Osaka University Knowledge Archive

Title	Modeling motion uncertainty of moving obstacles for robot motion planning
Author(s)	三浦, 純; 白井, 良明
Citation	Proceedings - IEEE International Conference on Robotics and Automation. 2000, 3, p. 2258-2263
Version Type	VoR
URL	https://hdl.handle.net/11094/14068
rights	c2000 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE..
Note	

Osaka University Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

Osaka University

Proceedings of the 2000 IEEE
International Conference on Robotics & Automation
San Francisco, CA • April 2000

Modeling Motion Uncertainty of Moving Obstacles for Robot Motion Planning

Jun Miura and Yoshiaki Shirai
Dept. of Computer-Controlled Mechanical Systems,
Osaka University, Suita, Osaka 565-0871, Japan

Abstract

This paper describes a method of modeling the motion uncertainty of moving obstacles and its application to mobile robot motion planning. The method explicitly considers three sources of motion uncertainty: path ambiguity, velocity uncertainty, and observation uncertainty. The model is represented by a probabilistic distribution over possible position on a path of a moving obstacle. Using this model, the best robot motion is selected which minimizes the expected time of reaching the destination. By considering not the range but the distribution of the uncertainty, more efficient behaviors of the robot are realized.

1 Introduction

Motion planning is one of the fundamental functions of mobile robots. As mobile robots extend their application areas from factory to office or home, they have to cope with moving obstacles such as human or other robots. Therefore, mobile robot motion planning in dynamic environments has recently been studied extensively [1].

In the case where a robot cannot communicate with moving obstacles, the robot needs to predict the future motion of them. Most of past research can be classified, in terms of the knowledge of obstacle motion, into two categories. In one category, the obstacle motion is completely *unknown*, thus, a reactive motion selection is only reasonable way for a robot to cope with moving obstacles [2]; not the optimality but the safety of robot motion is an important issue there. In the other category, the obstacle motion is completely *known*; thus, an optimal motion can be generated by employing a *planning in space-time*[3]. In [4], for example, the robot predicts future motion of obstacles by assuming that they will continue to move at the current velocity, and plans the next best action in the space-time. Such prediction and planning are repeatedly performed. Fiorini and Shiller [5] proposed a local motion planning method using the concept of *velocity obstacles*.

In between these categories, several works consider the uncertainty in obstacle motion. Most of them, however, consider only the range of uncertainty (e.g., [6][7]); the robot generates a plan which is safe regardless of the actual obstacle motion. Such an uncertainty modeling may result in an inefficient robot motion if the positional distribution of an obstacle is not uniform within the range (see Fig. 1 for an example).

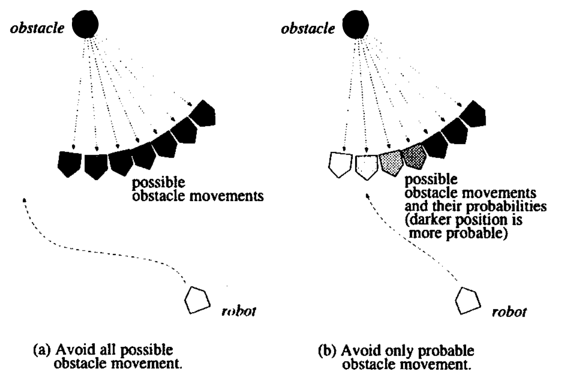


Fig. 1: An example case where considering the bias in the motion uncertainty is effective.

In order to model the motion uncertainty of obstacles, we hierarchically decompose it into two levels: the ambiguity in path selection and the motion uncertainty on each path. This is analogous to "path-velocity" decomposition [8] in motion planning.

First, let us consider the path ambiguity. In a usual environment where both static and moving obstacles exist, we can predict the motion of moving obstacles to some extent. They never move randomly; each has its own start and goal points and the path connecting them should be generated in some rational manner (e.g., by a minimum-length criterion). In a typical office environment, for example, flow of people is restricted by the placement of walls, doors, furniture, and so on. To enumerate possible paths of a moving obstacle, we use the *tangent graph* [9]. Using the given knowledge of static obstacles, a tangent graph is generated. Each path is represented as a set of consecutive segments on the graph.

