Robotica (2011) volume 29, pp. 1059–1073. © Cambridge University Press 2011
doi:10.1017/S0263574711000142

Parameterized collision region for centralized motion planning of multiagents along specified paths

Jeong S. Choi†, Younghwan Yoon‡*, Myoung H. Choi§ and Beom H. Lee†

†*Department of Electrical Engineering, Seoul National University, ASRI, Kwanak-ku, Seoul, Korea.*
E-mails: Jsforce2@snu.ac.kr, bhlee@snu.ac.kr
‡*Automation R&D Center, LS Industrial Systems, Anyang, Korea. E-mail: lucidite@gmail.com*
§*Division of Electrical and Computer Engineering, Kangwon National University, Chuncheon-si, Gangwon-do, Republic of Korea*

(Received in Final Form: February 5, 2011. First published online: April 15, 2011)

SUMMARY
This paper presents closed-form analytic solutions for collision detection among multiagents traveling along specified paths. Previous solutions for centralized multiagent systems have mainly used iterative computational approaches for collision detection, which impose a heavy computational burden on the systems. In this paper, we formalize a new mathematical approach to overcoming the difficulty on the basis of simple continuous curvature (SCC) path modeling and a collision representation tool, extended collision map (ECM) method. The formulation permits all the potential collisions to be detected, represented, and parameterized with physical and geometric variables. The proposed parameterized collision region (PCR) method is a simple but precise, computationally efficient tool for describing complicated potential collisions with time traveled. Several simulations are presented to validate the proposed approach for use in centralized collision detectors and to compare the results with those of the iterative computational method and the proximity query package (PQP) method that are available.

KEYWORDS: Multirobot systems; Mobile robots; Motion planning; Collision detection; Centralized approach.

1. Introduction

Collision detection is a core competence of a multiagent system (MAS) which is defined as a coupled network of problem-solver entities that work together to find answers to problems that are beyond the individual capabilities.[1] It requires effective algorithms to enable agents to quickly and exactly check, with each other, for potential crashes while moving to a goal or executing their particular tasks. A practical algorithm for real-time and exact collision detection is often a major bottleneck in centralized MASs since most of computation times were spent detecting potential collisions,[2–8] and it has been required to reduce them for real-time motion planning. In addition, the solution for collision detection is an essential prior condition for other following procedures such as conflict resolution and plan optimization; hence collision detection is considered an important issue in motion planning.

Early approaches to motion planning are summarized in ref. [9, 10], and most recent contributions to the field are addressed in ref. [11, 12]. In the literature, multiagent motion planning is traditionally classified into two approaches, namely, *centralized* and *distributed*. In the centralized approaches, a supervisory system plans all motions of every agent interactively with a wide-range communication system and a position detecting system. The merit of the centralized approach is its ability to guarantee completeness and optimality of the system, but its demerits are the high dimensionality of the configuration spaces (or C-spaces) and the high computational complexity.[9] Akella and Hutchinson[13] have shown that the multiagent optimization problem is nondeterministic polynomial-time hard (NP-hard). On the other hand, the distributed approaches allow each agent to control itself independently, and such approaches are clearly less computationally intensive, but intrinsically incapable of satisfying the completeness and optimality requirements. To date, the distributed methods have been popular because many systems in the real world are naturally distributed and appear to work well. However, small attention has been steadily given to the merits of the centralized approach that can be very helpful for commercial multiagent applications, such as the KIVA system.[14]

A centralized MAS can compute the feasible motions of agents by dealing with the two main problems that need to be addressed, namely, the path planning problem (PPP) and the velocity planning problem (VPP).[13,15] Over the past decades, while studies for centralized motion planning have been conducted on the PPP and the VPP, there are few studies on collision detection. For practical centralized MASs, however, it has been addressed to develop a collision detection method that is computationally efficient in the studies[3,6] since a heavy computational burden (e.g., more than 90% of the total computation time)[2,13] was imposed on the motion planners. Most of the studies on collision detection in robotics have used the algorithms for distributed systems[16–23] in which moving agents are assumed to be

* Corresponding author. E-mail: lucidite@gmail.com

static and immediate reaction to the agents is generated. For this reason, they are of limited use in detecting all possible collisions from a long-term point of view.

Early researches on collision detection between two agents moving along a specified path can be found in collision-free motion planning of robot manipulators.[24–28] Kant and Zucker[15] showed that potential collisions are obtained by intersecting the swept volume of the obstacles, with the given path of a robot in Cartesian space and the collision detection result can be represented in the path-time space. Lee and Lee[29] have investigated the same problem and developed an effective collision detection algorithm, referred to as the *collision map*, which consists of the trajectory of a robot and several collision regions (CRs) that are obtained from a geometric analysis along time and a similar notion of *coordinated space* has been proposed in ref. [30]. However, these studies have encountered two difficulties: how to determine a trade-off solution between precision and the computation time and how to accurately describe the CRs representing potential collisions. To overcome these difficulties, the CRs have been approximated as rectangles[15,29] or ellipses,[28] at the expense of precision, and the approximated solutions have been found in the studies.[31,32]

The following papers have demonstrated the renewed interest in improving the precision in collision detection because precision is directly related to the quality of the motion planning. Chang *et al*.[33] have discretized the collision map and used an iterative computational method to obtain the time-optimal solution; Lavalle and Hutchinson[7] and Akella and Hutchinson[13] have presented a similar approach to perform a discrete-time analysis that computes the volume swept by each robot. Peng and Akella[5] have identified the potential collisions along paths by combining the disjunctive constraints for the robots to formulate a mixed integer nonlinear programming problem, and Ji *et al*. have used a discretization technique to represent complicated CRs for multiagents.[34] Owen and Montano have introduced the dynamic object velocity set method to detect and represent potential collisions.[35] These studies intensively deal with developing high-precision descriptions in addressing the collision detection problem at the expense of computation time.

In recent years, little progress has been made on collision detection regarding the aforementioned studies. Instead, several collision checking packages, such as the proximity query package (PQP),[36] software library for interference detection (SOLID),[37] V-Clip,[38] I-Collide,[39] SWIFT++,[40] and Callisto,[41] have been used as collision checkers for multiagents,[5] mainly contributing to sampling-based studies.[3,4,42] The sampling-based approach has proven to be very efficient for high-dimensional problems, thus, garnering a great amount of attention, most famously with the probabilistic road map (PRM)[2] and the rapidly-exploring random tree (RRT).[43] The collision checking packages were originally developed to compute the distance between static polyhedral bodies and thus suit the sampling-based approach well; however, these packages do not well suit other approaches in which a description of an aspect of collisions regarding the time traveled is required. In recent papers,[5,8,44–47] the merits of coupled and decoupled approaches have been addressed, and thus it is necessary to develop practical collision detectors that are both precise and computationally efficient for nonprobabilistic approaches.

This paper presents closed-form analytic solutions for collision detection among multiagents traveling along specified paths. To reduce the computational burdens and overcome the imprecision problems concurrently, a mathematical approach was formalized for performing the collision detection task. For the problem formulation, simple continuous curvature (SCC) paths first introduced by Scheuer and Fraichard[48] were used in the path generation. Potential collisions in the paths were classified into four intersection types, and these were formulated by geometric analysis based on the extended collision map (ECM) method. From the formulation, the potential collisions were detected, represented, and parameterized with physical and geometrical constraints. In simulation studies carried out, 126 potential collisions among 12 agents with a traveled time of 24.4 s were computed in under 0.015 s. The proposed parameterized collision region (PCR) method enables the detection and description of the potential collisions using simple closed-form equations and the two difficulties in collision detection, namely, heavy computational burdens and low-precision descriptions, can be overcome by the deterministic method.

This paper is organized as follows: Section 2 describes the important aspects of the overall problem in detail. Section 3 presents the formulation of the potential collision detection problem and summarizes the current obtained results. Simulations and comparisons with previous methods are presented in Section 4, and Section 5 describes in detail the proposed approach and conclusions are given in Section 6.

2. Problem Description

2.1. Definitions

In order to generate greater precise in study of collision detection among multiagent motion, we first define *motion* for robots $\mathbf{p}(t) = (x(t), y(t))$, which is a position parameterized by time. Two functions for the length of travel and the path being followed are defined as follows:

$$\lambda : t \in [t^s, t^e] \mapsto \lambda(t) = l \in [0, l^e],$$
$$\boldsymbol{\rho} : l \in [0, l^e] \mapsto \boldsymbol{\rho}(l) = \mathbf{p} = (x, y) \in Q_{free}, \qquad (1)$$

where $\boldsymbol{\rho}(0) = \mathbf{p}^s$, $\boldsymbol{\rho}(l^e) = \mathbf{p}^e$ and Q_{free} is free C-space.[11] From the definitions, motion can be defined as follows:

$$\boldsymbol{\rho} \circ \lambda : t \in [t^s, t^e] \mapsto \boldsymbol{\rho}(\lambda(t)) = \mathbf{p}(t) \in Q_{free}. \qquad (2)$$

The traveled length is given by a speed function determined by kinodynamic constraints (or maximum velocity and acceleration) as follows:

$$\upsilon(t) = \lambda'(t), |\upsilon(t)| \leq \upsilon_{\max}, |\upsilon'(t)| \leq a_{\max}. \qquad (3)$$

The configuration of a robot system is characterized by the complete specification of each position at every point of that

https://doi.org/10.1017/S0263574711000142 Published online by Cambridge University Press

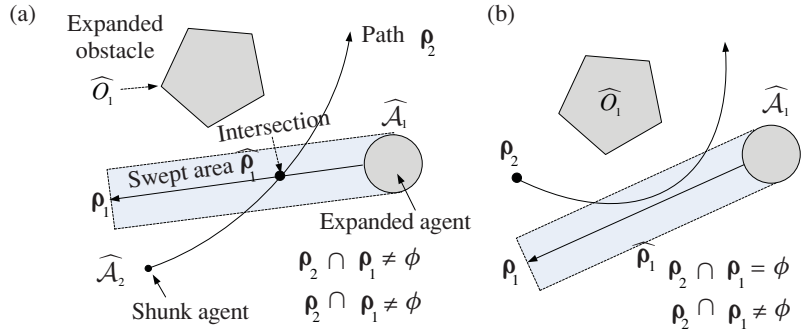

Fig. 1. (Colour Online) Two paths and one obstacle in C-space (a) There is an intersection between paths and a potential collision and (b) no intersection, but a potential collision.

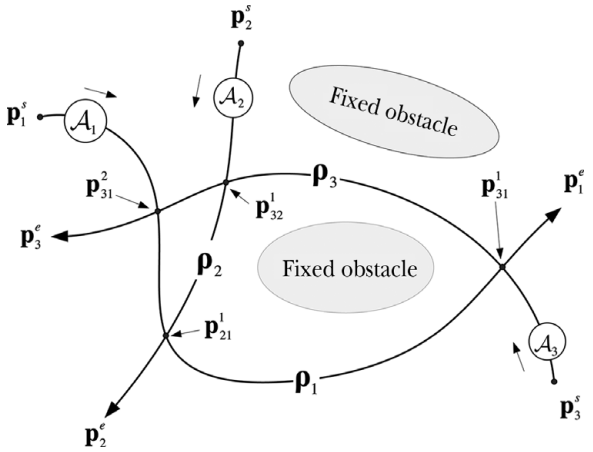

Fig. 2. Three agents with each path and four intersections (for definitions, see Section 2.1).

system; the C-space is the space of all possible configurations of the system. Thus, a configuration is simply a point in the abstract C-space.[11] On the basis of these concepts, the followings are defined:

$\mathcal{A}_i$: The ith agent with a radius r_i in multiagents $\mathcal{A}_i (i = 1 \ldots N)$. On the basis of the C-space concept, a robot is shrunk (or reduced) to a point and meanwhile an obstacle expands and is treated as a new obstacle (see Fig. 1). Let $\widehat{\mathcal{A}_j}$ and $\widehat{\mathcal{A}_i} (i \neq j)$ be the expanded obstacle and the shrunk robot, respectively. Under circular modeling, the radius for the obstacle and the robot is zero, $\widehat{r_j} = 0$, and the sum of their radii, $\widehat{r_i} = r_i + r_j$, respectively.

