Generation of Search Behavior by a Modification of Q-MDP Value Method

Ryuichi Ueda

Abstract We modify Q-MDP value method and observe the behaviors of a robot with the modified method in an environment, where state information of the robot is essentially indefinite. In Q-MDP value method, an action in every time step is chosen based on a calculation of expectation values with a probability distribution, which is the output of a probabilistic state estimator. The modified method uses a weighting function with the probability distribution in the calculation so as to give precedence to the states near the goal of the task. We applied our method to a simple robot navigation problem in an incomplete sensor environment. As a result, the method makes the robot take a kind of searching behavior without explicit implementation.

Keywords Q-MDP value method · Particle filters · Belief states · Partially observable Markov decision process

1 Introduction

Uncertainty of state recognition is an inevitable problem for autonomous agents in the real world. On the other hand, agents can sometimes choose appropriate actions even where its state is quite uncertain. For example, a person can reach his/her bedroom even if his/her house is in total darkness.

In robotics, online methods for solving POMDP (partial observable Markov decision process) problems have been proposed in [1–3]. These methods give an appropriate action of a robot by real-time search based on the partial knowledge of the state of the robot and its surroundings, and a mathematical model of the task that the robot is trying to complete. These online methods can be applied to POMDP problems with high dimensionality.

R. Ueda (✉)
Advanced Institute of Industrial Technology, 1-10-40 Higashi Ohi,
Shinagawa-ku, Tokyo, Japan
e-mail: ueda-ryuichi@aiit.ac.jp
URL: http://aiit.ac.jp/english/

E. Menegatti et al. (eds.), *Intelligent Autonomous Systems 13*,
Advances in Intelligent Systems and Computing 302,
DOI 10.1007/978-3-319-08338-4_1

When a problem is to be defined in a low-dimensional state space, we can use abundant computing resources, or we can choose some offline methods. In an offline method, some parts of a POMDP problem is solved previously. Robots can utilize the solution to choose their actions. Coastal navigation by Roy et al. is a straightforward example [4]. This method is classified into the group of the AMDP (augmented Markov decision process) methods in [5]. In this study, a robot used range sensors for measurement of the distance from the walls in an indoor environment and the outputs were used for self-localization. The behavior of the robot was planned in a space spanned beforehand by four variables: the position on an X–Y coordinate system, orientation, and an additional scalar value. The additional one represents the level of the uncertainty of self-localization. Since the planner, which is based on the dynamic programming [6], considers the easiness of self-localization, a robot with the planned policy moves along the walls. With the behavior, the robot can keep an appropriate distance from the walls for self-localization.

To use an AMDP method, we must prepare a state transition model that considers when or where the robot can obtain sensor information. It is not always known previously.

Previous knowledge about obtainment of information is not required in Q_{MDP} value method [7], which is written as *Q-MDP value method* in this paper. This method was proposed by Littman et al. two decades ago. In this method, a decision-making policy is obtained without consideration of the information uncertainty. Instead, uncertainty is considered in online calculations. The robot with this method chooses an action that is stochastically effective to the task of the robot when a probability distribution that represents the self-localization uncertainty is given. When the state is clear, the robot can choose the identical action that is chosen by the decision-making policy that does not consider the uncertainty.

We have applied Q-MDP value method to some tasks of robot soccer [8, 9]. Since the robot has poor computing resources, offline methods are suitable. Since the transition of state uncertainty is then unpredictable in bruising soccer games, we have chosen Q-MDP value method rather than any AMDP method. On the other hand, we never expected the Q-MDP value method to generate such a skillful behavior as obtained in Roy's research. That is because uncertainty of states is never considered in the planning phase.

In this paper, we show that another kind of skillful behavior can be generated by a small modification of the Q-MDP value method. The robot with the modification method overcomes a lack of information by its search behavior, which is not explicitly planned in offline calculations.

This paper is composed of the following sections. The modification of Q-MDP value method is proposed in Sect. 2. In Sect. 3, we define a problem and some conditions for simulation in which we observe and evaluate behaviors of a robot generated by the modification method. We discuss results of the simulation in Sect. 4 and conclude this paper in Sect. 5.

