MDP based Active Localization for multiple robots

Jyotika Bahuguna, B. Ravindran and K. Madhava Krishna

***Abstract*— In environments with identical features, the global localization of a robot, might result in multiple hypotheses of its location. If the situation is extrapolated to multiple robots, it results in multiple hypotheses for multiple robots. The localization is facilitated if the robots are actively guided towards locations where it can use other robots as well as obstacles to localize itself. This paper aims at presenting a learning technique for the above process of active localization of multiple robots by co-operation. An MDP framework is used for learning the task, over a semi-decentralized team of robots hereby maintaining a bounded complexity as opposed to various multi-agent learning techniques, which scale exponentially with the increase in the number of robots.**

I. INTRODUCTION

To navigate reliably in an enviornment and to perform tasks like map-building, mobile robots must know where they are. Therefore, estimating the position of a robot based on sensing and motion is one of the fundamental problems of mobile robotics. The various flavours of the localization problem are characterized by the type of knowledge that is available intially and at run-time. Local localization is the correction of the dead reckoning errors in the robot's position due to odometry. In this case, the initial pose of the robot is known. Global self-localization addresses the problem of localization with no apriori information i.e. a robot is not told its initial pose but instead has to determine it from scratch. A hypothesis is a probable location (x,y) in the map that a robot might be in. In environments which possess relatively few features that enable a robot to unambiguously determine its location, global localization algorithms can result in multiple hypotheses about the location of the robot. This is inevitable as the local environment seen by a robot repeats at other parts of the map. The global localization method is called passive localization if the robots motion is not aimed at facilitating localization. It is termed as 'active localization' if the robots plans every movement such that localization is facilitated. The complexity of the problem can be used to its advantage if there are multiple robots trying to localize. By using active localization, the robots can use each other's relative pose as well as obstacles to localize.

Jyotika Bahuguna is a student at Robotics Research Lab,Department of Computer Science and Engineering, IIIT Hyderabad, Andhra Pradesh, India jyotika@research.iiit.ac.in

B. Ravindran is with Department of Computer Science and Engineering, IIT Madras , Chennai, India ravi@cse.iitm.ac.in

K. Madhava Krishna is with the Robotics Research Lab, Department of Computer Science and Engineering, IIIT Hyderabad, Andhra Pradesh, India mkrishna@iiit.ac.in

In general, the work on active localization has been limited when compared to passive localization. The pioneering work has been from [1] and [2]. In [1], a method of active localization based on maximum information gain was presented. Dudek et al. [2] presented a method for minimum distance traversal for localization that works in polygonal environments without holes that they have shown to be NP-Hard. In [3], the enviornment's ambiguity in the terms of observation and robot's ambiguity in pose is modelled as a POMDP and an optimal strategy to move to the target location is devised. In [4], this method is extended by estimating actions which allow the robot to improve its position estimation. All the above methods tackle the problem for a single robot. [5] devises a Bayesian approach to enable the robots to calculate the relative poses of every other robot. Since it only estimates the relative pose, it is unable to perform global localization. [6] uses Partially Observable Stochastic Games to solve a multi-agent co-operative problem in a fully distributive fashion. Since it is decentralized the cross product of the action and the observation space is considered, which exponentially increases the complexity with the increase in the number of robots. In [7], the problems of large joint action and state spaces is mitigated by using a multi-agent MDP that reduces the search space for an optimal action set by considering just the upper and lower bounds of the single agent MDPs. But this method requires isolated training of each agent seperately before undertaking the task as a team. The method presented here enables all the agents to get trained simultaneously, leading to faster convergence. The only optimal known solution to the above problem has been provided by [8], [9]. An extension with a learning framework to the same is intended in this paper. The improvement over [8], [9] is its semi-distributed nature as opposed to completely centralized nature of the former. Since the agents are loosely coupled, they take advantage of robot-robot detection if applicable, and continue trying to localize with the help of enviornment, if not. This also brings in a certain level of autonomy with respect of the number of agents. The paper is organized as follows: Sec II explains the problem of Active Localization in detail, Sec III describes the framework used, Sec IV explains the basic algorithm and the MDP formulation, Sec V talks about the results in simulation and Sec VI concludes by stating the scope of further work.