$\mathbf{p}_{ij}^k$: The kth intersection of $\boldsymbol{\rho}_i$ and $\boldsymbol{\rho}_j$, i.e., $\boldsymbol{\rho}_i \cap \boldsymbol{\rho}_j = \mathbf{p}_{ij}^k$ $(i, j = 1 \ldots N, k = 1 \ldots K)$ in a set of intersections I_i in $\boldsymbol{\rho}_i$ (See Fig. 2). It is a reference point for a potential collision between the two paths. In the C-space, $\widehat{\boldsymbol{\rho}_i}$ is the whole area swept by $\widehat{\mathcal{A}_i}$ along $\boldsymbol{\rho}_i$ as shown in Fig. 1. Note that if $\boldsymbol{\rho}_i \cap \widehat{\boldsymbol{\rho}_j} \neq \phi$, a potential collision can be said to exist when there is no intersection between two paths, i.e., $\boldsymbol{\rho}_i \cap \boldsymbol{\rho}_j = \phi$.

Let a segment of a path intersected by another agent at an intersection between two agents be *a collision segment*. To formulate potential collisions involving the collision segments, the following definitions were utilized on the basis of our earlier work.[29,34]

CR_{ij}^k: The CR corresponding to $\mathbf{p}_{ij}^k$ in a set of CRs, CR_i. Its length and time range is expressed using $[l^-, l^+]$ and $[t^a, t^b]$, respectively (for details, see Section 3.1).

λ_{ij}^k: A contour equation of CR_{ij}^k that can be decomposed into an upper and a lower function, $\lambda_{ij}^{k+}(t)$ and $\lambda_{ij}^{k-}(t)$, $\forall t\ t \in [t^a, t^b]$.

From the definitions above, in the presence of multiple agents, the collision-free condition for a single agent $\mathcal{A}_i$ is expressed as follows:

$$\forall t \in \left[t_i^s, t_i^e\right] \mathbf{p}_i(t) \cap \widehat{\mathbf{p}_j(t)} = \phi, \quad i \neq j\ (j = 1 \ldots N). \tag{4}$$

We extend the last formula to multiagents, finally achieving the following expression:

$$\forall t\ \ \mathbf{p}_i(t) \cap \widehat{\mathbf{p}_j(t)} = \phi, \quad i \neq j\ (i, j = 1 \ldots N), \tag{5}$$

which is the multiagent collision-free condition. In brief, the definitions lead to the following multiagent motion planning problem:

Multiagent Motion Planning Problem: *Given N multiagents with an initial configuration and a goal configuration, and physical constraints, find the agent motions $\mathbf{p}_i(t)\, t_i^s \leq t \leq t_i^e (i = 1 \ldots N)$ such that they obey the collision-free condition defined by Eq.* (5).

Having clarified the problem, we now turn our attention to the formulation of collision detection for Eq. (5) in the following sections.

2.2. SCC path modeling and assumptions

The condition for collision-free motion of $\mathcal{A}_i$ in Eq. (4) can be rewritten using the definitions for the traveled length and the CR:

$$\forall t\ \lambda_i(t) \cap CR_i = \phi, \tag{6}$$

which indicates that $\lambda_i(t)$ does not touch or cross any CRs for the entire traveled time of $\mathcal{A}_i$. In the minimum-time trajectory generation,[5,13,15,29,32–34] it is necessary to formulate CR_i in the last equation. In previous studies, however, a CR has been obtained by approximation[15,28,29,31,32] or discretization,[5,7,13,33,34] and thus we focus on the exact formulation of CR_i. The following assumptions were made:

https://doi.org/10.1017/S0263574711000142 Published online by Cambridge University Press

Assumption 1: Paths are planned using straight lines and circular arcs.
Assumption 2: Agents move with constant speed near intersections.
Assumption 3: Agents are represented as circles.

Assumption 1 is based on the concept of SCC paths, which are continuous paths consisting of two simple components (lines and circular arcs), as proposed by Scheuer and Fraichard.[48] Many papers dealing with robot motion have applied this concept to describe the paths of mobile robots, especially car-like robots.[5,35,49] Moreover, this kind of smooth path modeling method, composed of lines and curves, has been used in previous studies (e.g., Hermite curves[32,50] and the sequences of line segments[51]). It is interesting to consider the collision detection problem with arbitrary paths but this is not our present concern.

Assumption 2 could be accepted as reasonable for practical applications since the motion of mobile robots are mainly planned with constant speed except for near start and end point in a centralized system. This assumption is called the constant velocity (CV)[52] and the exact range to which it is applied is determined by $[l^-, l^+]$. It is possible to use various geometric primitives[53] instead of a circle in *Assumption 3*, but the circular model satisfactorily formulates our problem without losing generality.

3. Parameterized Collision Region

The overall goal of this paper is to parameterize the CR for fast collision detection and exact collision description. This section introduces the basic concept of the CR that was originally defined in the collision map[29,34] and its detailed formulation. For ensuring some simplicity in the formulation, the potential collisions are classified into four intersection types, and local coordinates systems are used in analyzing the potential collisions at the intersections.

3.1. Overview of the extended collision map

The ECM is the extension of the collision map, which is a two-dimensional figure representing a trajectory and information regarding potential collisions of two agents simultaneously.[29] It is assumed that the two agents have their own priorities, and that the higher priority agent $\mathcal{A}_1$ adheres to its original trajectory. Thus, the collision detection aspects are examined from the viewpoint of the lower priority robot $\mathcal{A}_2$. Figure 3 indicates a scheme of representing a potential collision in the collision map that is generated whenever two paths cross each other. For simplification, the concept of C-space is introduced. Since two agents collide with each other when the distance between their centers becomes smaller than the sum of their radii, $\mathcal{A}_1$ is expanded to a new agent $\widehat{\mathcal{A}_1}$ whose radius is the sum of the radii, and $\mathcal{A}_2$ is shrunk to a point agent $\widehat{\mathcal{A}_2}$ whose radius is zero.

Next, the potential collision information is obtained by inspecting the intersection between $\widehat{\mathcal{A}_1}$ and $\boldsymbol{\rho}_2$ with time because the two agents have a potential collision if $\boldsymbol{\rho}_2$ is intersected by $\widehat{\mathcal{A}_1}$. The union of these collision lengths along time can be drawn as a connected region, as shown in Fig. 3, and this is called a *CR*. If a trajectory touches or crosses

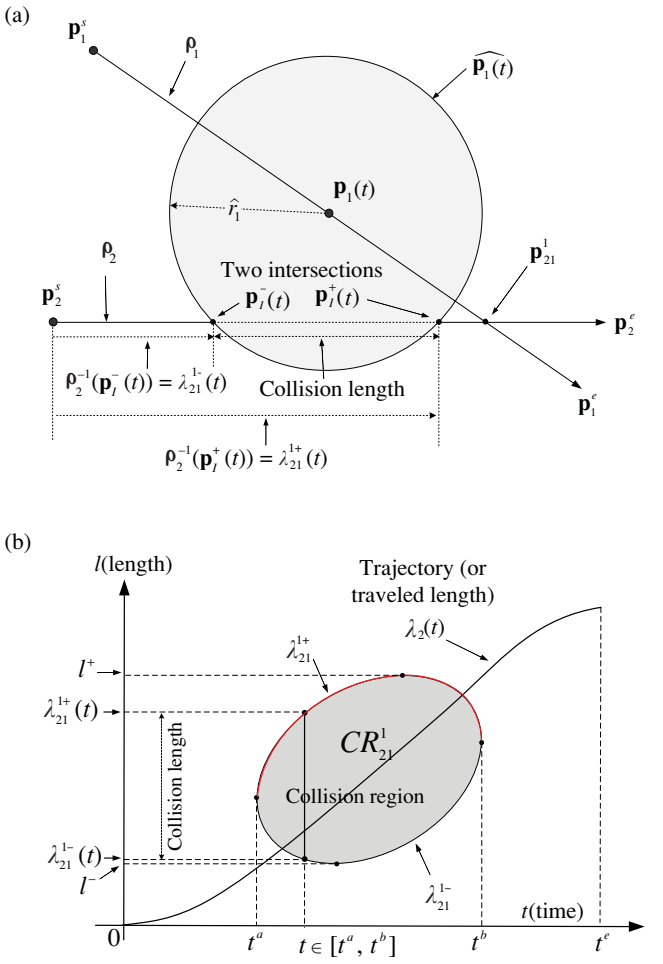

Fig. 3. (Colour Online) (a) Collision detection by geometric analysis between two agents. (b) CR generation by collecting the collision lengths over time (for notations, see Section 2.1).

a CR in a collision map, two agents will physically collide with each other. More details for collision detection using the collision map will be discussed in the following sections.

The collision map has been extended to multiagent motion planning, and it is referred to as an ECM.[34] The CRs in the ECM are generated by sampling continuous paths and trajectories of two agents and by computing the distances between the sampling points. In ref. [34], two limits are removed from the original concept: the numbers of agents and the shapes of the paths. That is, the ECM is designed to describe potential collisions among a large number of agents traveling along arbitrary paths.

In the previous study,[34] the ECM method has been developed for decoupled motion planners, and the detected collisions in the ECM were resolved with a priority assignment method so that the burden of avoiding collisions between two agents fell on the lower priority agent. In the decoupled approach, the highest priority agent did not need its own ECM for collision avoidance, and all agents did not need to consider or represent the potential collisions with the lower priority agents. This paper, however, aims to provide all information on potential collisions for all kinds of motion planners. For this reason, we present all ECMs including

https://doi.org/10.1017/S0263574711000142 Published online by Cambridge University Press

that for the highest priority agent and describe all potential collisions with the lower priority agents.

3.2. Procedures for PCR formulation

In the C-space, two agents are modeled, one as an expanded agent and the other as a point agent, respectively (see Fig. 3). From this modeling and the definitions in Section 2, CR_{ij}^k is formulated by following four procedures in sequence: (1) specifying two motions $\mathbf{p}_i(t)$, $\mathbf{p}_j(t)$ and intersections $\mathbf{p}_{ij}^k (i > j)$ from the given paths and kinodynamic constraints (see Section 2); (2) formulating two contour equations for time t at CR_{ij}^k based on collision map generation, $\exists t\ [\lambda_{ij}^{k-}(t), \lambda_{ij}^{k+}(t)]$; (3) deriving a time range (or domain) from the formula ${}_{cs}t_{ij}^k = [t_{ij}^{k,a}, t_{ij}^{k,b}]$; and (4) parameterizing $CR_{ij}^k(\cdot)$ with only given physical and geometrical constraints.

In SCC path modeling,[48] there are four possible cases for describing an intersection between two paths $\boldsymbol{\rho}_i$, $\boldsymbol{\rho}_j$ because every part of the paths can be modeled either as a straight line or as a circular arc. For this reason, we classified the intersections in the paths into four types: *SS*, *SA*, *AS*, and *AA*, where S and A represent a straight line and a circular arc, respectively, and a similar approach is also found in ref. [51]. Two characters for the type representation indicate the path types of two collision segments sequentially. For example, if a path is a straight line and the other is a circular arc, then the intersection between them is expressed as SA (straight–arc) type. According to the procedures and classification, the following sections will present a mathematical analysis of the potential collisions.

3.3. Coordinate systems {A} and {B}

Multiagents generally have complicated potential collisions when their paths are planned in common space. To systematically describe them, we defined a global coordinate system whose origin is the starting point of a path, and local coordinate systems are attached to every intersection on the path as shown in Fig. 4. This is because it is convenient to analyze each potential collision with respect to the local coordinate attached to the intersections and to describe the collisions together in the global coordinates.

