A New Analytical Representation to Robot Path Generation with Collision Avoidance through the Use of the Collision Map

Seung-Hwan Park and Beom-Hee Lee

Abstract: A new method in robot path generation is presented using an analysis of the characteristics of multi-robot collision avoidance. The research is based on the concept of the collision map, where the collision between two robots is presented by a collision region and a crossing curve TLVSTC (traveled length versus servo time curve). Analytic collision avoidance is considered by translating the collision region in the collision map. The 4 different translations of collision regions correspond to the 4 parallel movements of the actual original robot path in the real world. This analysis is applied to path modifications where the analysis of collision characteristics is crucial and the resultant path for collision avoidance is generated. Also, the correlations between the translations of the collision region and robot paths are clarified by analyzing the collision/non-collision areas. The influence of the changes of robot velocity is investigated analytically in view of collision avoidance as an example.

Keywords: Collision avoidance, collision map, mobile robot, path generation.

1. INTRODUCTION

Collision avoidance among robots is becoming an important issue, especially in an environment where there are many robots operated in a common workspace and exposed to obstacles. In such case, a robot can be an obstacle to another robot. So far, many studies have been done with respect to anticipating the movement of robots so that collision situation is removed in order to ensure completion of assigned tasks.

These studies have been carried out in various fields such as probability, vision, behavior-base and fuzzy logic. In particular, methods using the geometric properties have given diverse and useful results. Tsubouchi et al. [1-3] discussed the method of iterated forecast and planning which predicted the motion of the robot from its situation, planned the following motion and iterated these steps. Generally, human beings reach their goal through the optimal path without colliding with mobile obstacles including other human beings. The method proposed herein basically imitates this usual human behavior. Yamamoto et al. [4] and Fiorini et al. [5] investigated the collision avoidance problem against a dynamic obstacle by using the concept of velocity obstacle. If the velocity vector set was included in the velocity obstacle, there was assumed to be a possibility of collision. Then the robot must wait or the path of the robot has to be changed. Abe et al. [6] extended this concept to avoid collision for multiple mobile robots. Angel P. del Pobil et al. [7,8] modeled robots and obstacles as combinations of spheres to detect collision and Czarnecki [9] embodied the results in the 3-dimensional collision map. Minguez et al. [10] studied a geometry-based environment design, which is so-called the Nearness Diagram. Although this method was applied to environment where there was very complex but not having a moving obstacle, it could be extended to dynamic environment.

Ando [11] proposed a path planning method for collision-free motion by using the concept of global and local search. Qu et al. [12] considered a kinematic model for robots, which was used to derive feasible trajectories and corresponding steering controls, and developed a new collision-avoidance condition for the dynamically changing environment. Additionally, Li et al. [13] proposed a fast and efficient centralized planner which used a hierarchical sphere tree structure to group robots dynamically.

On the other hand, Miura et al. [14], and Miyata et al. [15] used the probability method to estimate the waiting time of the robot for a predicted collision. They first designated the path selection probability

Manuscript received March 21, 2005; revised August 16, 2005; accepted December 7, 2005. Recommended by Editorial Board member Sangdeok Park under the direction of Editor Jae-Bok Song. This work was supported by the BK21 Information Technology at Seoul National University, Automation and Systems Research Institute (ASRI) in Seoul National University, one of the 21st Century Frontier R&D Program funded by the Ministry of Commerce, Industry, and Energy of Korea.

Seung-Hwan Park and Beom-Hee Lee are with the School of Electrical Engineering, Seoul National University, San 56-1, Sillim-dong, Gwanak-gu, Seoul 151-742, Korea (e-mails: sinkyv@chol.com, bhlee@asri.snu.ac.kr).

according to the moving obstacles, and then used this probability to calculate the expected time to the destination for each path to find the optimal path. Finally, they maneuvered the robot to its goal through the selected optimal path. Tadokoro et al. [16] statistically predicted the human motion that could avoid collision with another human. They used a GA (Genetic Algorithm) to obtain the optimal movement. Also, Suwannatat et al. [17] and Nair et al. [18] performed the polar transform for timed-images from a vision system. They observed the changes in the timed-images to extract information about the moving obstacles. This information was used to avoid the dynamic obstacles.

Recently, there have been studies which are based on behavior-base and fuzzy logic to simplify repetitive mechanical motions and to make robot motions close to human motions. Parker et al. [19] reported a behavior-based method which made the robot execute predefined motions if it recognized a pertinent situation. Aoki et al. [20] used the steering and velocity control inputs based on fuzzy logic so that the robot may select an optimal behavior automatically. These control inputs were finely adjusted and combined by the reinforcement-learning algorithm. Zhang et al. [21] studied a dual neural network to avoid obstacles. Their method was based on the dynamically-updated inequality constraints and the physical constraints. Also, Yang et al. [22] used a neural network as a torque controller to non-holonomic mobile robots for their collision-free navigation.

As the number of robots and dynamic obstacles increases, the calculation load will become increase greatly in most of the above methods. In contrast, the method using the collision map [23] has reasonable calculation load and can check collision directly from the graph. Also the overall calculation load does not increase much even if the number of dimensions or robots and obstacles increases. This is due to prioritized planning. Through prioritized planning, which was used in the collision map, a single planning problem in high dimension space can be divided into sequential planning problems in low dimension space [24]. In addition, a collision can be detected by only the distance between two robots in the collision map. Thus, small calculation load becomes one of merits of the collision map, and also the main feature of the proposed method using the collision map.

We focus on the suggestion of a new algorithm for collision avoidance using the collision map. In the collision map, both collision region and TLVSTC (traveled length versus servo time curve) are used to detect a collision. Here, we have analyzed the translations of the collision region for the first time. The translations of the path are considered as path modifications in conjunction with the modification of the collision map. In addition, we apply this method to the designation of the collision and the non-collision area to verify the effectiveness of our proposed method in collision avoidance.

The presentation of this paper proceeds as follows. In Section 2, the basic concepts of the collision map is presented with translations of the collision region. The translation of the collision region is explained in terms of the path modifications of the robot. In Section 3, we explain the way of determining the collision and the non-collision area. The effect of the robot velocity changes is investigated in this section, and the simulation results for verification of our analysis are presented in Section 4. Finally, conclusions are made in Section 5.

