# Simplified Cooperative Collision Avoidance Method Considering the Desired Direction as the Operation Objective of Each Mobile Robot

ABE Yasuaki and MATSUO Yoshiki

Department of Mechanical and Control Engineering,
Graduate School of Science and Engineering, Tokyo Institute of Technology

**Abstract:** In a previous study, the authors have proposed the Cooperative Collision Avoidance (CCA) method which enables mobile robots to cooperatively avoid collisions, by extending the concept of the Velocity Obstacle to multiple robot systems. The method introduced an evaluation function considering an operation objective so that each robot can choose the velocity which optimizes the function. As the evaluation function could be of an arbitrary type, this method is applicable to a wide variety of tasks. However, it complicates the optimization of the function especially in real-time. In addition, construction of the evaluation function requires an operation objective of the other robot which is very hard to obtain without communication. In this paper, the CCA method is improved considering such problems for implementation. To decrease computational costs, the previous method is simplified by introducing two essential assumptions. Then, by treating the desired direction of locomotion for each robot as the operation objective, an operation objective estimator which estimates the desired direction of the other robot is introduced. The only measurement required is the other robot's relative position, since the other information can be obtained through the estimation. Hence, communicational devices that are necessary for most other cooperative methods are not required. Moreover, mobile robots employing the method can avoid collisions with uncooperative robots or moving obstacles as well as with cooperative robots. Consequently, this improved method can be applied to general dynamic environments consisting of various mobile robots.

**Keywords:** Multiple Mobile Robot Systems; Collision Avoidance; Cooperative Collision Avoidance; Velocity Obstacle; Common Velocity Obstacle.

## 1. Introduction

We have previously proposed the Cooperative Collision Avoidance method that extends the Velocity Obstacle (VO) method for multiple mobile robot systems. In the scheme, an evaluation function intended to consider an operation objective of a robot was introduced and optimized by each robot. As the type of the evaluation function was truly nonrestricted, the function could represent almost any operation objective of a robot, however, obtaining the optimal velocity became very hard on the contrary. Searching overall the possible velocity region was used in the previous study and performed well on a simulator, but its computational cost was extremely high. Besides, construction of the evaluation function required operation objective of the other robot to maximize total performance for both robots. If communication was available, this would be not very difficult, however, it is not reasonable to assume that mobile robot systems are always equiped with such capability.

Thus, in this paper, we simplify the previous method to decrease the computational cost, and adopt an operation objective estimator to achieve implicit communication, so that the method can be fitted on actual mobile robot systems. Moreover, mobile robots employing the method can avoid collisions with uncooperative robots or moving obstacles as well as with cooperative robots.

The organization of this paper is as follows. Section 2 briefly reviews related works and Section 3 states what is collision avoidance. The basic principles of the present method are derived from the VO method in Section 4. However, the principles are rather analytical and hard to implement. Therefore, simplification of the method under some reasonable assumptions are proposed in Section 5. Computational simulations are conducted and examined in Section 6 followed by concluding Section 7.

## 2. Related Works

Mobile robot systems have a wide range of applications such as object transportation, field exploration, and security control. Therefore, they have been attracting much research interest. A central issue for mobile robots is collision avoidance and many researches have been focused on it. For static environments it has been studied well and some effective methods have already been proposed. However, though considering moving obstacles including other mobile robots or humans are often inevitable in practice, methods for such cases have not been established.

### 2.1. Velocity Based Navigation Methods

Navigation methods which plan in the velocity field are suitable for collision detection and collision avoidance with moving objects. SIMMONS proposed a collision detection method called CVM [5], where robot's velocity is divided into translational velocity and rotational velocity. Besides, when non-holonomic robots are considered and/or robots can locomote at a high speed, dynamics of the robots must be treated explicitly. The dynamic window approach is one of such methods and can be easily adapted to the velocity field navigation [8, 13].

### 2.2. Collision Avoidance Methods

A well-known method is to determine a reference trajectory just considering stationary obstacles, tnen to perform a reactive collision avoidance procedure considering moving obstacles [8, 12, 13]. Although this approach has a notable feature of yielding a sub-optimal trajectory, the trajectory may poorly perform collision avoidance especially in congested environments. Another approach is to utilize a potential function with respect to relative position between robots.

However, design of this method tends to be ad hoc, so that realization of complicated behavior is difficult. On the other hand, the VO method is collision detection method in the velocity field based on the relative velocity between robots [3, 4, 10], which determines a velocity set leading to future collisions. As the VO method exactly describes a collision condition between robots, many researchers prefer to adopt this method [1, 2, 7, 9].