Let $\{L\}_i^s$ and $\{L\}_{ij}^k$ be the coordinates attached to the start point $\mathbf{p}_i^s$ and the intersection $\mathbf{p}_{ij}^k$, respectively. For simple notation, $\{L\}_i^s = \{A\}$ and $\{L\}_{ij}^k = \{B\}$ are denoted in the coordinate systems. The local time ${}^Bt$ is set to be zero when $\mathcal{A}_j$ passes $\mathbf{p}_{ij}^k$. If $\boldsymbol{\rho}_i$ has H intersections with other paths, $\{A\}$ includes multiple local coordinates $\{B\}_h$ $(h = 1, \ldots H)$.

Next, the mapping of a point from $\{B\}$ to $\{A\}$ is considered, and this is essential in systematically representing potential collisions and in manipulating the mathematical quantities describing them. Recall that a CR is generated from the traveled time along $\boldsymbol{\rho}_j$ and the traveled length along $\boldsymbol{\rho}_i$, and that the motions of the two agents are planned individually; $\{B\}$ can be represented relative to $\{A\}$ with ${}^BT$ and ${}^BL$ as follows: first, the traveled time from the origin of $\{A\}$ to that of $\{B\}$ along $\boldsymbol{\rho}_j$ is given by simple subtraction: ${}^BT = {}^At - {}^Bt$. Similarly, the distance is given by ${}^BL = {}^Al - {}^Bl$. Thus, in a time-length space, the transformation mapping from an arbitrary point ${}^B\mathbf{q}$ to ${}^A\mathbf{q}$ is described with the following matrix

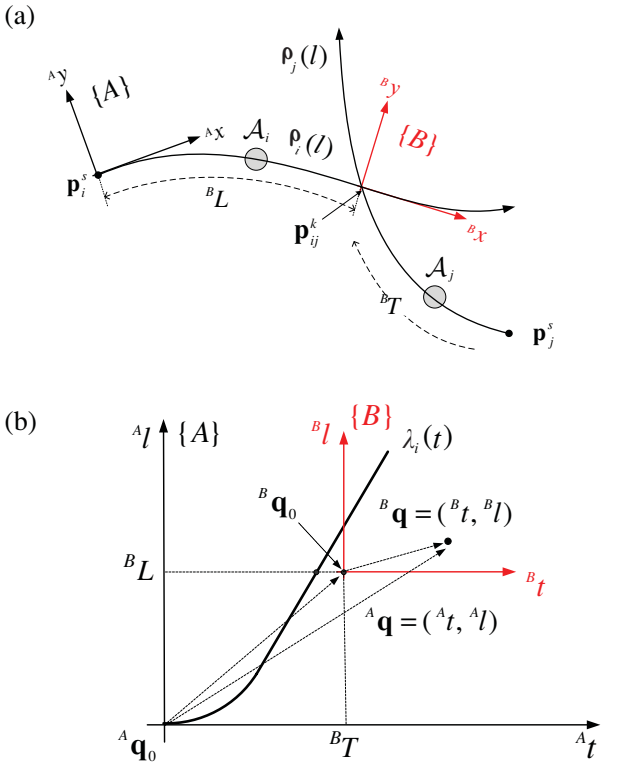

Fig. 4. (Colour Online) Local coordinates system in (a) Cartesian space and in (b) time-length space (the notations used are presented in Table I).

operation:

$$ {}^A\boldsymbol{q} = \mathbf{M}_1 \cdot {}^B\boldsymbol{q} + \mathbf{M}_2 \quad \text{or} \quad \begin{bmatrix} {}^At \\ {}^Al \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^Bt \\ {}^Bl \end{bmatrix} + \begin{bmatrix} {}^BT \\ {}^BL \end{bmatrix}, \tag{7} $$

where $\mathbf{M}_1$ and $\mathbf{M}_2$ are a rotation and a translation matrix, respectively. Since the rotation matrix is the 2×2 identity, it is possible to map a potential collision from $\{B\}$ to $\{A\}$ by only involving a translation. Note that the origin of $\{B\}$ usually lies out of the trajectory (or traveled length) $\lambda_i(t)$ as shown in Fig. 4 because the time information of the origin $({}^BT, {}^BL)$ is calculated from the motion of $\mathcal{A}_j$, not from that of $\mathcal{A}_i$. All the notations defined in Section 2 can be expressed relative to $\{A\}$ or $\{B\}$, e.g., ${}^A\boldsymbol{\rho}_i$ and ${}^B\boldsymbol{\rho}_i$. These coordinate systems are used here to locally describe the four types of CRs, and then to express them globally. The four types are presented on the basis of the systems in $\{B\}$ as follows.

3.3.1. Straight–straight (SS) type (Fig. 5). In the SS type, two agents $\mathcal{A}_i$, $\mathcal{A}_j$ each follow straight-line paths $\boldsymbol{\rho}_i$, $\boldsymbol{\rho}_j$ as illustrated in Fig. 5. For simple notations, assign $i = 2$, $j = 1$, and $k = 1$. It is possible to replace i, j, and k indicating CR_{ij}^k with a superscript B since the local coordinate $\{B\}$ indicates parameters involving CR_{ij}^k, e.g., $\theta_{21}^1 = {}^B\theta$, ${}_{cs}l_{21}^1 = {}_{cs}^Bl$, ${}_{cs}t_{21}^1 = {}_{cs}^Bt$, $\lambda_{21}^1 = {}^B\lambda$. In addition, we can omit B since all notations in the following Sections from 3.3.1 to 3.3.2 are represented only with respect to $\{B\}$. Using the notations, the contour of the expanded agent is simply denoted as

$$ \widehat{\mathbf{p}_1(t)} = \widehat{\boldsymbol{\rho}_1(\lambda_1(t))} = \{(x(t), y(t)) | (x(t) - p_{1x}(t))^2 + (y(t) - p_{1y}(t))^2 = \hat{r}_1^2\}, \tag{8} $$

https://doi.org/10.1017/S0263574711000142 Published online by Cambridge University Press

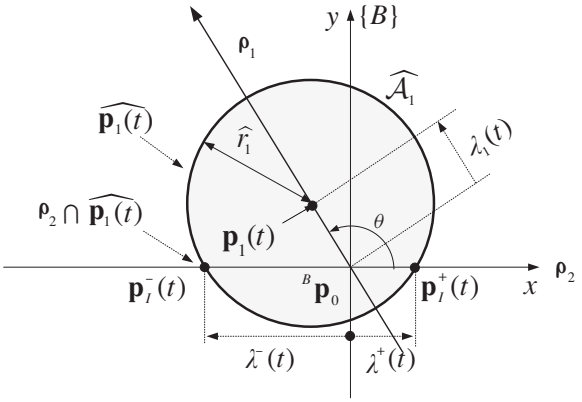

Fig. 5. Straight–straight (SS) type.

where

$$\mathbf{p}_1(t) = \{(p_{1x}(t),\, p_{1y}(t)) | p_{1y}(t) = \tan\theta \cdot p_{1x}(t)\} = (\lambda_1(t) \cdot \cos\theta, \lambda_1(t) \cdot \sin\theta) \quad (9)$$

and under assumption 2, $\lambda_1(t) = v_1 \cdot t$, where v_1 is a constant speed of $\mathcal{A}_1$.

We now specify the accurate range of a collision segment to which the proposed model is applied. From the definitions in Section 2 and Table II, the length range at t (or collision length) is expressed as

$$\exists t\ {}_{cs}l(t) = [\lambda^-(t), \lambda^+(t)] = [\boldsymbol{\rho}_2(\mathbf{p}_I^-(t)), \boldsymbol{\rho}_2(\mathbf{p}_I^+(t))],$$
$$\lambda^-(t) \le \lambda^+(t)\, s.t.\, \mathbf{p}_I^{\pm}(t) = \boldsymbol{\rho}_2 \cap \widehat{\boldsymbol{\rho}_1(\lambda^1(t))}. \quad (10)$$

In the SS type, $\boldsymbol{\rho}_2$ coincides with the x-axis and thus

$$\boldsymbol{\rho}_2 = \{(p_{2x},\, p_{2y}) | p_{2x} \in \mathcal{R},\, p_{2y} = 0\}. \quad (11)$$

Since $p_{2y} = 0 \Leftrightarrow \forall t\ p_{2y}(t) = 0$, we replace $(x(t), y(t))$ in Eq. (8) with (p_{2x}, p_{2y}) to solve Eq. (10), and obtain

$$p_{2x}^{\pm}(t) = p_{1x}(t) \pm \sqrt{\hat{r}_1^2 - p_{1y}^2(t)} \quad \text{and} \quad \mathbf{p}_I^{\pm}(t) = (p_{2x}^{\pm}, 0). \quad (12)$$

Thus,

$$\exists t\ [\lambda^-(t), \lambda^+(t)] = [p_{2x}^-, p_{2x}^+] = \left[p_{1x}(t) - \sqrt{\hat{r}_1^2 - p_{1y}^2(t)},\, p_{1x}(t) + \sqrt{\hat{r}_1^2 - p_{1y}^2(t)}\right]. \quad (13)$$

On the basis of the formulas above, we derive the length range in a time-independent form and the time ranges of a CR, i.e., ${}_{cs}l$ and ${}_{cs}t$. Two methods, relying on geometry and calculus, can be used for derivation, and both methods give exactly the same result. We here formulate the ranges with calculus. Combining Eqs. (8) and (9) and substituting $(p_{2x}, 0)$ into the equation gives

$$(1 + \tan^2\theta)p_{1x}(t)^2 - 2p_{2x}p_{1x}(t) + p_{2x}^2 - \hat{r}_1^2 = 0, \quad (14)$$

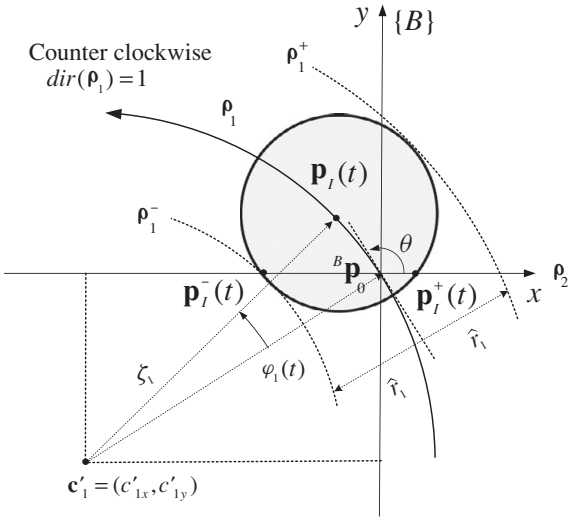

Fig. 6. Straight–arc (SA) type.

which is a quadratic equation in $p_{1x}(t)$. When two agents collide with each other, $p_{1x}(t)$ has one double or two real roots. By letting the discriminant of Eq. (14) $(p_{2x}^2 - (1 + \tan^2\theta)(p_{2x}^2 - \hat{r}_1^2))$ be equal to or greater than zero, the following is obtained:

$${}_{cs}l = [l^-, l^+] = [\min_t(p_{2x}^-(t)), \max_t(p_{2y}^+(t))] = [-\hat{r}_1/\sin\theta,\ \hat{r}_1/\sin\theta]. \quad (15)$$

The time range of CR is defined as

$${}_{cs}t = [t^a, t^b] = [\min(t), \max(t)]\, s.t.\, \forall t\ \boldsymbol{\rho}_2 \cap \widehat{\boldsymbol{\rho}_1(\lambda_1(t))} \neq \phi, \quad (16)$$

and it can be derived from the condition that the radicand $(\hat{r}_1^2 - p_{1y}^2(t))$ in Eq. (12) should be equal to or greater than zero. This is because if the radicand is negative, it means that the intersection defined in Eq. (16) does not exist, i.e., $\boldsymbol{\rho}_2 \cap \widehat{\boldsymbol{\rho}_1(\lambda_1(t))} = \phi$. The physical meaning of positive radicand is that two agents contact or collide with each other. From this condition, ${}_{cs}t$ is obtained as

$${}_{cs}t = [t^a, t^b] = [-\hat{r}_1/(v_1 \cdot \sin\theta), \hat{r}_1/(v_1 \cdot \sin\theta)]. \quad (17)$$

3.3.2. Straight–arc (SA) type (Fig. 6). This section examines the case in which $\mathbf{p}_1(t)$ is modeled as a circular arc. Since the path $\boldsymbol{\rho}_1$ is a straight line in this type, two intersections between them are given by Eq. (12). Now we formulate $\mathbf{p}_1(t)$ using three parameters $\mathbf{c}_1$, ς_1, and $\varphi_1(t)$, based on polar coordinates as follows (for the definitions, see Table II):

$$\mathbf{p}_1(t) = \left\{p_{1x}(t),\, p_{1y}(t) | (p_{1x}(t) - c_{1x})^2 + (p_{1y}(t) - c_{1y})^2 = \varsigma_1^2\right\}. \quad (18)$$

Since two symmetric arcs satisfy the last equation mathematically, they can be distinguished by using $\mathbf{c}'_1 = (c'_{1x}, c'_{1y})$

https://doi.org/10.1017/S0263574711000142 Published online by Cambridge University Press

Table I. Notation and meanings used in Fig. 4.