II. Active Localization

In order to localize, the robot uses its sensor readings of the enviornment. In highly symmetrical enviornments (Figure 1), like office spaces with similar cubicles, or corridors with symmetric rooms or evenly spaced pillars,the sensor readings for a position are replicated at various positions in the map. This results in multiple hypotheses for a robot's pose. With every hypothesis is associated a value that defines the probability of robot being in that hypothesis. The probabilities are calculated using Markov localization method [13]. Markov localization addresses the problem of state estimation from sensor data. It maintains a probability distribution over the space of all possible hypothesis of a robot. The probabilistic representation allows it to weigh these hypotheses in a mathematically sound way. To actively localize, a robot should move towards the locations which promise the highest information gains. When a group of robots are involved in the scenario, robot-robot detections can be used along with the map feature detection to speed up the localization. The robot is considered localized when it converges to a single hypothesis.

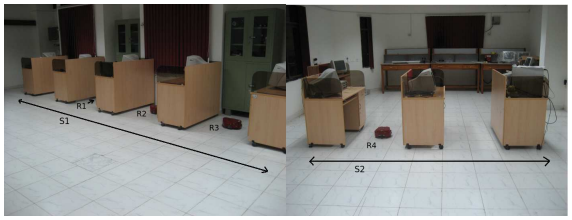

Fig. 1. Symmetrical enviornments: Cubicles in an office

III. Markovian Decision Processes

A finite MDP is defined as a 4-tuple (S,A,P,R) , in which S is the finite set of states, A is the finite set of actions, P is the transition probability function and R is the reward function. The dynamics of the enviornment is defined by the transition probability function

$$P : S \times A \times S \rightarrow [0, 1] \tag{1}$$

The reward function is defined as a real value bounded function

$$R : S \times A \times S \rightarrow R \tag{2}$$

$R^a_{ss'}$ is the reward of taking action *a* in state *s* and landing in state *s'*. A value function $V^{\pi}(s)$ is to be estimated, which determines the value of state *s* under a policy π, i.e the expected return when starting in *s* and following the policy π thereafter. The solution of the value function is given by the Bellman optimality equation. For all $s \in S$:

$$V(s) = max_{a \in A}(\Sigma_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V(s')]) \tag{3}$$

The Bellman equations can be solved using iterative algorithms, where we start with an initial guess V_0 and iterate for every state *s*:

$$V_{k+1}(s) = max_{a \in A}(\Sigma_{s' \in S} P^a_{ss'}[R^a_{ss'} + \gamma V_k(s')]) \tag{4}$$

As *k* goes to infinity, V_k converges to the optimal policy, V^*.

A. Framework

In this work, we model the problem of active localization as a MDP. A location hypothesis of a robot is a probable location (x,y) in the map, a robot can be in. Consider a workspace populated by robots, each of them having such multiple location hypotheses. A probabilty distribution over all location hypotheses of a robot constitutes a belief state for that robot. And belief states for all robots constitute the state space for the MDP. The problem is to move these robots to locations so that each of the robot localizes in an optimal fashion. Frontiers [10] are boundaries separating the seen and the unseen. The frontiers are considered as good places to move to, since they are easier to compute and provide a sufficient set of places to move, to converge to a single hypothesis. The orientation is considered to be known through a compass installed on the robot. The robots can detect each other with help of an IR transreciever which provides the angle and the distance from the robot detected. This experimental setup has been tested and can be safely assumed to work quite robustly. The belief state calculation involves the method of Markov localization. It starts by assuming the belief of a robot as an uniform distribution over all positions in the map. The query to the sensors yield a set of readings which are tried to match with the readings at different locations in the map. This belief can be multimodal if there are no unique features in the map. If its unimodal, the robot is considered to be localized. Clustering is performed over all positions in the map, which have a belief value greater than some pre-defined threshold. The number of clusters define the number of hypotheses for a robot and the probability associated with them forms the belief value. These set of hypotheses for a robot is defined as a belief state of the robot. The action space consists of actions that lead to the different frontiers visible from a position. The reward is calculated intuitively on the basis of the information gain on transition from one belief state to another. This aims at reducing entropy. Considering the Markov property, the rewards can be zero but never negative. It is because of Markov property of the enviornment, that one-step dynamics are sufficient to predict the next state and next reward, given the current state and action.

