The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Analytic Collision Anticipation Technology Considering Agents' Future Behavior

Jeong S. Choi, Gyuho Eoh, Jimin Kim, Younghwan Yoon, Junghee Park, and Beom H. Lee, *Fellow*, *IEEE*

Abstract— **This paper presents a collision anticipation method that shows when and where collisions will occur in configuration time space by considering the future behavior of agents. Previous solutions to collision anticipation have mainly focused on generating immediate reactive solutions to time -varying environments because of the high inaccuracy of sensors and a heavy computation burden. However, the recent rapid growth in sensor and estimation technology has led to a need for new systems that consider agents' future behavior explicitly. Based on this need, we formalized a mathematical approach to collision anticipation and proposed a tool, the 3-D triangular collision object(TCO), that informs a robot where and when collisions will occur over all possible heading angles. This formulation greatly reduced computation loads. Considering that this result is especially advantageous to fast moving robots, a full-speed collision-free (FSCF) motion planner is proposed based on the TCO. In real experiments, analytic solutions from the planner were modified to compensate for sensor error, and finally yielded safe motions for fast moving robots.**

Keywords– Analytic solution, obstacle avoidance, fast mobile robot

I. Introduction

OBSTACLE avoidance competence is essential to all kinds of mobile robotic systems, and the collision-free condition is the most important prerequisite for feasible robot motion. The obstacle avoidance problem is a canonical problem in robotics and a considerable number of studies have been conducted on the problem [1-12]. In spite of these efforts, it is still controversial to plan motion for robots safely and efficiently [13, 14] because of deficits in collision anticipation. Thus, it is challenging to develop a collision anticipation technique capable of improving safety by considering the future behavior of agents with real-time effectiveness for practical applications.

Collision anticipation has been considered as a prerequisite in developing effective path planners. Approaches to anticipation mainly depend on what kind of information is provided, how accurate the information is, and how long the information is valid. Thus, path planning is directly related to environmental conditions. According to the combination of types of obstacles and environments (static or dynamic obstacles, and completely or partially known environments), path planning can be classified into several cases. For the case of static obstacles, obstacle avoidance is a time-invariant problem, which makes the navigation problem very simple. Various methods for this case have been introduced, including C-space, visibility graph, and cell decomposition.

On the other hand, for the problem of moving obstacle avoidance, it is difficult to compute feasible paths and to find the optimal solution. In the past decade, three competitive approaches to collision anticipation have been introduced. Borenstein and Koren [1] presented the vector field histogram (VFH) method to simultaneously detect unknown obstacles and analyze the information statistically; their method was later extended into VFH+ and VFH*. Fiorini and Shiller [2] proposed another useful concept, the velocity obstacle (VO), representing a potential collision with respect to time-varying configuration space (velocity space) defined by a collision cone (CC) and a velocity vector. This method has been employed in many subsequent papers. Fox *et al.* introduced the dynamic window (DW) approach [3], and variations include concepts such as global DW [4]. In this method, the kinetics of the robot is taken into account by searching a well-chosen velocity space; thus, a new motion direction is generated by applying an objective function to all the admissible candidates in the dynamic window.

Recent studies have tried to compensate for some limitations of the original studies, or extend them to real applications. Based on the VO, probabilistic VO (PVO) has been applied to uncertain environments [5] and generalized velocity obstacles (GVOs) for car-like robot navigation have been introduced in [6]. For multi-agent navigation, the concept of reciprocal VOs (RVOs) and a virtual plane using the VO were proposed in [7] and [8], respectively. The VFH approach has been modified into the vector polar histogram (VPH) approach using a laser scanner to consider the inaccuracy problem [9], and a comparative study involving the VPH has been shown in [10]. The nearness diagram (ND) [11] using the divide and conquer concept is also based on the VFH. In [12], the DW has been integrated into a focused D* algorithm for a partially unknown environment.

This work was supported in part by a Korea Science and Engineering Foundation NRL Program grant funded by the Korean government (No.R0A-2008-000-20004-0), and in part by the Brain Korea 21 Project, and in part by the Industrial Foundation Technology Development Program of MKE/KEIT [Development of Collective Intelligence Robot Technologies].

Jeong S. Choi, Gyuho Eoh, and Jimin Kim are with ASRI and Seoul National University (e-mail:{ jsforce2,torin00,geni0620}@snu.ac.kr).

Younghwan Yoon is now with LS Industrial Systems, Republic of Korea, (lucidite@gmail.com)

Junghee Park is now with the Department of Electronics Engineering and Information Science, Korea Military Academy, (grstorm7@gmail.com).

Beom H. Lee, a Fellow of IEEE, is now a Professor with the Department of Electrical Engineering, Seoul National University (bhlee@snu.ac.kr).