Notation	Meaning
$\{A\}$, $\{B\}$	Local coordinates attached to $\mathbf{p}_i^s$ and $\mathbf{p}_{ij}^k$, respectively. The x-axis is defined as the tangent line of $\boldsymbol{\rho}_i$.
${}^A\mathbf{q}$, ${}^B\mathbf{q}$	A point represented with respect to origin of $\{A\}$ and $\{B\}$, respectively.
${}^BT$, ${}^BL$	Traveled time from $\mathbf{p}_j^s$ to $\mathbf{p}_{ij}^k$ along $\boldsymbol{\rho}_j(l)$
	Traveled length from $\mathbf{p}_i^s$ to $\mathbf{p}_{ij}^k$ along $\boldsymbol{\rho}_i(l)$

and $\mathbf{c}''_1 = (c''_{1x}, c''_{1y})$, denoted as

$$\begin{cases} \mathbf{c}'_1 = (\varsigma_1 \cos(\theta + \pi/2), \varsigma_1 \sin(\theta + \pi/2)), \text{counter clockwise} \\ \mathbf{c}''_1 = (\varsigma_1 \cos(\theta - \pi/2), \varsigma_1 \sin(\theta - \pi/2)), \text{clockwise} \end{cases} \tag{19}$$

where *clockwise* and *counter clockwise* represent the direction of circular path $\boldsymbol{\rho}_2$. Note that the position of the virtual agent, with respect to Cartesian coordinates, can be described by a rotation matrix operation:

$$[p_{1x}(t), p_{1y}(t)] = \begin{bmatrix} 1 - \cos\varphi_1(t) & dir(\boldsymbol{\rho}_1) \cdot \sin\varphi_1(t) \\ -dir(\boldsymbol{\rho}_1) \cdot \sin\varphi_1(t) & 1 - \cos\varphi_1(t) \end{bmatrix} \begin{bmatrix} c_{1x} \\ c_{1y} \end{bmatrix}, \tag{20}$$

where the direction of $\boldsymbol{\rho}_1$ is defined as $dir(\boldsymbol{\rho}_1) = \begin{cases} 1 : counter\ clockwise \\ -1 : clockwise \end{cases}$.

To derive ${}_{cs}l$ from geometrical analysis, the following is defined:

$$\begin{aligned} \boldsymbol{\rho}_1^{\pm} &= \{(x(t), y(t)) | (x(t) - c_{1x})^2 + (y(t) - c_{1y})^2 \\ &= (\varsigma_1 \pm \widehat{r_1})^2\}, \end{aligned} \tag{21}$$

and two intersections are found between x-axis and $\boldsymbol{\rho}_1^{\pm}$, which gives

$$[l^-, l^+] = \begin{cases} \left[\sqrt{(\varsigma_1 + \widehat{r_1})^2 - c_{1y}^2} - |c_{1x}|, -\sqrt{(\varsigma_1 - \widehat{r_1})^2 - c_{1y}^2} + |c_{1x}|\right], c_{1x} \geq 0 \\ \left[-\sqrt{(\varsigma_1 - \widehat{r_1})^2 - c_{1y}^2} + |c_{1x}|, \sqrt{(\varsigma_1 + \widehat{r_1})^2 - c_{1y}^2} - |c_{1x}|\right], c_{1x} < 0. \end{cases} \tag{22}$$

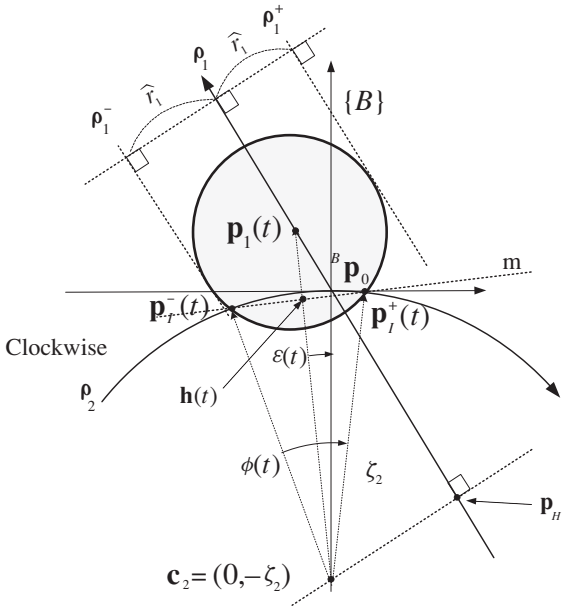

Fig. 7. Arc–straight (AS) type.

Then ${}_{cs}t$ is derived from substituting the last equation into $\widehat{r_1} \geq |p_{1y}(t)|$ in Eq. (12), and thus

$$[t^a, t^b] = \left[\left(\sin^{-1}\left(\frac{-\widehat{r_1} - c_{1y}}{\varsigma_1}\right) - \theta_{SA}\right)\Big/\omega_1, \left(\sin^{-1}\left(\frac{\widehat{r_1} - c_{1y}}{\varsigma_1}\right) - \theta_{SA}\right)\Big/\omega_1\right], \tag{23}$$

where $\theta_{SA} = \tan^{-1}(c_{1y}/(dir(\boldsymbol{\rho}_1) \cdot c_{1x}))$ and $\omega_1 = \upsilon_1(t)/\zeta_1$.

3.3.3. Arc–straight (AS) type (Fig. 7). Types AS and AA differ from SS and SA in that $\mathbf{p}_I^{\pm}(t)$ may not lie on the x-axis, $\mathbf{p}_I^{\pm}(t) = (p_{2x}^{\pm}, p_{2y}^{\pm})$, $p_{2y}^{\pm} \neq 0$ since $\boldsymbol{\rho}_2$ is a circular arc, and thus it does not coincide with the x-axis. Given that $\mathbf{c}_2 = (c_{2x}, c_{2y})$ and ς_2 are the center and the radius of the

Table II. Local notation and meanings used in figures for four types.

Notation	Meaning
θ	Angle from the direction of $\boldsymbol{\rho}_2$ to that of $\boldsymbol{\rho}_1$ at ${}^B\mathbf{p}_0$
$\mathbf{c}_i$, ζ_i	Center and radius of curvature of $\boldsymbol{\rho}_i$
$\varphi_1(t)$	Angle from the direction of ${}^B\mathbf{p}_0$ to that of $\mathbf{p}_1(t)$ at $\mathbf{c}'_1$ $\varphi'_1(t) = \upsilon_1(t)/\zeta_1$,
$\mathbf{p}_I^{\pm}(t)$	Two intersections between $\boldsymbol{\rho}_2$ and $\widehat{\mathbf{p}_1(t)}$, i.e., $\boldsymbol{\rho}_2 \cap \widehat{\mathbf{p}_1(t)}$
$\lambda^{\pm}(t)$	Traveled length from ${}^B\mathbf{p}_0$ to $\mathbf{p}_I^{\pm}(t)$ along $\boldsymbol{\rho}_2$
$\boldsymbol{\rho}_1^{\pm}$	Expanded and shrunk (or translated) circular path of $\boldsymbol{\rho}_1$ by $\widehat{r_1}$
$\phi(t)$, $\varepsilon(t)$, $\mathbf{h}(t)$	$\phi(t) = \lvert\angle\mathbf{p}_I^-(t)\mathbf{c}_2\mathbf{p}_I^+(t)\rvert$, $\varepsilon(t) = \angle\mathbf{p}_1(t)\mathbf{c}'_2{}^B\mathbf{p}_0$, $\mathbf{h}(t)$ is the intersection between line $\overleftrightarrow{\mathbf{p}_I^-(t)\mathbf{p}_I^+(t)}$ and line $\overleftrightarrow{\mathbf{c}'_2\mathbf{p}_1(t)}$

https://doi.org/10.1017/S0263574711000142 Published online by Cambridge University Press

curvature of $\boldsymbol{\rho}_2$, the following is obtained:

$$\boldsymbol{\rho}_2 = \left\{(p_{2x}, p_{2y}) | (p_{2x} - c_{2x})^2 + (p_{2y} - c_{2y})^2 = \varsigma_2^2\right\}, \tag{24}$$

and the center can be denoted as one of

$$\mathbf{c}'_2 = (0, -\varsigma_2), \quad \mathbf{c}''_2 = (0, \varsigma_2). \tag{25}$$

As shown in Fig. 7, the part of $\boldsymbol{\rho}_2$ intersected by $\widehat{\mathbf{p}_1(t)}$ is also a circular arc. Thus, using $\phi(t)$ and $\varepsilon(t)$, we have

$$\exists t[\lambda^-(t), \lambda^+(t)] = [\varsigma^2(\varepsilon(t) - 0.5\phi(t)), \varsigma^2(\varepsilon(t) + 0.5\phi(t))], \tag{26}$$

where $\phi(t) = 2\cos^{-1}(\overline{\mathbf{c}_2 h(t)}/\varsigma_2)$ and $\overline{\mathbf{c}_2 h(t)}$ is the distance between $\mathbf{c}_2$ and $h(t)$. To convert this equation from polar to Cartesian coordinates, $\phi(t)$ and $\varepsilon(t)$ need to be expressed with the parameters $\mathbf{p}_1(t)$, $\mathbf{p}_2(t)$, $\widehat{r_1}$, and ς_2. The common chord m of Eqs. (8) and (24) is expressed as

$$\begin{aligned} &2p_{1x}(t)x + 2(p_{1y}(t) - c_{2y})y \\ &\quad - \left(p_{1x}(t)^2 + p_{1y}(t)^2 - \widehat{r_1}^2\right) = 0. \end{aligned} \tag{27}$$

Substituting Eq. (25) and the last equation into the distance equation from a point to a line, gives $\overline{\mathbf{c}_2 h(t)}$ in Eq. (26) and therefore

$$\phi(t) = 2\cos^{-1} \times \left(\frac{|2(p_{1y}(t) - c_{2y})c_{2y} - \left(p_{1x}(t)^2 + p_{1y}(t)^2 - \widehat{r_1}^2\right)|}{2\varsigma_2\sqrt{p_{1x}(t)^2 + (p_{1y}(t) - c_{2y})^2}} \right). \tag{28}$$

Referring to Fig. 7 gives

$$\varepsilon(t) = \tan^{-1}(p_{1x}(t)/(p_{1y}(t) - c_{2y})). \tag{29}$$

To derive the length range ${}_{cs}l$ in the AS type, two lines, $\boldsymbol{\rho}_1^{\pm}$, are used in a similar fashion to the SA type and gives

$$[l^-, l^+] = \begin{cases} [\varsigma_2(\theta_{AS}^+ - \theta_{AS}), \varsigma_2(\theta_{AS} - \theta_{AS}^-)], 0 \le \theta < \pi/2, \pi \le \theta < 3\pi/2 \\ [\varsigma_2(\theta_{AS}^- - \theta_{AS}), \varsigma_2(\theta_{AS} - \theta_{AS}^+)], \pi/2 \le \theta < \pi, 3\pi/2 \le \theta < 2\pi, \end{cases} \tag{30}$$

where $\theta_{AS}^+ = \cos^{-1}(\varsigma_2/(\overline{\mathbf{c}_2\mathbf{p}_H} + \widehat{r_1}))$, $\theta_{AS} = \pi - \theta$, and $\theta_{AS}^- = \cos^{-1}(\varsigma_2/(\overline{\mathbf{c}_2\mathbf{p}_H} - \widehat{r_1}))$.