### 2.3. Cooperative Methods

Cooperative approaches are also attracting research interest. FUJIMORI et al. proposed a cooperative collision avoidance method where predefined rules are applied on each robot based on the relative positions between them [6, 11]. However, as this method entirely depends on an assumption that both robots obey the same algorithm, collision avoidance with uncooperative robots is not guaranteed, which makes the method ineffective in real environments.

## 3. Scope of the Study

Firstly, in this paper, we consider moving objects. Moving objects can be classified into mobile robots and moving obstacles. Mobile robots are equiped with the method to be proposed but moving obstacles are not. Mobile robots have limited abilities on sensing and computation and have individual "operation objectives" such as wandering, going to the goal position, and so on. Although moving obstacles may have their own operation objectives, they are not considered in this paper. Considering moving obstacles has an important aspect that mobile robots without omni-directional sensing ability can be explicitly considered. For example, if Robot A can detect Robot B but Robot B can not sense Robot A, Robot B will locomote freely and be regarded as a moving obstacle by Robot A.

Secondly, we clarify the class of the method to be proposed. Major aspects that characterize control methods for multiple mobile robot systems are degree of concentration, real-time capability and implementability to actual robotic systems. As we aim at implementation, our method should be a distributed on-line method.

Finally, the type of collision avoidance should be mentioned. Roughly, collision avoidance can be classified into global collision avoidance and local collision avoidance. Global collision avoidance can be regarded as a routing problem aiming at avoiding congested places, while local collision avoidance treats collisions at hand. Our objective is to design local collision avoidance, therefore we do not bother about the global long time performance.

## 4. The Fundamentals of the CCA Method

The Cooperative Collision Avoidance (CCA) method utilizes the concept of Velocity Obstacle (VO), so that a brief summary of the conventional VO method comes first and is followed by derivation of the CCA method.

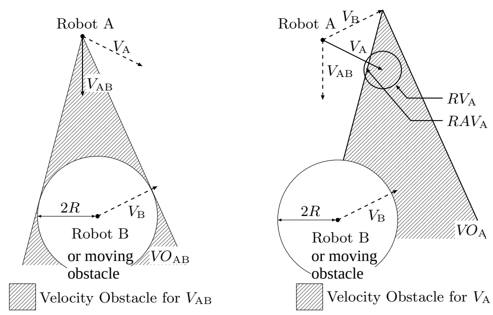
### 4.1. The Conventional Velocity Obstacle Method

The VO method is a well-known collision detection method proposed in [10]. It provides a velocity set called Velocity Obstacle, where velocities inside the set mean future collisions. Since detailed description of the VO method can be found in [10], only the results are introduced in the following.

In fig. 1, Robot A and B (or a moving obstacle) are moving at velocities of $V_{\mathrm{A}}$ and $V_{\mathrm{B}}$, respectively. Each robot's shape is represented by a disk of radius $R$. Then, Robot A's radius can be regarded as 0 by enlarging Robot B's radius by Robot A's actual radius. Clearly, this conversion does not affect the collision condition. Viewing from coordinates fixed on Robot B, Robot A seems to be moving at a relative velocity of $V_{\mathrm{AB}} = V_{\mathrm{A}} - V_{\mathrm{B}}$. Hence, if $V_{\mathrm{AB}}$ points at the hatched region in fig. 1(a) and both robots keep their velocities, Robot A and B will collide in the future. Thus the region is called a "Velocity Obstacle".

$VO_{\mathrm{A}}$ in fig. 1(b) is the VO for velocity $V_{\mathrm{A}}$ defined on global coordinates, which can be derived from $VO_{\mathrm{AB}}$ by adding $V_{\mathrm{B}}$ to $V_{\mathrm{AB}}$ and $VO_{\mathrm{AB}}$. However, this procedure implicitly assumes that $V_{\mathrm{B}}$ does not vary. The assumption may not be satisfied, if Robot B is a mobile robot.

When Reachable Velocity $RV_{\mathrm{A}}$ which means a feasible velocity set considering robot's dynamics and kinematics is given, Reachable Available Velocity $RAV_{\mathrm{A}} = \overline{VO_{\mathrm{A}}} \cap RV_{\mathrm{A}}$ yields possible collision avoidance velocities.




(a)VO regarding $V_{\mathrm{AB}}$ (b)VO regarding $V_{\mathrm{A}}$

Fig. 1. Introduction of the Velocity Obstacle method

### 4.2. The Basic Principles of the CCA Method

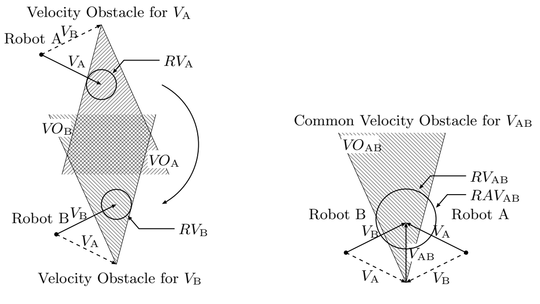
Although the VO method assumes that the other robot moves in a constant velocity, the assumption is inadequate for dynamic environments. In this subsection, this assumption is eliminated by assuming that the moving object is a mobile robot employing the same algorithm.