Concerning the motion uncertainty on a path, let us consider the following simple example. Suppose you

0-7803-5886-4/00/$10.00© 2000 IEEE

are going to cross a street and a car is approaching you. You have to decide when to begin crossing the street (i.e., before or after the car passes). When the car is far away, predicting when the car will pass in front of you suffers from a large uncertainty because it is a prediction of a far future, and because the observation uncertainty is large for a far object. As time advances, however, the situation will be more certain and, at some time point, you will be able to make a decision with confidence. As seen from this example, for modeling motion uncertainty, we consider two sources of uncertainty: the velocity uncertainty of an obstacle and the observation uncertainty of the robot.

Using the proposed model, the robot can enumerate points on its path where it may meet each moving obstacles, and can calculate the probabilistic distribution of the moving obstacle coming to such a point. This distribution is used for the robot to estimate the expected time to the destination. The best robot motion is then selected which minimizes the expected time.

2 Modeling Motion Uncertainty on a Path

2.1 Modeling Velocity Uncertainty

We assume the following on the motion of obstacles: each moving obstacle has the possible range of its velocity, represented as $[v_{min}, v_{max}]$; it changes the velocity at every *time step* ΔT; the velocity for a time step is constant, and is randomly *and* independently selected within the above range[1].

Under these assumptions, we can predict the future position of a moving obstacle as follows. Let x_0 and σ_0^2 be the current position and the variance of an obstacle and v_k be the velocity at the kth time step. Then the position x_i after i steps is given by:

$$x_i = x_0 + \sum_{k=1}^{i} v_k \Delta T. \tag{1}$$

Since every v_k follows the same but independent uniform distribution within the above velocity range, the distribution of x_i can be *approximated* by a normal distribution (by *central limit theorem* [10]); the variance σ_{step}^2 of the motion added by one step is calculated as that of uniform distribution of width $v_{max} - v_{min}$, which is $(v_{max} - v_{min})^2/12$. The probability density function $p(x;i)$ of the obstacle being at x after moving for i steps is then given by

$$\begin{aligned} p(x;i) &= \frac{1}{\sqrt{2\pi\sigma_i^2}} exp\left(-\frac{(x-\bar{x}_i)^2}{2\sigma_i^2}\right), \qquad (2) \\ \bar{x}_i &= x_0 + i\bar{v}\Delta T, \\ \sigma_i^2 &= \sigma_0^2 + i\sigma_{step}^2, \end{aligned}$$

where $\bar{v} = (v_{max} + v_{min})/2$ is the mean velocity of the obstacle.

[1] More constraining knowledge could be used depending on the actual environment and the problem settings.

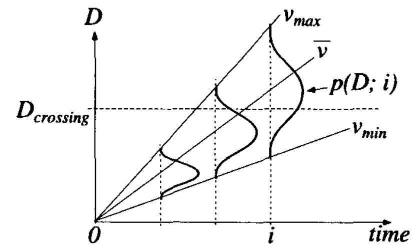


Fig. 2: Prediction of arrival time.

2.2 Predicting the Arrival Time of Moving Obstacle at a Crossing

Motion of an obstacle affects that of the robot near the crossings of their paths. Thus it is necessary for the robot to calculate the distribution of the arrival time of the obstacle at a crossing. Using the velocity uncertainty model described above, the distribution is calculated as follows (see Fig. 2).

In the figure, the vertical axis indicates the moving distance of the obstacle from the current position; the horizontal axis indicates the time (or time step). $D_{crossing}$ is the distance to a specific crossing on the path. Since the positional distribution of the obstacle at some time point is calculated by eq. (2), the probability $P(i)$ of the obstacle reaching the crossing at the ith time step can be approximated by:

$$P(i) = \alpha p(D_{crossing}; i), \tag{3}$$

where α is a normalization constant.

2.3 Modeling Observation Uncertainty

Another source of uncertainty in predicting obstacle motion is the observation uncertainty. We suppose a vision-based mobile robot which uses stereo vision to detect obstacles and to measure their position and velocity. We use the uncertainty model of stereo vision that we have previously developed [11]. The model uses a normal distribution to represent the positional uncertainty of an object due to vision uncertainty.