As it can be deduced, the MDP framework divides the map into a hierarchical structure that reduces the size of the state space. This is because it is now considered in terms of the belief states which is a subset of all the possible

positions in the map.The mapping of these logical clusters, i.e belief states, to actual positions in the map is done by dividing the map into a grid structure. The granularity of the grid structure can be assumed to be some appropriate value for e.g, the smallest cubicle in the map. The granularity should assure that no two clusters fall in the same grid. The learning algorithm would enable the robots to learn how to move, if they find themselves in a certain grid in the map, such that it encounters either other robots or unique features in the map thereby facilitating localization. Intuitively the robots learn the features in the map as well as the locations where it might find other robots. To avoid enumeration of all the possible belief states, a Cerebellar Model Articulation Controller (CMAC) is used to implement the value function *V(b)*. A CMAC [14] uses tile coding and the weighted sum of the tiles yield the V(b) for a belief state. It in turn acts as a function approximator for the belief states which dont get visited during the training. Since the cardinality of a belief state represents the dimension of the state space, the number of tilings to be used become enormous. But since only few of the values in the belief state are non-zero at every instant (since sensor readings are clustered into hypotheses) , number of tilings dont extend more than 10, say. Hence 10 tilings can be safely used. The tiles in the tiling are assumed to be uniformly spaced since no prior information about the nature of input space is known.

IV. Formulation

The algorithm is semi-distributed in nature. Most of the multi-agent learning algorithms suffer by high computational complexity because of the joint state and action spaces for the robots which increase exponentially with the increase in the number of robots. The robots in this framework are loosely coupled to each other. They update the same value function, but the information about other robots in the enviornment is only limited to robot-robot detection. Thereby, the robots dont need to know the positions of all other robots in the map as well as the actions they take. In this way, the method provides an effective solution of learning from the experiences of the other robots, but avoids the large state and action spaces at the same time. This feature of the algorithm also captures the robot-robot interaction effectively hence driving the robots to move to frontiers where the probability of finding other robots is maximum.

A. Basic Algorithm

The pseudocode for the algorithm is provided below. The notations used are:
V(a) $\rightarrow$ The goodness value for the action 'a'
b(s_n) $\rightarrow$ Is the vector of the belief state 'b', nth element 's_n'
R_{map} $\rightarrow$ The reward due to map information.
b' $\rightarrow$ The next belief state b-prime.
V(b) $\rightarrow$ The goodness value of the belief state b.
$R_{map+r-r_detection}$ $\rightarrow$ The reward due to map information and information gain due to robot-robot detection.

1) Make observations, Compute the belief state 'b'
2) Calculate the projected gain that each action would yield

$$V(a1) \leftarrow \Sigma_{n\in N}[b(s_n) * (R_{map} + (\lambda * V(b')))] \quad (5)$$

3) Choose the action 'a' which has the highest V value.
4) Execute 'a'
5) If finds another robot
   - Update belief state.
   - Calculate reward $R_{map+(r-r)detection}$.
6) Update V(b)

$$V(b) \leftarrow \alpha(R_{map+r-r_detection}+\gamma*V(b')-V(b))+V(b) \quad (6)$$

7) Repeat until the robot is localized

A set of robots are set up in the enviornment. The above algorithm is executed on all the robots independently except the step 6. The robots start with sensing their surroundings and calculating the belief state, as explained above. The robots then perform a one-step look ahead by calculating the action which would yield the maximum information gain. Since its a look-ahead, the only information gain available is due to the map features, hence the reward R_{map}. The robots execute the action which seems to be most beneficial in step (3). Now the robot looks for any possibilities of robot-robot detections. If any robot is found in the vicinity, the angle and distance from the robot is supplied. With this information, the robot tests all its hypotheses and checks which of them are now feasible. (For, e.g, one of the hypothesis might suggest that the robot is detected beyond a wall, discard it). Update the new belief state and calculate the reward. Since this reward is due to both the map features (from step 2) and presence of other robots, its termed as $R_{map+(r-r)detection}$. This reward is jointly fed back by all robots to the commonly shared CMAC. The above set of steps are repeated, until all robots get localized. The training runs consists of various initial configurations for varied number of robots.
To avoid getting stuck in the local minimas at step (3), the exploration strategy of ϵ- greedy with a slight modification is adopted. One could use strategies like softmax action selection but ϵ- greedy is used for computational convenience. The value of ϵ is high in initial training iterations but is decreased as the training proceeds. This is done to avoid the unlearning. Since the higher frequency of low rewards can over weigh a low frequency of high reward, it is ensured that random actions are not exceuted too frequently.