#### 4.2.1 The Common Velocity Obstacle

Figure 2(a) shows $VO_{\mathrm{A}}$ and $VO_{\mathrm{B}}$ in the same velocity field. Noting that the arrangement of $VO_{\mathrm{A}}$ for Robot A and that of $VO_{\mathrm{B}}$ for Robot B are just symmetrical each other, both can be depicted as a single figure as fig. 2(b), by rotating and overlaying one onto the other. Then, the VO is renamed as $VO_{\mathrm{AB}}$ and called a "Common Velocity Obstacle", because a collision will occur if $V_{\mathrm{AB}}$ points inside of it. Collision avoidance can be regarded as that Robot A and B vary their relative velocity $V_{\mathrm{AB}}$ cooperatively so that $V_{\mathrm{AB}}$ points outside $VO_{\mathrm{AB}}$. It is worth noting that $RV_{\mathrm{AB}}$ is the additive set of $RV_{\mathrm{A}}$ and $RV_{\mathrm{B}}$ and larger than $RV_{\mathrm{A}}$ or $RV_{\mathrm{B}}$. As the shape of $VO_{\mathrm{AB}}$ is the same as $VO_{\mathrm{A}}$, $VO_{\mathrm{B}}$, broader $RV_{\mathrm{AB}}$ directly leads to broader $RAV_{\mathrm{AB}}$.

#### 4.2.2 Evaluation Functions for Velocity Determination

As described above, neither the VO method nor the CCA method designate a certain collision avoidance velocity. Therefore, some means to determine the collision avoidance veloc-




(a)VO regarding $V_A$, $V_B$ (b)VO regarding $V_{AB}$

Fig. 2. Introduction of the Cooperative Collision Avoidance method

ity is required. In the previous studies [1, 2], we proposed an evaluation function based method, where the operation objective, the VO and the RV are all described as functions of velocity. Then, the total evaluation function consists of these sub functions, where the velocity that minimizes the function is regarded to be optimal. However, to obtain an optimal velocity is not a simple problem, because the function is not analytic in many cases. Accordingly, a searching method was used in the previous works, however, its computational cost was high.

In the following discussion, symbolic function $F$ is used to represent a procedure which derives the optimal velocity, because the discussion does not concern the specific method. Generally, arguments and returns of $F$ can be written as

$$(\dot{p}_{A,k+1},\, \dot{p}_{B,k+1})^T = F(p_{A,k},\, \dot{p}_{A,k},\, p_{B,k}, \dot{p}_{B,k},\, r_{A,k},\, r_{B,k}),$$

where $p$ denotes position of a robot, $r$ is a vector representing an operation objective of a robot. For example, when the goal position is given as the operation objective, $r = (r_x,\, r_y)$ is a vector of two dimensions.

4.2.3 Estimation of the Operation Objective of the Other Robot

Since the mobile robots considered in this paper do not have a communicative capability, the last argument of $F$, the operation objective of the other robot, can not be obtained directly. In this subsection, an estimation method based on Jacobian of $F$ is proposed.

Assume that Robot A and B obey the following two equations

$$(\dot{p}_{A,k+1},\, \tilde{\dot{p}}^A_{B,k+1})^T = F(p_{A,k},\, \dot{p}_{A,k},\, p_{B,k}, \dot{p}_{B,k},\, r_{A,k},\, \tilde{r}^A_{B,k}),$$
$$(\dot{p}_{B,k+1},\, \tilde{\dot{p}}^B_{A,k+1})^T = F(p_{B,k},\, \dot{p}_{B,k},\, p_{A,k}, \dot{p}_{A,k},\, r_{B,k},\, \tilde{r}^B_{A,k}),$$

respectively, where $\tilde{\dot{p}}^A_{B,k+1}$ is the predicted velocity of Robot B at the next sampling time by Robot A, and $\tilde{r}^A_{B,k}$ is Robot B's operation objective estimated by Robot A.