Every time the robot measures the position of an obstacle, the position data is statistically integrated with the previous data to reduce the uncertainty. Here we explain how to estimate the positional uncertainty of an obstacle after observation at the next step.

Let $N(\mu_0, \sigma_0^2)$ be the *predicted* distribution of obstacle position x before the next observation; this distribution is calculated from the current distribution and the predicted motion uncertainty added by the next step. Let x_{obs} be the position measured by the next observation. Assuming that the variance σ_{obs}^2 of x_{obs} is constant regardless of the true value of x, x_{obs} follows $N(\mu_0, \sigma_0^2 + \sigma_{obs}^2)$. From these values, mean μ_1 and variance σ_1^2 of the distribution after integrating the next observed data are estimated as follows; σ_1^2 is given by $\sigma_0^2\sigma_{obs}^2/(\sigma_0^2 + \sigma_{obs}^2)$. μ_1 cannot be predicted

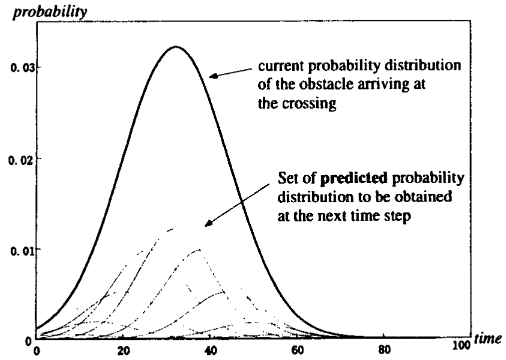


Fig. 3: Reduction of prediction uncertainty.

deterministically; instead, its *distribution* can be estimated as $N(\mu_0, \sigma_0^2/(\sigma_0^2 + \sigma_{obs}^2))$. Refer to [11] for the details. We use this distribution of μ_1 to enumerate a set of possible states after the next step.

2.4 Gradual Reduction of Prediction Uncertainty

By combining the velocity and the observation uncertainty, we can model the gradual reduction of prediction uncertainty, which is described by an example in Sec. 1. Fig. 3 shows the current probability distribution of an obstacle arriving at a crossing, and a set of *predicted* probability distributions (weighted with its occurrence probability) to be obtained after one time step passes and the new observation result is integrated[2]. The set of distributions covers all possible situation which is represented by the current distribution. Only one of which, however, will actually occur. The variance of each distribution in the set is smaller than that of the current one, that is, the situation will become more certain.

3 Motion Planning for Fixed Path Moving Obstacle

3.1 Planning the Next Motion

The robot basically follows a path on the tangent graph to minimize the moving distance to the destination as long as there is no influence from moving obstacles. If the robot has to consider avoidance of collision with them, the robot selects a certain number of nodes as the candidates of an intermediate goal and enumerates a set of candidate motions which cover the directions to the candidate nodes (see Fig. 4). After each time step, the robot observes obstacles, estimates their positional uncertainty, and performs one-step look-ahead search for the next motion. Once a

[2]Note that the velocity range of the obstacle is discretized with some granularity for a computational purpose.

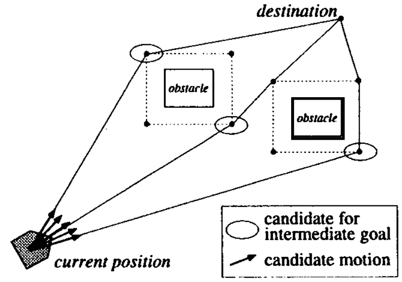


Fig. 4: Generating candidate motions.

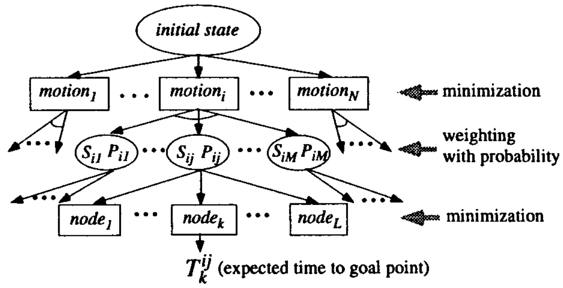


Fig. 5: Selecting the next motion.

node is known to be far superior to the others, the commitment is made to the node.

