Robotica (2006) volume 24, pp. 295–303. © 2005 Cambridge University Press
doi:10.1017/S0263574705002122 Printed in the United Kingdom

Analysis of robot collision characteristics using the concept of the collision map

Seung Hwan Park and Beom Hee Lee*

(Received in Final Form: June 20, 2005, first published online 17 November 2005)

SUMMARY

Robot collision characteristics are analyzed by using the idea of the collision map. This analysis consists of the translations of the collision region on the collision map and they correspond to parallel movements of the original robot path. These translations are investigated in several cases and applied to general situations in which two robots are moving or working in a common workspace. Also, the collision characteristics are analyzed for a few special situations where the analysis of collision characteristics is crucial and the resultant solution for collision avoidance can be obtained.

KEYWORDS: Collision characteristics; Collision map; Mobile robot; Motion planning.

I. INTRODUCTION

Industrial robots, such as assembling machines or welding machines, are not mobile generally and are installed to perform their tasks without any possibilities of collision with obstacles and/or other robots. But recently, the demands for personal or service robots have been increasing to meet the demands for a more convenient lifestyle. Thus robots must perform better than ever before. These personal or service robots must decide and execute their works independently to achieve their goals in an obstacle-existing environment having dynamic obstacles as well as stationary obstacles.

Nowadays, it is becoming more frequent to operate many robots simultaneously in a common workspace for the effective task execution. In these cases, not only is the collision avoidance of robot with obstacles important but also the collision avoidance among the robots because a robot can be treated as an obstacle from the viewpoint of another robot. Thus, many studies have proposed strategies for robots to complete their tasks without any collision with obstacles and/or other robots.

These studies have been carried out in various fields. Especially, the methods using the geometric properties have given diverse and useful results. In addition, there have been much research based on probability, vision, behavior-based method and fuzzy logic. Direct controller design method and multivalue codes method using binary codes have been studied also.

In the studies applying the geometric properties, 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 of iterated forecast and planning imitated this usual human behavior. In this method, a robot selects the optimal path among many paths and tests whether or not it will collide with any other obstacles in the selected path. Yamamoto et al.[4] and Fiorini et al.[5] researched the avoidance against a dynamic obstacle by using the concept of velocity obstacle. In this method, the velocity vector set after some time is presented from the present position of the robot with consideration of the robot dynamics. Then, this velocity vector set was used to test whether or not this set was included in the velocity obstacle which was the possible area of collision. If the set was included in the velocity obstacle, there was a possibility of collision in this situation. Then, the robot must wait some time, or the path of the robot has to be changed. Besides, Czarnecki[6] embodied the 3-dimensional collision map and Angel P. del Pobil et al.[7,8] modeled robots and obstacles as combinations of spheres to detect collision.

Miura et al.[9] and Miyata et al.[10] used the probability method to estimate the waiting time of the robot for a predicted collision; this time was used in collision avoidance. They first designated the robot path selection probability according to the path selection of the moving obstacles, then used this probability to calculate the expected time to destination for each path to find the optimal path, and finally moved the robot to its goal through the selected optimal path. Tadokoro et al.[11] statistically predicted the human motion that could avoid collision with a human obstacle of uncertain motions; they used GA (Genetic Algorithm) to obtain the optimal movement.

Suwannatat et al.[12] and Nair et al.[13] performed polar transform by using a vision system. They observed the changes in images with time to extract the information about the moving obstacles. This information was used to avoid the dynamic obstacles.

There have been studies based on behavior-based method or fuzzy logic which tries to reduce simple, repetitive mechanical motions to make robot motions close to human motions. Parker et al.[14] researched behavior-based method which made the robot execute predefined motions if it recognized a pertinent situation. Also they made the system robust to changes by inserting some motivations of behavior in each robot. Aoki et al.[15] used the steering control inputs

* Corresponding author: E-mail: bhlee@asri.snu.ac.kr
ASRI, School of Electrical Engineering, Seoul National University, San 56-1, Shinrim-Dong, Kwanak-Gu, Seoul, (South Korea), 151-742

and the velocity control inputs based on fuzzy logic so that the robot can select the optimal behavior automatically. These control inputs were finely adjusted or combined by the reinforcement-learning algorithm. As the more unusual case, Alain[16] planned the robot motion amidst moving obstacles using the multivalue codes based on binary codes and the Karnaugh board.

If the number of dimensions, robots, or dynamic obstacles increases, the calculation load increased greatly in the most of the above methods. In contrast, the method of the collision map[17] has small 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 agents (robots or obstacles) increases. In this paper, we analyze the collision characteristics between two robots by using the graph of the collision map. The translations of the path are considered as path modifications in this analysis. These path translations are applicable to general situations to control robots in which two robots move and/or work in a common workspace. In addition, we apply this method of translations to some special situations to understand why collision avoidance is difficult in these situations and how collision avoidance can be achieved with the method of translations.

This paper is organized as follows: Section II presents the idea of the collision map. In section III, the characteristics of robot collision are analyzed with the translations of the collision region in the collision map. In section IV, a few special situations are investigated as applications of this analysis. Finally, section V presents concluding remarks.

II. COLLISION MAP AND COLLISION AVOIDANCE

We introduce the idea of collision map for collision avoidance. The following subsection presents some assumptions on mobile robots and briefly discusses the concept of collision map for two robots. The potential collision between two robots is predicted by using TLVSTC (traveled length versus servo time curve) and the collision region.

II.A. Basic assumptions

- The mobile robot is a uni-directional robot. This means that the robot moves in one direction and cannot change its direction abruptly. But the robot is free from the non-holonomic condition. Thus, the robot can change its direction slowly in the stationary state as well as the dynamic state. And the robot moves in a straight line path.
- The mobile robot is treated as a circle or sphere to simplify the problem; however other geometries can be used to represent the mobile robot. For example, a cylindrical model or a hexahedral model can be used. But the mathematical expressions of these models become position-dependent in the global coordinate frame and large computational load is generated. In contrast, a sphere model is rotationally invariant and is completely defined by its radius and the coordinates of its center; thus, the complexity of the detection of collisions can be reduced.