Jacobian of $F$ can be written as follows:

$$\frac{\partial F}{\partial q} = \begin{bmatrix} \frac{\partial \dot{p}_{A_x}}{\partial q} \\ \frac{\partial \dot{p}_{A_y}}{\partial q} \\ \frac{\partial \dot{p}_{B_x}}{\partial q} \\ \frac{\partial \dot{p}_{B_y}}{\partial q} \end{bmatrix} = \begin{bmatrix} \frac{\partial \dot{p}_{A_x}}{\partial p_A} & \frac{\partial \dot{p}_{A_x}}{\partial \dot{p}_A} & \cdots & \frac{\partial \dot{p}_{A_x}}{\partial r_A} & \frac{\partial \dot{p}_{A_x}}{\partial \tilde{r}_B} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial \dot{p}_{B_y}}{\partial p_A} & \frac{\partial \dot{p}_{B_y}}{\partial \dot{p}_A} & \cdots & \frac{\partial \dot{p}_{B_y}}{\partial r_A} & \frac{\partial \dot{p}_{B_y}}{\partial \tilde{r}_B} \end{bmatrix}$$

where $q = (p_A,\, \dot{p}_A,\, p_B, \dot{p}_B,\, r_A,\, r_B)$. Using this Jacobian, relationship between the prediction error and the estimation error can be approximated by

$$(\dot{p}_{B,k+1} - \tilde{\dot{p}}^A_{B,k+1})^T \simeq \begin{bmatrix} \frac{\partial \dot{p}_{B_x}}{\partial r_{B_x}} & \frac{\partial \dot{p}_{B_x}}{\partial r_{B_y}} \\ \frac{\partial \dot{p}_{B_y}}{\partial r_{B_x}} & \frac{\partial \dot{p}_{B_y}}{\partial r_{B_y}} \end{bmatrix}_k \cdot (\tilde{r}^A_{B,k+1} - \tilde{r}^A_{B,k})^T.$$

Therefore,

$$\tilde{r}^{AT}_{B,k+1} \simeq \begin{bmatrix} \frac{\partial \dot{p}_{B_x}}{\partial r_{B_x}} & \frac{\partial \dot{p}_{B_x}}{\partial r_{B_y}} \\ \frac{\partial \dot{p}_{B_y}}{\partial r_{B_x}} & \frac{\partial \dot{p}_{B_y}}{\partial r_{B_y}} \end{bmatrix}_k^{-1} \cdot (\dot{q}_{B,k+1} - \tilde{\dot{p}}^A_{B,k+1})^T + \tilde{r}^{AT}_{B,k},$$

where $r = (r_x,\, r_y)$ is assumed. Using this updating equation, the operation objective of the other robot can be estimated.

## 5. The Simplified CCA Method

To implement the present method to real robotic systems, the computational cost is a major problem. In addition, as Jacobian of $F$ can not be obtained in most cases, the estimation method proposed above is inadequate in practice. In this section, the previous CCA method is simplified under some reasonable assumptions and a straightforward estimation technique is introduced.

### 5.1. Assumptions

In this section, we make the following assumptions to simplify the previous CCA method:

• The operation objective of each robot is given as a desired direction.

• Robots are $\Delta R$–equivalent omni-directional robot.

The first assumption is based on an idea that the goal position does not affect local navigation. The second concept of $\Delta R$–equivalent omni-directional robot is such that the robot can be regarded as an omni-directional robot by virtually enlarging its radius by $\Delta R$. Therefore, kinematic and/or dynamic model of robots such as differential drive robot, etc. can be abstracted in the following discussion.

A trajectory of an ideal omni-directional robot during a time period of $k\Delta t \le t < (k+1)\Delta t$ can be described by

$$(x_{\rm n}(t),\, y_{\rm n}(t)) = \left(x_k + \frac{x_{k+1} - x_k}{\Delta t}(t - k\Delta t),\, y_k + \frac{y_{k+1} - y_k}{\Delta t}(t - k\Delta t)\right),$$

where $\Delta t$ is a sampling period and $(x_k,\, y_k)$ is position of the robot at time $t = k\Delta t$. When a robot satisfies the following condition:

$$\max_t \sqrt{\{x_{\rm n}(t) - x(t)\}^2 + \{y_{\rm n}(t) - y(t)\}^2} \le \Delta R,$$

the robot is called a $\Delta R$–equivalent omni-directional robot. Clearly, a meaning of $\Delta R$ is maximum error allowable for the radius of the robot. Using this concept, for example, a differential-drive robot can be regarded as an omni-directional robot with a proper $\Delta R$ as a sufficient condition for collision avoidance, where kinematics and dynamics of the robot are implicitly considered. The only constraint applied to the $\Delta R$–equivalent omni-directional robot is the maximum velocity.