The state value function V(s) is estimated instead of state-action value function ,i.e Q(s,a). This has twofold benefits. One, it reduces the computational complexity since the Q(s,a) would require maintaining multiple CMAC's, one for every kind of action. And second, by estimating V(s), one can assimilate the experience and capture the interaction of the multiple robots effectively. The method tries to estimate a goodness value of every belief state and guiding the robots towards the belief states with higher goodness value.
The immediate feedback of a (state, action) pair is incorporated with the help of one step look-ahead in step (2) of the algorithm.

V. Tests and Results

The experiment was carried out with 20 robots on Map B, Fig 7. The belief state evolution as explained in section IV-A, is shown for one robot.

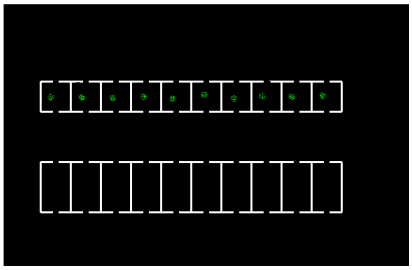

Fig. 2. Green dots represent the probability of all hypotheses of the robot

In Figure 2, the robot takes an initial scan and calculates its belief state. Each cluster of green dots represent one hypothesis, and hence one element in the belief state.

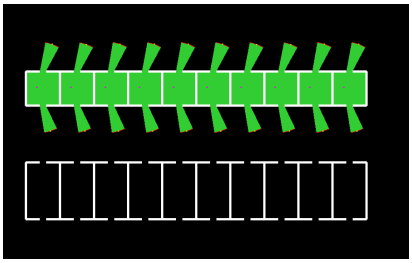

Fig. 3. The sensor scan is taken from all hyptheses locations

In Figure 3, the robot considers a scan from all its possible hypotheses.

In Figure 4, after the scan, the robot moves south because Q value for the action south was highest. This is evidently due to the memory of finding other robots during training.

In Figure 5, after the robot moves south, it detects an another robot and is able to localize. The belief state becomes unimodal.

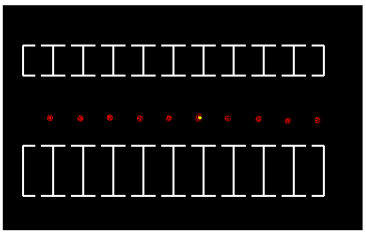

Fig. 4. The red dots indicate the probability before robot-robot detection

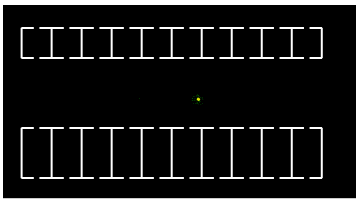

Fig. 5. The green dots indicate the probability after robot-robot detection

There were two kind of tests performed.

1) The method was tested on different kind of maps to verify the proof of concept. These maps were designed in order to test the algorithm in practical situations. The learning pattern for different kind of maps are shown below:

a) Map A: Figure 6

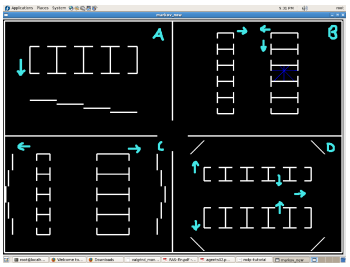

Fig. 6. Map A: Learning pattern shown by arrows

As seen in Figure 6, if the robots are present in the section A, intuitively the best action to be taken is south since it leads to the unqiue features of the map that would help to localize in the absence of the other robots. On the other hand, if all robots are guided to south, the probabilty of robot-robot detections is also high. The algorithm learns to choose the action south if the robot is in any of the cubicles in the section A.

In section B of the map that has no unique features the algorithm learns to move the robots in a direction (shown in the figure) so that there are more robot-robot detections. In cases, where there were not enough robots in the opposite row of cubicles, the robots learn to move towards the sides.

In section C, the robots clearly learn to move towards the obstacles. It is apparent due to the fact that there is a unique feature outside every cubicle which surpasses the gain due to robot-robot detections.