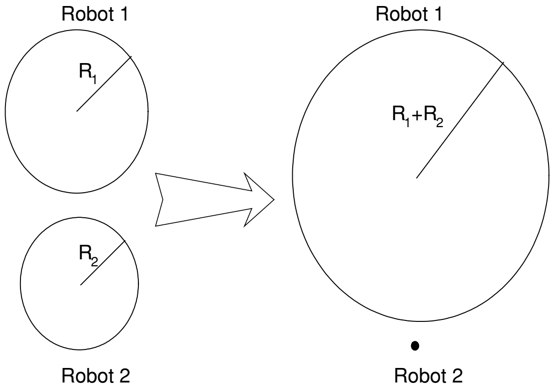

Fig. 1. Simplifications of Two Robots.

- The mobile robot has its own velocity profile.[18] The maximum velocity and acceleration are designated in this profile. Equations (1) and (2) show these limits:

$$|v(k)| \le \varepsilon^v; \varepsilon^v > 0 \tag{1}$$

$$|a(k)| \le \varepsilon^a;\ \varepsilon^a > 0 \tag{2}$$

where $v(k)$ and $a(k)$ are the velocity and acceleration of a robot. Additionally, ε^v and ε^a are the limit vectors for the velocity and acceleration of a robot, respectively. If a robot reaches its maximum velocity designated in the velocity profile, it maintains its current velocity until negative acceleration is imposed. Especially, we assume in this paper that the robot velocity profile is a type of trapezoid; that is, it has a constant acceleration and no time is consumed in the change of acceleration.

II.B. Collision map

The robot that has 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 as shown in Fig. 1. Because robot 1 has 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 the two robots move along linear paths, as shown in Fig. 2. The method of collision map may be applicable to arbitrary path shapes. But in this paper, robot paths are restricted to linear paths for simplicity. These two robots have a potential for 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 robot 1 is called the ‘collision length’. In Fig. 2, the portion between $\lambda_1(k)$ and $\lambda_2(k)$ is the collision length. The existence of this overlapped part was examined at every instant of the servo sampling time. These collision lengths are collected to construct the ‘collision region’. If the traveled length versus servo time curve (TLVSTC) of robot 2 meets this region, it indicates that the two robots will collide under the original trajectories as shown in Fig. 3. In this figure, the

http://dx.doi.org/10.1017/S0263574705002122

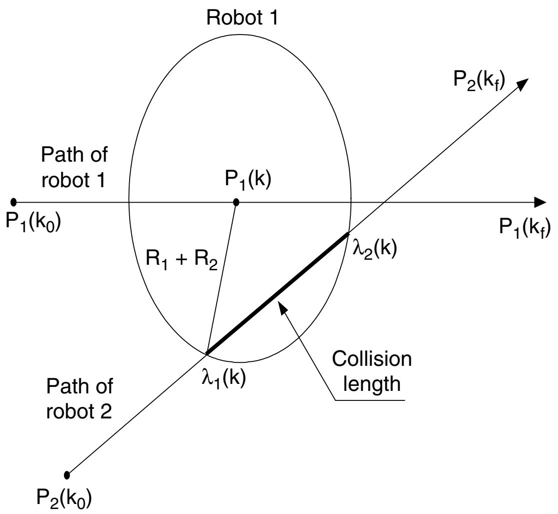

Fig. 2. Paths of Two Robots.

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 by using Fig. 2. In Fig. 2, $p_1(k)$ 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:

$$p_2(k) = p_2(k_0) + \lambda(p_2(k_f) - p_2(k_0)) \tag{3}$$

where $0 \leq \lambda \leq 1$, $p_2(k_0)$ 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 from $p_1(k)$ to the path of robot 2 in Eq. (3) is less than or equal to the radius of robot 1, $(\mathrm{R}_1 + \mathrm{R}_2)$. Thus we first solve the following equation.

$$(R_1 + R_2)^2 = \|p_1(k) - p_2(k)\|^2 \tag{4}$$

If we replace $p_2(k)$ in Eq. (4) with Eq. (3), then we have

$$\begin{aligned}(R_1 + R_2)^2 = {}& \{p_1(k) - p_2(k_0) - \lambda(p_2(k_f) - p_2(k_0))\}^T \\ & \bullet \{p_1(k) - p_2(k_0) - \lambda(p_2(k_f) - p_2(k_0))\}.\end{aligned} \tag{5}$$

More explicitly,

$$\begin{aligned}(R_1 + R_2)^2 = {}& \|p_1(k) - p_2(k_0)\|^2 - 2\lambda(p_1(k) - p_2(k_0))^T \\ & \bullet (p_2(k_f) - p_2(k_0)) + \lambda^2\|p_2(k_f) - p_2(k_0)\|^2\end{aligned} \tag{6}$$

Eq. (6) 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 with the path of robot 2 or starts leaving the path of robot 2; it has two real solutions, which means that robot 1 encroaches on the path of robot 2 and two robots may collide.

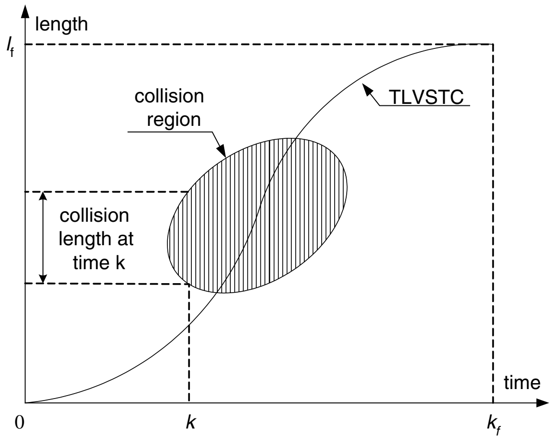

Fig. 3. TLVSTC and the Collision Region.

II.C. Collision avoidance

Two methods are proposed to avoid the collision detected by the collision map discussed earlier. In the beginning, we assumed that the robots cannot change their paths in these methods. To avoid collision, the TLVSTC of robot 2 should not meet the collision region in Fig. 3. 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 time. Thus, the collision box is introduced to solve the problem of collision avoidance. This is shown in Fig. 4.

In Fig. 4, 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. In contrast, l_s and l_e are the minimum and maximum values of the boundary values of the collision length in the collision region. The above parameters are used to find the coordinates of the edges of the collision box and these coordinates are used to control the trajectory of robot 2 so that robot 2 avoids collision with robot 1.

