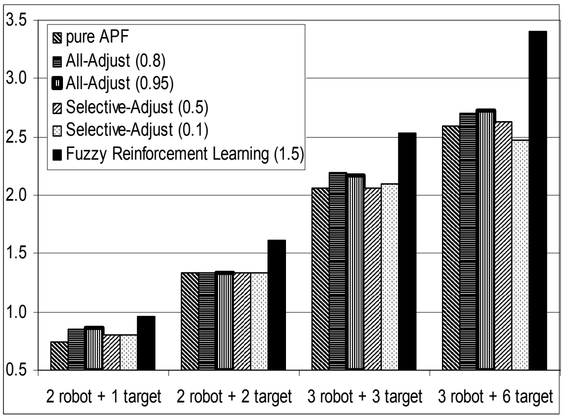

Fig. 7. Average Tracked Target

Observing Figure 7, we may draw a conclusion that pure potential field based control and two heuristics perform worse than the learning controller. Furthermore, while we need to decide important parameters for the human designed controllers, the learning controller can enable the robots to automatically generate desired cooperation. The resultant performance is as good as, or even better than, the deliberatively designed controllers.

*2) Learning Analysis:*
In our approach, the fuzzy reinforcement learning controller aims to find the optimal fuzzy rules which appropriately link fuzzy states "target is found" and "target is tracked" to fuzzy actions "track target" and "leave target". The learning results should be two optimal fuzzy rules each for one fuzzy state. The simulation results show that in most cases the robot learns the following two fuzzy rules: 1) IF *target is found*, THEN *track target*; and 2) IF *target is tracked*, THEN *leave target*. This result is accordant to human intuition on how to cooperative track targets.

As introduced in Section 3C, different learning stop threshold value may lead to different learning results. Now we compare the controllers with different stop learning thresholds by examining the successful rate of learning the optimal results. Figure 8 presents the frequency that the robots learn the two optimal fuzzy rules. The higher the frequency, the better the learning performance is. For learning controllers with threshold 1.0, 1.5, 3.0, and 10.0, the learning usually ends before the end of the simulation episode. However, if the threshold is infinity, the robots will never stop learning (no coordination of learning at all). For this case, the final $V(r_{s,a})$ in the last step is used to indicate the learned fuzzy rules.

Observing Figure 8, we find that for different robot group size, the optimal threshold value is different. Small threshold value suits small robot group, while large threshold suits large group. This may be explain by the fact the large robot group will have more interference; therefore require large stop learning threshold to help kick out the sub-optimal rules. However, if the threshold is too big, the robot will "hesitate" to fix the good fuzzy rules; therefore concurrent learning robots may have more "struggles". For our simulation scenario, the optimal threshold value is "1.5".

Figure 9 shows the average number of targets tracked for the entire length of one episode (15000 simulation steps) for two representative learning controllers. This gives an indication of the "overall" performance. The controller with stop learning threshold "1.5" works better than the controller with infinite threshold (never stop learning). This may be because the former controller can effectively avoid the undesired learning results as local maxima, or the cyclic switching among control policies.

## VI. CONCLUSION AND FUTURE WORK

Multi-robot concurrent learning on how to cooperatively work is one of the ultimate goals of robotics and artificial intelligence research. In this paper, we propose a distributed fuzzy reinforcement learning controller that applies fuzzy

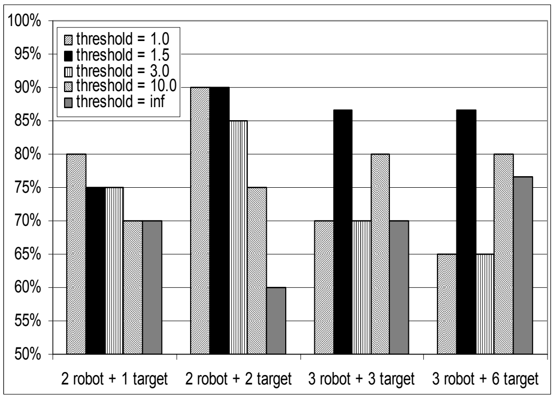

Fig. 8. Success Rate of Learning Optimal Fuzzy Rules

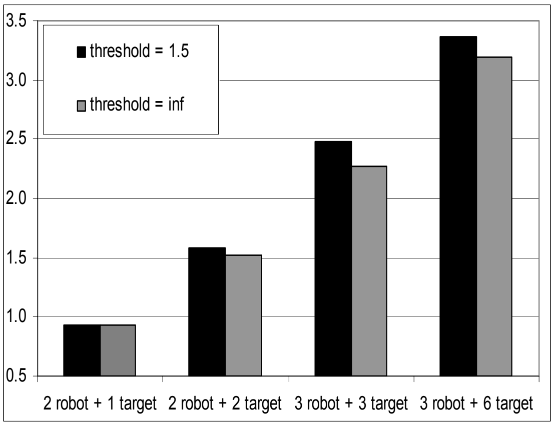

Fig. 9. Comparison of Different Learning Stop Threshold

logic and reinforcement learning. This controller can enable the robot to generate cooperative behaviors based on fuzzy states and actions. In addition, we propose a natural inspired distributed learning control algorithm to coordinate the concurrent learning processes. This algorithm can help avoid the generation of local sub-optimal control policy or the cyclic switching of control policy without the need for explicit intercommunications among the robots. Our approach is tested in multi-robot tracking of multiple moving targets; simulation shows that the learning controller can achieve the performance as good as, or even better than, the controllers deliberatively designed.