## 5.2. The Relative Velocity Diagram

To obtain optimal collision avoidance velocities as analytic form is challenging issue in the CCA method. The Relative Velocity Diagram (RVD) introduced in this subsection is one answer for the problem. By using the RVD, computational cost is extremely reduced and the CCA method using the RVD is called the Simplified CCA method.

Figure 3 shows an example of desired direction $r_A$, velocity constraints $RV_A$ and $VO_\mathrm{A}$ for Robot A. In this case, only two points of $P_{B1}$ and $P_{B2}$ are the optimal velocity candidates. Clearly, $P_C$ is always a possibly optimal velocity, however, $P_C$ is covered with the VO when collision is predicted and cannot be chosen as the optimal velocity. In addition, $P_{B1}$ or $P_{B2}$ always have better evaluations than $P_A$, because if $P_A$ is prefered than others, $P_C$ must not be covered with the VO and this means collision free. Therefore, only $P_{B1}$ and $P_{B2}$ are the optimal velocity candidates.

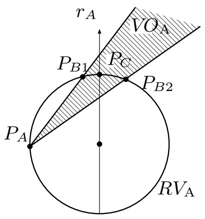
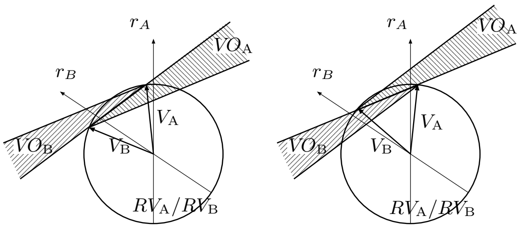



Fig. 3. Derivation of the RVD (Step 1)

When optimal $V_\mathrm{A}$ is $P_{B1}$, corresponding $VO_\mathrm{B}$ for $V_\mathrm{B}$ is depicted as fig. 4(a), while fig. 4(b) means optimal $V_\mathrm{A}$ of $P_{B2}$. It is worth noting that $VO_\mathrm{A}$ and $VO_\mathrm{B}$ are symmetry and their edges are always aligned. To clearfy the meanings



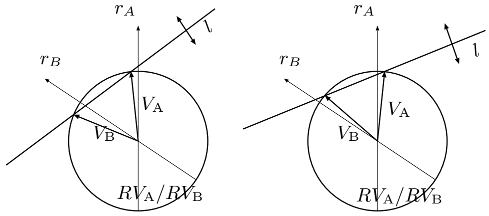

(a)Case 1 (b)Case 2

Fig. 4. Derivation of the RVD (Step 2)

of the fact, essential components in fig. 4 can be rewritten as fig. 5, where boundary lines between $VO_\mathrm{A}$ and $VO_\mathrm{B}$ are depicted as line $l$. This figure is called the Relative Velocity Diagram (RVD), because line $l$ represents the direction of relative velocity $V_\mathrm{AB}$ and plays the central role in the diagram. This figure means that $V_\mathrm{A}$ and $V_\mathrm{B}$ are determined by appropriately placed line $l$, and the pair $V_\mathrm{A}$ and $V_\mathrm{B}$ are guaranteed to be collision avoidance velocities. Thus, the choosing collision avoidance velocities problem is now proved by determining location of line $l$.

To determine the optimal placement of line $l$, an evaluation function which provides the highest evaluation to the desired velocity of each robot is introduced so that collision avoidance velocities lead to not only collision avoidance but also achiveing the operation objective of each robot. For example, we choose an evaluation function for each robot as

$$\begin{array}{ll}\text{Robot A:} & C\cos(\varphi - \psi)\\ \text{Robot B:} & C\cos(\varphi + \psi - r_\mathrm{B}),\end{array}$$




(a)Case 1 (b)Case 2

Fig. 5. The Relative Velocity Diagram (RVD)

using $\psi$ shown in fig. 6, where $C$ is a positive constant and $r_\mathrm{A}$ is assumed to be 0 without loss of generality. Beside, $\alpha = -\varphi$ and $\beta = \varphi - r_\mathrm{B}$ are introduced for simplicity, then the following is a total evaluation function:

$$C\{\cos(\psi + \alpha) + \cos(\psi + \beta)\}.$$

Clearly, this evaluation function has maximum at $\psi = -\frac{\alpha+\beta}{2} = \frac{r_\mathrm{B}}{2}$. Differences between $\angle V_\mathrm{A}$ and $r_\mathrm{A}$, and $\angle V_\mathrm{B}$ and $r_\mathrm{B}$ are

$$\begin{array}{ll}\text{Robot A:} & \varphi - \psi = \varphi - \frac{r_\mathrm{B}}{2}\\ \text{Robot B:} & \varphi + \psi - r_\mathrm{B} = \varphi - \frac{r_\mathrm{B}}{2},\end{array}$$

respectively. Hence, when two robots equally avoid a collision, the total evaluation function is maximized.