In section D, the learning pattern is the most intuitive. The robots first move towards the obstacles, i.e north for the upper row of cubicles and south for the lower row. This is because of the huge gain yielded by the action towards the obstacles when the robot was in any one of the corner cubicles and got instantly localized. If the robot still remains unlocalized, it moves towards the space between two cubicles in the hope of finding other robots. If it still remains unlocalized, it moves sideways to get localized.

The above pattern was verified by a testcase consisting of 7 robots, one in every row of cubicles. The corresponding pattern for every section stated above was observed.

b) Map B: Figure 7

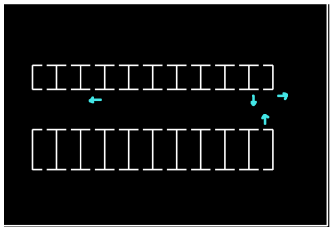

Fig. 7. Map. With almost no unique features in map to aid localization

As it can be seen, the above map is challenging since there are almost no unique map features to aid localization. Hence it serves as an ideal platform to test the algorithm. The robots learn to move in directions shown by the arrows. The space between the cubicles is where there is highest probability of having robot-robot detections. When there are no other robots, the frontiers on the sides are selected which lead the robots to the end of the row where they get localized with the help of the wall.

The similar pattern has been observed in Map A. Since the learning pattern is consistent over different kind of maps, it can be safely assumed that the method would work on all general maps.

TABLE I

TREND OF LOCALIZATION TIME WITH RESEPECT TO THE NUMBER OF ROBOTS FOR MAP B

No. of robots	No. of Iterations
6	12
7	8
11	9
15	5
20	1

2) The second test performed was to show the flexibility of the framework to infer the results for different number of robots without re-training. The framework, if trained for an appropriate number of robots can be used for a varied range of number of robots without re-training each time. To prove this, the *MAP B* was trained with test cases for 10 robots (i.e 5 in each row of cubicle) and 6 robots (3 in each row of cubicle). The objective is to train the framework to look out for map features when there are no robots around to aid localization and move to locations of higher robot-robot detections, in general. The former is achieved by training it extensively with lesser number (6) of robots and the later by training with large number of robots (10).
The testing is done with a range of number of robots and number of iterations required to localize were recorded.

The trend is plotted in the graph 8. The graph shows that on an average the number of iterations required to localize all the robots decreases with the increase in the number of robots. This shows that the framework learns to take advantage of the presence of other robots. On the other hand, it does not get handicapped in the absence of other robots and uses the map features available to localize.

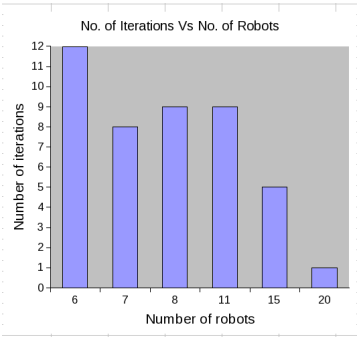

Fig. 8. Trend showing number of iterations VS Number of robots

This as a matter of course, also proves that the framework need not be re-trained everytime the number of robots to be localized changes. The training sample space should consist of test cases with appropriate number of robots after which the framework can handle the entire range of robots in between.

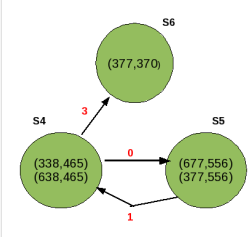

Fig. 9. Transition of states, blue numbers above the arrow, represents the action taken

An interesting observation was made regarding the rewards that were used as feedbacks for the CMAC. The reward is calculated as the information gain which is basically the difference in entropies. For actions which yield no increase in the information gain, the difference in entropies is a small negative number instead of zero. Assuming the Markov property, which states that there can no negative rewards, the information gain should be set to zero and fed back to the CMAC. But if this quantity is fed back in its pristine form, it acts as negative feedback that prevents the robots from getting stuck in a prolonged and recursive loop of actions not leading to any progress.

One such situation has been shown in Figure 9. The robot starts from state S4, executes an action '0' and transits to state S5. The robot then executes an action '1' and ends up in state S4. It remains stuck in this loop for 4 iterations, until action '3' is selected ,robot transists to S6 and localizes. The corresponding Q(a) values have been plotted in Graph 10.