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

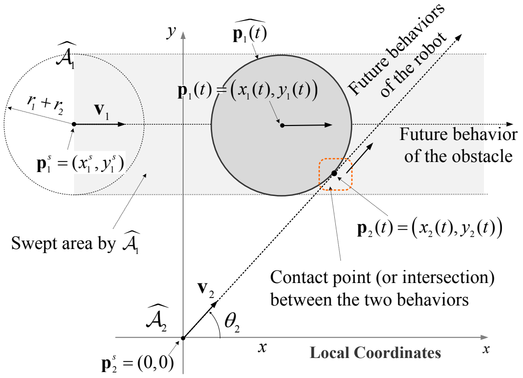

y
$\widehat{\mathbf{p}_1(t)}$
Future behaviors of the robot
$\widehat{\mathcal{A}}_1$
$\mathbf{p}_1(t) = (x_1(t), y_1(t))$
$r_1 + r_2$
$\mathbf{v}_1$
Future behavior of the obstacle
$\mathbf{p}_1^s = (x_1^s, y_1^s)$
$\mathbf{p}_2(t) = (x_2(t), y_2(t))$
Swept area by $\widehat{\mathcal{A}}_1$
Contact point (or intersection) between the two behaviors
$\mathbf{v}_2$
$\widehat{\mathcal{A}}_2$
θ_2
x
Local Coordinates
x
$\mathbf{p}_2^s = (0,0)$

Fig. 1.Two future behaviors of a robot and an obstacle under the current velocity. The behaviors are represented with respect to a local coordinate system (the notations used are presented in Table I) .

Fraichard [13] insisted that in most cases, and especially in a dynamic environment, safety is still not guaranteed, and proposed three criteria for safety motion: *dynamics*, *future behavior*, and *infinite time-horizon*. In the studies [13, 14] that included benchmarking, it was argued that since the VO, DW, and ND methods violate the second or the third criterion, safety in the presence of moving obstacles cannot be guaranteed, although the VO is superior to the others because it takes into account the future behavior of objects. Fraichard suggested that the future motion of moving obstacles should explicitly be considered to develop a novel collision avoidance scheme.

To this end, based on the VO scheme and the results in [13, 14], we developed a collision anticipation technology to improve safety in motion. This paper proposed a 3-D triangular collision object(TCO) which tells the robot where and when collisions will occur over all possible heading angles. In addition, we formalized a mathematical approach to the tool, instead of iterative computation methods, for real-time effectiveness. Our method exploits the fact that recent commercial sensors have the ability to widely detect the environment, update information quickly, and provide information with high accuracy (e.g., over 30m, 250°, 25 ms/scan, ± 30mm), and thus it is possible to make full use of the acquired information for collision anticipation.

The proposed analytic tool is especially advantageous for fast moving robots. This is because the sum of a robot's braking and traveled distance during reaction is often too large to ensure safety when it depends only on an immediate reaction strategy. In addition, immediate reaction methods make robots highly prone to frequently changing both their direction and speed, which may lead to significant impracticality or inefficiency during traveling time. These phenomena can be found in many previous solutions, although obstacles are moving with a constant linear velocity. Considering these points, we proposed a full-speed collision -free motion planner that is a simple example of the proposed method, producing stable velocity vectors for near collisions. To overcome limitations of the analytic solutions, the vectors were modified using measured sensor errors.

TABLE I
PHYSICAL MEANING OF NOTATIONS IN FIG.1AND FIG.2

Notation	Meaning
$\mathcal{A}_i$	An moving agent ($\mathcal{A}_1$:obstacle, $\mathcal{A}_2$: robot)
$\widehat{\mathcal{A}}_i$	$\mathcal{A}_i$ represented in configuration space (or c-space)
$r_i, \mathbf{v}_t, \mathbf{p}_i(t)$	Radius, velocity, and motion for $\mathcal{A}_i$, $v_i = \lVert \mathbf{v}_i \rVert$
$\hat{r}_i, \widehat{\mathbf{p}_i(t)}$	Radius and motion for $\mathcal{A}_i$ in c-space
$\mathbf{p}_i^s, \mathbf{p}_i^e$	Start and end (or goal) point for $\mathcal{A}_i$
$\mathbf{p}_{ij}^c(\theta_i)$	Intersection between $\mathbf{p}_i(t)$ with θ_i and $\widehat{\mathbf{p}_j(t)}$, $\mathbf{p}_i(t) \cap \widehat{\mathbf{p}_j(t)}$ For simple notation, $\mathbf{p}_{21}^c(\theta_2) = \mathbf{p}^c(\theta_2)$

This paper is organized as follows: Section II describes the important aspects of the overall problem. Sections III and IV present our formulation of the obstacle avoidance problem and the concept of FSCF motion, respectively. Experiments and comparisons are presented in Section V. Section VI describes the proposed approach, and our conclusions are given in Section VII.