2. ANALYSIS OF THE ROBOT COLLISION AVOIDANCE

2.1. Collision map and collision avoidance

We first consider a two-robot system. The robot with a higher priority is called 'robot 1,' and the other 'robot 2.' The radii of the two robots are R_1 and R_2 respectively. If we use the obstacle space scheme, robot 1 can be represented as the robot that has the radius of R_1+R_2, and robot 2 can be considered as a point robot. Because robot 1 has the higher priority, this robot will not change its original trajectory. On the contrary, robot 2 must modify its trajectory if there is any possibility of collision. It is assumed that two robots move along linear paths, as shown in Fig. 1. The concept of the collision map can be applied to arbitrary shape paths. But in this paper, robot paths are restricted to linear paths for simplicity. These two robots have a potential collision under the original trajectories if the path of robot 2 meets robot 1, which has the radius of R_1+R_2. In this case, the part of robot 2 path that overlaps with robot 1 is called the 'collision length', which is denoted by the portion between $\lambda_1(k)$ and $\lambda_2(k)$ in Fig. 1. The existence of this overlapped part is examined at every instant of

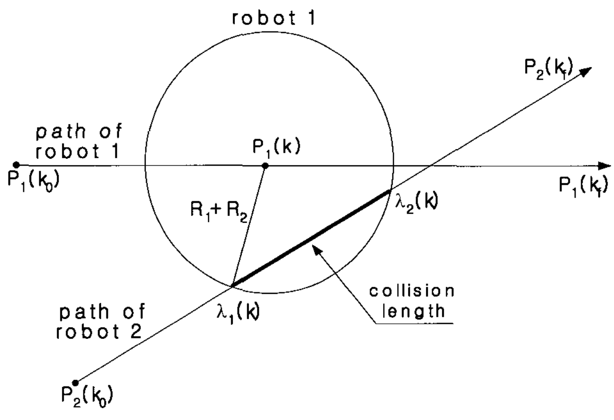


Fig. 1. Paths of two robots.

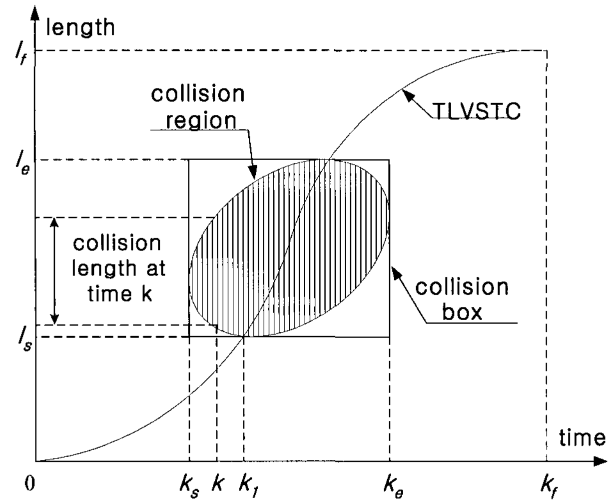


Fig. 2. TLVSTC and the collision region.

the sampling time. These collision lengths are collected to construct the 'collision region.' If the TLVSTC (traveled length versus servo time curve) of robot 2 meets this region, it indicates that the two robots will collide under the original trajectories as shown in Fig. 2. In this figure, the vertical axis represents the traveled length of robot 2 and the horizontal axis represents the elapsed time.

The collision between robot 1 and robot 2 can be analyzed algebraically from Fig. 1. In Fig. 1, $p_1(k)$ is the center point of robot 1 at time k. If we represent the position of robot 2 at time k as $p_2(k)$, the original trajectory of robot 2 is:

$$p_2(k) = p_2(k_0) + \lambda(p_2(k_f) - p_2(k_0)), \qquad (1)$$

where $0 \le \lambda \le 1$, $p_2(k_0)$ and $p_2(k_f)$ are the initial and final position of robot 2, respectively.

The collision between two robots occurs at time k when the distance between $p_1(k)$ and $p_2(k)$ is less than or equal to the radius of robot 1, (R_1+R_2). Thus, we first solve the following equation.

$$(R_1 + R_2)^2 = \|p_1(k) - p_2(k)\|^2 \qquad (2)$$

If we replace $p_2(k)$ with (1), then we have:

$$\begin{aligned}(R_1 + R_2)^2 &= \{p_1(k) - p_2(k_0) - \lambda(p_2(k_f) - p_2(k_0))\}^T \\ &\quad \bullet \{p_1(k) - p_2(k_0) - \lambda(p_2(k_f) - p_2(k_0))\}.\end{aligned} \qquad (3)$$

More explicitly,

$$\begin{aligned}(R_1 + R_2)^2 &= \|p_1(k) - p_2(k_0)\|^2 - 2\lambda(p_1(k) - p_2(k_0))^T \\ &\quad \bullet (p_2(k_f) - p_2(k_0)) + \lambda^2 \|p_2(k_f) - p_2(k_0)\|^2.\end{aligned} \qquad (4)$$

(4) is a quadratic equation in λ. Thus it has three types of solutions. First, it may not have any real solutions, which means that there is no collision between two robots; second, it has one double real solution which is generated when robot 1 starts overlapping or starts leaving the robot 2 path; finally, it has two real solutions, which means that robot 1 encroaches on the path of robot 2 and two robots may collide.

For collision avoidance, the TLVSTC of robot 2 should not meet the collision region in Fig. 2. We know that it is difficult to mathematically represent the boundary line of the collision region because it is a set of boundary values of the collision length at each sampling time. Thus, the collision box is introduced as shown in Fig. 2. In this figure, k_s is the time that robot 1 starts encroaching the path of robot 2 and k_e is the time that robot 1 leaves the path of robot 2. l_s and l_e are the minimum and maximum value of the boundary values of the collision length in the collision region, respectively. We can compute the edge coordinates of the collision box by using the above parameters and they are used to modify the robot 2 trajectory so that robot 2 avoids collision with robot 1.There are two methods that can be used to avoid collision, namely,

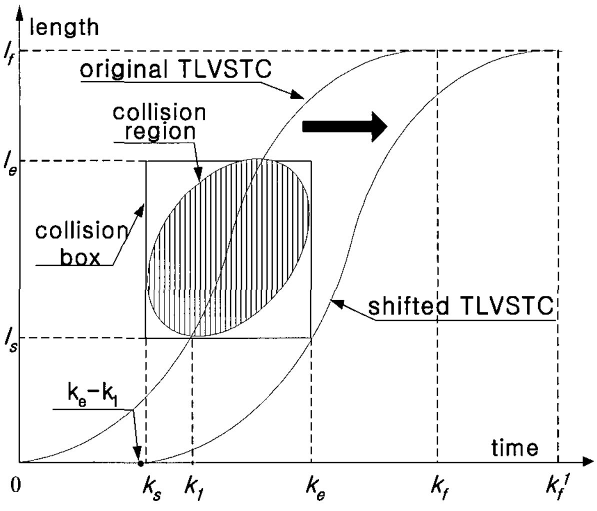


Fig. 3. Collision avoidance through time delay.