2 Modification of Q-MDP Value Method

2.1 Problem Definition

Q-MDP value method and our modification can be applied to a POMDP (partial observable Markov decision process) problem that is defined as follows. In general, for POMDPs not only the state of controlled objects, but also the optimal policies are unknown in advance. However, we only handle the POMDP problems in which the optimal policies are known when the state is known.

2.1.1 States

The state of a robot and its environment can be explained by n state variables: $x_1, x_2, \ldots, x_n$. Each state $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ belongs to a state space $\mathcal{X}$. There are *final states* in the state space. The set of final states is denoted by $\mathcal{X}_\mathrm{f} \subset \mathcal{X}$. When the state becomes one of the final states, the task defined below is regarded as finished.

2.1.2 Actions and State Transitions

The robot chooses one action from the action set: $\mathcal{A} = \{a_1, a_2, \ldots, a_m\}$ at each time step. When the robot executes an action, a state transition occurs. State transitions are stochastic and their model is represented by the probability density $p^a_{\boldsymbol{x}\boldsymbol{x}'}$ where $\boldsymbol{x} \in \mathcal{X}$ is the state where the action $a \in \mathcal{A}$ is chosen and $\boldsymbol{x}' \in \mathcal{X}$ is one of the posterior states after the action a.

2.1.3 Evaluation of Behavior Based on Optimal Control Problem

The evaluation of the above set of an action and states: $\boldsymbol{x}, a, \boldsymbol{x}'$ is given by a real number. In this paper, this number is named as a penalty $r^a_{\boldsymbol{x}\boldsymbol{x}'} \in \Re$.

We consider a function $V : \mathcal{X} \to \Re$. $V(\boldsymbol{x})$ is the sum of penalties from $\boldsymbol{x}$ to a final state when a decision-making policy $\pi : \mathcal{X} \to \mathcal{A}$ is given. The purpose of decision making is to minimizes $V(\boldsymbol{x})$ in each state $\boldsymbol{x}$.

2.1.4 Belief State

In a POMDP problem, $\boldsymbol{x}$ is unknown. Instead, the state information is given as a probability density function $b : \mathcal{X} \to \Re$. The probability that the actual state is in a set $Y \subset \mathcal{X}$ is given as

$$B(Y) = \int_Y b(\boldsymbol{x}) d\boldsymbol{x}. \tag{1}$$

b is called *belief state* of the robot [10].

2.1.5 Recognition of Final States

The robot is informed whether the task is finished or not from someone or some sensors. In other words, the robot does not need to judge from the probability density function b whether the actual state is in $\mathcal{X}_\mathrm{f}$ or not.

This assumption is not arbitrary and can be applied to some tasks. In the case of robot soccer, for example, the finish of the task (scoring) is notified by the referee. In the example of the dark house mentioned in the introduction, the person may notice that he/she is in a bedroom when he/she touches a surface of a bed.

Under this assumption, the actual state is not a final state when the robot knows that task is not finished. We can therefore add the following operation of the belief state after each action:

$$b(\boldsymbol{x}) \longrightarrow 0 \qquad (\forall \boldsymbol{x} \in \mathcal{X}_\mathrm{f}) \tag{2}$$

if the task is not finished after the action.

2.2 Q-MDP Value Method

When a policy π and the value function V that is derived from π is known, Q-MDP value method chooses a stochastically optimal action based on the following function:

$$Q_\mathrm{MDP}(a) = \int_{\boldsymbol{x}} b(\boldsymbol{x}) \int_{\boldsymbol{x}'} p^a_{\boldsymbol{x}\boldsymbol{x}'} \left(V(\boldsymbol{x}') + r^a_{\boldsymbol{x}\boldsymbol{x}'} \right) d\boldsymbol{x}' d\boldsymbol{x}. \tag{3}$$

The action $a \in \mathcal{A}$ that minimizes $Q_\mathrm{MDP}(a)$ is regarded as the optimal action.

When the penalty $r^a_{\boldsymbol{x}\boldsymbol{x}'}$ in Eq. (3) is a constant, Eq. (3) can be considered as a probabilistic version of the artificial potential method [11]. To choose an action that minimizes the expected value of *energy* of the value function V after the action is the strategy of Q-MDP value method.

Q-MDP value method has the problem of local minima as is the case with almost all of artificial potential methods. The robot with Q-MDP method cannot choose an appropriate action when the expected value of energy cannot be improved after any action.