The time range ${}_{cs}t$ is given by the boundary condition of the collision between two circles, denoted as Eqs. (8) and (24). More specifically, the collision begins when two circles touch each other internally or externally while $\widehat{\mathbf{p}_1(t)}$ crosses $\boldsymbol{\rho}_2$, which is expressed as

$$p_{1x}(t)^2 + (p_{1y}(t) - c_{2y})^2 = (\varsigma_2 \pm \widehat{r_1})^2. \tag{31}$$

Applying Eq. (9) to the last equation gives

$$\begin{aligned} &v_1^2(\sin^2\theta + \cos^2\theta)t^2 - 2(v_1 \cdot \sin\theta \cdot \varsigma_2)t \pm 2\varsigma_2\widehat{r_1} \\ &\quad - \widehat{r_1}^2 = 0 \end{aligned} \tag{32}$$

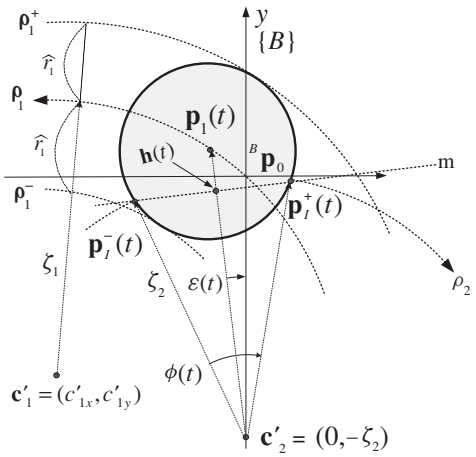

Fig. 8. Arc–arc (AA) type.

and the roots of this equation give

$$\begin{aligned} [t^a, t^b] = &\left[\left(\alpha_{AS} - \sqrt{\alpha_{AS}^2 - v_1^2\beta_{AS}}\right)/v_1^2,\right. \\ &\left.\left(\alpha_{AS} + \sqrt{\alpha_{AS}^2 - v_1^2\beta_{AS}}\right)/v_1^2\right], \end{aligned} \tag{33}$$

where $\alpha_{AS} = v_1 \cdot \sin\theta \cdot \varsigma_2$ and $\beta_{AS} = \pm 2\varsigma_2\widehat{r_1} - \widehat{r_1}^2$.

3.3.4. Arc–arc (AA) type (Fig. 8). We now consider the intersection type in which two collision segments are modeled as different arcs, denoted as Eqs. (18) and (24). In this type, two intersections between $\boldsymbol{\rho}_2$ and $\widehat{\mathbf{p}_1(t)}$ are given by Eq. (26) and the length range of CR is geometrically obtained by $\boldsymbol{\rho}_2 \cap \widehat{\boldsymbol{\rho}_1}$ $\left(\text{or } \boldsymbol{\rho}_1^{\pm}\right)$. Using the cosine formula, we have

$$[l^-, l^+] = \begin{cases} [\varsigma_2(\theta_{AA}^+ - \theta_{AA}), \varsigma_2(\theta_{AA} - \theta_{AA}^-)], c_{1x} \ge 0 \\ [\varsigma_2(\theta_{AA}^- - \theta_{AA}), \varsigma_2(\theta_{AA} - \theta_{AA}^+)], c_{1x} < 0 \end{cases}, \tag{34}$$

where

$$\theta_{AA}^+ = \cos^{-1}\left(\varsigma_2^2 + \overline{\mathbf{c}_1\mathbf{c}_2}^2 - \left(\varsigma_1 + \widehat{r_1}\right)^2\right)/(2\overline{\mathbf{c}_1\mathbf{c}_2}(\varsigma_1 + \widehat{r_1})),$$

$$\theta_{AA} = \cos^{-1}\left(\varsigma_2^2 + \overline{\mathbf{c}_1\mathbf{c}_2}^2 - \varsigma_1^2\right)/(2\overline{\mathbf{c}_1\mathbf{c}_2}\varsigma_1),$$

$$\theta_{AA}^- = \cos^{-1}\left(\varsigma_2^2 + \overline{\mathbf{c}_1\mathbf{c}_2}^2 - (\varsigma_1 - \widehat{r_1})^2\right)/(2\overline{\mathbf{c}_1\mathbf{c}_2}(\varsigma_1 - \widehat{r_1})).$$

To find ${}_{cs}t$ of the AA type, we subtract Eq. (18) from Eq. (31) and apply $c_{1x}^2 + c_{1y}^2 = \varsigma_1^2$ and $c_{2y}^2 = \varsigma_2^2$ to the equation, which gives

$$c_{1x}p_{1x}(t) + (c_{1y} - c_{2y})p_{1y}(t) = \pm\varsigma_2\widehat{r_1} + 0.5\widehat{r_1}^2. \tag{35}$$

Substituting Eq. (20) into the last equation and simplifying gives

$$\begin{aligned} &dir(\boldsymbol{\rho}_1)c_{1x}c_{2y}\sin\varphi_1(t) + (c_{1y}c_{2y} - \varsigma_1^2)\cos\varphi_1(t) \\ &\quad = \pm\varsigma_2\widehat{r_1} + 0.5\widehat{r_1}^2 + c_{1y}c_{2y} - \varsigma_1^2, \end{aligned} \tag{36}$$

https://doi.org/10.1017/S0263574711000142 Published online by Cambridge University Press

Table III. Four parameterized collision region in $\{B\}$.

		$\boldsymbol{\rho}_1$	
Path and type		Straight line	Circular arc
$\boldsymbol{\rho}_2$	Straight line	${}^{B,SS}CR_{21}(\widehat{r_1}, \upsilon_1, \theta)$	${}^{B,SA}CR_{21}(\widehat{r_1}, \upsilon_1, \theta, \varsigma_1)$
	Circular arc	${}^{B,AS}CR_{21}(\widehat{r_1}, \upsilon_1, \theta, \varsigma_2)$	${}^{B,AS}CR_{21}(\widehat{r_1}, \upsilon_1, \theta, \varsigma_1, \varsigma_2)$

which can be solved by using linear combinations of sine waves of the same period $\varphi_1(t)$. Finally, since $\varphi_1(t) = \omega_1 t$, it is possible to obtain

$$[t^a, t^b] = \left[\left(\sin^{-1}\left(\frac{\gamma_{AA}^-}{\sqrt{\alpha_{AA}^2 + \beta_{AA}^2}}\right) - \theta_{AA}\right)\Big/\omega_1, \times \left(\sin^{-1}\left(\frac{\gamma_{AA}^+}{\sqrt{\alpha_{AA}^2 + \beta_{AA}^2}}\right) - \theta_{AA}\right)\Big/\omega_1\right], \tag{37}$$

where

$$\alpha_{AA} = dir(\boldsymbol{\rho}_1)c_{1x}c_{2y}, \beta_{AA} = c_{1y}c_{2y} - \varsigma_1^2, \gamma_{AA}^+ = \varsigma_2\widehat{r_1} + 0.5\widehat{r_1}^2 + \beta_{AA},$$
$$\gamma_{AA}^- = -\varsigma_2\widehat{r_1} + 0.5\widehat{r_1}^2 + \beta_{AA}, \theta_{AA} = \tan^{-1}(\beta_{AA}/\alpha_{AA}).$$

3.4. Parameterization of CRs

So far, we formulated four types of CR with respect to the local coordinate $\{B\}$ of which origin is ${}^B\mathbf{p}$. In this section, we discuss the parameterization of CRs and their mapping from the local to the global coordinate. In Section 2, we defined a CR as:

$$CR_{ji}^k = \left\{(t, l) | t \in [t^a, t^b], \lambda_{ij}^{k-}(t) \le l \le \lambda_{ij}^{k+}(t)\right\}. \tag{38}$$

Since the intersections are classified into four types (SS, SA, AS, and AA), it is needed to distinguish the type of CR. For convenience, label these as ${}^{SS}CR_{ji}^k$, ${}^{SA}CR_{ji}^k$, ${}^{AS}CR_{ji}^k$, and ${}^{AA}CR_{ji}^k$, sequentially, and we can parameterize them all with physical and geometric constraints initially given. For instance, from Eqs. (9), (13), (17), and (39), ${}^{B,SS}CR_{21}$ can be locally obtained and parameterized only with $\widehat{r_1}$, υ_1, and θ; that is, we have a parameterized CR, ${}^{B,SS}CR_{21}(\widehat{r_1}, \upsilon_1, \theta)$. In similar fashion, all the CR types can be parameterized as shown in Table III.

To solve multiagent motion planning problems, we need to express all CRs among agents with respect to a global coordinates $\{A\}$ of each agent. It is possible to describe all parameterized CRs (PCRs) locally obtained in a single ECM using

$$\Psi : {}^BCR_{ij}^k(\cdot) \in {}^BCR_i \mapsto \Psi\left({}^BCR_{ij}^k(\cdot)\right) = {}^ACR_{ij}^k({}^BT, {}^BL, \cdot) \in {}^ACR_i,$$
$$\text{where } {}^BT = \lambda_j^{-1} \circ \boldsymbol{\rho}_j^{-1}({}^B\mathbf{p}), {}^BL = \boldsymbol{\rho}_j^{-1}({}^B\mathbf{p}), \tag{39}$$

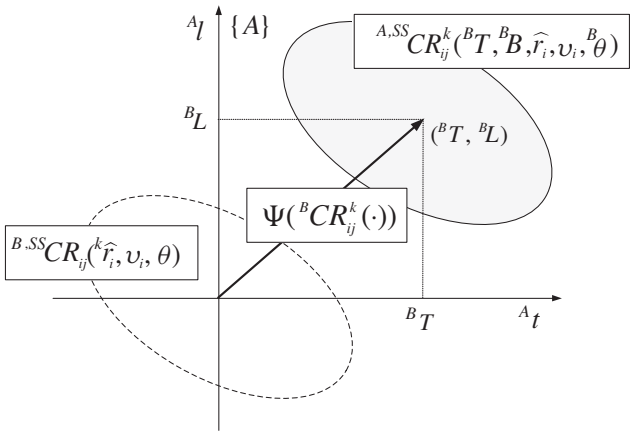

Fig. 9. Translation operator.

which is a translation mapping from $\{B\}$ to $\{A\}$ defined in Eq. (7) (see Fig. 9), and the two translation parameters ${}^BT$, ${}^BL$ are added in global representation. From the mapping functions, we summarize our results and construct Table IV. The PCR represented globally aids in describing all collisions among multiagents in a very simple way, and they are conceptually clearer than the previous methods.

4. Simulation and Experimental Results

4.1. Simulation results

We performed simulation experiments to evaluate the effectiveness of the results. A Window XP based PC with a Pentium IV 2.33 GHz processor was used for the experiments. Figure 10 illustrates 12 multiagents with specified paths, and Table V shows the simulation conditions of the physical constraints, SCC path structures, and the number of CRs of each agent. To model a congested environment with heterogeneous multiagents, various size and different velocity profiles were assigned to them. From the initial paths depicted in the figure and the path geometries for Eq. (5), 63 intersections could easily be found because the SCC-based paths were denoted mathematically, and the translation vectors, defined in Eqs. (7) and (39), were given by the information on the intersections. Then, by computing the time ranges of the collision segments from the four equations in Table IV and by putting them into the proposed contour equations, we finally obtained 126 CRs as shown in Fig. 11.

For comparison, we applied the iterative computational method introduced in our earlier work[34] and the PQP method often used in previous studies[2–6] in which a sampling resolution of 0.01 s was used to illustrate the concept. Total computation time for three approaches is presented

https://doi.org/10.1017/S0263574711000142 Published online by Cambridge University Press

Table IV. Parameterized collision region.