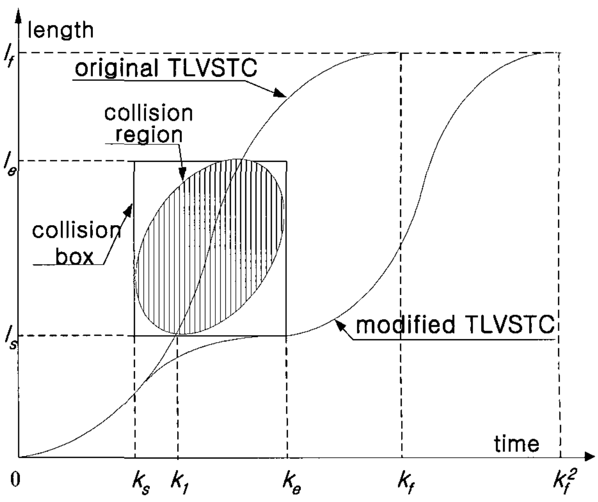


Fig. 4. Collision avoidance through speed reduction.

time delay and speed reduction. Time delay is the method that delays the start time of robot 2 to avoid the collision by the value k_e-k_1 as shown in Fig. 3. Consequently, robot 2 reaches its goal at the time k_f^1 that is delayed for k_e-k_1 from k_f. In contrast, all robots are assumed to start moving simultaneously in speed reduction. Here, the moving speed of robot 2 is changed to avoid collision. The velocity profile of robot 2 is modified so that the robot 2 trajectory does not touch the collision region. It should be noted that if speed reduction is used, there may be an instance when the velocity of robot 2 becomes zero as it proceeds. Thus, this method of speed reduction results in lower performance in terms of arrival time than that of time delay as can be seen in Fig. 4.

2.2. Translations of the collision region

We consider the collision avoidance of the robot in terms of the translation of the collision region. The translation of the collision region corresponds to the translation of the robot path in reality. When the TLVSTC (traveled length versus servo time curve) of robot 2 crosses the collision region, there exists a collision in the original trajectories of the two robots. The change or translation of the robot 2 path has not been considered yet in the original concept of the collision map. This is a suitable assumption for industrial robots because their paths are fixed and their workspace is restricted generally. On the contrary, service robots are generally movable, and thus, their paths can be selected freely for collision avoidance for better performance.

In considering the translations of the collision region, we treat the collision box as the collision region. Moving directions are classified into 4 cases and these are discussed in the following. First, we translate the collision region to the right/left and then to the up/down direction. These translations are represented by case 1 through case 4 in Fig. 5. The collision region located at the center indicates the original case. The collision region is composed by a bunch of line segments called collision lengths. The collision length is the part of the robot 2 path that overlaps with robot 1 as discussed in Fig. 1. In Fig. 5, case 1 indicates the right-shifted collision region by $\Delta t(d_1)$ from the original case. Also, case 2 indicates the left-shifted collision region by $\Delta t(d_2)$ from the original case. Robot 1 should meet robot 2 path as later as d_1 for case 1 and as earlier as d_2 for case 2. Thus, the robot 2 path must translate as much as d_1 away from the start point of robot 1 for case 1 and as much as d_2 toward the start point of robot 1 for case 2. These are shown in Fig. 6.

Now we discuss the translation of the collision region to the vertical direction. These are case 3 and case 4 as shown in Fig. 5. Case 3 corresponds to the down-shifted collision region by d_3 from the original case. Also, case 4 corresponds to the up-shifted collision region by d_4 from the original case. The corresponding translations of the robot 2 path are shown in Fig. 7. We assume that robots are moving in straight line paths. In the collision map, any paths can be denoted by the parameter λ from 0 to 1 irrespective of their shapes (see (1)), and can be used as robot paths. Also, a collision is detected by only the distance between two robots. Thus, this method can be applied to arbitrary shape paths with more computational burden. Here, the translations of each collision region not to cross TLVSTC enable the robots to avoid collisions obviously. The reason which we use straight line paths is to show our analysis and results more clearly.

The distances d_1 and d_2 in cases 1 and 2 are calculated from the velocity profile of robot motion as shown in Fig. 8. There are three possibilities where the time difference k_2-k_1 is located in the velocity

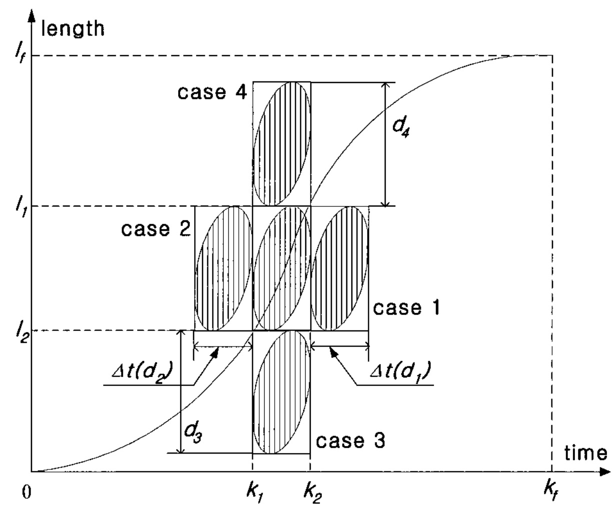


Fig. 5. Translations of the collision region ($\Delta t(d_i)$ indicates the travel time required for the robot to move the distance d_i on the robot path).

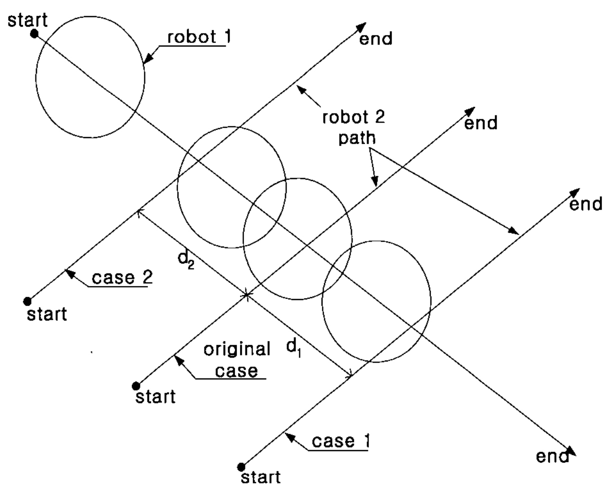


Fig. 6. Translations of the robot 2 path in cases 1 and 2.