2.3 Modification of Q-MDP Value Method with an Attentional Function

We think that a subtle modification of Q-MDP method is effective when Eq. (2) can be applied to the task. Our idea of the modification is the introduction of attention to some important areas in the state space. For example,

$$Z(a) = \int_{\boldsymbol{x}\in\mathcal{X}\setminus\mathcal{X}_{\mathrm{f}}} w(\boldsymbol{x}) \int_{\boldsymbol{x}'} p^a_{\boldsymbol{x}\boldsymbol{x}'} \left(V(\boldsymbol{x}') + r^a_{\boldsymbol{x}\boldsymbol{x}'}\right) d\boldsymbol{x}' d\boldsymbol{x}, \text{ where} \tag{4}$$

$$w(\boldsymbol{x}) = \frac{b(\boldsymbol{x})}{V(\boldsymbol{x})} \tag{5}$$

can be used instead of Eq. (3). In Eq. (3), the decision is weighted by $b(\boldsymbol{x})$. That is, the larger $b(\boldsymbol{x})$ a state has, the more influential the state on the decision. In Eq. (4), on the other hand, the weight is $b(\boldsymbol{x})/V(\boldsymbol{x})$. Since the value $V(\boldsymbol{x})$ denotes the total cost from $\boldsymbol{x}$ to a final state, not only the probability density, but also the *closeness* to a final state gives a large weight to a state for decision making.

We can expect that the robot with Eq. (4) moves as it checks preferentially a part of the probability distribution of b near the final state. If the robot cannot reach in a final state, the part of the distribution can be erased with Eq. (2). This selectivity is not shown in Eq. (3). When the state is certain, moreover, the decision making with Eq. (4) is optimal as a problem of MDP (Markov decision process) as well as Q-MDP value method.

3 Generation and Evaluation of Behavior with Simulation

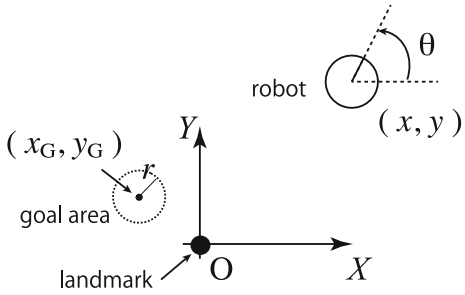
3.1 A POMDP Problem with Indefiniteness of State Recognition

We define a task and a robot environment where the state of the robot is essentially indeterminate.

3.1.1 Environment, Task, and Action

As shown in Fig. 1, There is a mobile robot and a point landmark in the environment. The point landmark is set at the origin of the X–Y coordinate system in the figure.

Fig. 1 Environment



The state of the robot is denoted by $\boldsymbol{x} = (x, y, \theta)$, where (x, y) is the position of the robot in the X–Y coordinate system and θ denotes the orientation of the robot from the X-axis.

The task given to the robot is to touch the goal point $(x_\mathrm{G}, y_\mathrm{G})$ with its body as soon as possible. When the radius of the robot is r, the state is a final state when (x, y), the center position of the robot, is in the circle whose radius and center are r and $(x_\mathrm{G}, y_\mathrm{G})$, respectively.

The robot can choose one action from the following action set $\mathcal{A} = \{\mathrm{ccw}, \mathrm{cw}, \mathrm{fw}\}$, where

- ccw: the robot rotates by $\omega + \sigma_\omega$,
- cw: the robot rotates by $-\omega + \sigma_\omega$ and,
- fw: the robot goes forward $v + \sigma_v$ in the θ direction.

σ_ω and σ_v in this list represent Gaussian noises in each motion. The robot can run through on the point landmark.

3.1.2 Recognition of the Robot

The robot recognizes its state as a probability density function b, which is defined in Eq. (1). b is referred to as a belief state.

The belief state changes when the robot moves or observes something. When the robot executes an action, the probability distribution represented by b moves and diffuses based on the amounts of displacement and noises. In regard to the observation, the robot can observe the point landmark, and whether the task is finished or not.

The robot can observe the point landmark at any state. The robot measures (ℓ, φ) by one observation. ℓ is the direction from the landmark to the robot, and φ is the relative orientation of the landmark from the robot. Gaussian noises contain in ℓ and φ. The noises on ℓ and those on φ are independent of each other. This information can be reflected to the belief state b based on Bayes' theorem.