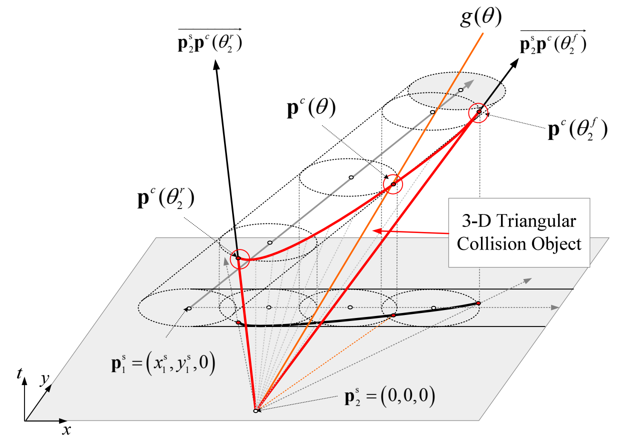
II. PROBLEM DESCRIPTION

The problem of avoiding moving obstacles can be decomposed into three sub-problems: when and where collisions will occur under current physical constraints (or velocities), defining a set of feasible motions, and which of the motions is optimal in terms of time, length, or energy. This paper focuses on the first problem. To formulate our problem, the following assumptions were made:

- *Assumption 1*: Agents (or robots and obstacles) are modeled as a circle in a two-dimensional(2-D) plane.
- *Assumption 2*: Agents are moving with a specific velocity, and the velocity is kept for a short time.
- *Assumption 3*: Robots are holonomic systems with kino-dynamic constraints.

Figure 1 illustrates the problem addressed in this paper. Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be a moving obstacle and a mobile robot, respectively. In the concept of configuration space, an obstacle is expanded as a new obstacle $\widehat{\mathcal{A}}_1$ meanwhile a robot is shrunk to a point $\widehat{\mathcal{A}}_2$. Under assumption1 and this concept, it follows that $\hat{r}_1 = r_1 + r_2$ and $\hat{r}_2 = 0$. For convenience, we use a local coordinates whose origin is the starting point of the robot and its x-axis is set to be the same as the direction of the obstacle. In the local coordinates, velocity vectors are denoted as $\mathbf{v}_1 = (v_1, 0)$ and $\mathbf{v}_2 = (v_2 \cos\theta_2, v_2 \sin\theta_2)$.

Let $\mathbf{p}_i(t)$ be a motion for $\mathcal{A}_i (i = 1, 2)$ which is a set of positions parameterized by time. In the coordinate, the local time is set to zero when the robot is located at the origin $\mathbf{p}_2^s$, and the boundary of $\widehat{\mathcal{A}}_1$ at time t is expressed as $\mathbf{p}_1^s(t)$. Under assumption2, it can be said that *a possible collision between two agents moving with a specific velocity exists if and only if* $\forall t \geq 0$ $\widehat{\mathbf{p}_1(t)} \cap \mathbf{p}_2(t) \neq \phi$. Our problem is to compute the potential collisions with the formulation, and represent them as a set of contact points indicating collision time (*when*) and heading angles (*where*).



$\overrightarrow{\mathbf{p}_2^s\mathbf{p}^c(\theta_2^r)}$
$g(\theta)$
$\overrightarrow{\mathbf{p}_2^s\mathbf{p}^c(\theta_2^f)}$
$\mathbf{p}^c(\theta)$
$\mathbf{p}^c(\theta_2^f)$
$\mathbf{p}^c(\theta_2^r)$
3-D Triangular Collision Object
$\mathbf{p}_1^s = (x_1^s, y_1^s, 0)$
$\mathbf{p}_2^s = (0,0,0)$
t
y
x

Fig. 2. Representation of two future behaviors in Fig. 1 with respect to configuration-time (CT) space and a 3-D triangular collision object (TCO). The local time is zero when the robot is at its staring point. The two contact lines represent the boundary of full-speed collision-free solutions and the curve indicates the collision information; more explicitly, when and where the collision will occur (the notations used are presented in Table I).

It is possible to use various geometric primitives instead of a circle in assumption 1, but the circular model is of great advantage in formulating the proposed problem in a simple way. As stated in Section I, it has been proven that assumption 2 is helpful for increasing safety in robot motion. The kinodynamic constraints [15] for a holonomic robot in assumption 3 should be taken into account, especially for high-speed robots that require a significant stopping distance.

III. 3-D Triangular Collision Object (TCO)

Several formulations for moving obstacle avoidance have been presented in our recent study [16] in which robot motion was decomposed into three phases: *approach, contact, and detachment* [16]. In this study, the collision-free solution for robot motion was given by deriving several inequalities and optimizing them. However, full information on potential collisions was obtained using a computational search method, and thus we provide closed-form equations to fully describe information on potential collisions between two agents.