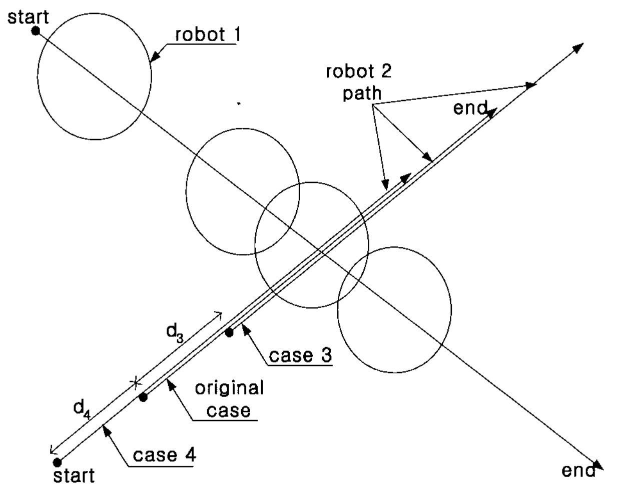


Fig. 7. Translations of the robot 2 path in the cases 3 and 4.

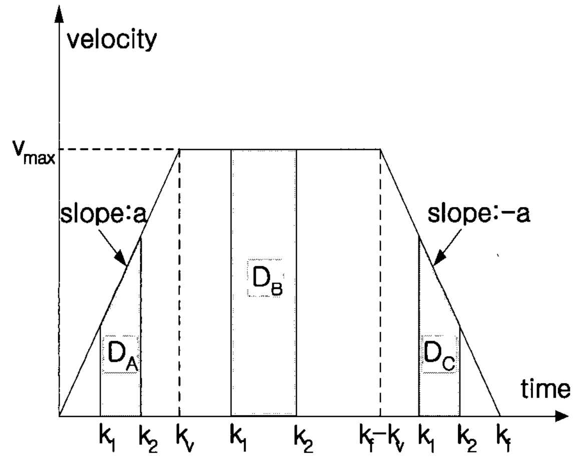


Fig. 8. Calculation of the distance.

profile of a robot. In Fig. 8, the area of D_A represents the distance corresponds to the time difference k_2-k_1 in the constant accelerating motion. If the time difference k_2-k_1 is in constant velocity section, the area of D_B represents the distance related to this situation, and finally the area of D_C represents the distance corresponds to the time difference k_2-k_1 in the constant decelerating motion. If the time difference k_2-k_1 is located across two or three of the above sections, then we divide that time difference into several parts and apply the calculation to each part separately. The results are shown in Eqs. (5) to (7) for D_A, D_B, and D_C, respectively. In these equations, a is the acceleration of the robot, v_{max} is the maximum velocity of the robot, k_v is the time when the robot velocity reaches its maximum value, and k_f is the arrival time of the robot to its goal.

$$D_A = \frac{1}{2}a(k_2^2 - k_1^2) \tag{5}$$

$$D_B = v_{\max}(k_2 - k_1) = ak_v(k_2 - k_1) \tag{6}$$

$$D_C = \frac{1}{2}a(k_2 - k_1)\{(k_f - k_1) + (k_f - k_2)\} \tag{7}$$

3. COLLISION AREA AND ITS APPLICATIONS

3.1. Collision and non-collision areas

The translations of the collision region imply the translations of the robot 2 path. In this section, we generate the collision-free path of robot 2 by using the method of translation. In Fig. 5, if the collision region is located above the TLVSTC, its lower-right edge is the point where the translated collision regions touch with the TLVSTC. These situations correspond to cases 2 and 4. If the collision region is located below the TLVSTC, its upper-left edge is the contact point with the TLVSTC. These correspond to cases 1 and 3. In fact, a collision does not occur only if the collision region remains away from the TLVSTC. Thus, if we translate the collision region to some direction while the region contacts with the TLVSTC, we can determine the direction and extent of the path translation for collision avoidance. This result is shown in Fig. 9. In this figure, the area A represents the collision area. If the starting point of robot 2 is located in area A during the path translation, two robots collide near the cross point of their paths. On the contrary, the areas B and C represent the non-collision areas. If the starting point of robot 2 is located in these areas during the path translation, two robots can move to their goals without collision. We can select any position in these areas for the robot 2 path to guarantee that no collision will occur. The starting points of cases 1 through 4 are shown on s-type curves in Fig. 9. As we mentioned above, these s-type curves have the shape which is similar to the original TLVSTC.

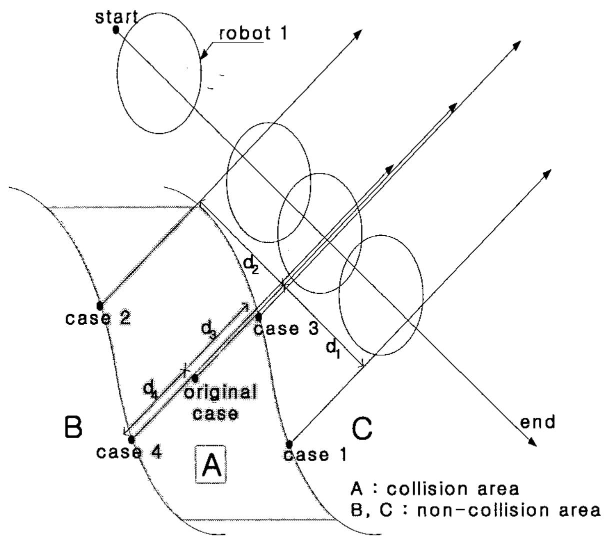


Fig. 9. Determination of the collision/non-collision areas.

3.2. Investigations on the changes of robot 1 velocity

3.2.1 When the velocity of robot 1 becomes higher than the standard situation

In our analysis, it was assumed that velocities of both robots are relatively similar. This is referred to as the standard situation. If the velocity of robot 1 becomes higher than that in the standard situation, the collision map is subject to some changes. First, the collision region becomes narrower because robot 1 passes through the robot 2 path faster. This indicates that the smaller translation of the collision region needs for collision avoidance. Thus, more free-space is provided for the collision-free motion of the robot.