The detailed planning algorithm is as follows (see Fig. 5). For each motion $i\,(i = 1, \ldots, N)$, the robot first predicts the set of possible states $\{S_{ij} | j = 1, \ldots, M\}$ with their probability P_{ij}, which are to be obtained after the motion, as described in the previous section. Then for each state S_{ij}, the robot calculates the expected time T_k^{ij} of reaching the destination when selecting candidate node $k\,(k = 1, \ldots, L)$[3] and selects the best (minimum-time) candidate node k_{ij}^* as:

$$k_{ij}^* = \arg\min_{k=1}^{L} T_k^{ij}. \tag{4}$$

Then the expected time T_i of reaching the destination when taking candidate motion i is given by

$$T_i = \sum_{j=1}^{M} P_{ij} T_{k_{ij}^*}^{ij}. \tag{5}$$

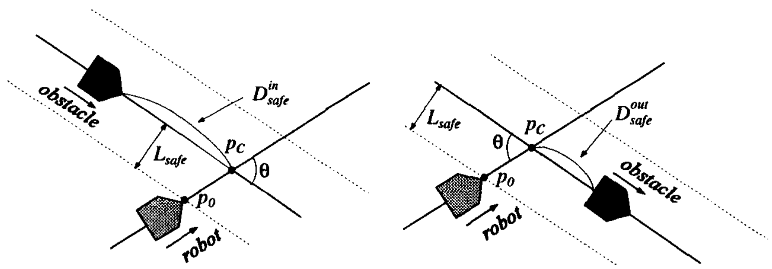
Finally the best motion i^* is selected as:

$$i^* = \arg\min_{i=1}^{N} T_i. \tag{6}$$

3.2 Calculating Expected Time to Destination

Collision Avoidance by Stopping We set a safety distance L_{safe} and controls the robot so as not to enter

[3]The next subsection will explain how to calculate the expected time.



(a) Pass before the obstacle. (b) Pass after the obstacle.

Fig. 6: Conditions for collision avoidance by stopping.

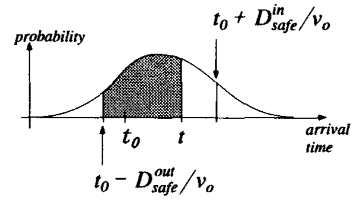


Fig. 7: Expected time to wait.

within L_{safe} from any obstacles. Basically the robot moves at a constant speed on the shortest path. If the path of the robot and that of an obstacle intersect, and if the robot knows the obstacle will come to the distance less than L_{safe}, the robot stops before the intersection point (crossing) and waits for the obstacle to pass by[4]

Calculating Waiting Time The period during which the robot has to wait is calculated as follows. Let us consider Fig. 6. The two paths intersect at p_c with angle θ. The robot waits at point p_0 whose distance to the path of the obstacle is L_{safe}. Let t_0 be the time at which the robot reaches p_0. To calculate the waiting period, we first calculate two distances, D^{in}_{safe} and D^{out}_{safe}. D^{in}_{safe} indicates the distance of the obstacle to the crossing p_c such that the robot can pass the crossing before the obstacle if the obstacle is further than D^{in}_{safe} at t_0 (see Fig. 6(a)). D^{out}_{safe} is the distance from the crossing such that the robot can pass the crossing after the obstacle if the obstacle is further than D^{out}_{safe} at t_0 (see Fig. 6(b)). Assuming that near the crossing, the obstacle and the robot moves at constant speed v_o and v_r respectively, these two distances are given by:

$$D^{in}_{safe} = \frac{L_{safe}}{\sin\theta}\left\{\sqrt{\frac{v_r^2+v_o^2-v_r v_o\cos\theta}{v_r^2}}+\frac{v_o}{v_r}\right\}, \quad (7)$$

$$D^{out}_{safe} = \frac{L_{safe}}{\sin\theta}\left\{\sqrt{\frac{v_r^2+v_o^2-v_r v_o\cos\theta}{v_r^2}}-\frac{v_o}{v_r}\right\}. \quad (8)$$