The motions based on agents' future behavior are represented as a 3-D line or an object in configuration-time (C-T) space as shown in Fig.2. The motion for $\widehat{\mathcal{A}_1}$ is expressed as

$$\mathbf{p}_1(t) = (x_1(t), y_1(t), t) = (x_1^s + v_1 t,\ y_1^s,\ t) \tag{1}$$

From (1), the boundary of the obstacle is expressed as

$$\widehat{\mathbf{p}_1(t)} = \{(x,y,t) \mid (x-(x_1^s+v_1 t))^2 + (y-y_1^s)^2 = \hat{r}_1^{\,2}\} \tag{2}$$

Similarly, the motion for $\widehat{\mathcal{A}_2}$ is simply denoted as

$$\mathbf{p}_2(t) = k(t)(L_2\cos\theta_2, L_2\sin\theta_2, T_2) \tag{3}$$

where L_2 and T_2 are the overall traveled length and time, respectively, and $k(t) = t / T_2\ (0 \le t \le T_2)$.

As shown in Fig.2, collecting the points given by (2) and (3) with time forms an oblique cylinder and a line segment. Therefore, the contact or crossing point between them represents the exact information about when and where collisions will take place. We can have two (or the right and left) contact points and a line connecting crossing points with geometry and above formulas, which form a 3-D object in C-T space called a *3-D triangular collision object (TCO).*

We first derive a closed-form equation for the two line segments, tangent lines between the start point and the cylinder, in terms of heading angle. Substituting (3) represented by θ into (2) gives

$$(k(t)L_2\cos\theta - (x_1^s + v_1\cdot k(t)T_2))^2 + (k(t)L_2\sin\theta - y_1^s)^2 = \hat{r}_1^{\,2} \tag{4}$$

Simplifying (4) about $k(t)$ yields

$$(\alpha^2+\beta^2)k(t)^2 - 2(\alpha x_1^s + \beta y_1^s)k(t) + (x_1^s)^2 + (y_1^s)^2 - \hat{r}_1^{\,2} = 0 \tag{5}$$

where $\alpha = L_2\cos\theta - v_1 T_2$ and $\beta = L_2\sin\theta$.

The last equation is a quadratic equation in $k(t)$, more precisely, and thus its discriminant, usually expressed as D or $D/4$, should be equal to zero to satisfy the condition that the line segment is in contact with the oblique cylinder. If the equation has two real roots $t_a, t_b (t_a \ne t_b)$, then the line segment crosses the oblique cylinder, and physically the robot will collide with the moving obstacle. If the equation has two imaginary roots, the robot and the obstacle have no potential collision with each other. Thus, we have

$$D/4 = (\alpha x_1^s + \beta y_1^s)^2 - (\alpha^2+\beta^2)((x_1^s)^2 + (y_1^s)^2 - \hat{r}_1^{\,2}) = 0. \tag{6}$$

Since α and β in (5) is a function of $\cos\theta$ and $\sin\theta$, respectively, and $\sin\theta = \sqrt{1-\cos^2\theta}$, (6) is a quartic equation in $\cos\theta$ whose general solutions can be obtained using Ferrari's method. The solution by Ferrari's method is expressed in a complicated manner in terms of coefficients of the generalized quartic equation, not initial parameters of our formulation. For this reason, several procedures were used in the mathematical simplification, and we finally obtained

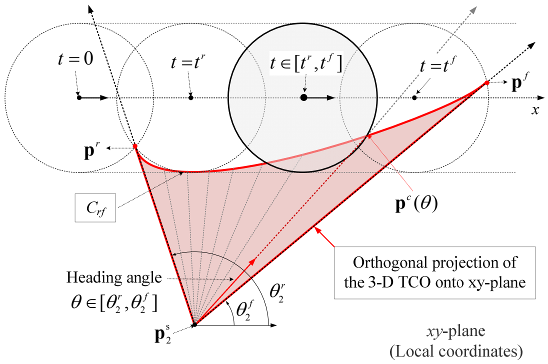


Fig. 3. Orthogonal projection of the 3-D triangular collision object (TCO) onto the *xy*-plane. When the direction of the robot crosses in the projection of TCO, the robot collides with the obstacle and its time information is given by the curve in the TCO.