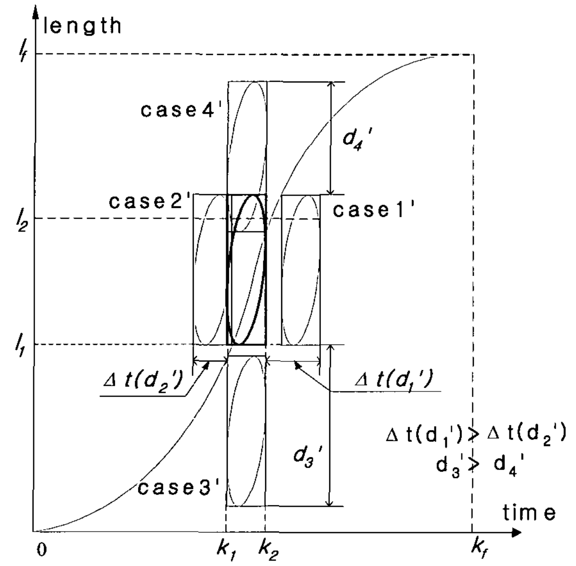


Fig. 10. Collision map when the velocity of robot 1 is higher than the standard situation($\Delta t(d_i')$ indicates the travel time required for the robot to move the distance d_i' on the robot path).

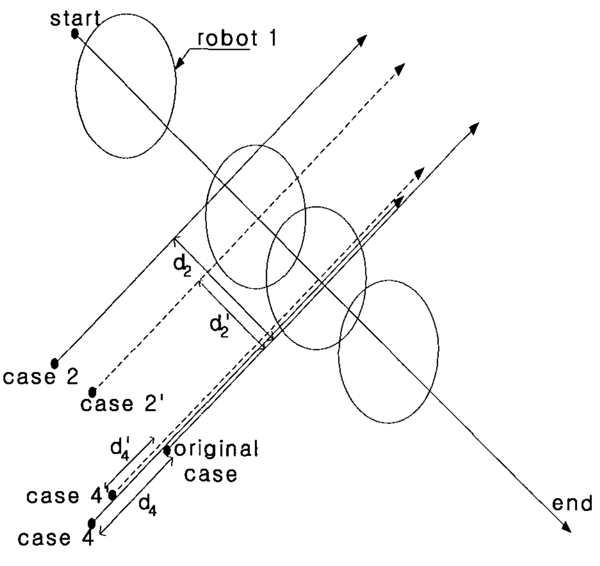


Fig. 11. Robot 2 paths when the velocity of robot 1 is higher than the standard situation.

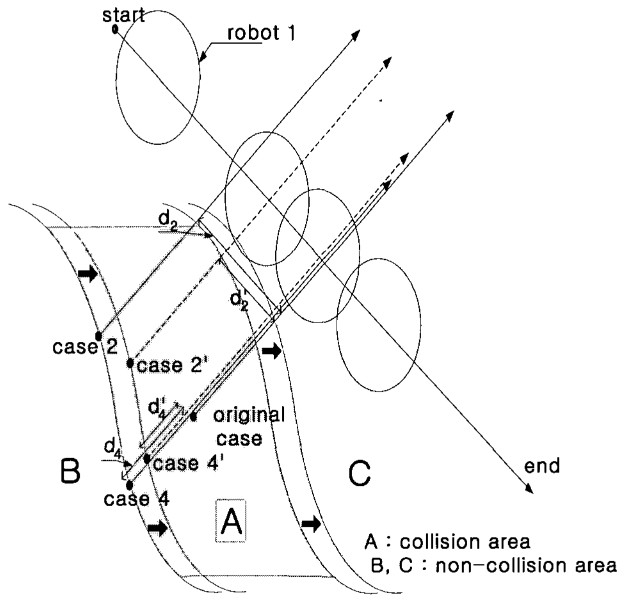


Fig. 12. Collision/non-collision areas when the velocity of robot 1 is higher than the standard situation.

Second, the time when the collision region generates is advanced. We now analyze the situation in Fig. 10, where the original collision region is crossed by the TLVSTC. The translational amount of the collision region for collision avoidance is different from case to case. As can be seen in Fig. 10, the values d_2' and d_4' are smaller than d_1' and d_3'. Thus, case 2′ or case 4′ will be a better choice if we want to translate the robot 2 path as small as possible. The translated robot paths are shown in Fig. 11, where the solid lines denoted by cases 2 and 4 correspond to the standard situation in the velocity profile. On the contrary, the dotted lines denoted by cases 2′ and 4′ correspond to the situation where the robot 1 velocity becomes higher than that in the standard situation. The amounts of the translations are denoted as d_2' and d_4', which are smaller than those of the standard situation, d_2 and d_4. If we further investigate the situation in Fig. 10, we find that the translated regions above the TLVSTC are preferred to the regions below the TLVSTC. Thus, we can reconstruct collision and non-collision areas for this situation as shown in Fig. 12. In this figure, two TLVSTC-like curves are moved to the right direction. If we select case 2′ or case 4′, less amount of translation will be needed for collision avoidance.

3.2.2 When the velocity of robot 1 becomes lower than the standard situation

If the velocity of robot 1 becomes lower than that of the standard situation, the collision map is subject to changes also. First, the collision region becomes wider because robot 1 passes through the robot 2 path in longer period of time. Second, there will be a time delay for the generation of the collision region. Another example is given in Fig. 13. The translational

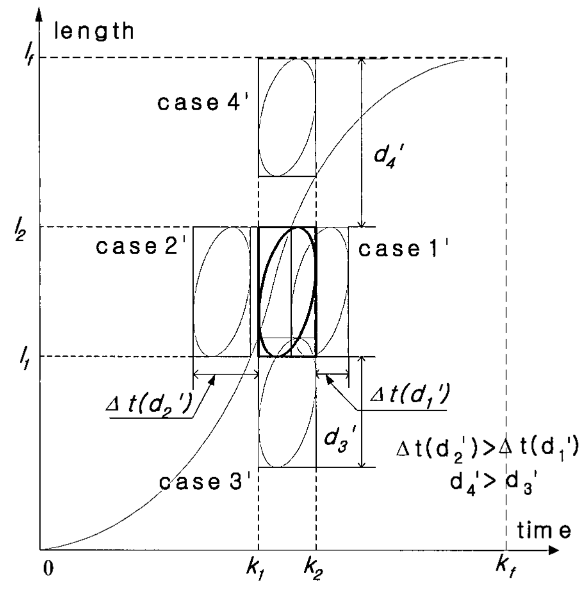


Fig. 13. Collision map when the velocity of robot 1 is lower than the standard situation($\Delta t(d_i')$ indicates the travel time required for the robot to move the distance d_i' on the robot path).