If the obstacle is within the range $[p_c - D^{in}_{safe}, p_c + D^{out}_{safe}]$ at t_0, the robot has to wait for the obstacle exiting from the range. From this condition, we can obtain the following. (1) If the time of the obstacle arriving at the crossing is within the range $[t_0 - D^{out}_{safe}/v_o, t_0 + D^{in}_{safe}/v_o]$, the robot has to wait. (2) In addition, for arrival time t within the range, the robot has to wait for the duration of $t-(t_0-D^{out}_{safe}/v_o)$ (see Fig. 7); this is explained as follows. If the obstacle arrives at p_c at t, we know that it was at the distance of $v_o(t-t_0)$ to p_c at time t_0. Thus the robot has to wait while the obstacle moves by the distance $D^{out}_{safe} + v_o(t-t_0)$. Dividing this distance by v_o leads to the above waiting time.

Expected Time to Destination From the above result and the probability distribution $P(i)$ of the obstacle arriving at the crossing (see eq. (3)), we can calculate the expected time of the robot reaching the destination on a certain path. The expectation of the extra time needed for waiting, T_{wait}, is calculated by:

$$\begin{aligned} T_{wait} &= \sum_{t\in[t_{min},t_{max}]} P(t)(t-(t_0-D^{out}_{safe}/v_o)), \quad (9)\\ t_{min} &= t_0 - D^{out}_{safe}/v_o, \\ t_{max} &= t_0 + D^{in}_{safe}/v_o. \end{aligned}$$

The expected time to the destination is then calculated as the sum of T_{wait} and the time needed in the case where the robot encounters no obstacles.

The motion uncertainty of the robot is not considered in the above discussion. However, given that the uncertainty is represented by a probabilistic distribution, it can be easily incorporated by further calculating the expectation over the possible range of robot uncertainty.

3.3 Simulation Result

Fig. 8 shows a simulation result. There are a static obstacle and a moving obstacle in the environment; the robot considers two paths, among which the left one is shorter. The figure shows the movements of the robot and the obstacle until the robot reached the goal point. Since the left path is shorter, the robot started toward the left path; as the situation became more certain, the evaluation of the right path went up, while that of the left one fell down. So the robot gradually shifted its direction towards the right path and, at time $t = 17$, it committed to the right path and followed it to the goal point. The parameters used in this simulation are: obstacle velocity range is $[4.5 \pm 1.0][cm/s]$; the robot velocity is $7.5[cm/s]$;

[4] Other avoidance methods such as potential methods can be used instead, by adopting the corresponding procedure of calculating the expected time to reach the destination.

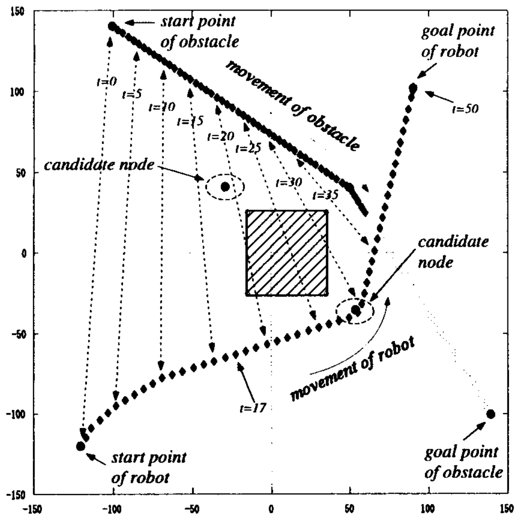
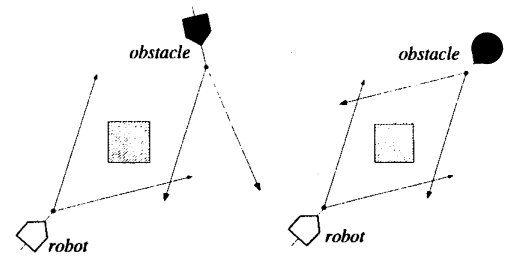


Fig. 8: A simulation result.



(a) Left route is much better. (b) Both routes are comparable.

Fig. 9: Two cases of path ambiguity of obstacle.

the variance of uncertainty in measuring distance is $6.25e-7 \cdot d^4$ (d is the distance to the obstacle); the length of the left and the right path are 317.0[cm] and 332.0[cm], respectively. The number of candidate motions is 5.