***a)* Time delay.** This is the method that delays the start time of robot 2 to avoid the collision for the value which

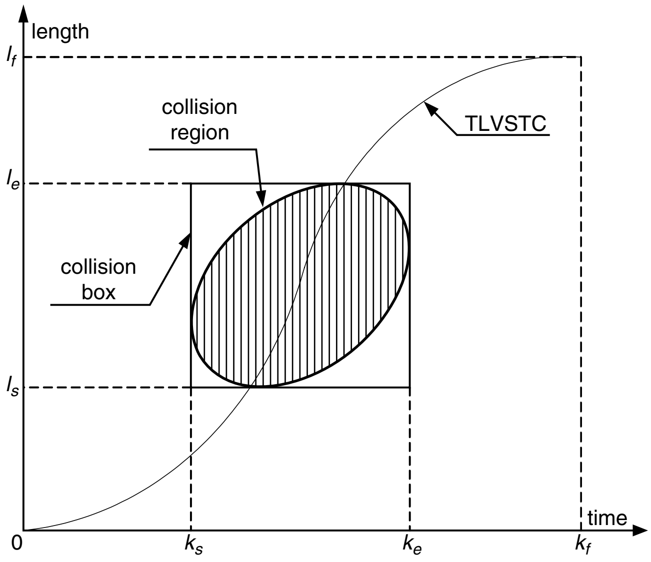

Fig. 4. Collision Map with Collision Box.

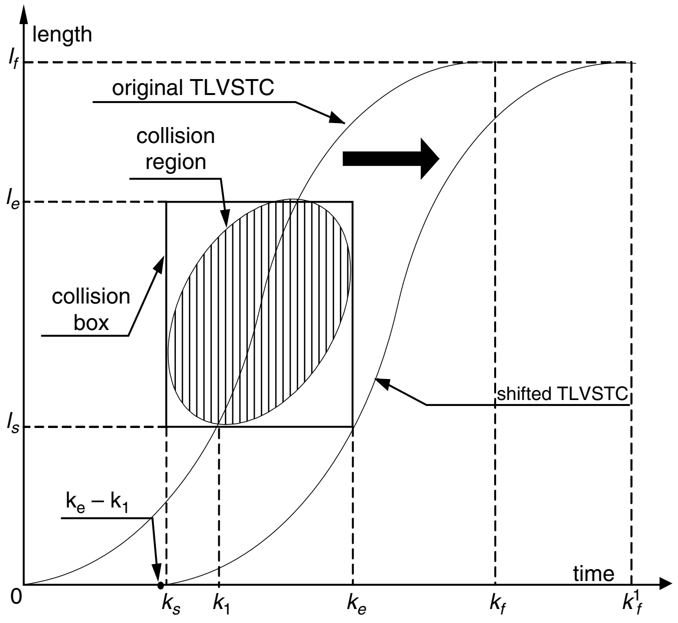

Fig. 5. Collision Avoidance with Time Delay.

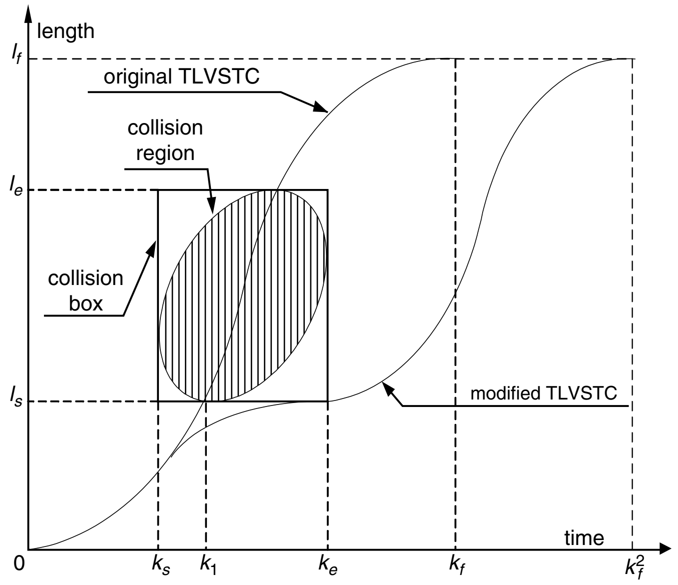

Fig. 6. Collision Avoidance with Speed Reduction.

is the difference between k_e and k_1 as shown in Fig. 5. Consequently, robot 2 reaches its goal at time k_f^1. This is the time delayed for k_e-k_1 from k_f.

***b*) Speed reduction.** In contrast with the case that applied the method of time delay, the case in which all robots are assumed to start moving simultaneously was considered. Here, the moving speed of robot 2 changes as it tries to avoid collision. The velocity profile of robot 2 is modified so that the trajectory of robot 2 does not touch the collision region. Among the various methods of velocity profile modification, one is shown in Fig. 6, where there is an instant when the velocity of robot 2 becomes zero as it proceeds. Thus, this method of speed reduction results in lower performance of the robots with respect to arrival time than that of time delay, generally.

Table I. Analysis of Collision Characteristics.

Part	Translation of region	Translation of path	Translated length
A	Right & left	Along path of robot 1	Related to robot 1
B	Up & down	Along path of robot 2	Related to robot 2
C	Upper-left & lower-right	Toward end of robot 1 & Away end of robot 1	Related to both robots
D	Upper-right & lower-left	Away start of robot 1 Toward start of robot 1	Related to both robots

III. COLLISION CHARACTERISTICS THROUGH TRANSLATIONS OF COLLISION REGION

In this section, we investigate the collision characteristics of the robot using the translation of the collision region. The translation of the collision region means the translation of the path of a robot. When the TLVSTC of robot 2 crosses with the collision region once, there exists a collision in the original trajectories of the robots. The change or translation of robot 2 path was not considered for collision avoidance. This is a suitable assumption for industrial robots because their path is 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. Here, we consider the translation of the collision region. From now on, we only consider the collision box as the collision region. In this section, we move the collision region of the collision map to various directions. Moving directions are classified into 4 categories and these are abbreviated in Table I. We assume that robot 2 path should change for collision avoidance because of its lower priority. The translation of the path does not mean the global change of the length and/or shape of the path; it only means the shift of the path for the case of mobile service robots. Thus the same TLVSTC in the collision map can be used without change.