TABLE II
EXPRESSION FOR SYMBOLS IN (7)

Symbol	Expression
$S,\ H,\ d$	$S = v_1 T_2\ ,\ H^2 = x_1^2 + y_1^2 - r^2,\ d = L_2(y_1^2 + x_1^2)^2$
a_1, a_2	$a_1 = r^2 x_1^2 + y_1^4 + x_1^2 y_1^2 - r^2 y_1^2, a_2 = -2rx_1 y_1,$
b_1	$L_2^2(3x_1^4 y_1^4 + 3x_1^6 y_1^2 + r^2 y_1^6 - r^2 x_1^6 + x_1^2 y_1^6 + x_1^8 + r^2 x_1^2 y_1^4 - r^2 x_1^4 y_1^2)$
b_2	$-x_1^6(r^2 + y_1^2) + x_1^4(r^4 + 5r^2 y_1^2 - 2y_1^4) + x_1^2(5r^2 y_1^4 - y_1^6 - 6r^4 y_1^2) - y_1^6 r^2 + r^4 y_1^4$
c_1	$L_2^2(6rx_1^3 y_1^5 + 2rx_1 y_1^7 + 6rx_1^5 y_1^3 + 2rx_1^7 y_1 - 4r^3 x_1^3 y_1^3 - 2r^3 x_1 y_1^5 - 2r^3 x_1^5 y_1)$
c_2	$4r^5(x_1^3 y_1 - x_1 y_1^3) + 2r(x_1^5 y_1^3 - x_1^3 y_1^5 + x_1^7 y_1 - x_1 y_1^7) + 6r^3(x_1 y_1^5 - x_1^5 y_1)$

Note. For simple notation, superscript s is omitted; i.e., $(x_1^s)^2 = x_1^2$, $(y_1^s)^2 = y_1^2$ and $\hat{r}_1 = r$.

$$\cos\theta = \frac{(a_1 \pm_t a_2 H)S \pm_s \sqrt{b_1 + b_2 S^2 \pm_t (c_1 + c_2 S^2) / H}}{d} \qquad (7)$$

: Table II gives the expressions for the symbols in (7). The two $\pm_t$ must have the same sign, while the sign of $\pm_s$ is independent. This equation gives four real or imaginary solutions, and each solution yields two angle candidates from the inverse cosine function; i.e., *arccos*. It is necessary to remove invalid solutions from the eight solution candidates by using several conditions in our formation.

Imaginary solutions were unconditionally removed from the eight candidates, and the validity of the real solutions was checked by substituting them into (5). If the equation had two real roots, the candidate was also removed. The next condition is that the double root from the equation should obey $t \in [0, T_2]$ because we deal only with future potential collisions in a time range defined by (3). Using the two conditions, only valid solutions (specifically one or two) were selected. The two line segments can be expressed in terms of the solutions θ_2^r and θ_2^f representing heading angles from the origin to the rear and front contact point, respectively. In addition, the contact times between the agents can be calculated by substituting the solutions into (5) and solving the quadratic equation.

The concave curve segment in the TCO is expressed with two real roots $t_a(\theta), t_b(\theta)$ in (5) because the minimum of the two physically indicates when and where two agents will first contact with each other. Let $t^c(\theta) = \min_\theta(t_a(\theta), t_b(\theta))$ be the first contact time, and the curve between the two angles θ_2^r, θ_2^f is denoted as:

$$\begin{aligned} C_{rf} &= \{\mathbf{p}^c(\theta) \mid \mathbf{p}^c(\theta) = \widehat{\mathbf{p}_1(t^c(\theta))} \cap g(\theta), \theta \in [\theta_2^r, \theta_2^f]\} \\ g(\theta) &= \{(x, y, t) \mid x / \cos\theta = y / \sin\theta = t \cdot v_2\} \end{aligned} \qquad (8)$$

where $\mathbf{p}^c(\theta_2^r) = \mathbf{p}^r$ and $\mathbf{p}^c(\theta_2^f) = \mathbf{p}^f$. From (7) and (8), the TCO was finally formulated from the agents' velocity vectors (or future behavior), and describes when and where collisions will occur with respect to the local coordinates. The TCO is actually a surface in 3-D space, and thus can be represented as a simple 2-D figure (specifically, the projection of the TCO) as shown in Fig.3. If we keep the time information in the projection, the figure is a simple but useful tool to provide full information on potential collisions over all heading angles.

IV. FULL-SPEED COLLISION-FREE MOTION

Robots can use the collision information from the TCO for their own navigation purposes. We present a simple example of TCO-based navigation in which only the heading angle is controlled to avoid moving obstacles in front of a high-speed robot, and call it a full-speed collision-free (FSCF) motion. Similar methods to quickly generate linear paths in dynamic environments were recently introduced in [17]. This method is based on the holonomic and kinodynamic constraints in assumption 3. More specifically, a fast moving vehicle has in general a large inertia and a very short time to collision, and thus it is often better to avoid near collisions with only direction changes than emergency braking or complicated algorithms. Our formulation and tool is especially advantageous in this kind of navigation.

Solutions for FSCF motion can be directly calculated using only (7). For real applications, however, several technical problems should be considered. First, an inaccurate sensor measurement makes the robot dangerous, which leads to the introduction of an extended obstacle model or solution modification that the inaccuracy is applied to. Another problem is the discontinuity of a function for the robot path as shown in Fig.4. It is necessary to assume that a robot can move holonomically or to modify our solutions by one of the smooth path planning methods; e.g., kinodynamic constraints [18]. We used a simple algorithm based on a B-spline curve [19] to smoothly connect two configuration points for robots.