4 Modeling Path Ambiguity

This section deals with the case where there are multiple possible paths for a moving obstacle. Suppose an obstacle is moving on a segment towards a node (*branching point*) to which n possible paths are connected. Let P_{path_j} ($j = 1, \ldots, n$) is the probability of taking the ith path; if the robot does not have any prior knowledge of obstacle motion, $P_{path_j} = 1/n$.

To path ambiguity cases, we can also apply the motion selection method described in Sec. 3.1 with a little modification. For each possible path, we can calculate the expected time T_{ij} when taking candidate motion i using eq. (5). Then the expected time T_i of motion i is calculated by

$$T_i = \sum_{j=1}^{n} P_{path_j} T_{ij}. \tag{10}$$

Finally the best action is selected using eq. (6). But this method may not be appropriate in some cases.

In the case of Fig. 9(a), for example, the action towards the left path will be selected because this motion is apparently better than the other regardless of the actual path of the obstacle. In the case of Fig. 9(b), however, the above selection method is not effective; that is, each motion of the robot is good for one obstacle path but bad for the other. Since the motion is selected by one-step look-ahead search using the expected time to the destination, and since any motion is not far superior to the others, the motion towards directly one of the candidate nodes may be selected. But such a motion is very inefficient when the obstacle happens to take the path on the same side.

A reasonable strategy is, thus, to defer the commitment to a path until the obstacle takes one of the possible paths. The problem is, however, that it cannot be deterministically determined when the obstacle comes to the branching point. To cope with this problem, we calculate and use the probabilistic distribution of the obstacle reaching the point, just as in the case of the obstacle reaching a crossing (see Sec. 2.2).

Let us examine the case where there is one moving obstacle and where there are two possible paths (*left* or *right*) from the next branching point for the obstacle. Let P_{left} and P_{right} be the prior knowledge of the probability of the obstacle taking the left or the right path, respectively. Also let $P(i)$ be the probability that the obstacle arrives at the branching point at time step i (see eq. (3)) and t_{min} and t_{max} be the earliest and the latest time of arrival, respectively. Assuming that the path of the obstacle is immediately determined when the obstacle passes the branching point, the probabilities P^i_{left} and P^i_{right} of the robot detecting the obstacle on the left and the right path, respectively, at time step i are calculated as:

$$P^i_{left} = P_{left}P(i), \quad P^i_{right} = P_{right}P(i). \tag{11}$$

Fig. 10 illustrates how the situation gradually becomes certain as the time advances in this model.

The expected time of the robot reaching the destination is calculated as follows. Let T^i_{left} and T^i_{right} be the time of the robot reaching the destination when taking the best action against the left and the right path of the obstacle, respectively, after moving for the duration of i time steps without making commitment to any path. Then the expected time T of this strategy is given by

$$T = t_{min}\Delta T + \sum_{i=t_{min}}^{t_{max}} (P^i_{left}T^i_{left} + P^i_{right}T^i_{right}). \tag{12}$$

This expected time is compared with that of the action selected by the previous, one-step look-ahead strategy (see Sec. 3.1) and the better action is selected. This formulation can be extended to a general, n-paths case.

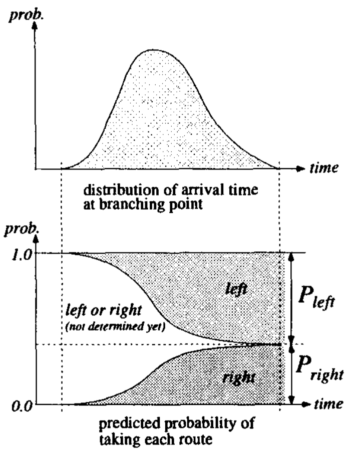


Fig. 10: Predicting probability of an obstacle taking each path.