III.A. Right and left translational case

First, we translate the collision region to the right and the left direction. These translations are represented by case 1 and case 2 in Fig. 7, respectively. 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 robot 2 path that overlaps with robot 1 as shown in Fig. 2. At some time, where robot 1 locates in the path of robot 2 decides the position of the collision length at that time. In Fig. 7, case 1 has the collision length of the same position ($l_1 \sim l_2$) at the different time k_2 as that of the original case at time k_1. Also, case 2 has the same collision length at time k_3 as that of the original case at time k_1. The time differences are $k_2 - k_1$ and $k_1 - k_3$. The distances that robot 1 can move at these time differences are d_1 and d_2, as presented in Fig. 7. Thus the path of robot 2 must translate as much as d_1 away from the start point of robot 1 for case 1 or as much as d_2 toward the start point

http://dx.doi.org/10.1017/S0263574705002122

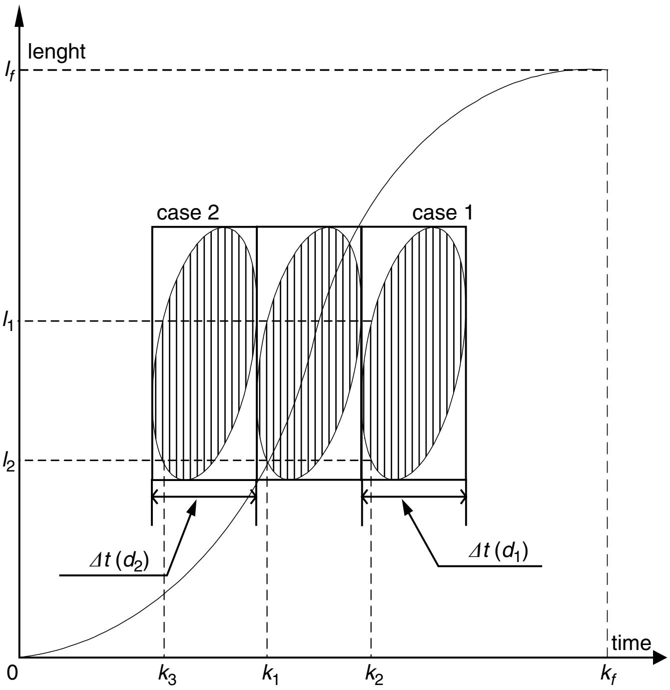

Fig. 7. Right and Left Translations of the Collision Region ($\Delta t(d_i)$ indicates the travel time required to move the distance d_i on the robot path).

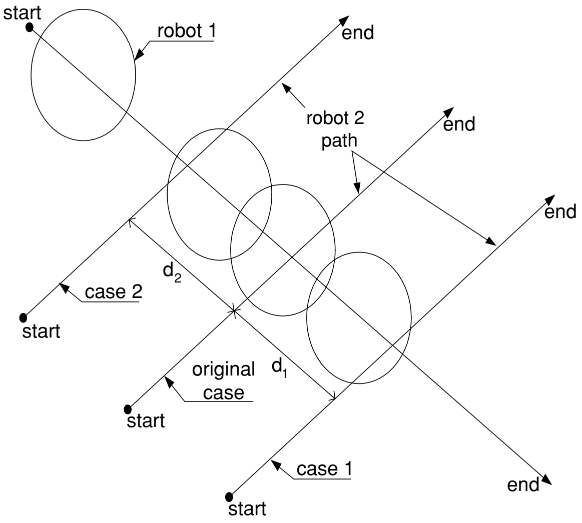

Fig. 8. Translations of Robot 2 Path in Cases 1 and 2.

of robot 1 for case 2 to translate the collision regions. These translations of the path are shown in Fig. 8.

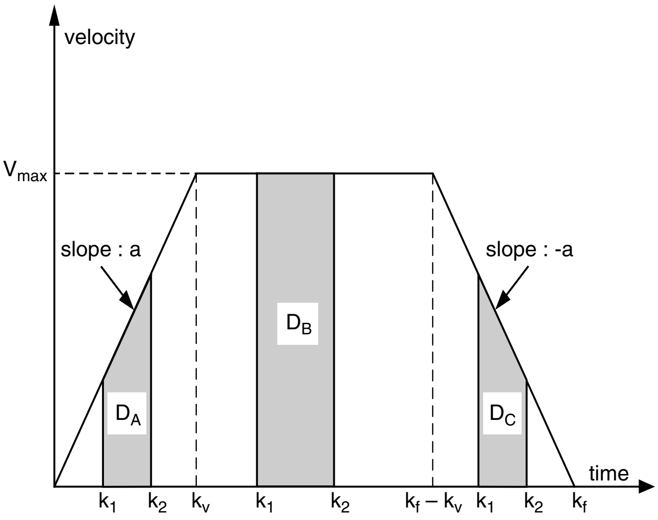

Fig. 9. Calculation of the Distance.

The distances d_1 and d_2 are calculated by using the concept of Fig. 9. There are three possibilities where the $k_2 - k_1$ is located in the graph with respect to the velocity profile of a robot. In Fig. 9, D_A is the distance related to the situation in which the time difference is in the increasing velocity section. If the time difference $k_2 - k_1$ is in the regular velocity section, D_B represents the distance related to this situation, and finally D_C is the distance calculated when the time difference $k_2 - k_1$ is located in the decreasing velocity section. If the time difference is laid 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. Because the graph in Fig. 9 is the velocity-time graph, the moving distance is calculated as the area between k_1 and k_2. The results are shown in Eqs. (7) to (9) for D_A to D_C, respectively. In these equations, a is the acceleration of a robot, $v_{\max}$ is the maximum velocity of a robot, k_v is the time when the velocity of a robot reaches its maximum value, and k_f is the arrival time of a robot to its goal.

$$D_A = \frac{1}{2}a\left(k_2^2 - k_1^2\right) \tag{7}$$

$$D_B = v_{\max}(k_2 - k_1) = ak_v(k_2 - k_1) \tag{8}$$

$$D_C = \frac{1}{2}a(k_2 - k_1)\{(k_f - k_1) + (k_f - k_2)\} \tag{9}$$

III.B. Up and down translational case

Now we translate the collision region to the vertical direction, up and the down. These are case 3 and case 4 as shown in Fig. 10. The collision region located at the center indicates the original case. Unlike the above cases, case 3 has the collision length in the different position ($l_3 \sim l_4$) at the same time k_1 as the original case. Also, case 4 has the collision length in the different position ($l_5 \sim l_6$) at the same time. The differences of the positions are d_3 and d_4. When the collision length occurs at the same time, it means that robot 1 starts crossing the path of robot 2 at the same time. When the collision length exists at a different position, it means that robot 1 encroaches on the path of robot 2 at the different point. Thus, the path of robot 2 must translate forward as much as d_3 along the path of robot 2 for the case 3 or backward as much as d_4 along the path of robot 2 for the case 4 to translate the collision regions. These translations of the path are shown in Fig. 11. In this figure, the paths of the robot 2 overlap with each other in some region.