Step 1 (Local PCR): Formulating potential collisions with PCRs with respect to $\{B\}$			
Type	Local PCR	Time range $\forall t \in [t^a, t^b]$	Contour equation $\exists t\ [\lambda^-(t), \lambda^+(t)]$
SS	${}^{B,SS}CR_{21}(\widehat{r_1}, \upsilon_1, \theta)$	(17)	(13) where (9)
SA	${}^{B,SA}CR_{21}(\widehat{r_1}, \upsilon_1, \theta, \varsigma_1)$	(23)	(13) where (20)
AS	${}^{B,AS}CR_{21}(\widehat{r_1}, \upsilon_1, \theta, \varsigma_2)$	(33)	(26) where (9)
AA	${}^{B,AS}CR_{21}(\widehat{r_1}, \upsilon_1, \theta, \varsigma_1, \varsigma_2)$	(37)	(26) where (20)
Step 2 (Global PCR): Mapping local PCRs from $\{B\}$ to $\{A\}$ by Eq. (39)			

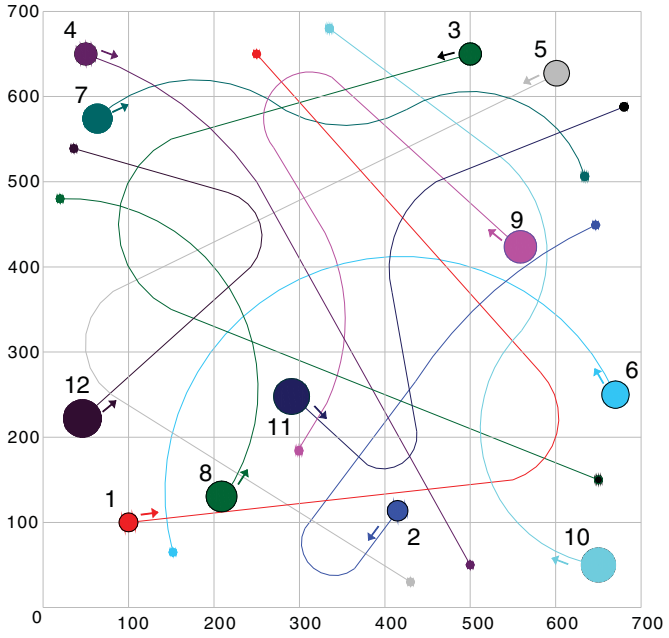

Fig. 10. (Colour Online) An example of the SCC paths for 12 agents. The workspace is 7 m × 7 m. The numbers indicate the ID of each agent.

in Table VI (12 agents, 126 CRs). Clearly, the computation time of the proposed analytical method is much lower than that of the other methods. The result of the PQP method, which was originally developed to check collisions between static polyhedral objects, is the sum of the computation time at every sampling time due to the fact that the other results represent the elapsed times of moving agents with a predefined path.

In the analytical method, the amount of time taken for computation for intersection detection increased with the square of the number of agents, but the total computation time depended primarily on the number of CRs (increasing almost linearly with the number), as shown in Fig. 12. This is a marked contrast to the results obtained from the computational and PQP method in which the total computation time proportionally increased with the number of agents. Such a result was obtained because collision detection in the analytical method was performed by intersection detection and CR generation in sequence, imposing approximately 75%–95% of the computational load on the latter. Conversely, the computational approach simultaneously detected collisions and generated CRs by sampling all the paths and comparing the distances between the sampling points. Incidentally, the computation times of the computational and PQP methods increased as the travel times along the paths increased, mostly due to the detail that the number of sampling points was given by the travel time and resolution. We determined the amount of memory used in describing the collisions, and found that the dependency on sampling resolution in the description was attributed to the notable difference between the results. More specifically, a maximum of seven float variables were required to describe a collision in the PCR method, as presented in Tables IV and VII, whereas, in the computational method, a collision was described with a set of points whose number depended on resolution.

For a more detailed comparison between the two approaches, the analytical and the computational collision detection results are presented in Fig. 13. In the analytical result, all potential collisions involving agent 3 were simply described with the boundary lines from the 18 PCR equations presented in Table VII and with unerring precision as shown in the figure. On the other hand, in the iterative computational method, the CRs consisting of a set of points were obtained by intensive computation. Since the boundary information is sufficient for collision representation, the results show that the iterative computational method is inefficient in terms of both computation time and memory usage. Moreover, when we applied a lower resolution of 0.1 s to reduce its computation time, the method yielded considerable inaccuracies, as shown in the figure; the iterative algorithm encountered the precision/time trade-off problem stated in Section 1.

Figure 14 shows examples of three special cases found in the collision detection simulations where the potential collision cannot be detected by using a single PCR. The first case is that there is a potential collision between two agents, although their paths have no intersection, as stated in Fig. 1 (see also Fig. 14(a)). This case can be solved using the C-space scheme, i.e., expanding one of the paths, and the formulas for PCRs. The second case is the overlapping of two or more CRs (see Fig. 14(b)), which occurs when the minimum distance between each part of two paths in a range of two intersections is less than the radius of the expanded agent. In this case, two or more PCRs are overlapped and merged into a single PCR, and thus many PCRs are required to describe the potential collision. The last case is the change of the path type within a collision segment (see Fig. 14(c)), which makes the original PCR inaccurate. By finding virtual intersections between paths, generating additional PCRs with virtual intersection information, and merging them, a correct CR can be generated, as shown in the figure.

https://doi.org/10.1017/S0263574711000142 Published online by Cambridge University Press

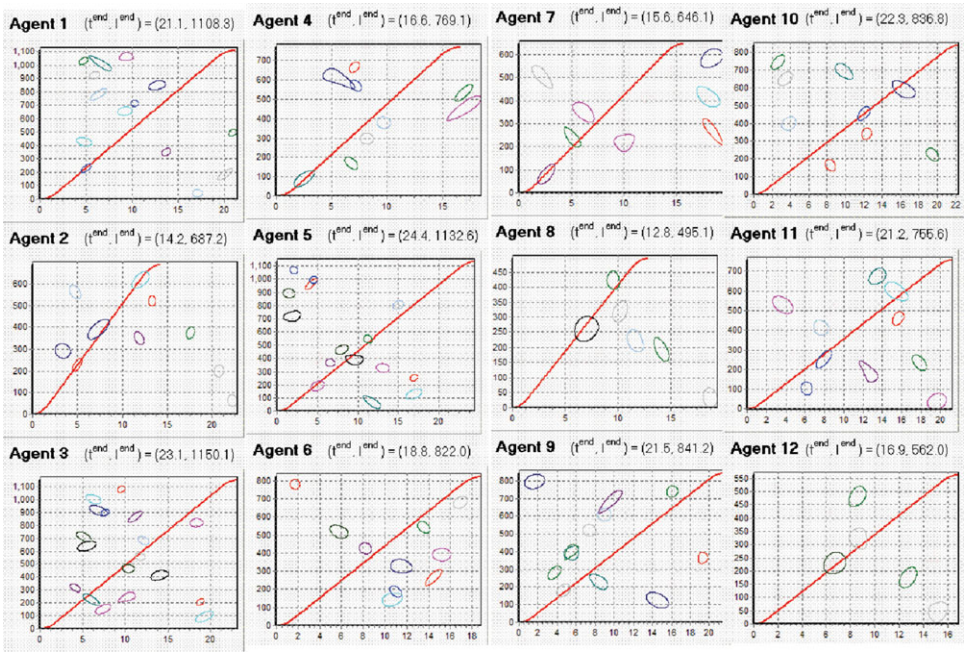

Fig. 11. (Colour Online) ECMs including 126 CRs and trajectories of 12 agents. The *x*- and *y*-axes indicate the time (s) and the length (cm), respectively. The trajectories were generated on the basis of the kinodynamic constraints defined in Eq. (3).

4.2. Experimental results

Experiments of four real robots (e-puck systems) were performed on the basis of the proposed approach (see Fig. 15). The mission for the robots is foraging in a small clustered environment where other robots and fixed obstacles coexist. We developed a real-time position detector consisting of an overhead camera, an image grabber (GP-3780), and a supervisory computer (Intel Core2 Quad 2.4 GHz). The positions of robots were detected by observing each ID number on the robots. To compensate for uncertainty of position detection (error bound: ± 15 mm) and robot motion, we updated robot motion at every two seconds by using the images from the camera with 30 FPS.

As presented in the figure, the original PCR (bold line, $t = 0$ s) was updated into a new one, and as a result, the trajectory was also modified for collision avoidance. Prioritization and the minimum-time delay method[32–34] were used as a collision resolution method in the experiment (details of collision resolution are not within the scope of this paper). In this way, an on-line replanning was systematically

Table V. Physical conditions and SCC path structure.

Agent no.	1	2	3	4	5	6	7	8	9	10	11	12
Radius (m)	0.11	0.12	0.13	0.14	0.15	0.16	0.17	0.18	0.19	0.20	0.21	0.22
Maximum velocity (m/s)	0.58	0.56	0.54	0.52	0.50	0.48	0.46	0.44	0.42	0.40	0.38	0.36
Maximum acceleration (m/s^2)	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30
SCC path structure	SAS	SASA	SAS	AS	SAS	A	AAA	A	SASAS	AAS	SASAS	SAS
No. of CRs	14	10	18	9	15	10	8	6	12	9	10	5

Note. The SCC path structure indicates the sequence of subpath types based on SCC path modeling, e.g., agent 1 moves along the three subpaths: a straight line (S), an arc (A), and a straight line (S) in order.

Table VI. Comparision of computation times.

No. of agents	No. of CRs	Maximum of traveled times (s)	Analytical method		Computation method		PQP method (ms)
			Time (ms)	Memory (KB)	Time (ms)	Memory (MB)	
5	16	24.4	1.4	1	7,750	3	97
12	16	24.4	1.5	1	48,125	3	1,220
12	126	24.4	14.3	4	51,110	10	1,220
30	218	22.3	18.6	14	193,763	17	3,791
50	718	23.9	66.3	46	553,279	54	10,755
100	2478	26.8	255.7	159	2032,841	202	53,619

Note. Cylinder-like objects, each with 100 triangles, were used for the method using the PQP library.

https://doi.org/10.1017/S0263574711000142 Published online by Cambridge University Press

Table VII. PCRs describing 18 collisions of agent 3 in Fig. 13.

	Parameterized collision regions		
1	${}^{SS}CR^1_{3\,10}(19.3, 0.9, 0.3, 0.4, -0.9)$	10	${}^{SS}CR^2_{3\,12}(5.5, 6.4, 0.4, 0.4, -5.2)$
2	${}^{SS}CR^1_{3\,9}(7.4, 1.5, 0.3, 0.4, -1.0)$	11	${}^{SA}CR^1_{3\,6}(12.1, 6.8, 0.3, 0.5, -1.9, 2.7, 1.0)$
3	${}^{SS}CR^1_{3\,1}(18.8, 2.1, 0.2, 0.6, -1.1)$	12	${}^{SA}CR^2_{3\,8}(5.1, 7.1, 0.3, 0.4, -4.1, 2.2, 1.0)$
4	${}^{SA}CR^1_{3\,7}(5.9, 2.3, 0.3, 0.5, 2.3, 1.5, 1.0)$	13	${}^{SA}CR^3_{3\,9}(18.3, 8.2, 0.3, 0.4, -1.5, 2.3, -1.0)$
5	${}^{SA}CR^2_{3\,9}(10.4, 2.5, 0.3, 0.4, 1.2, 0.5, 1.0)$	14	${}^{SS}CR^2_{3\,4}(11.1, 8.7, 0.3, 0.5, -0.7)$
6	${}^{SA}CR^1_{3\,4}(4.2, 3.2, 0.3, 0.5, 2.1, 4.0, -1)$	15	${}^{SS}CR^1_{3\,2}(7.6, 9, 0.3, 0.6, -5.0)$
7	${}^{AS}CR^1_{3\,12}(14, 4.2, 0.4, 0.4, -1.2, 1.1)$	16	${}^{SS}CR^1_{3\,11}(6.8, 9.2, 0.3, 0.4, -4.2)$
8	${}^{AA}CR^1_{3\,8}(10.4, 4.7, 0.3, 0.4, -1.7, 1.2, 1.0, 2.2)$	17	${}^{SA}CR^2_{3\,10}(6.3, 10.0, 0.3, 0.4, -4.3, 1.6, 1.0)$
9	${}^{AS}CR^1_{3\,5}(11.8, 5.6, 0.3, 0.5, -1.7, 1.1)$	18	${}^{SA}CR^2_{3\,1}(9.6, 10.8, 0.2, 0.6, -5.0, 0.8, 1.0)$

realized in our supervisory system; all the robots succeeded their foraging mission without any collision. Besides, in our method, the overall travel time to resources (8.1 s) was nearly the half (15.2 s) of that in the potential field which is a canonical reactive method. The main reason is that the strong coordination of the proposed method prevented a serious loss of time for collision avoidance in the crowded environment.