The robot can also reflect the information whether the task is finished or not to b. This assumption is mentioned in Sect. 2.1.

In this condition, the state $\boldsymbol{x}$ is almost indefinite in a task. The robot just knows the relative pose from the landmark when it observes the landmark. The information whether the task is finished gives subtle and implicit information about the X–Y coordinate of the robot.

3.1.3 Implementation of a Particle Filter for Self-localization

In the simulation, the belief state and its operations are implemented by a particle filter [12]. The particle filter that is used by the robot has the following particle set:

$$\Xi = \{\xi^{(i)} = (\boldsymbol{x}^{(i)}, w^{(i)}) | i = 1, 2, \ldots, N\} \tag{6}$$

where $\boldsymbol{x}^{(i)}$ and $w^{(i)}$ are the state and the weight of a particle $\xi^{(i)}$ respectively.

With the particles, Eq. (1) is approximated as

$$B(X) = \sum_{i=1}^{N} w^{(i)} \delta(\boldsymbol{x}^{(i)} \in X). \tag{7}$$

When the robot obtains a set of measurement values (ℓ, φ) from the landmark observation, the weight of each particle is changed to

$$w^{(i)'} = \alpha^{-1} w^{(i)} \mathcal{N}(\ell - \ell^{(i)}, \sigma_\ell^2) \mathcal{N}(\varphi - \varphi^{(i)}, \sigma_\varphi^2) \tag{8}$$

where

$$\alpha = \sum_{i=1}^{N} w^{(i)'}, \text{ and} \tag{9}$$

$$\mathcal{N}(x, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{x^2}{2\sigma^2}\right]. \tag{10}$$

$(\ell^{(i)}, \varphi^{(i)})$ are the direction and the orientation of the landmark from $\boldsymbol{x}^{(i)}$. σ_ℓ and σ_φ are the standard deviations of the measurement of ℓ and φ respectively.

Particles frequently converge on some points where the robot does not exist. In such a case, Bayes' theorem cannot be applied to the approximated b accurately. To reset this situation, we use the sensor resetting algorithm [13]. When α in Eq. (8) is smaller than a threshold value after an observation of the landmark, preknowledge is discarded through the replacement of all particles. In the replacement, the particles are distributed in the area where the robot can obtain the observation with a high probability.

We implement Eq. (2). The recognition of whether the task is finished or not is reflected to the particles as

$$w^{(i)'} = \begin{cases} \alpha^{-1} 10^{-5} w^{(i)} & (\boldsymbol{x}^{(i)} \in \mathcal{X}_\mathrm{f}) \\ \alpha^{-1} w^{(i)} & (\text{otherwise}) \end{cases} \tag{11}$$

after every action. This means that we can erase the particles in the area of final states. 10^{-5} is an alternative value for zero.

3.2 Value Function and Penalty

In the simulation trials, we compare some methods. All of the methods use a value function that is defined by

$$V(\boldsymbol{x}) = \begin{cases} \varphi_{\mathrm{G}}/\omega + (\ell_{\mathrm{G}} - r)/v & (\ell_{\mathrm{G}} > r) \\ 0 & (\text{otherwise}) \end{cases} \tag{12}$$

for decision making. This function can be considered as a potential function in an artificial potential method. ℓ_{G} and φ_{G} denote, respectively, the relative distance and orientation of the goal from the robot. Though the form of the optimal value function of this task is more complex than that of this function, it is sufficient for comparison.

We also define that the robot is given a penalty $r^a_{\boldsymbol{x}\boldsymbol{x}'} = 1$ after every action. To minimize the sum of this penalty in the task, the robot must finish the task in as small number of steps as possible.

3.3 Parameters on the Simulation

We choose values of the parameters in the simulation as shown in Table 1. To tell the truth, this set of values is selective for obtaining the behavior that we have interest in. We should investigate the influence of parameter choices for the behavior of the robot in the future.