III.C. Upper-left and lower-right translational case

We translate the collision region to the upper-left and the lower-right direction corresponding to case 5 and case 6, respectively, as shown in Fig. 12. They are the mixtures of

http://dx.doi.org/10.1017/S0263574705002122

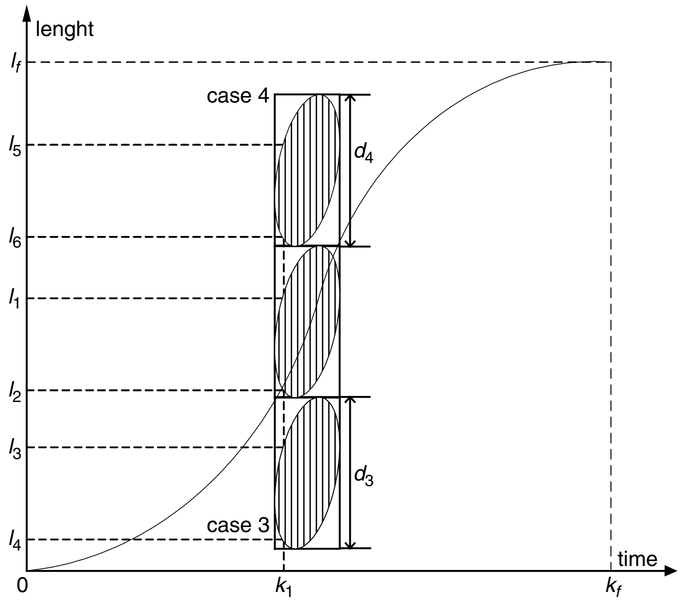

Fig. 10. Up and Down Translations of the Collision Region.

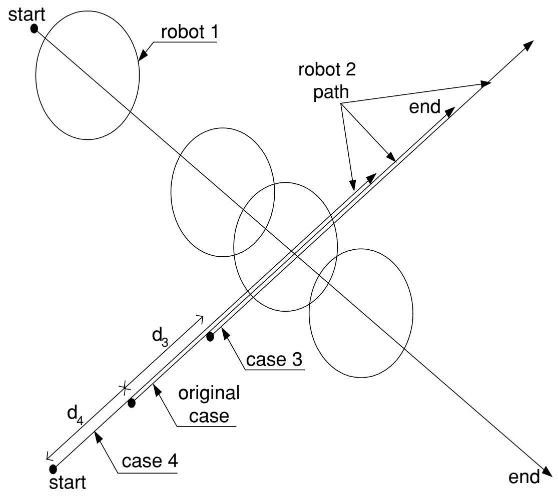

Fig. 11. Translations of Robot 2 Path in Cases 3 and 4.

cases 1, 3, and 2, 4. The collision region moves along the TLVSTC from case 1 to case 3 or from case 2 to case 4. Case 5 designates that the collision region locates in the middle position between cases 1 and 3. Similarly, the collision region locates between cases 2 and 4 in case 6. Thus, these cases are different along the axis of time as well as the axis of length from the original case. The distances due to the difference along the axis of time are d_{51} and d_{61}, and the distances along the axis of length are d_{52} and d_{62}. The collision region translates as much as d_{51} to the same direction as case 1 and as much as d_{52} to the direction of cases 3 in the case 5. The similar translation occurs in case 6 as shown in Fig. 12. The translations of the collision region are reflected in the translations of the path of robot 2. This is shown in Fig. 13.

III.D. Upper-right and lower-left translational case

Finally, the collision region is translated to the upper-right and the lower-left direction, corresponding to case 7 and case 8, respectively, as shown in Fig. 14. The collision region moves from case 2 to case 3 or from case 1 to case 4. Actually, these cases are not worthy of discussing because there is a collision at each case. However, we want to observe the feature of the change of robot 2 path due to the change of collision region. Thus, we refer to these cases in this section. Like cases 5 and 6, these cases are also different along the axis of time as well as the axis of length from the original case. The collision region translates as much as d_{71} to the same direction as case 2 and as much as d_{72} to the direction of case 3 in case 7. The similar translation occurs in case 8, as shown in Fig. 14, and the translations of robot 2 path are shown in Fig. 15.

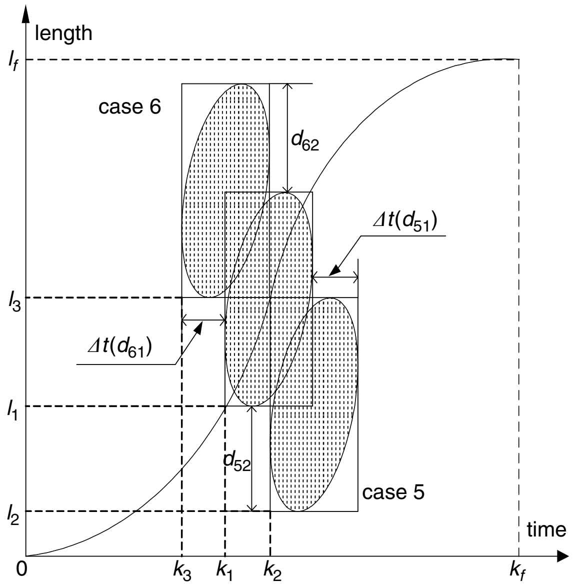

Fig. 12. Upper-Left and Lower-Right Translations of the Collision Region ($\Delta t(d_{ij})$ indicates the travel time required to move the distance d_{ij} on the robot path).