5. Discussion

Centralized multiagent motion planning is an NP-hard problem, and therefore solving it exactly is often impractical. In particular, collision detection has often been a major computational bottleneck in the planning because of its heavy computational burden. In addition to the computational tractability, precision is another important requirement for collision detection since the approximated CRs yields motions of poor quality when two agents move in opposite directions or share large portions of the same path.[46] An analytical method has been proposed here, which can satisfy these requirements. On the basis of the ECM method and SCC path modeling, the four intersection types were formulated and closed-form equations were presented to describe potential collisions as PCRs. In our simulations, all potential collisions among multiagents were simply described as the PCRs, as presented in Section 4. Thus, the proposed PCR method facilitates the direct mapping from complicated potential collisions in Cartesian space to simple ellipse-like shapes in time-length space.

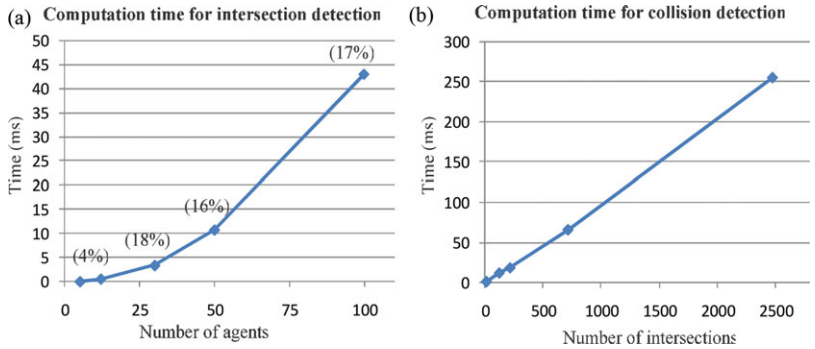

Fig. 12. (Colour Online) Amount of time taken for (a) intersection detection with the number of agents and (b) collision detection along predefined paths with the number of intersections in the proposed method. The number in the parenthesis indicates the ratio of the computation time for intersection detection to that for collision detection in the right figure.

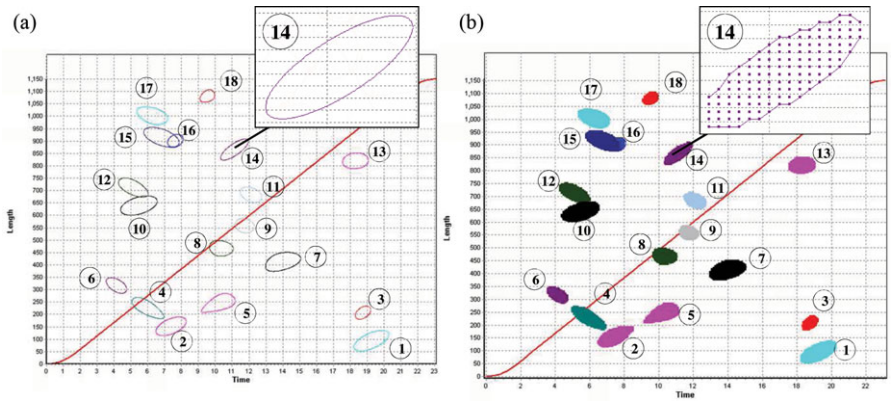

Fig. 13. (Colour Online) ECM for agent 3 generated by the (a) analytical method and (b) the computational method.

https://doi.org/10.1017/S0263574711000142 Published online by Cambridge University Press

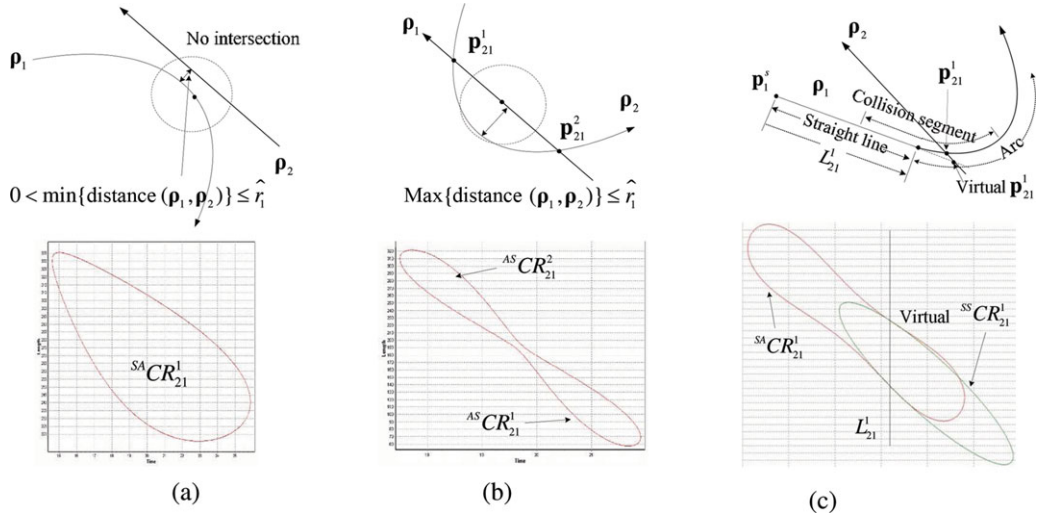

Fig. 14. (Colour Online) Special cases of collision detection (a) two paths have no intersection, but a potential collision exists, (b) two CRs are overlapped, and (c) the path type is changed near an intersection.

The following summary outlines the contributions this work has presented to the field of collision detection. First, the PCR method gave an exact solution for collision detection and description under the assumptions in Section 2. Previous studies[5,13,15,29] have addressed the need for an exact description of potential collisions required for finding optimal or time-efficient solutions. However, since the formulation was considered to be difficult, alternative methods were utilized, such as approximation, discrete-time analysis, or linearization, in which a problem with attaining precision was experienced. The PRMs that currently dominate multiagent motion planning also experience this problem because they inherently depend on sampling resolution. In this paper, the problems were solved by the PCR method, as shown in Fig. 13, and this method was based on the closed-form equations presented in Table IV. Therefore, the analytical method presents great possibilities and contributions to the field of collision detection and description.

Heavy computational burdens were greatly reduced by the proposed approach owing to the fact that the PCR method is based on analytical solutions rather than possessing a dependency on iterative computations. The observation follows that the accurate detection of potential collisions significantly increases the computation time,[5,13,34] and sampling-based methods utilizing collision checking packages spend considerable execution times for the collision detection in multiagents.[2–4] The proposed method yielded an elapsed time less than $0.1\% \sim 0.5\%$ of those for the computational and the PQP method. Additionally, under 0.1% memory was required for collision description, which is greatly helpful to reduce transmission time of a large-scale message on collisions among multiagents. Such values were obtained because intensive computation of the distance between all the sampling points with respect to the time traveled was unnecessary in our approach. Instead, two essential pieces of information on collision (specifically,

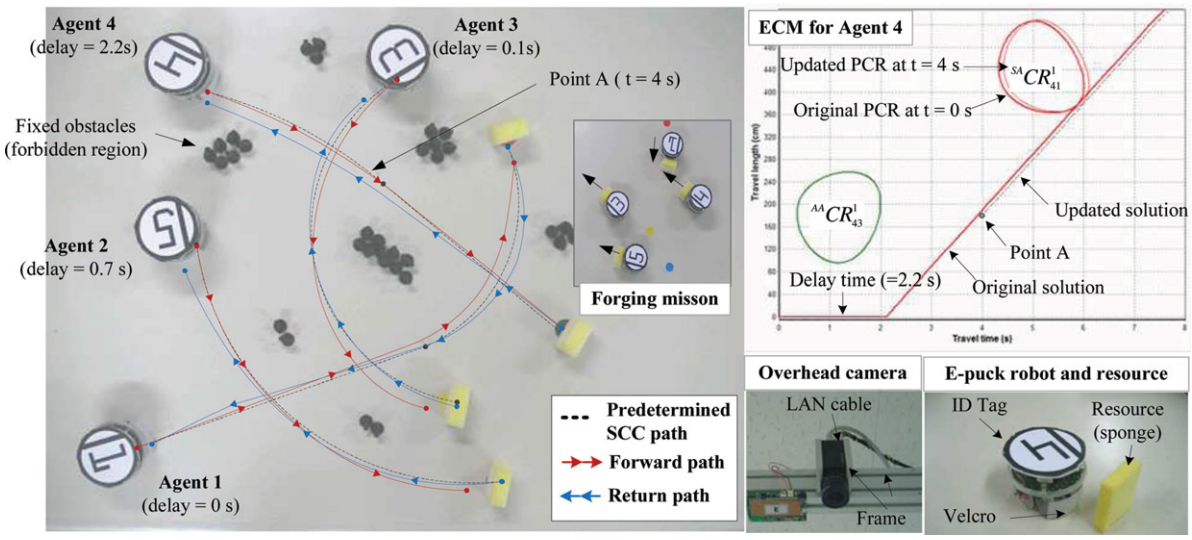

Fig. 15. (Colour Online) Experimental results of four e-puck robots. Initial paths were generated on the basis of the SCC modeling. Real paths were detected and modified by a supervisory system with an overhead camera. The delay times are the collision-free solutions in our approach. The solutions were updated from the ECMs for the robots.

https://doi.org/10.1017/S0263574711000142 Published online by Cambridge University Press

where and when a collision will occur) were directly obtained from SCC models and the equations for time domains.

A collision detector depending on the number of CRs (or intersections) rather than that of agents was proposed, which reduces the high dimensionality of C-space in a large number of robots with a few intersections. A recent study[45] addressed a dimensionality problem: two robots in separate rooms are considered a system with twice the degree of freedom, even though there are no intersections between them. This problem was solved by the proposed method in which collision detection was decomposed into two components: intersection detection and CR generation. In the method, computation time for intersection detection increased as the number of agents increased; however, depending on the number of intersections, most time was spent on CR generation, as shown in Fig. 12. In short, these findings are conducive for a centralized system managing a large number of agents as well as communicating with other systems. Such usefulness presented by the proposed method is due to the simplicity in collision description; the efficiency in terms of computation time, memory usage, and transmission time; and the nearly independent nature of the method from the number of agents.

The formulation of PCRs in Section 3 shows the possibility of combining the proposed approach and previous methods producing non-SCC paths. From the formulation, we can find that a whole path do not necessarily have to be modeled as a SCC path. This is because PCRs can be locally defined by only the information near intersections, and the travel time and length for global mapping can be easily computed regardless of the shape of path. This signifies that the PCR method can be applicable to even non-SCC paths if the part near an intersection is approximated as a line, an arc, or a serial connection of them. Generally, a part near an intersection is relatively small in a whole path, and thus the approximation of the small part does not cause a significant error in collision detection. If the SCC modeling is not at all possible, the computational approach of our previous work[34] is partially applied to local collision detection. In short, we can combine the proposed approach for local PCR definition and previous methods for global mapping.