3.4 Methods Compared in the Simulation

We compare the following decision-making methods:

(A) the modified Q-MDP value method that uses Eq. (4),
(B) Q-MDP value method,

Table 1 Parameters for the simulation

Initial state	$x = 1000\,\mathrm{mm}$, $y = 0\,\mathrm{mm}$, $\theta = -90°$
Goal position	$x_{\mathrm{G}} = 0\,\mathrm{mm}$, $y_{\mathrm{G}} = 200\,\mathrm{mm}$
Radius of the robot	$r = 50\,\mathrm{mm}$
Displacement on an action	$w = 5°$, $v = 10\,\mathrm{mm}$
Amounts of noise on an action	$\sigma_w = 0.5°$, $\sigma_v = 1\,\mathrm{mm}$
Frequency of a landmark observation	every 5 steps
Frequency of an observation of task finished/unfinished	every step
Amounts of noise on a landmark observation	$\sigma_\ell = 0.1\,\ell$ (10 % of ℓ), $\sigma_\varphi = 10°$
Number of particles	1000
Threshold value to invoke a sensor resetting	10^{-6}
Ranges of the environment	$-4000 \leq x \leq 4000\,\mathrm{mm}$, $-4000 \leq y \leq 4000\,\mathrm{mm}$

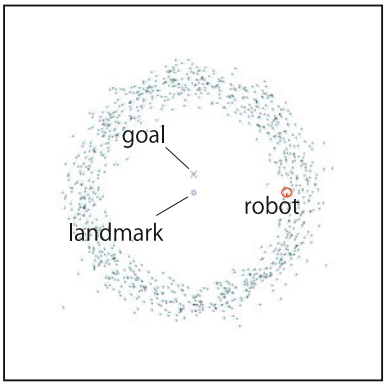


Fig. 2 Distribution of particles on trials

(C) decision making based on the weighted mean state of the particles, and
(D) decision making based on the actual state.

Since the actual state is indefinite, the method C will not work. We can expect that Q-MDP value methods (A and B) can give proper actions when the robot is far from the landmark and the goal. In Fig. 2, we show a state of the robot and particles in a trial after several observation of the landmark. The particles are distributed in a torus-shape area on the X–Y coordinate system. θ of each particle reflects the relative orientation between the landmark and the robot. When the particle distributes as shown in this figure, Q-MDP value methods chooses an action that makes the robots come close to the landmark because the expected value calculated from Eq. (12) will be reduced. However, we can also expect that decision making will be difficult when the robot is near the landmark and the goal due to the indefiniteness of the state.

4 Discussion with the Simulation Results

4.1 Results of Comparison

100 trials are examined for each method. In each trial, we count the number of steps of the robot from the start state to a goal state. When the number of steps reaches to 500(step), we regard the trial as a failed one and stop the trial. In Table 2, the success rates, the average steps in success trials, and the average distance from the robot to the goal at 500th step in failed trials are shown.

As shown in this table, the robot cannot come near the goal and the landmark with the method C, whereas the robot with A and B can be near the goal even if a trial is failed. The difference between the results of A and B is the success rate. Though the absolute position of the robot on the X–Y coordinate system is uncertain, the method A, the modified Q-MDP value method, can bring the robot to the goal in 87 trials.

Table 2 Comparison of the results

Methods	Successful trials (%)	Avg. # of steps (step)	Avg. final dist. on failed trials (mm)
(A) modified Q-MDP	87	289	260
(B) Q-MDP	17	242	223
(C) weighted average	0	–	993
(D) state known	100	121	–

We show two cases of deadlock in failed trials. (a) and (b) are obtained with the method A and B respectively. In these situations, the belief states are trapped in local minimum points. Both Eqs. (3) and (4) have local minima.

Through the trials, the distribution of particles is symmetric around the landmark when the deadlocks happen in the case of the Q-MDP value method as shown in (b). On the other hand, the distribution is not symmetric at deadlocks in the case of the modified one. It would appear that this tendency is one of the reasons for the difference in success rate between the Q-MDP value method and the modified one. The pattern of distribution shown in (b) is generated frequently in this task. The modified Q-MDP value method can choose a sequence of appropriate actions even when the distribution of particles is symmetric around the landmark. In that case, the particles near the goal have heavier weights than the others and the motion of the robot is dragged by the particles with heavy weights (Fig. 3).

4.2 Generation of Search Behavior

We show a typical behavior of the robot with the method A in Fig. 4. Though the value function defined in Eq. (12) never considers the uncertainty of state recognition, the behavior of the robot compensates for the state indefiniteness. The robot sweeps a circle around the landmark and goes into the goal that is on the circle.