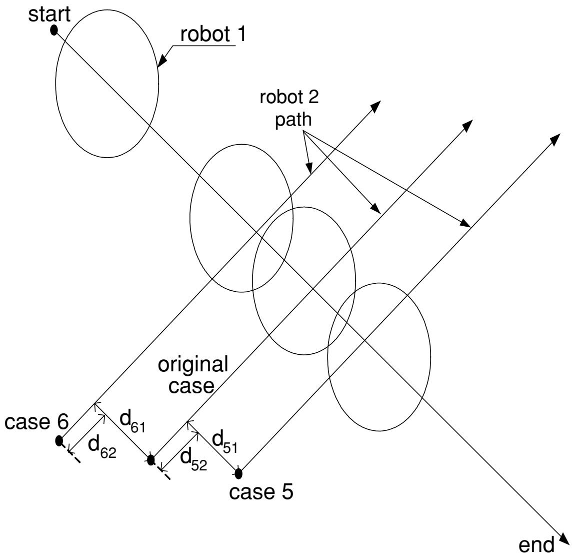

Fig. 13. Translations of Robot 2 Path in Cases 5 and 6.

IV. INVESTIGATIONS OF SOME SPECIAL SITUATIONS

In most general situations, the method of time delay and/or speed reduction can be used successfully for collision

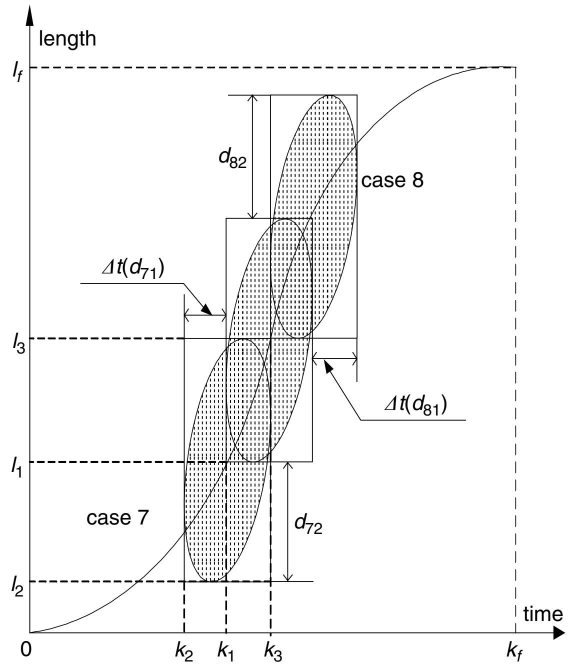

Fig. 14. Upper-Right and Lower-Left Translations of the Collision Region ($\Delta t(d_{ij})$ indicates the travel time required to move the distance d_{ij} on the robot path).

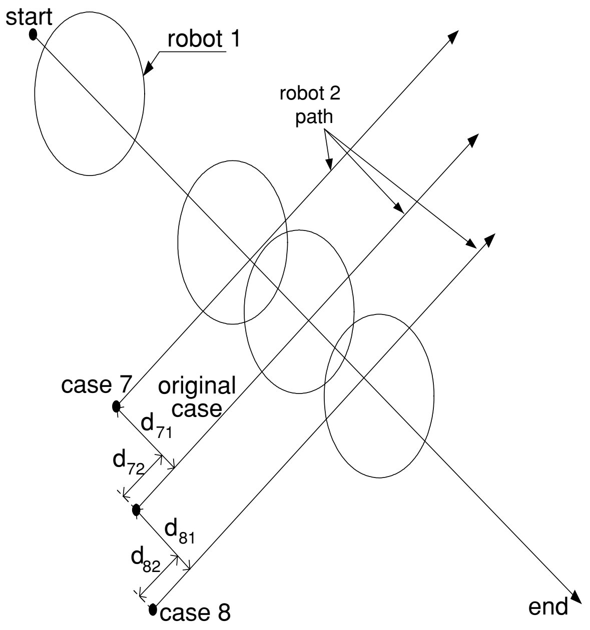

Fig. 15. Translations of Robot 2 Path in Cases 7 and 8.

avoidance. But in cases with difficult conditions, these methods are not sufficient for collision avoidance. We now investigate these special situations with the idea of collision map. In this section, the circle written by R1 in several figures represents robot 1, the higher priority robot, and that written by R2 means robot 2, the lower priority robot.

IV.A. Overlapping path situations for both robots

In this subsection, we analyze the overlapping path situations for both robots when robot 1 path covers robot 2 path completely. The collision regions are generated so that they cut through the whole collision map in these situations.

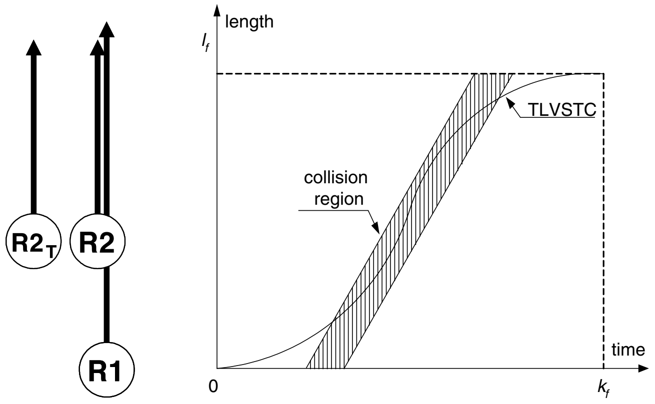

Fig. 16. Situation that Robot 1 is Following Robot 2 and Related Collision Map.

***a*) When robot 1 is following the robot 2.** This situation is illustrated in Fig. 16. In this situation, it is assumed that robot 1 is faster than robot 2 so that collision occurs in the original paths of both robots. The collision region in Fig. 16 spreads across the whole area of collision map, and there is a time constraint which ensures the collision between the robots after some time. Thus, the methods that increase the total arrival time of robot 2 such as time delay and/or speed reduction cannot be applied to solve this situation. Also, we note that if the angle made by the paths of both robots is 0° or 180°, then case 1 in Fig. 8 coincides with case 3 or case 4 in Fig. 11, respectively. Similarly, case 2 coincides with case 4 or case 3, respectively. Thus, the path translation methods can be applied only if the distance of translation is large enough that the start point of robot 2 is located behind the start point of robot 1 or ahead of the end point of robot 1. We introduce another path modification, in which the robot 2 path translates in the direction perpendicular to the robot 1 path with the extent produced by adding the radiuses of two robots. With this translation, there is no crossing between the paths of both robots and the collision region disappears in the collision map. The circle written by $R2_T$ is the translational result.