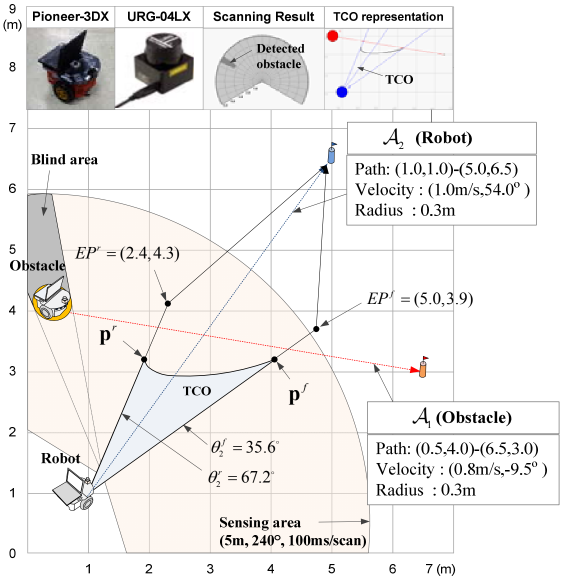


Fig. 4. An experimental setup for a FSCF motion. Two pioneers have their own paths and initial velocities. A laser scanner (URG-04LX) installed in the robot provides its scanning result (sensing area) at every update. The obstacle is modeled as an extended circle considering real sensor inaccuracy. The motion command for the obstacle (speed, heading angle) is not changed for its navigation, but the motion command for the robot is computed by the motion planner. When the robot reaches the rear or front escape point (EP^r, EP^f), which is the intersection between its straight line path and the boundary of the area swept by the obstacle, its path is modified to head towards the goal.

V. Experimental Results

We performed experiments to measure sensor inaccuracy and evaluate the validity of the proposed solutions (see Fig. 4). Two mobile robots (Pioneer 3DX) equipped with a laser scanner (URG-04LX-UG01) and Windows XP-based notebook computer with a Pentium IV 2.33 GHz processor were used. For comparison, a VPH [10] with a laser scanner was applied to the experiments. Figure 5 shows the time-varying measurements and estimated results of the size of the obstacle (m) and the velocity vector (direction and speed) from a Kalman filtering analysis performed for preliminary navigation. To ensure safety in the navigation, upper bounds or two bounds were imposed on the solution modification. The radius of the robot and obstacle was set to 0.3 m from the measurement results instead of the real radius of 0.28 m.

From the geometrical parameters shown in Fig.4 and derived formula (7), the motion planner produced eight solution candidates: 20.5, 320.6, 35.6, 305.5, 6.5, 334.5, 67.2, and 273.9°, and selected 67.2° and 35.6° as valid solutions, θ_2^r and θ_2^f, based on the two conditions presented in Section III. The planner then modified the solutions into $\theta_2^{r'} = 70.4^{\circ}$ and $\theta_2^{f'} = 30.9^{\circ}$ from the bounds of 0.47 m/s and 0.53 m/s shown in Fig. 5. Note that the lower and upper bounds were applied to modify θ_2^r and θ_2^f, respectively. The reason is that if an obstacle is considered to be slower than the real speed because of sensor errors, the robot following the path defined by θ_2^f does not safely escape the swept area. Similarly, the lower bound was used for the modification of the solution θ_2^r. In the modified TCO, the expected contact times were 2.59 s and 4.45 s when $v_1 / v_2 = 0.8 / 0.1$. Note that if the relative velocity is the same, the solutions are also the same in our formulation.

Figure 6 illustrates real traveled paths of the agents in the x-y plane, and the variation of speed and direction (heading angle) with time traveled. The total traveled time is presented in Table III. As for the relative velocities $0.4 / 0.5$ and $0.8 / 1.0$, the two agents actually collided with each other in the experiments. That is why the sum of its computation time, which depended on the grid number of sensing area, and the reaction time (about 150 *ms*), were too large to safely avoid the obstacle when its speed was high. In addition, the significant limit of motion from the kinodynamic constraints was directly attributed to the collision of the robot based on the immediate reactive method.

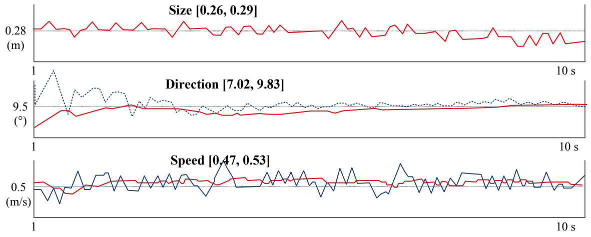


Fig. 5. Measured (dashed lines) and estimated results (bold lines) by Kalman filtering with time. The difference between the measured and estimated result decreases with time. The numbers in the brackets indicate the lower and upper bounds of the estimated results.