Nonetheless, several issues require further study. First, in spite of the possibility of combination, it is still needed to model initial paths as SCC-paths, so that we directly exploit the advantages of PCRs without any additional procedures. In practical applications, such as robotic warehouse systems (e.g., KIVA[14]), the need can be realized since commercial robots are generally preprogrammed systems in which the type of path is selectable. Further studies are proposed to tackle new motion planning methods using the relationships between the parameters and the shape of the CR. Substantial detail should be placed on this study due to the fact that even small changes in the shape may result in considerable advantages in motion planning. Finally, intensive investigation must be conducted for new deterministic methods resolving detected collisions with the PCR (or giving solutions for Eq. (6)).

6. Conclusion

The results of this study indicate that the collision detection problem, which is a subproblem of multiagent motion planning (NP-hard problem) and which imposes a heavy computational burden on a centralized MAS, can be solved with an analytical method. On the basis of the ECM method and the SCC path modeling, four intersection types were formulated, and closed-form equations were presented to describe potential collisions as PCRs. The use of the PCRs enabled complicated potential collisions to be simply and systematically represented with respect to time traveled. Moreover, the PCR method provided precision in describing collision, efficiency in terms of computation time and memory usage, and nearly independent nature of method from the number of agents. Thus, the precision/time trade-off problem previously incurred in collision detection can be solved, and high dimensionality of C-space can be reduced by the PCR method. Our approach has demonstrated good potential as being a core technique of a deterministic, precise, and time-efficient motion planner for a centralized MAS such as a robotic warehouse system.

Acknowledgments

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

References

1. E. H. Durfee, V. R. Lesser and D. D. Corkill, "Trends in cooperative distributed problem solving," *IEEE Trans. Knowl. Data Eng.* **1**(1), 63–83 (1989).
2. D. Hsu, L. E. Kavaraki, J. C. Latombe, R. Motwani and S. Sorkin, "On Finding Narrow Passage With Probabilistic Roadmap Planners," *Parallel and Distributed Proceedings of IPPS/SPDP*, Orlando, FL (1998) pp. 141–153.
3. S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *Int. J. Robot. Res.* **20**(5), 378–400 (2001).
4. G. Sanchez and J. C. Latombe, "On delaying collision checking in PRM planning - Application to multi-robot coordination," *Int. J. Robot. Res.* **21**(1), 15–26 (2002).
5. J. Peng and S. Akella, "Coordinating multiple robots with kinodynamic constraints along specified paths," *Int. J. Robot. Res.* **24**(4), 295–310 (2005).
6. J. Peng and S. Akella, "Coordinating Multiple Double Integrator Robots on a Roadmap: Convexity and Global Optimality," *Proceedings of the International Conference on Robotics and Automation*, Barcelona, Spain (2005) pp. 2751–2758.
7. S. M. Lavalle and S. A. Hutchinson, "Optimal motion planning for multiple robots having independent goals," *IEEE Trans. Robot. Autom.* **14**(6), 912–925 (1998).
8. M. Saha and P. Isto, "Multi-Robot Motion Planning by Incremental Coordination," *Proceedings of the International Conference on Intelligent Robotics and Systems*, Beijing, China (2006) pp. 5960–5963.
9. J. C. Latombe, *Robot Motion Planning* (Kluwer Academic Publishers, Boston, 1991).
10. K. Fujimura, *Motion Planning in Dynamic Environment* (Springer-Verlag, NewYork, 1991).
11. H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavaraki and S. Thrun, *Principles of Robot Motion* (The MIT Press, Massachusetts, 2005).
12. S. M. Lavalle, *Planning Algorithms* (Cambridge University Press, Cambridge, 2006).
13. S. Akella and S. Hutchinson, "Coordinating the Motions of Multiple Robots with Specified Trajectories," *Proceedings of*

the International Conference on Robotics and Automation, Washington, DC (May 2002) pp. 624–631.
14. E. Guizzo, "Three engineers, hundreds of robots, one warehouse," *IEEE Spectr.* **45**(7), 22–29 (2008).
15. K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *Int. J. Robot Res.* **5**(3), 72–88 (Fall, 1986).
16. C. W. Warren, "Multiple Robot Path Coordination using Artificial Potential Fields," *Proceedings of IEEE International Conference on Robotics and Automation*, Cincinnati, OH (1990) pp. 500–505.
17. P. S. Lee and L. L. Wang, "Collision avoidance by fuzzy logic for AGV navigation," *J. Robot. Syst.* **11**(8), 743–760 (1994).
18. K. M. Krishna and H. Hexmoor, "Reactive Collision Avoidance of Multiple Moving Agents by Cooperation and Conflict Propagation," *Proceedings of the International Conference on Robotics and Automation*, New Orleans, LA (2004) pp. 2141–2146.
19. K. M. Krishna and H. Hexmoor, "Reactive Collision Avoidance of Multiple Moving Agents by Cooperation and Conflict Propagation," *Proceedings of the International Conference on Robotics and Automation*, New Orleans, LA (2004) pp. 2141–2146.
20. K. Azarm and G. Schmit, "Conflict-Free Motion of Multiple Mobile Robots Based on Decentralized Motion Planning and Negotiation," *Proceedings of IEEE International Conference on Robotics and Automation*, Albuquerque, NM (Apr. 1997) pp. 3526–3533.
21. M. M. Quottrup, T. Bak and R. I. Zamanabadi, "Multi-robot planning: A timed automata approach," *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 5, New Orleans, LA (2004) pp. 4417–4422.
22. D. Fox, W. Burgard and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.* **4**(1), 23–33 (1997).
23. H. Ma, D. J. Cannon and S. R. T. Kumara, "A Scheme Integrating Neural Networks for Real-Time Robotics Collision Detection," *Proceedings of IEEE International Conference on Robotics and Automation*, Nagoya, Japan (1995) pp. 881–886.
24. J. Y. S. Luh and C. E. Campbell, "Minimum distance collision-free path planning for industrial robots with a prismatic joint," *IEEE Trans. Autom. Control* **AC-29**(8), 675–680 (Aug 1984).
25. E. G. Gilbert, D. W. Johnson and S. S. Keerthi, "A Fast Procedure Computing the Distance between Complex Objects in Three-Dimensional Space," *IEEE J. Robot. Autom.* **4**(2), 193–203 (Apr. 1988).
26. M. A. Gill and A. Y. Zomaya, *Obstacle Avoidance in Multi-Robot Systems* (World Scientific, Singapore, 1998).
27. E. Freud and H. Hoyer, "Real-Time Path Finding in Multirobot Systems Including Obstacle Avoidance," *Int. J. Robot. Res.* **7**(1), 42–70 (1988).
28. R. A. Basta, R. Mehrotra and M. R. Varanasi, "Collision Detection for Planning Collision-Free Motion of Two Robot Arms," *Proceedings of IEEE International Conference on Robotics and Automation*, Philadelphia, PA (1988) pp. 638–640.
29. B. H. Lee and C. S. G. Lee, "Collision-free motion planning of two robots," *IEEE Trans. Syst. Main Cybern.* **17**(1), 21–31 (Jan./Feb. 1987).
30. Y. Shin and Z. Bien, "Collision-free trajectory planning for two robots," *Robotica* **7**, 205–212 (Jul.–Sep. 1989).
31. S. H. Park and B. H. Lee, "Analysis of robot collision characteristics using the concept of the collision map," *Robotica* 24, 295–303 (May 2006).
32. J. B. Park and B. H. Lee, "Roadmap-Based Collision-Free Motion Planning for Multiple Moving Agents in a Smart Home Environment," *Lecture Notes in Computer Science* (Springer-Verlag, Berlin, Jun. 2007) pp. 70–80.
33. C. Chang, M. J. Chung and B. H. Lee, "Collision avoidance of two general robot manipulators by minimum delay time," *IEEE Trans Syst Main Cybernetics* **24**(3), 517–522 (Mar. 1994).
34. S. H. Ji, J. S. Choi and B. H. Lee, "A computational interactive approach to multi-agent motion planning," *Int. J. Control Autom. Sys.* **5**(3), 295–306 (Jun. 2007).
35. E. Owen and L. Montano, "A Robocentric Motion Planner for Dynamic Environments using the Velocity Space," *Proceedings in IEEE International Conference on Intelligent Robots and Systems*, Beijing, China (2006) pp. 4368–4374.
36. E. Larsen, S. Gottschalk, M. C. Lin and D. Manocha, "Fast Distance Queries with Rectangular Swept Sphere Volumes," *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, CA (Apr. 2000) pp. 3719–3726.
37. G. Bergen, "Efficient collision detection of complex deformable models using AABB trees," *J. Graph. Tools* **2**(4), 1–14 (1997).
38. B. Mirtich, "V-clip: Fast and robust polyhedral collision detection," *ACM Trans. Graph.* **17**(3), 177–208 (1998).
39. J. D. Cohen, M. C. Lin, D. Manocha and M. K. Ponamgi, "I-COLLIDE: An Interactive and Exact Collision Detection System for Large-Scale Environments," *Symposium on Interactive 3D graphics*, Nashville, TN, USA (1995) pp. 189–196.
40. A. Ehmann and M. C. Lin, "Accurate and fast proximity queries between polyhedra using convex surface decomposition," *Comput. Graph. Forum* **20**, 500–511 (2001).
41. D. Nieuwenhuisen, "Callisto" January, 2010, Accessed http://www.nieuwenhuisen.nl/callisto/callisto.php, 28 March 2011.
42. R. J. Geraerts, Sampling-Based Motin Planning: Analysis and Path Quality," *Ph.D. Thesis* (Utrecht University, Utrecht, Netherlands, 2006).
43. S. M. LaValle, "Rapidly-Exploring Random Trees: A New Tool for Path Planning," *Technical Report No.* 98-11 (Computer Science Department, Iowa State University, Iowa, USA, 1998).
44. M. Peasgood, C. M. Clark and J. McPhee, "A complete and scalable strategy for coordinating multiple robots within roadmaps," *IEEE Trans. Robotics* **24**(2), 283–292 (2008).
45. J. Berg, J. Snoeyink, M. Lin and D. Manocha, "Centralized Path Planning for Multiple Robots: Optimal Decoupling into Sequential Plans," *Proceedings of Robotics: Science and Systems*, Seattle, USA (2009) pp. 1–8.
46. J. Berg and M. H. Overmars, "Prioritized Motion Planning for Multiple Robots," *Proceedings of the International Conference on Intelligent Robots and Systems*, Edmonton, Canada (2005) pp. 2217–2222.
47. Y. Li, K. Gupta and S. Payandeh, "Motion Planning of Multiple Agents in Virtual Environments using Coordination Graphs," *Proceedings of the International Conference on Robotics and Automation*, Barcelona, Spain (2005) pp. 378–383.
48. A. Scheuer and T. Fraichard, "Continuous-Curvature Path Planning for Car-Like Vehicles," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, Grenoble, France (Sep. 1997) pp. 997–1003.
49. W. D. Esquivel and L. E. Chiang, "Nonholonomic path planning among obstacles subject to curvature restrictions," *Robotica* **20**, 49–55 (2002).
50. I. Waheed and R. Fotouhi, "Trajectory and temporal planning of a wheeled mobile robot on an uneven surface," *Robotica* **27**(4), 481–498 (2008).
51. S. Leroy, J. P. Laumond and T. Simeon, "Multiple Path Coordination for Mobile Robots: A Geometric Algorithm," *Proceedings of the International Joint Conference on Artificial Intelligence*, Stockholm, Sweden (1999) pp. 1118–1123.
52. R. Jiang, X. Tian, L. Xie and Y. Chen, "A Robot Collision Avoidance Scheme Based on the Moving Obstacle Motion Prediction," *Proceedings of International Colloquium on Computing, Communication, Control, and Management*, Guangzhou City, China (2008) pp. 341–345.
53. M. Lin and S. Gottschalk, "Collision Detection Between Geometric Models: A Survey," *Proceedings of IMA Conference on Mathematics of surface*, Birmingham, UK (1998).

https://doi.org/10.1017/S0263574711000142 Published online by Cambridge University Press