However, in our learning controller, the fuzzy states and actions are defined by the designer and are specific to the task and application. If other tasks are selected, e.g., cooperative table carrying, we have to design other specific fuzzy states and actions accordingly. Obviously, it would be much better if the fuzzification of the normal state and actions can be generic and effective for all kinds of control problem. This is an important research issue to be studied. In addition, due to the interference among the concurrent learning robots, the distributed learning controller sometimes generates unsatisfying results even though we have proposed a distributed learning coordination algorithm. How to perfectly coordinate concurrent learning processes by minimal intercommunications is another critical research topic for our future research.

## REFERENCES

[1] Y. U. Cao, A. S. Fukunaga, A. B. Kahng, and F. Meng, *Cooperative mobile robotics: antecedents and directions*, in proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol:1, pp:226-234, 1995.

[2] P. Tangamchit, J. M. Dolan, and P. K. Khosla, *The necessity of average rewards in cooperative multirobot learning*, in proceedings of IEEE International Conference on Robotics and Automation, 2002.

[3] M. J. Mataric, *Reinforcement learning in the multi-robot domain*, Autonomous Robots 4(1), pp 73-83. 1997.

[4] R. S. Sutton, and A. G. Barto, *Reinforcement learning: an introduction*, MIT Press, Cambridge, MA. 1998.

[5] H. T. Chu, and B. R. Hong, *Cooperative behavior acquisition in multi robots environment by reinforcement learning based on action selection level*, in proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol:2, pp:1397-1402. 2000.

[6] J. A. Boyan, and A. W. Moore, *Generalization in reinforcement learning: safely approximating the value function*, in Advances in Neural Information Processing Systems 7. 1995.

[7] W. D. Smart, and L. P. Kaelbling, *Practical reinforcement learning in continuous spaces*, in proceedings of the 7th International Conference on Machine Learning. 2000.

[8] K. Doya, *Temporal difference learning in continuous time and space*, in Advance in Neural Information Processing Systmes 8, pp:1073-1079. MIT Press, Cambridge, MA. 1996.

[9] Stephan H.G. ten Hagen, *Continuous state space Q-learning for control of nonlinear systems*, PhD thesis, Computer Science Institute, University of Amsterdam. 2001.

[10] L. Jouffe, *Fuzzy inference system learning by reinforcement methods*, in IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol. 28, No. 3. 1998.

[11] X. W. Yan, Z. D. Deng, and Z. Q. Sun, *Genetic Takagi-Sugeno fuzzy reinforcement learning*, in Proceedings of the IEEE International Sympposium on Intelligent Control. 2001.

[12] C. Ye, N. H. C. Yung, D. Wang, *A fuzzy controller with supervised learning assisted reinforcement learning algorithm for obstacle avoidance*, in IEEE Transcations on Systems, Man, and Cybernetics, Part B: Cybernetics, Vol. 33, No.1. 2003.

[13] L. P. Kaelbling, M. L. Littman, and A. W. Moore, *Reinforcement learning: a survey*, in Artificial Intelligence Research, Vol: 4, pp237-285. 1996.

[14] K. I. Kawakami, K. Ohkura, and K. Ueda, *Adaptive role development in a homogeneous connected robot group*, in proceedings of IEEE International Conference on Systems, Man, and Cybernetics, Vol:3, pp:254-256. 1999.

[15] E. Uchibe, M. Nakamura, and M. Asada, *Co-evolution for cooperative behavior acquisition in a multiple mobile robot environment*, in proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Victoria, B. C., Canada. 1998.

[16] M. Asada, E. Uchibe, and K. Hosoda, *Cooperative behavior acquisition for mobile robots in dynamically changing real worlds via vision-based reinforcement learning and development*, Artificial Intelligence, Vol.110, pp.275-292, 1999.

[17] S. Ikenoue, M. Asada, and K. Hosoda, *Cooperative behavior acquisition by asynchronous policy renewal that enables simultaneous learning in multiagent environment*, in proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.2728-2734, 2002.

[18] L. E. Parker, *Distributed algorithm for multi-robot observation of multiple moving targets*, Autonomous Robots, vol. 12(3), May, 2002.

[19] Z. Liu, M. H. Ang, and W. K. G. Seah, *Searching and tracking for multi-robot observation of moving targets*, in proceedings of the 8th Conference on Intelligent Autonomous Systems, March, 2004.

[20] G. Faria, and R. A. F. Romero, *Incorporating fuzzy logic to reinforcement learning*, in Proceedings of the IEEE International. Conference on Fuzzy Systems. 2000.