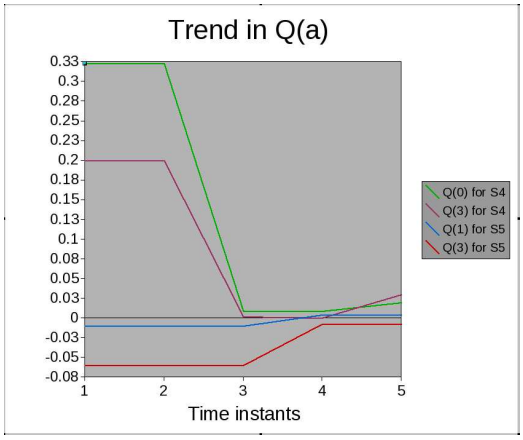

Fig. 10. Variation in the Q(a) behavior

With reference to the graph in Figure 10, at time instant 1, the robot is in state S4 and Q(0) is greater than Q(3). Hence the robot selects the action '0' and transists to state S5. At time instant 2, the robot is in state S5 and executes the action '1' to reach back to state S4. At time instant 3, the difference in the Q values of action '0' and '3' decreases. But Q(0) is still greater than Q(3). The action '0' is selected again. At time instant '5' ,Q(3) finally surpasses Q(0). Action '3' is executed, the robot reaches state S6 where it localizes. This happens because of the negative feedback that gets added during the transitions between S4 and S5.

VI. CONCLUSIONS

A. Conclusions

The test results discussed in previous sections show that the method is an effective way to perform Active localization for multiple robots without dealing with huge joint state and action spaces. Since the algorithm aims at learning the unique features in the map, whether be it obstacles or other robots, it performs a kind of feature learning and hence is scalable to any number of robots or lack of them. That is to say, that the system doesnt need to be retrained for every change in the number of robots. Once it is trained for an appropriate number of robots required to capture the basic features of the map, the system would work for large range of number of robots without re-training.

REFERENCES

[1] W. Burgard D. Fox and S. Thrun. Active Markov Localization for Mobile Robots.*Robotics and Autonomous Systems, (1998)*

[2] K. Romanik G. Dudek and S. Whitesides. Localizing a Robot with Minimum Travel.*SIAM J. Computing 27, no. 2, 583.604, (1998).*

[3] M.L. Littman L. P. Kaelbing and A.R. Cassandra. Planning and Acting in Partially Observable Stochastic Domains. *Technical report, Brown University, 1995.*

[4] A.Cassandra L. Kaelbling and J. Kurien. Acting Under Uncertainty: Discrete Bayesian Models for Mobile Robot Navigation.*In Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems,1996.*

[5] M J Mataric A Howard and G S Sukhatme. Putting the I in the Team: Ego-Centric Approach to Multi-Robotic Localization.*Proceedings of IEEE International Conference on Robotics and Automation, 2003.*

[6] Emery-Montemerlo, R. and Gordon, G. and Schneider, J. and Thrun, S.: Approximate Solutions For Partially Observable Stochastic Games With Common Payoffs. *Proceeding of Autonomous Agents and Multi-Agent Systems,2004*

[7] Mohammad Ghavamzadeh and Sridhar Mahadevan: A Multiagent Reinforcement Learning Algorithm by Dynamically Merging Markov Decision Processes *First International Conference on Autonomous Agents and Multiagent Systems (AAMAS) , Bologna, Italy, 2002*

[8] Shivudu Bhuvanagiri, K. Madhava Krishna, Supreeth Achar: Coordination in ambiguity: coordinated active localization for multiple robots. *AAMAS (Demos) 2008: 1707-1708*

[9] Shivudu Bhuvanagiri, K. Madhava Krishna: Active global localization for multiple robots by disambiguating multiple hypotheses. *IROS 2008*

[10] Brian Yamauchi. A Frontier-Based Approach for Autonomous Exploration. *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, CA, July 1997.*

[11] M. Mataric, Reinforcement learning in multi-robot domain. *Autonomous Robots ,4(1);73-78, 1997*

[12] R.S Sutton and A.G Barto. Reinforcement learning , An introduction *The MIT Press, Cambridge, MA 1998*

[13] Dieter Fox. Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Navigation *Doctoral thesis, University of Bonn, Germany Press, Dec 1998*

[14] J.S Albus. A new approach to manipulator control: The cerebellar model articulation contoller (CMAC) *Journal of Dynamic Systems, Measurement and Control 1975*