To explain this behavior, we show the distributions of particles at some steps of this trial in Fig. 5. At first, the robot went to the direction of the landmark as shown in (a) and (b). Between (b) and (d), the robot got away from the landmark and went on

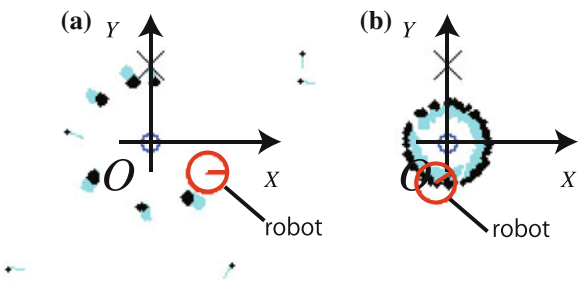


Fig. 3 Deadlocks shown in failed trials

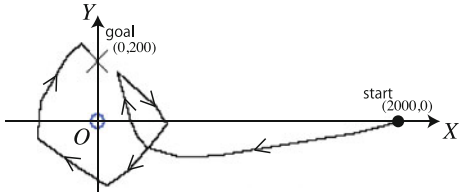


Fig. 4 A trajectory with the modified Q-MDP value method

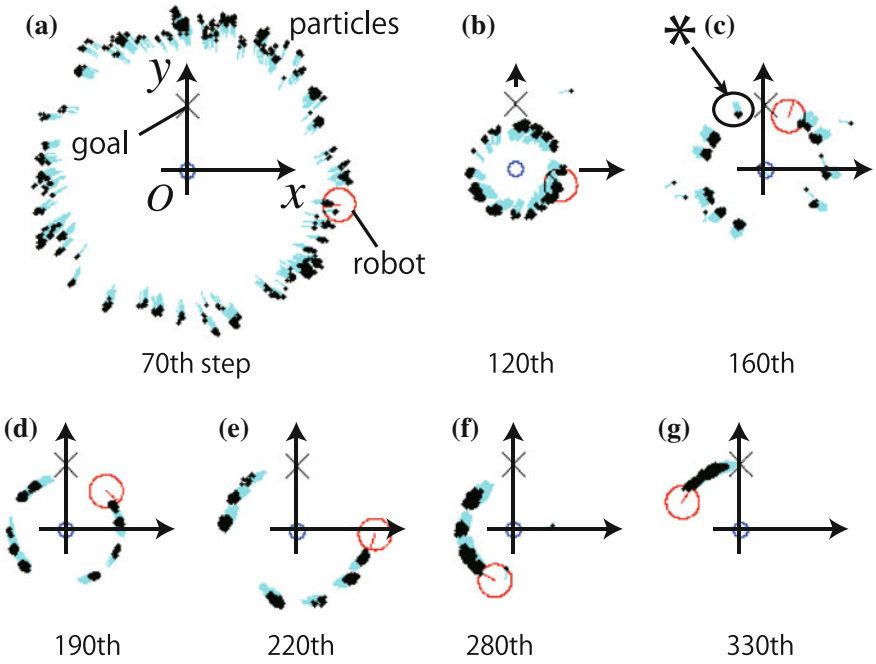


Fig. 5 Distributions of particles on the trial

the circle. Here, we pay attention to the situation in (c). Particles that formed a cluster marked with "*" was given heavy weights by the modified Q-MDP method. In this case, the action that brought these particles into the goal was chosen. Therefore, the robot started a clockwise turn at (c). We think that this choice would be different from the decision with Q-MDP value method, since Q-MDP value method does not weight particles in such a selective manner.

After (c), the modified Q-MDP method chose the sequence of actions that brought particles in the goal from one to the next, and the robot whose state was somewhere in the distribution of particles, reached the goal.