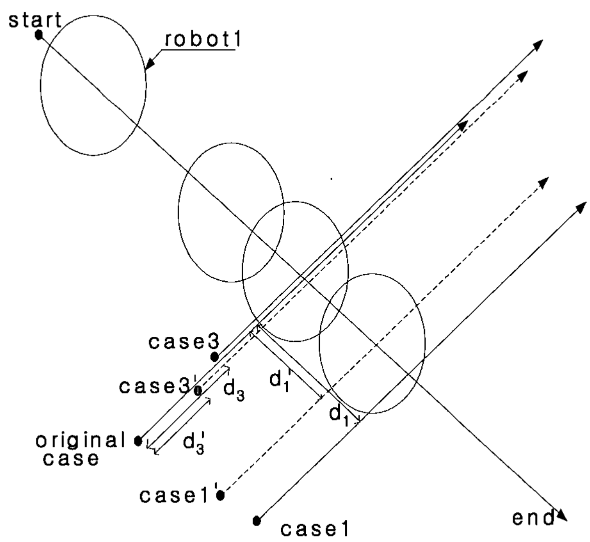


Fig. 14. Robot 2 paths when the velocity of robot 1 is lower than the standard situation.

amount of the collision region is different from case to case. As can be seen in this figure, the changed value d_2' and d_4' are larger than d_1' and d_3'. Therefore, case 1′ or case 3′ will be better choices. The translated robot paths are shown in Fig. 14, where the solid lines denoted by cases 1 and 3 correspond to the standard situation, and the dotted lines denoted by cases 1′ and 3′ correspond to the situation where the robot 1 velocity becomes lower than that of the standard situation. The amounts of translations are represented as d_1' and d_3'. We also find in Fig. 13 that the translated regions below the TLVSTC are preferred to

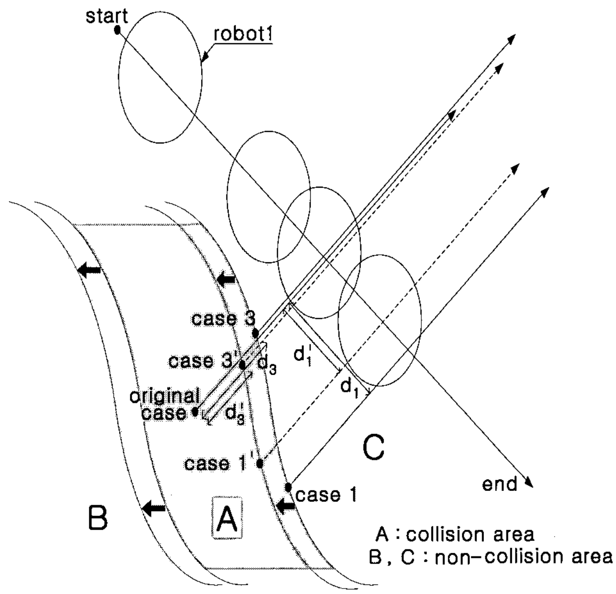


Fig. 15. Collision/non-collision areas when the velocity of robot 1 is lower than the standard situation.

the regions above the TLVSTC. Thus, we can also reconstruct collision and non-collision areas for this situation as illustrated in Fig. 15. In this figure, two TLVSTC-like curves are moved to the left direction. If we select case 1′ or case 3′, less amount of translation will be needed.

4. SIMULATION RESULTS

4.1. Simulator

We developed a simulator for the verification of our analysis as shown in Fig. 16. This simulator is consisted of a control section and a data section. The control section is located in the upper part of the simulator to simulate the various motions of robots. The data section is divided into 5 parts. Part A shows paths and motions of robots. In this part, R1 represents robot 1 which has the higher priority than

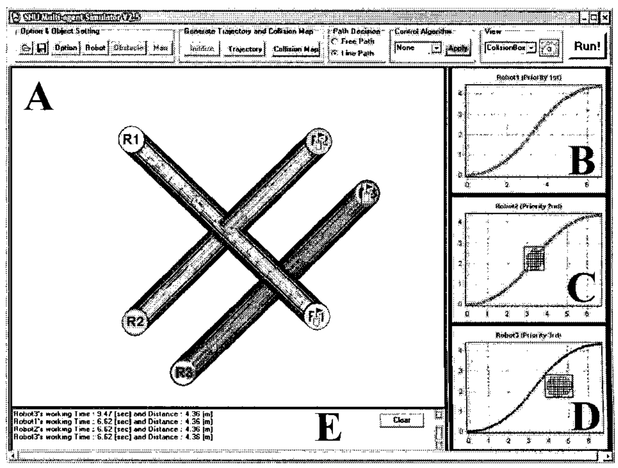


Fig. 16. Simulator.

Table 1. Numerical data of simulations.

Case	(x, y) Coordinate of Start Point	(x, y) Coordinate of End Point	Travel Time (sec)	Travel Distance (m)
Robot 1	(202, 120)	(516, 417)	6.62	4.36
Robot 2 (Original)	(208, 426)	(519, 122)		
Robot 2 (Case 1)	(289, 512)	(600, 209)		
Robot 2 (Case 2)	(123, 329)	(432, 21)		
Robot 2 (Case 3)	(281, 354)	(592, 51)		
Robot 2 (Case 4)	(128, 505)	(438, 200)		

robot 2. R2 indicates robot 2 which has a lower priority, and R3 shows the translated case of robot 2. Part B shows the collision map of robot 1, which has no collision region due to its higher priority. Part C shows the collision map for the original case of robot 2. In this part, TLVSTC crosses the collision region, thus a collision is predicted with original paths of robots. Part D represents the collision map after translation of the robot 2 path, where the collision region is translated so that it does not cross TLVSTC. Finally, part E shows numerical data for robot motions. More detailed analysis is discussed in the next section. For simplicity, part A, C and D are separated and rejoined in the subsequent discussions. The numerical data of simulations are shown in Table 1, where the maximum velocity and acceleration of robots are assumed to be 1.5m/s and 0.4m/s^2, respectively.