TABLE III

Traveled Times of Robot Navigation Shown in Fig.4

Method	$0.2 / 0.25$ (v_1 / v_2)	$0.4 / 0.5$	$0.8 / 1.0$
Proposed (FSCF) (s)	28.3	14.8	7.37
VPH (s)	33.2	collision	collision

On the other hand, in FSCF motion, the robot succeeded in navigating to the goal without any collisions even in the high-speed case. This is because the time to the computed contact point decreased with increased speeds of agents, which made our assumption 2 more realistic. For the low speed case, as with the VPH, the robot had enough time to react to environmental changes. Moreover, the relatively constant velocity vectors (or speed and direction) induced stable motions for the robots, and the proposed formulation allowed that the computational times were negligible.

VI. Discussion

Based on our results, the following summarizes the contributions of this study. First, we presented a collision anticipation tool that directly states when and where collisions will occur by considering the future behavior of agents. Previous tools such as VFH, VPH, and ND have mainly emphasized finding a set of safe directions by using only current positions of agents. The VO can present a set of safe velocities (directions and speeds) from the behavior, but cannot provide time information on potential collisions; this led to the introduction of a finite time horizon in the practical applications [20]. The TCO, however, directly and analytically provided full information on potential collisions with respect to C-T space, which could be helpful to new collision resolution schemes.

The method of using one of the solutions from the proposed tool yields more stable, safer motions compared to immediate reactive methods as shown in Fig. 6. In the experiments, the method was of great use when the time to collision was very small (or the speed for the robot was high). This is because the small time, which has been considered to be undesirable in most methods, contributed to reducing the gap between real and simulated robots, and keeping the motions during the avoidance. In addition, it has the real-time effectiveness that is essential to practical applications since a solution for safe robot motion can be analytically given.

Nonetheless, several issues require further study. Since the proposed TCO is fundamentally an analytic tool to represent potential collisions under the three assumptions, it should be addressed in more detail to extend the tool for compensation of sensor and motion errors (e.g., PVO [5]), and for the relaxation of assumption 2 (e.g., NLVO [21] and time horizon [20]). Next, it is necessary to develop advanced methods to fully exploit the information from the TCO. Although the example in this study, FSCF motion, used only a boundary of solutions in the TCO, it is possible to control speed and direction simultaneously for efficient collision avoidance in terms of traveled time with the TCO.

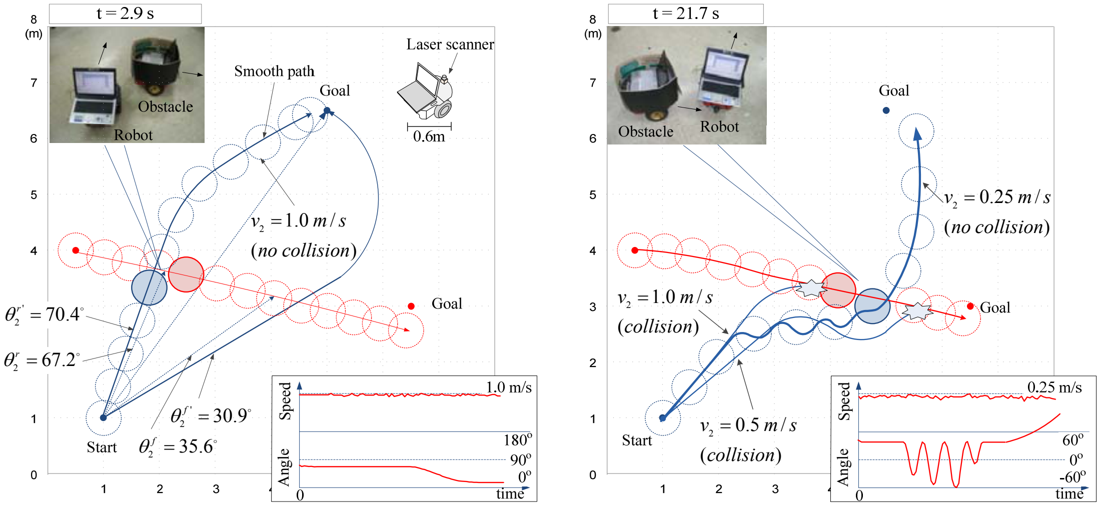


Fig. 6. Experimental results of the FSCF motion method (left) and the VPH method (right). The theoretical solutions θ_2^r, θ_2^f were modified by considering measured inaccuracies in Fig.5. The path segment from the escape to the goal point was also modified to be feasible by a B-spline based path planning method. The odometer errors were attributed to slight differences between the planned and real traveling paths. The initial velocities at the start points were set to be the values in Table III, not zero. The variations of speed and heading angle with traveling time are represented with simple graphs to investigate their trends. The pictures show the moments at which the agents were the closest to each other.