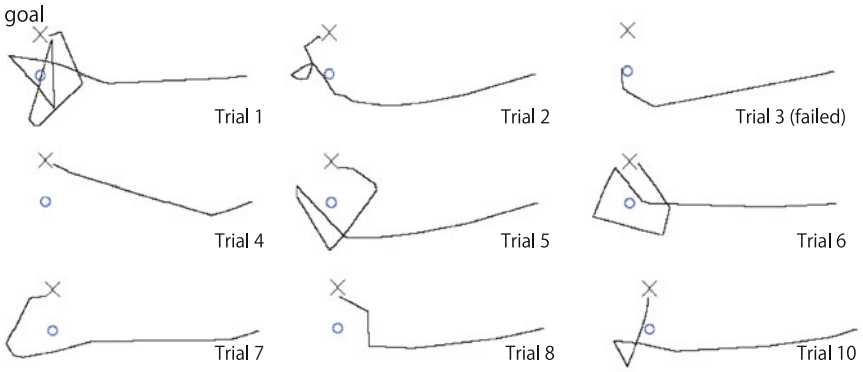


Fig. 6 Other trajectories with the modified Q-MDP value method (Fig. 4 shows the trajectory of the 9th trial)

We also show the trajectories obtained in the first ten trials with the modified Q-MDP value method in Fig. 6. Behaviors that compensate uncertainty of the state could be seen in almost trials with the modified Q-MDP value method. The difference in the success rate between Q-MDP and the modified Q-MDP value methods suggests it. However, there were failed trials in which the robot got trapped into a state that had a local minimum value $V(\boldsymbol{x})$ with both the Q-MDP value methods. The modified Q-MDP value method does not completely remove the problem of local minima though this method can reduce the frequency of the trapping.

5 Conclusion

We modified Q-MDP value method and applied it to a simulated POMDP problem in which the state of the robot was essentially indefinite. From the simulation, we can study the following:

- the robot shows a kind of search behavior that compensates for the lack of information about its state with the modified Q-MDP value method, and
- the modified Q-MDP value method improves the success rate of trials in the simulation from 17 to 87 % when compared to Q-MDP value method.

In the future works, we should apply the modified Q-MDP value method in this task with various parameters and other tasks so as to investigate the following:

- whether we can find other complex behaviors of the robot with the modified Q-MDP value method, and
- the condition that the modified Q-MDP value method is stable.

References

1. Silver, D., Veness, J.: Monte-Carlo Planning in Large POMDPs. In: NIPS. Volume 23. (2010) 2164–2172
2. Bonet, B., Geffner, H.: Solving POMDPs: RTDP-BEL vs. Point-based Algorithms. In: IJCAI. (2009) 1641–1646
3. Ong, S.C., Png, S.W., Hsu, D., Lee, W.S.: Planning under Uncertainty for Robotic Tasks with Mixed Observability. The International Journal of Robotics Research **29**(8) (2010) 1053–1068
4. Roy, N., Burgard, W., Fox, D., Thrun, S.: Coastal Navigation - Mobile Robot Navigation with Uncertainty in Dynamic Environments. In: Proc. of IEEE ICRA. (1999) 35–40
5. Thrun, S., Burgard, W., Fox, D.: Probabilistic ROBOTICS. MIT Press (2005)
6. Bellman, R.: Dynamic Programming. Princeton University Press, Princeton, NJ (1957)
7. Littman, M.L., et al.: Learning Policies for Partially Observable Environments: Scaling Up. In: Proceedings of International Conference on Machine Learning. (1995) 362–370
8. Ueda, R., Arai, T., Sakamoto, K., Jitsukawa, Y., Umeda, K., Osumi, H., Kikuchi, T., Komura, M.: Real-Time Decision Making with State-Value Function under Uncertainty of State Estimation. In: Proc. of ICRA. (2005)
9. Jitsukawa, Y., et al.: Fast Decision Making of Autonomous Robot under Dynamic Environment by Sampling Real-Time Q-MDP Value Method. In: Proc. of IROS. (2007) 1644–1650
10. Thrun, S., et al.: Probabilistic ROBOTICS. MIT Press (2005)
11. Latombe, J.C.: Robot Motion Planning. Kluwer Academic Publishers, Boston, MA (1991)
12. Fox, D., Thrun, S., Burgard, W., Dellaert, F.: Particle Filters for Mobile Robot Localization. A. Doucet, N. de Freitas, and N. Gordon, editors, Sequential Monte Carlo Methods in Practice (2000) 470–498
13. Lenser, S., Veloso, M.: Sensor resetting localization for poorly modelled robots. In: Proc. of IEEE ICRA. (2000) 1225–1232