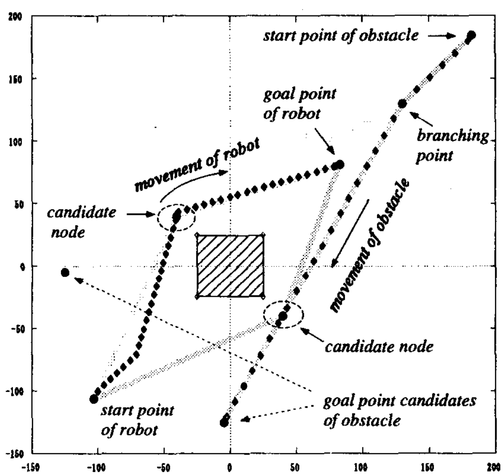


Fig. 11: Simulation result for the case of path ambiguity of a moving obstacle.

Fig. 11 shows a simulation result for the case where the robot considers the path ambiguity of a moving obstacle. Until time $t = 6$, the robot moved toward the center of the two candidate nodes and at time $t = 7$, it shifted to the left path because it detected the obstacle took the right path at that time. The parameters used in this simulation are: obstacle velocity range is $[15.3 \pm 1.0][cm/s]$; the robot velocity is $7.5[cm/s]$; the other parameters are the same as in the previous simulation shown in Fig. 8.

5 Conclusions and Discussion

We have proposed a method of modeling the motion uncertainty of moving obstacles to be used for robot motion planning. We consider three sources of the motion uncertainty, path ambiguity, velocity uncertainty, and observation uncertainty, to construct a probabilistic uncertainty model. Using this model, we can represent the gradual reduction of the uncertainty in motion prediction, which we usually experience in many situations. Based on this probabilistic model, the motion planner repeatedly selects the best motion in a decision-theoretic manner.

Currently we assume that the robot can determine the path of an obstacle at a branching point just after the obstacle passes the point. However, this assumption may not be effective in some cases, especially when the branching point is far from the robot and the observation uncertainty is large. An extension to the current modeling is to consider the observation uncertainty in determining the path of the obstacle.

References

[1] T. Tsubouchi and M. Rude. Motion Planning for Mobile Robots in a Time-Varying Environment. *J. of Robotics and Mechatronics*, Vol. 8, No. 1, pp. 15–24, 1996.

[2] S. Ishikawa. A Method of Indoor Mobile Robot Navigation by Using Fuzzy Control. In *Proc. 1991 IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, pp. 1013–1018, 1991.

[3] K. Fujimura. Time-Minimum Routes in Time-Dependent Networks. *IEEE Trans. on Robotics and Automation*, Vol. 11, No. 3, pp. 343–351, 1995.

[4] T. Tsubouchi and S. Arimoto. Behavior of a Mobile Robot Navigated by an Iterated Forecast and Planning Scheme in the Presence of Multiple Moving Obstacles. In *Proc. 1994 IEEE Int. Conf. on Robotics and Automation*, pp. 2470–2475, 1994.

[5] P. Fiorini and Z. Shiller. Motion Planning in Dynamic Environments Using Velocity Obstacles. *Int. J. of Robotics Research*, Vol. 17, No. 7, pp. 760–772, 1998.

[6] A. Elnagar and A. Basu. Local Path Planning in Dynamic Environments with Uncertainty. In *Proceedings of 1994 IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pp. 183–190, 1994.

[7] A. Inoue, K. Inoue, and Y. Okawa. On-line Motion Planning of an Autonomous Mobile Robot to Avoid Multiple Moving Obstacles Based on the Prediction of Their Future Trajectories. *J. of Robotics Soc. of Japan*, Vol. 15, No. 2, pp. 249–260, 1997. (in Japanese).

[8] K. Kant and S.W. Zucker. Towards Efficient Trajectory Planning: the Path-Velocity Decomposition. *Int. J. of Robotics Research*, Vol. 5, No. 3, pp. 72–89, 1986.

[9] Y.-H. Liu and S. Arimoto. Path Planning Using a Tangent Graph for Mobile Robots Among Polygonal and Curved Objects. *Int. J. of Robotics Research*, Vol. 11, No. 4, pp. 376–382, 1992.

[10] M. Fisz. *Probability Theory and Mathematical Statistics*. Wiley, 1963.

[11] J. Miura and Y. Shirai. Vision and Motion Planning for a Mobile Robot under Uncertainty. *Int. J. of Robotics Research*, Vol. 16, No. 6, pp. 806–825, 1997.