From a view point of the RVD, the first assumption is not necessary. The essential fact is that velocity candidates are limited to $P_{B1}$, $P_{B2}$. In such a case, the RVD can be composed and a similar optimization approach on the RVD can be used, even for a different type of operation objective.

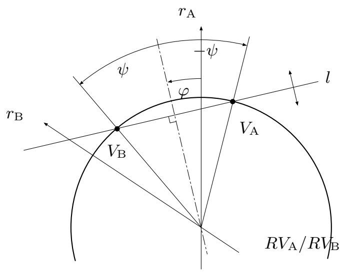



Fig. 6. Optimal solution using the RVD

## 5.3. Estimation of the Operation Objective of the Other Robot

As described above, the difference between the desired direction and the locomoting direction is equal for each robot.

$$r_\mathrm{A} - \angle V_\mathrm{A} = r_\mathrm{B} - \angle V_\mathrm{B}$$

$r$ : desired direction; $\angle$V : locomotiong direction

Therefore, the operation objective of the other robot can be estimated using the following equation.

$$\tilde{r}_\mathrm{B,k+1} = (1-\alpha)\,\tilde{r}_\mathrm{B,k} + \alpha\,(r_\mathrm{A} - \angle V_\mathrm{A,k} + \angle V_\mathrm{B,k})$$

where $0 < \alpha \leq 1$ is a decay factor which controls stability of the estimation and rapidity of the convergence.

When a moving obstacle going straight ($\angle V_\mathrm{B} = r_\mathrm{B}$) is considered, the estimation equation is rewritten into

$$\tilde{r}_\mathrm{B,k+1} = (1-\alpha)\tilde{r}_\mathrm{B,k} + \alpha(r_\mathrm{A} - \angle V_\mathrm{A,k} + r_\mathrm{B}),$$

and this equation converges to $\tilde{r}_\mathrm{B} = r_\mathrm{A} - \angle V_\mathrm{A} + r_\mathrm{B}$. The estimated value does not converge to the actual direction, however, collision avoidance can be achieved by this scheme against not only cooperative robots but moving obstacles.

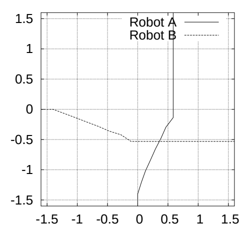
## 6. Simulations

In this section, the simplified CCA method is implemented for computational simulations. At first, to confirm the effectiveness of the method, simulations for a mobile robot and a moving obstacle are conducted. Then, to examine effects of parameters, simulations are conducted for various parameter set-ups.

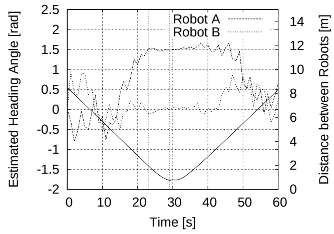
Two configurations, crossing and face to face, are employed in simulations. Measurement noise is applied to the position measurement at every sampling time. The noise is generated with a normal distribution whose mean value is 0 and standard deviation is $\sigma = dl^2$ where $d$ is a noise coefficient and $l$ denotes the distance between robots. This error model is based on actual image sensors. By adjusting a radius of robot virtually, the minimum distance during collision avoidance can be controlled. In this paper, the actual radius is scaled by 1.5. The other parameters used in simulations are shown in Table 1.

Table 1. Default parameters

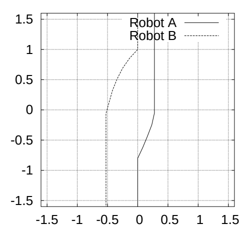
| | Notation | Value |
|---|---|---|
| $R$: | Robot radius | 0.25 [m] |
| $V_{\max}$: | Maximum velocity | 0.2 [m/s] |
| $l_{\text{sight}}$: | Sight radius | 2.0 [m] |
| $\alpha$: | Decay factor | 0.5 |
| $\Delta t$: | Sampling speed | 0.5 [s] |
| $d$: | Noise coefficient | 0.2 [1/m] |
| $R_{\text{margin}}$: | Margin factor | 1.5 |