***b*) When robot 1 is moving oppositely to robot 2.** This situation is illustrated in Fig. 17. In this situation, both robots move in the opposite direction so that collision occurs in the original paths. The collision region in Fig. 17 cut through the area of the collision map and there is also a time constraint that ensures collision after some time. Thus the method of time delay and/or speed reduction cannot solve this situation. Also the path translation methods discussed before can be applied only if the extent of translation is large enough that the start point of robot 2 is located behind the start point of robot 1 or the end point of robot 2 is located ahead of the end point of robot 1. We apply the above method of path modification in which the robot 2 path is translated in the direction perpendicular to the robot 1 path with the extent produced by adding the radiuses of both robots. With this translation, there is no collision region in the collision map. The circle written by $R2_T$ is the translated result of robot 2.

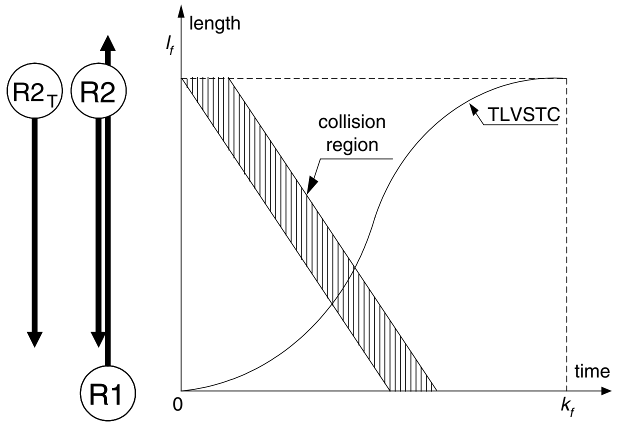

Fig. 17. Situation that Robot 1 is Moving Oppositely to Robot 2 and Related Collision Map.

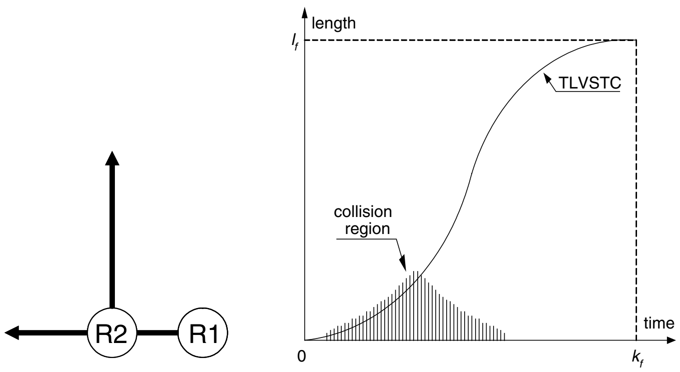

Fig. 18. Situation that Robot 1 is Passing through the Start Point of Robot 2 and Related Collision Map.

IV.B. Passing through the start point of robot 2 or the end point of robot 1

***a*) When robot 1 is passing through the start point of robot 2.** This situation occurs when robot 1 locates very close to the start point of robot 2 as shown in Fig. 18 so that both robots collide before robot 2 escapes from robot 1 path. Since robot 1 has higher priority, the method of time delay and/or speed reduction cannot provide a solution. Thus, the method of path translation must be used. As shown in Figs. 8 and 11, case 1 or case 3 is more useful for collision avoidance because a small extent of translation is needed in each case. If case 1 is applied, the robot 2 path and the collision map change as shown in Fig. 19. In this figure, the circle written by $R2_T$ is the translated result of robot 2.

***b*) When robot 2 is passing through the end point of robot 1.** Figure 20 shows the situation in which the end point of robot 1 locates in robot 2 path. Robot 1 stops at its end point, and robot 2 views it as an unexpected obstacle. As shown in Fig. 20, the related collision map is expanded along the time axis, and thus, the method of time delay and/or speed reduction is not adequate to apply for collision avoidance. On the other hand, the method of path translation can be useful in this situation. Case 1 or case 3 is proper for the collision avoidance. The result of applying case 3 is shown in Fig. 21. In this figure, the circle written by $R2_T$ is the translated result of robot 2.

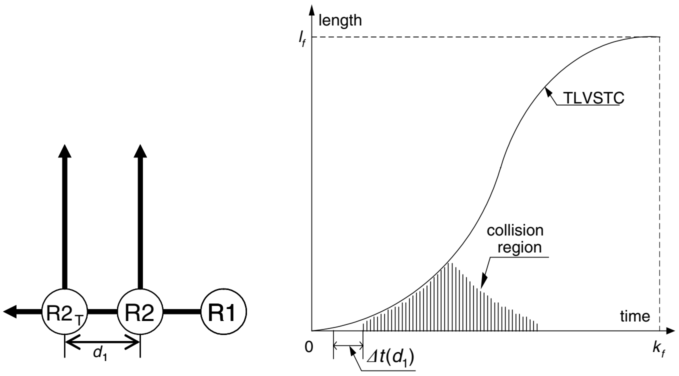

Fig. 19. Result after Applying the Translation of Case 1. ($\Delta t(d_i)$ indicates the travel time required to move the distance d_i on the robot path).

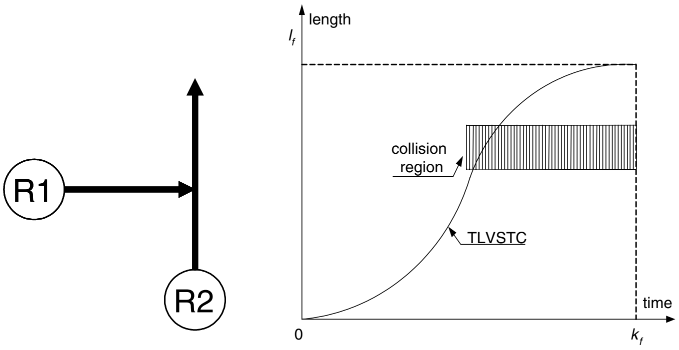

Fig. 20. Situation that Robot 2 is Passing through the End Point of Robot 1 and Realted Collision Map.