VII. Conclusion

The results of our study indicate that collisions between moving agents can be analytically anticipated and geometrically represented as 3-D TCOs in which the future behavior of agents is explicitly considered. The TCO is completely expressed with only closed-form equations, and can provide full information concerning when and where collisions will occur under motion conditions over all possible heading angles. The TCO-based method is computationally efficient and thus could be advantageous to real-time motion planners, especially those dealing with high-speed robots. Therefore, our approach has demonstrated good potential as a useful technique for advanced motion planning systems.

References

[1] J. Borenstein and Y. Koren, "The vector field histogram - fast obstacle avoidance for mobile robot," *Journal of Robotics and Automation*, vol. 7, no. 3, pp. 278-288, 1991.

[2] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using the relative velocity pardigm," *in IEEE Int. Conf. on Automation and Robotics*, vol. 1, pp. 560-566, 1993.

[3] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics and Automation Magazine*, pp. 23-33, March, 1997.

[4] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," *in Proc. of IEEE Int. Conf. on Robotics and Automation*, 1999.

[5] C. Fulgenzi, A. Spalanzani, and C. Laugier, "Dynamic obstacle avoidance in uncertain environment combining PVOs and occupancy grid," *in Proc. of IEEE Conf. on Robotics and Automation*, Roma, pp. 1610-1616, 2007.

[6] D. Wilkie, J. Berg, and D. Manocha, "Generalized velocity obstacles," *in Proc. of IEEE Conf. on Intelligent Robots and Systems*, St. Louis, pp. 5573-5578, 2009.

[7] J. Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," *in Proc. of IEEE Conf. on Robotics and Automation*, New York, pp. 1928-1935, 2008.

[8] F. Belkhouche, "Reactive path planning in a dynamic environment," *IEEE Trans. on Robotics*, vol. 25, no.4, August, 2009.

[9] D. An and H. Wang, "VPH: A new laser radar based obstacle avoidance method for intelligent mobile robots," *in 5th World Congress on Intelligent Control and Automation*, China, pp. 4681-4685, 2004.

[10] D. Shi, D. Dunlap, and E. Collins, "A comparion between a fuzzy behavioral algorithm and a vector polar histogram algorithm for mobile robot navigation," *in Proc. of IEEE Int. Symp. on Computational Intelligence in Robotics and Automation*, USA, pp. 260-265, 2007.

[11] J. Minguez and L. Montano, "Nearness diagram navigation: collision avoidance in troublesome scenarios," *IEEE Trans. on Robots and Automation*, vol. 20, no. 1, pp. 45-59, 2004.

[12] M. Seder and I. Petrovic, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," *in Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 1986-1991, 2007.

[13] T. Fraichard, "A short paper about motion safety," *in Proc. of IEEE Int. Conf. on Robotics and Automation,* Roma, pp. 1140-1145, 2007.

[14] L. Martinez-Gomez and T. Fraichard, "Collision avoidance in dynamic environments: an ICS-based solution and its comparative evaluation," *in Proc. of Int. Conf. on Robotics and Automation*, pp. 100-105, 2009.

[15] B. Donald, P. Xavier, J. Canny, and J. Reif, "Kinodynamic motion planning," *Journal of Association for Computing Machinery*, vol. 40, no. 5, pp. 1048-1066, 1993.

[16] J. H. Park, J. S. Choi, J. Kim and B. H. Lee, "Moving obstacle avoidance for a mobile robot," *in Proc. of IEEE Conf. on Control and Automation*, Christchurch, New Zealand, 2009.

[17] E. J. Beranbeu, "Fast generation of multiple collision-free and linear trajectories in dynamic environments," *IEEE Trans. on Robotics*, vol. 25, no. 4, August, 2009.

[18] B. Lau, C. Sprunk, and W. Burgard, "Kinodynamic motion planning for mobile robots using splines," *in Proc. of IEEE Int. Conf. on Robots and Intelligent Systems,* St. Louis, pp. 2427-2433 , 2009.

[19] M. Yamamoto, M. Iwamura, and A. Mohri, "Quasi-time time-optimal motion planning of mobile platforms in the presence of obstacles," *in Proc. of Int. Conf. on Robotics and Automation*, pp. 2958-2963, 1999.

[20] E. Prassler, J. Scholz, and P. Fiorini, "A robotics wheelchair roaming in a railway station," *in Proc. of Conf. on Field and Service Robotics,* Pittsburgh, 1999.

[21] Z. Shiller, F. Large, and S. Sekhavat, "Motion planning in dynamic environments: Obstacles moving along arbitrary trajectories," *in Proc. of Int. Conf. on Robotics and Automation*, pp. 3716-3721, 2001..