4.2. Verification results

4.2.1 Results for cases 1 and 2

The idea in Fig. 6 is verified for case 1 in Fig. 17 and case 2 in Fig. 18. Robot 1 moves from the upper-left position to the lower-right position, and robot 2 moves from the lower-left to the upper-right. In these figures, 3 robot paths and 2 collision maps for robot 2 are shown. The left map is for the original case of robot 2 (part C of Fig. 16) and the right map is for the translated case (part D of Fig. 16). The start and end positions of robots and their travel time and distance are also shown.

We note the change of the collision maps in Fig. 17 and 18. When the robot 2 path is translated, the collision region is also translated to the right or left direction horizontally. All collision regions in these figures have the same horizontal values. In these figures, the shapes of the collision regions may not be the same. This could happen due to the trapezoidal

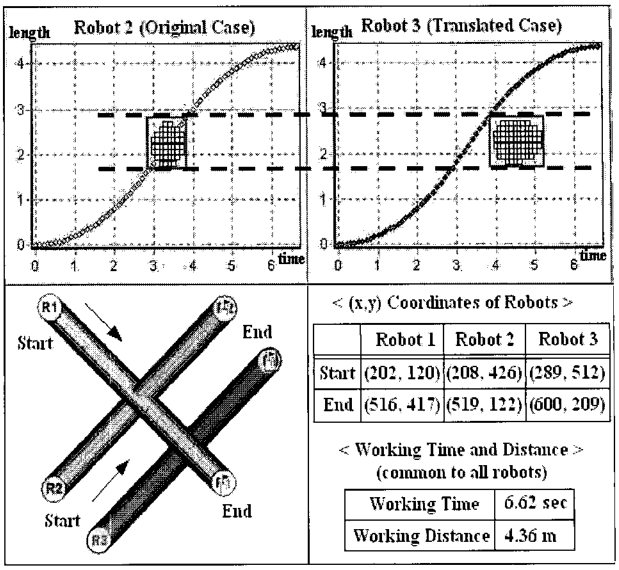


Fig. 17. Simulation result for case 1.

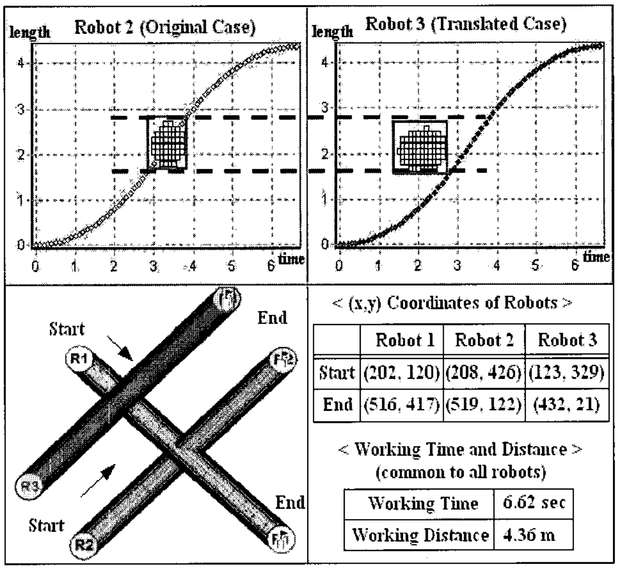


Fig. 18. Simulation result for case 2.

velocity profile of robot 1. This means that the longer time is needed to go through the robot 2 path and it would make the collision region wider.

4.2.2 Results for cases 3 and 4

The idea in Fig. 7 is shown in Fig. 19 for case 3 and Fig. 20 for case 4. As mentioned above, R3 is the translated case of robot 2. When the robot 2 path is translated, the collision region is also translated to up and down direction vertically. All collision regions in these figures have the same vertical positions. In Fig. 19, the collision region is located in the lower area of the TLVSTC. It means robot 2 passes through the crossing point faster than robot 1. On the contrary, robot 1 passes through that point faster than robot 2 in Fig. 20.

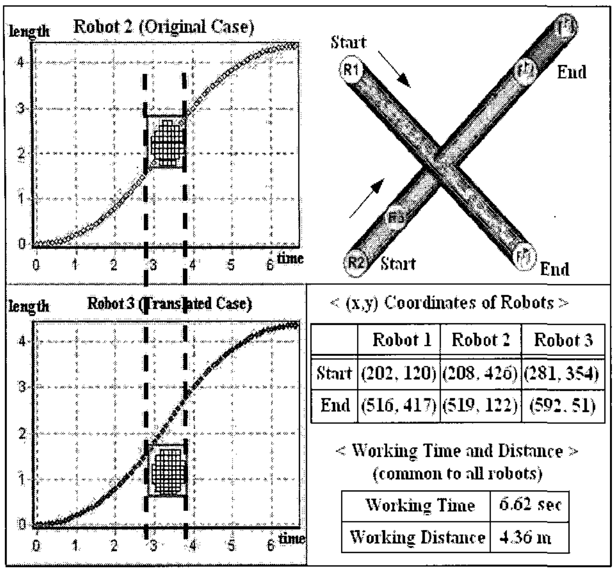


Fig. 19. Simulation result for case 3.

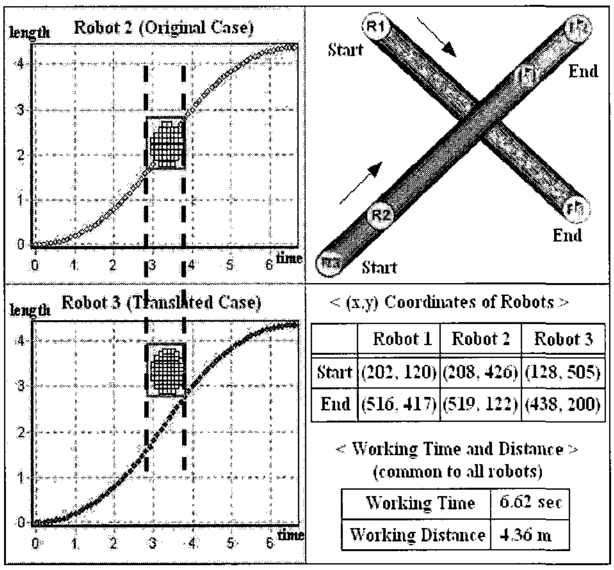


Fig. 20. Simulation result for case 4.