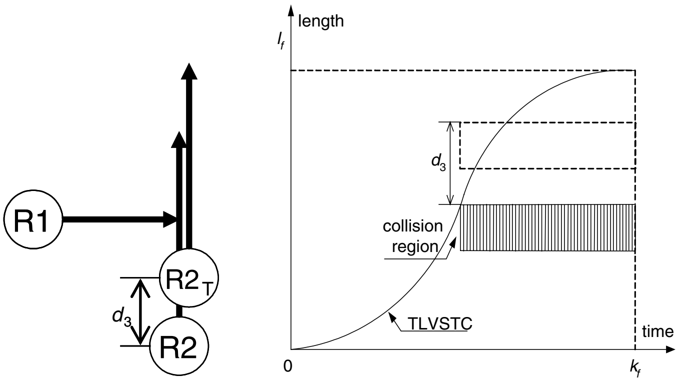

Fig. 21. Result after Applying the Translation of Case 3.

V. CONCLUSIONS

An analysis of robot collision characteristics was discussed. Circles modeled two robots used in this paper, and then they were abbreviated as a circle and a point robot. Various translations of collision region in the collision map were identified and they were used to predict collision between robots. Cases 1 to 8 showing these translations were classified and analyzed according to their conditions and characteristics. As mentioned earlier, a mobile service robot has relatively large workspace, and may perform better in some paths. These paths for the robot can be determined easily by expanding our analysis to more cases. Also, some special situations are investigated with the collision map. The conventional methods of collision avoidance such as time delay or speed reduction cannot be applied to solve these situations. Thus, the methods of path translations or other path modification methods were used for these special situations. In future work, we will analyze the cases in which the robot 1 velocity changes.

Acknowledgment

This work was supported partially by the BK21 Information Technology at Seoul National University, Automation and Systems Research Institute (ASRI) in Seoul National University, the HWRS-ERC at KAIST, and CIR, one of the 21st Century Frontier R&D Program funded by the Ministry of Science and Technology of Korea.

References

1. Takashi Tsubouchi and Suguru Arimoto, "Behavior of a Mobile Robot Navigated by an "Iterated Forecast and Planning" Scheme in the Presence of Multiple Moving Obstacles", *Proceedings of the IEEE International Conference on Robotics and Automation* (1994) pp. 2470–2475.
2. Takashi Tsubouchi, Soichiro Kuramochi and Suguru Arimoto, "Iterated Forecast and Planning Algorithm to Steer and Drive a Mobile Robot in the Presence of Multiple Moving Objects", *Proceedings of the IEEE International Conference on Intelligent Robots and Systems* (1995) pp. 33–38.
3. Takashi Tsubouchi, Akisada Hirose and Suguru Arimoto, "A Navigation Scheme with Learning for a Mobile Robot among Multiple Moving Obstacles", *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems* (1993) pp. 2234–2240.
4. Motoji Yamamoto, Masanori Shimada and Akira Mohri, "On-Line Navigation of Mobile Robot under the Existence of Dynamically Moving Multiple Obstacles", *Proceedings of the IEEE International Symposium on Assembly and Task Planning* (1995) pp. 13–18.
5. Paolo Fiorini and Zvi Shiller, "Motion Planning in Dynamic Environments using the Relative Velocity Paradigm", *Proceedings of the IEEE International Conference on Robotics and Automation* (1993) pp. 560–565.
6. C. A. Czarnecki, "Collision Free Motion Planning for Two Robots Operating in a Common Workspace", *IEEE International Conference on Control* (1994) **Vol. 2**, pp. 1006–1011.
7. Miguel Perez-Francisco, Angel P. del Pobil and Begona Martinez, "Fast Collision Detection for Realistic Multiple Moving Robots", *Proceedings of 8th International Conference on Advanced Robotics* (1997) pp. 187–192.
8. Angel P. del Pobil, Miguel A. Serna and Juan Llovet, "A New Representation for Collision Avoidance and Detection", *Proceedings of the IEEE International Conference on Robotics and Automation* (1992) pp. 246–251.
9. Junichi Miyata, Toshiyuki Murakami, and Kouhei Ohnishi, "An Approach to Tracking Motion of Mobile Robot for Moving Object", *26th Annual Conference of the IEEE of Industrial Electronics Society* (2000) **Vol. 4**, pp. 2249–2254.
10. Jun Miura, Hiroshi Uozumi and Yoshiaki Shirai, "Mobile Robot Motion Planning Considering the Motion Uncertainty of Moving Obstacles", *IEEE Transactions on Systems, Man, and Cybernetics* **4**, 692–697 (1999).
11. Satoshi Tadokoro, Masaki Hayashi, Yasuhiro Manabe, Yoshihiro Nakami and Toshi Takamori, "Motion Planner of Mobile Robots which Avoid Moving Human Obstacles on the Basis of Stochastic Prediction", *IEEE International Conference on Systems, Man, and Cybernetics* (1995) **Vol. 4**, pp. 3286–3291.
12. Tassaphan Suwannatat and Kosin Chamnongthai, "Moving Obstacle Path Detection for Mobile Robot", *5th International Symposium on Signal Processing and its Applications* (1999) pp. 583–586.
13. Dinesh Nair and J. K. Aggarwal, "Detecting Unexpected Moving Obstacles that Appear in the path of a Navigating Robot", *Proceedings of the IEEE International Conference on Image Processing* (1994) pp. 311–315.
14. Lynne E. Parker and Brad A. Emmons, "Cooperative Multi-Robot Observation of Multiple Moving Targets", *Proceedings of the IEEE International Conference on Robotics and Automation* (1997) pp. 2082–2089.
15. Takeshi Aoki, Toshiaki Oka, Tatsuya Suzuki and Shigeru Okuma, "Acquisition of Optimal Action Selection to Avoid Moving Obstacles in Autonomous Mobile Robot", *Proceedings of the IEEE International Conference on Robotics and Automation* (1996) pp. 2055–2060.
16. Alain Pruski, "Multi-Robot Path Planning among Moving Obstacles using Multivalue Codes", *Proceedings of Rensselaer's Second International Conference on Computer Integrated Manufacturing* (1990) pp. 588–592.
17. B. H. Lee and C. S. G. Lee "Collision-Free Motion Planning of Two Robots", *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-17**, No. 1, 21–31 (1987).
18. B. H. Lee, "Constraints Identification in Time-Varying Obstacle Avoidance for Mechanical Manipulators", *IEEE Transactions on Systems, Man, and Cybernetics* 19, No. 1, 140–143 (1989).