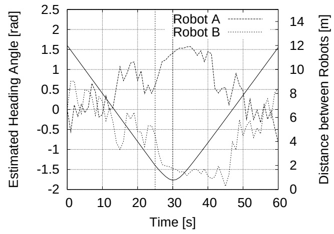
(a)Cross - Trajectory



(b)Cross - Estimated heading angle

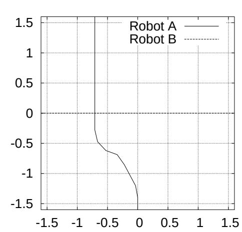


(c)Face to face - Trajectory

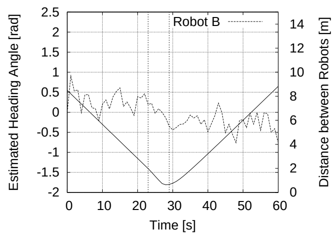


(d)Face to face - Estimated heading angle

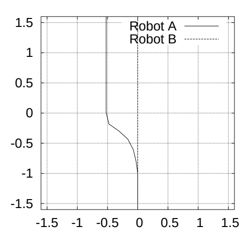
Fig. 7. Results for a mobile robot



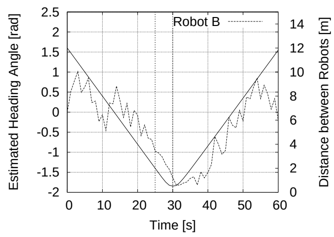
(a)Cross – Trajectory



(b)Cross – Estimated heading angle



(c)Face to Face – Trajectory



(d)Face to Face – Estimated heading angle

Fig. 8. Results for a moving obstacle (Robot B)

### 6.1. Basic Operations

Figure 7 shows the resulting trajectories and the estimated operation objectives during mobile robot versus mobile robot collisions. In fig. 7(a) and fig. 7(c), Robot A is moving upward from the bottom, while Robot B is moving from left to right and downward from the top, respectively. Figure 7(b) and fig. 7(d) show the estimated operation objectives, where actual operation objectives are $\left(\frac{\pi}{2}, 0\right)$ and $\left(\frac{\pi}{2}, -\frac{\pi}{2}\right)$ , respectively. Solid lines in the graph plots indicates the distance between the robots. When the robots are sufficiently close to each other, the estimation results are very close to the correct values. Otherwise, the estimation results are disturbed by the measurement noise.

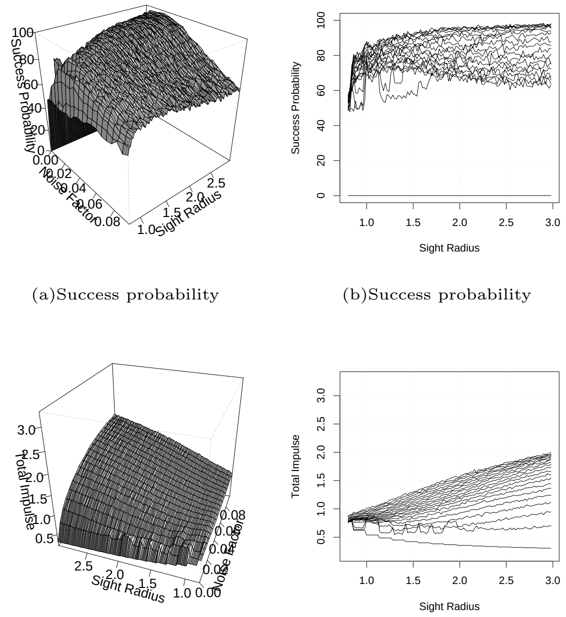
Figure 8 also shows the resulting trajectories and the estimated operation objectives. When Robot B behaves as a moving obstacle and just moves rightward in fig. 8(a) and downward in fig. 8(c). Vertical dashed lines in fig. 8(b) and fig. 8(d) indicate the moments at the beginning and the ending of the collision avoidance. During the collision avoidance, estimated operation objectives deviate from actual values, as described in the last section.

### 6.2. Influences of the Parameters

In this subsection, effects of some representative parameters such as the sight radius, the noise coefficient and the decay factor are examined. As the evaluation indices, summation of impulse that concerns with energy consumption and probability of success are used. For each lattice point in graph plots, 1,000 trial runs are made to smoothen the resulting surface.

Figure 9(a) and fig. 9(b) show success probability versus the noise coefficient and the sight radius. It is worth noting that a longer sight radius does not lead to a higher success probability. The cause of this phenomenon can be inferred such that an effect of the measurement noise is dominant when distance between the robots is long to some extent. The following fact confirms this expectation. The total impulse increases proportionally to the sight radius with the higher noise coefficient shown in fig. 9(c) and fig. 9(d), while it decreases according to the sight radius, when the measurement noise is zero or sufficiently small. Meanwhile, a smaller sight radius also causes low success probability shown in fig. 9(b). Hence, there should exist the optimal sight radius for a noise coefficient, but it has not been proved yet.