5. CONCLUSIONS

In this paper, we presented an analytic method to generate collision-free paths of robot through the use of the so-called "collision map." 4 different translations of the collision region were identified in the collision map and used to interpret the collision situation analytically. These 4 different translations were classified and analyzed for collision conditions and characteristics. From these translations, we could obtain collision-free robot paths. The path translation method was proposed to solve the collision avoidance problem. We also designated the collision and non-collision areas from our analysis. Finally, the changes of these areas were investigated when the robot velocity was changed. This approach can be applied to select a better path for collision avoidance where several start and goal points are located in parallel. Additionally, the collision map, which is a simple and powerful tool to detect a collision, is used to verify collision-free paths. On the other hand, it is hard to apply this approach to general situations. Actually, the main concern of this paper is to show how to handle the collision avoidance problem in terms of path translation using the proposed collision map analysis. Complement to this defect and actual implementations of this analysis are promising and needed for a future work.

REFERENCES

[1] 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," *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 2470-2475, 1994.

[2] T. Tsubouchi, S. Kuramochi, and S. Arimoto, "Iterated forecast and planning algorithm to steer and drive a mobile robot in the presence of multiple moving objects," *Proc. of the IEEE International Conference on Intelligent Robots and Systems*, pp. 33-38, 1995.

[3] T. Tsubouchi, A. Hirose, and S. Arimoto, "A navigation scheme with learning for a mobile robot among multiple moving obstacles," *Proc. of the IEEE/RSJ International Conference on Intelligent Robotics and Systems*, pp. 2234-2240, 1993.

[4] M. Yamamoto, M. Shimada, and A. Mohri, "On-line navigation of mobile robot under the existence of dynamically moving multiple obstacles," *Proc. of the IEEE International Symposium on Assembly and Task Planning*, pp. 13-18, 2001.

[5] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using the relative velocity paradigm," *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 560-565, 1993.

[6] Y. Abe and Y. Matsuo, "Collision avoidance method for multiple autonomous mobile agents by implicit cooperation," *Proc. of the IEEE/RSJ International Conference on Intelligent Robotics and Systems*, pp. 1207-1212, 2001.

[7] A. P. del Pobil, M. A. Serna, and J. Llovet, "A new representation for collision avoidance and detection," *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 246-251, 1992.

[8] M. Perez-Francisco, A. P. del Pobil, and B. Martinez, "Fast Collision detection for realistic multiple moving robots," *Proc. of 8th International Conference on Advanced Robotics*,

pp. 187-192, 1997.

[9] C. A. Czarnecki, "Collision free motion planning for two robots operating in a common workspace," *Proc. of IEEE International Conference on Control*, vol. 2, pp. 1006-1011, 1994.

[10] J. Minguez and L. Montano, "Nearness diagram (ND) navigation: Collision avoidance in troublesome scenarios," *IEEE Trans. on Robotics and Automation*, vol. 20, no. 1, pp. 45-59, 2004.

[11] S. Ando, "A fast collision-free path planning method for a general robot manipulator," *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 2871-2877, 2003.

[12] Z. Qu, J. Wang, and C. E. Plaisted, "A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles," *IEEE Trans. on Robotics and Automation*, vol. 20, no. 6, pp. 978-993, 2004.

[13] T.-Y. Li and H.-C. Chou, "Motion planning for a crowd of robots," *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 4215-4221, 2003.

[14] J. Miura, H. Uozumi, and Y. Shirai, "Mobile robot motion planning considering the motion uncertainty of moving obstacles," *Proc. of the IEEE International Conference on Systems, Man, and Cybernetics*, vol. 4, pp. 692-697, 1999.

[15] J. Miyata, T. Murakami, and K. Ohnishi, "An approach to tracking motion of mobile robot for moving object," *Proc. of 26th Annual Conference of the IEEE of Industrial Electronics Society*, vol. 4, pp. 2249-2254, 2000.

[16] S. Tadokoro, M. Hayashi, Y. Manabe, Y. Nakami, and T. Takamori, "Motion planner of mobile robots which avoid moving human obstacles on the basis of stochastic prediction," *Proc. of the IEEE International Conference on Systems, Man, and Cybernetics*, vol. 4, pp. 3286-3291, 1995.

[17] T. Suwannatat and K. Chamnongthai, "Moving obstacle path detection for mobile robot," *Proc. of 5th International Symposium on Signal Processing and its Applications*, pp. 583-586, 1999.

[18] D. Nair and J. K. Aggarwal, "Detecting unexpected moving obstacles that appear in the path of a navigating robot," *Proc. of the IEEE International Conference on Image Processing*, pp. 311-315, 1994.

[19] L. E. Parker and B. A. Emmons, "Cooperative multi-robot observation of multiple moving targets," *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 2082-2089, 1997.

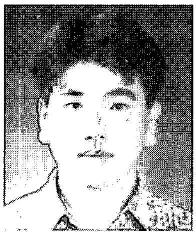
[20] T. Aoki, T. Oka, T. Suzuki, and S. Okuma, "Acquisition of optimal action selection to avoid moving obstacles in autonomous mobile robot," *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 2055-2060, 1996.

[21] Y. Zhang and J. Wang, "Obstacle avoidance for kinematically redundant manipulators using a dual neural network," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 34, no. 1, pp. 752-759, 2004.

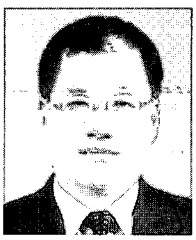
[22] S. X. Yang, T. Hu, X. Yuan, P. X. Liu, and M. Meng, "A neural network based torque controller for collision-free navigation of mobile robots," *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 13-18, 2003.

[23] B. H. Lee and C. S. G. Lee "Collision-free motion planning of two robots," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. SMC-17, no. 1, pp. 21-31, 1987.

[24] M. Erdmann and T. Lozano-Perez, "On multiple moving objects," *Proc. of the IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 1419-1424, 1986.



Seung-Hwan Park received the B.S. degree in Electrical Engineering from KAIST, Korea in 1995 and M.S. degrees in Electrical Engineering from Seoul National University, Korea in 1998, and now he is pursuing a Ph.D. degree in the School of Electrical Engineering at Seoul National University, Korea. His research interests include mobile robot control, sensor fusion and path planning.



Beom-Hee Lee received the B.S. and M.S degrees in Electronics Engineering from Seoul National University, Seoul, Korea in 1978 and 1980, respectively, and the Ph.D. degree in Computer, Information and Control Engineering from the University of Michigan, Ann Arbor, in 1985. From 1985 to 1987 he was with the School of Electrical Engineering at Purdue University, West Lafayette, IN, as an Assistant Professor. He joined Seoul National University in 1987, where he is currently a Professor of the School of Electrical Engineering. Since 2004 he is the Fellow of IEEE Robotics and Automation Society.