Figure 10 shows the success probability versus the noise coefficient and the decay factor $\alpha$. For a small decay factor, the estimation is precise but slow. Therefore, the success probability is not so high. On the contrary, a large decay factor leads to a fast but inaccurate estimation and also re-

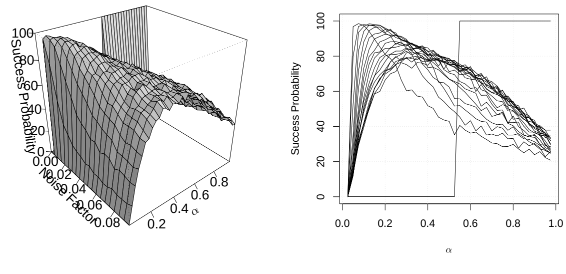



(a)Success probability (b)Success probability

(c)Total impulse (d)Total impulse

Fig. 9. Success probability/Total impulse vs Noise factor/Sight radius

sults in a low success probability. Although existence of the optimal decay factor is expected from the simulation results, it can not be determined from these graph plots. Because, the other parameters which are fixed to the default in the simulations may affect the optimal decay factor.




(a)Success probability (b)Success probability

Fig. 10. Success probability vs Decay factor

## 7. Conclusions

In this paper, the CCA method previously proposed by the authors was improved considering implementation. To simplify the CCA method, two reasonable assumptions were applied, where a concept of $\Delta R$–equivalent omni-directional robot was introduced. Using the simplified CCA method, the optimal velocity could be determined without time-consuming searching. In addition, the estimation of the operation objective of the other robot was also introduced. Thereby, the proposed method can be implemented to mobile robots that are not equipped with communication devices. The effectiveness of the proposed method for moving obstacles as well as mobile robots, was confirmed through computational simulations. The simulation results were evaluated by the total impulse and the success probability. Dependency of these indices on the essential parameters such as the sight radius, the noise coefficient and the decay factor were widely explored. Roles of these parameters were revealed to some extent, however, further detailed investigations and analytical explanations are necessary.

## References


[1] ABE, Y. and MATSUO, Y., "Collision Avoidance Method for Multiple Autonomous Mobile Agents by Implicit Cooperation," *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 2001.

[2] ABE, Y. and MATSUO, Y., "Cooperative Collision Avoidance Method for Multiple Autonomous Mobile Agents (in Japanese)," *Proc. 6th Robotics Symposia*, 2001.

[3] PRASSLER, E. et al., "MAid: A Robotic Wheelchair Operating in Public Environments," *Sensor Based Intelligent Robots*, LNAI 1724, pp. 68-95, 1999.

[4] FIORINI, P. and SHILLER, Z., "Motion Planning in Dynamic Environments Using Velocity Obstacles," *Intl. J. of Robotics Research*, Vol. 17, No. 7, pp. 760-772, 1998.

[5] SIMMONS, R., "The Curvature-Velocity Method for Local Obstacle Avoidance," *Proc. IEEE Intl. Conf. on Robotics and Automation*, vol.4, pp.3375-3382, 1996.

[6] FUJIMORI, A. et al., "Navigation of Multiple Mobile Robots with cooperative collision avoidance," *IFAC 14th Triennial World Congress*, vol.B, pp.107-112, 1999.

[7] LARGE, F. et al., "Towards Real-Time Global Motion Planning in a Dynamic Environment Using the NLVO Concept," *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, vol.1, pp.607-612, 2002.

[8] BROCK, O. and KHATIB, O., "High-Speed Navigation Using the Global Dynamic Window Approach," *Proc. IEEE Intl. Conf. on Robotics and Automation*, vol.1, pp.341-346, 1999.

[9] SHILLER, Z. et al., "Motion Planning in Dynamic Environment: Obstacle Moving Along Arbitrary Trajectories," *Proc. IEEE Intl. Conf. on Robotics and Automation*, vol.4, pp.3716-3721, 2001.

[10] FIORINI, P. and SHILLER, Z., "Time Optimal Trajectory Planning in Dynamic Environments," *Proc. IEEE Intl. Conf. on Robotics and Automation*, vol.2, pp.1553-1558, 1996.

[11] FUJIMORI, A. et al., "Cooperative Collision Avoidance between Multiple Mobile Robots," *J. of Robotic Systems*, vol.17, no.7, pp.347-363, 2000.

[12] ERDMANN, P. and LOZANO-PEREZ, T., "On Multiple Moving Objects," *Proc. IEEE Intl. Conf. Robotics and Automation*, pp.1419-1424, 1986.

[13] FRAICHARD, T. and LAUGIER, C., "Path-Velocity Decomposition Revisited and Applied to Dynamic Trajectory Planning," *IMPRECISE Proc. IEEE Intl. Conf. on Automat. and Rotot.*, Vol. 2, pp.40-45, 1993.